Fully-automated Plant Recognition Systems in Challenging Controlled and Uncontrolled Environments Using Classical and Deep Learning Methods

Dem Fachbereich Elektrotechnik der
Universität Siegen
zur Erlangung des akademischen Grades eines
Dr.-Ing.

eingereichte Dissertation

von
Herr M.Sc. Masoud Fathi Kazerouni
aus
Kazeroon

Datum der Einreichung: 2019/09/24

Referent: Prof. Dr.-Ing. Klaus-Dieter Kuhnert
Koreferent: Prof. Dr. rer. nat. Volker Blanz

Tag der mündlichen Prüfung: 2020/03/03

gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

This dissertation is dedicated to my lovely family.
To my mother, for her unconditional love and devotion.
To my father, the hero of my life, for his continuous support.
To my brothers, for always standing by me.

Eidesstattliche Erklärung

Die vorliegende Dissertation wurde von mir selbständig angefertigt. Die verwendeten Hilfsmittel und Quellen sind im Literaturverzeichnis vollständig aufgeführt. Eingetragene Warenzeichen und Copyrights werden anerkannt, auch wenn sie nicht explizit gekennzeichnet sind.

Siegen, 2019/09/23

Masoud Fathi Kazerouni

Abstract

Similar to other sectors, present-day agriculture relies on new advances in different fields such as machine learning, computer vision, robotics, botany, etc. In the modern world, new scopes have been introduced to agriculture, either directly or indirectly, to meet human needs, preserve the natural and environments and resources for the future. As an example, the sustainability of growth is dependent on a drop in cost under a particular threshold, and modernization of agriculture, in different aspects, is a demand to accelerate the process toward an acceptable growth. In order to improve agricultural productivity and increase benefits, one necessity is to transition from traditional methods to modern methods and availability of smart machines. In this way, it is feasible to build systems based on automation and control concepts and utilize precise algorithms for carrying out different tasks with fewer hands-on farms and protecting natural resources for the next generations. Hence, experts in robotics and electrical engineering are also involved with new aspects of agriculture and farming.

Accordingly, while researchers have been forced to compete for increasing precision and profitability in agricultural activities and improve present methods with respect to natural environments, it is also necessary to serve on new major fronts: accurate mitigation of weeds in fields, optimum water consumption, reducing labor costs and number of workers, 24-hour remote control of fields, etc. Hence, it is necessary to provide more useful information about plant species and apply the extracted information for further purposes. Accurate recognition of plants is an essential part of such information. This task cannot be neglected as it supports not only farmers but also botanists and environmentalists.

By considering the workplaces of farmers and botanists, it is feasible to divide the workspaces into two main subsets: controlled environments like laboratories with static conditions and uncontrolled environments like outdoor environments with dynamic conditions. Despite the importance of plant recognition, a considerable number of works has been proposed for recognizing plant species in stationary conditions based on constant background, light condition, the position of leaves, presence of single leaves, etc. In the real world, such constraints and assumptions do not lead to promising results. Therefore, consideration of other factors is essential to build efficient systems for natural plant recognition.

In this research, both workspaces have been considered to develop well-mechanized plant recognition systems. This work employs the modern combined methods for local feature extraction and precise recognition of plant species. To fulfill the goals in the controlled environment, six different plant recognition systems are developed and evaluated by conducting various experiments. It is noteworthy that the modern combined methods have been adopted as the foundation of the first phase of the natural plant recognition systems in the uncontrolled environment. However, the story changes in outdoor environments and there is no fixed condition for taking images of plants and leaves.

In uncontrolled environments, environmental and non-environmental factors affect the photographing process. Light intensity and illumination are two crucial environmental factors that have an impact on images, and these factors may vary over time. Images taken from one particular scene or object are not the same if it is captured in the morning or the evening. Furthermore, weather affects the color intensity in outdoor environments as the color of leaves depends on temperature, light and water supply, and changes to these factors are also inevitable with the change of month and season. Non-environmental factors like background and distance have also effects on the performance of plant recognition systems. Backgrounds of images of natural plants taken in outdoor environments are generally more complicated in comparison to backgrounds in controlled environments. Meanwhile, the distance, either short or long, between camera and plant is undoubtedly another big challenge in uncontrolled environments. In addition, there is no certainty that the images contain only one single leaf or there will be a number of leaves within images.

To develop a more efficient natural plant recognition system, we ride the wave of the tsunami of deep learning and build a novel plant recognition system based on a convolutional neural network. Due to the promising result and the superior performance, the system is then deployed as the main core of a mobile real-time system. To evaluate the system, a mobile robot and a semi-robot have been equipped with cameras to navigate and explore the outdoor environment in two different years, 2017 and 2018. While exploring, an image is captured and automatically processed by the deep natural plant recognition system to visualize the species of the target plant as a real-time system. The final results show that the real-time mobile plant recognition system can identify natural plant species independently of the used camera, distance, time of day and other environmental and non-environmental factors in uncontrolled environments.

Natural Plant Recognition System, Deep Learning, Dynamic Environment, Controlled Environment, Field Robot

Zusammenfassung

Ähnlich wie in anderen Sektoren wird die heutige Landwirtschaft durch aktuelle Fortschritte in anderen Bereichen wie z.B. maschinelles Lernen, Computer Sehen, Robotik, Botanik usw. beeinflusst. Die Moderne eröffnet neue Perspektiven für die Landwirtschaft, sowohl direkt als auch indirekt, um den menschlichen Bedürfnisse besser dienen zu können, dabei die natürlichen Lebensräume zu erhalten und die Ressourcen zu schonen, um nachhaltig zu wirtschaften. So ist beispielsweise die Nachhaltigkeit des Wachstums von einer deutlichen Senkung der Kosten unter einen bestimmten Schwellenwert abhängig. Damit wird die Modernisierung der Landwirtschaft unter verschiedenen Gesichtpunkten eine Forderung, die den Prozess eines akzeptablen Wachstum beschleunigen kann. Um die landwirtschaftliche Produktivität zu verbessern und den Nutzen zu steigern, ist es notwendig, von traditionellen Methoden auf moderne Methoden und insbesondere auf die Verfügbarkeit intelligenter Maschinen zurückzugreifen. Auf diese Weise ist es möglich, Systeme auf Basis von Automatisierungs- und Steuerungskonzepten zu bauen. Mit solchen präzise an die Aufgabe angepassten Systemen können die gleichen Aufgaben mit deutlich weniger menschlicher Arbeitskraft in den Betrieben bewältigt werden und es werden gleichzeitig die natürlichen Ressourcen für die nächsten Generationen geschont. Daher beschäftigen sich Experten für Robotik und Elektrotechnik auch mit neuen Aspekten der Landwirtschaft.

Die Forschung ist allgemein bemüht die Genauigkeit und Rentabilität der landwirtschaftlichen Tätigkeiten zu steigern und vorhandene Methoden in Einklang mit den gegenwärtigen, natürlichen Gegebenheiten zu verbessern. Dies bedeutet auch, dass es notwendig ist in neuen Gebieten tätig zu werden. Diese Gebiete sind unter anderem, präzisere Minimierung von Unkraut auf den Feldern, Optimierung des Wasserverbrauchs, Reduzierung von Arbeitskosten und Arbeitskräften, sowie ständige Fernsteuerung von Feldern. Um diesen neuen Anforderungen gerecht werden zu können, werden detaillierte Informationen über die verschiedenen Pflanzenarten benötigt. Dabei spielt vor allem die exakte Identifizierung der Pflanzenspezies eine zentrale Rolle. Schlussendlich werden davon die Landwirte, Botaniker und Umweltschützern gleichermaßen profitieren.

Die Arbeitsbereiche von Landwirten und Botanikern lassen sich in zwei Hauptgruppen unterteilen: Der erste Bereich findet sich in einer kontrollierten Umgebung. Eine solche Umgebung ist charakterisiert durch statische Bedingungen und lässt sich zum Beispiel in einem Labor realisieren. Der zweite Bereich ist die unkontrollierte Umgebung. Diese Umgebung ist durch dynamische Bedingungen geprägt und lässt sich beispielsweise in der Außenumgebung finden. Ein Großteil der Forschung zur Pflanzenerkennung findet bisher in einer kontrollierten Umgebung statt und dokumentiert Ergebnisse, wie die Position und das Vorhandensein einzelner Blätter. Diese Ergebnisse sind auf einen konstantem Hintergrund und gleichbleibende Lichtverhältnisse angewiesen. In der realen Welt überwiegen jedoch dynamische Bedingungen. Folglich kommt es in der Praxis häufig zu unzulänglichen Ergebnissen bei der Identifikation von Pflanzenarten. Für die Optimierung der Pflanzenerkennung in der Praxis ist es daher unbedingt erforderlich weitere Faktoren, die über die in den kontrollierten Umgebungen vorhandenen hinausgehen, zu berücksichtigen.

In dieser Forschung wurde in beiden Arbeitsbereichen die Entwicklung gut mechanisierbarer Pflanzenerkennungssysteme berücksichtigt. In dieser Arbeit werden moderne kombinierten Methoden zur lokalen Merkmalsextraktion und zur präzisen Erkennung von Pflanzenspezies eingesetzt. Um die Ziele in der kontrollierten Umgebung zu erreichen, werden sechs verschiedene Pflanzenerkennungssysteme entwickelt und durch verschiedene Experimente bewertet. Diese Methoden wurde auch in der ersten Phase in der unkontrollierten Umgebung verwendet. In der Außenumgebung gibt es jedoch keine festgelegten Bedingungen für die Aufnahme von Pflanzen und Blättern. Damit sind die Randbedingungen substantiel anders.

In unkontrollierten Umgebungen beeinflussen Umwelt- und Nicht-Umweltfaktoren den Prozess der Fotografie. Lichtintensität und Beleuchtung sind zwei wichtige Umweltfaktoren, welche die Bildentstehung beeinflussen. Sie können auch noch im Laufe der Zeit variieren. Die Bilder, die von einer bestimmten Szene oder einem bestimmten Objekt aufgenommen wurden, stimmen nicht überein, wenn sie morgens, mittags oder nachmittags aufgenommen wurden. Darüber hinaus beeinflusst das Wetter die Farbintensität in den Außenumgebungen, ebenfalls ist die Farbe der Blätter von der Temperatur, dem Licht und der Wasserversorgung abhängig. Änderungen dieser Faktoren sind auch mit dem Wechsel von Monat und Jahreszeit unvermeidlich. Nicht-Umweltfaktoren wie Hintergrund und Entfernung wirken sich zusätzlic auf die Leistung von Pflanzenerkennungssystemen aus. Hintergründe von Bildern natürlicher Pflanzen, die im Freien aufgenommen wurden, sind im Vergleich zu Hintergründen in kontrollierten Umgebungen im Allgemeinen komplexer. Der kurze oder große Abstand zwischen der Kamera und den Pflanzen ist zweifellos eine weitere große Herausforderung in unkontrollierten Umgebungen. Außerdem gibt es keine Gewissheit, dass die Bilder nur ein einziges Blatt enthalten, oder eine große Anzahl von Blättern widergegeben wird.

Um ein effizienteres natürliches Pflanzenerkennungssystem zu entwickeln, surfen wir mit auf der aktuellen Tsunami des Deep Learnings und bauen ein neuartiges Pflanzenerkennungssystem auf, das auf einem recursiven neuronalen Netzwerk basiert. Dieses System wird detailiert vorgestellt und evaluiert. Aufgrund der vielversprechenden Ergebnisse und der guten Leistung wird das System dann als Hauptkern eines mobilen Echtzeitsystems eingesetzt. Zur praxisnahen Bewertung des Systems wurden ein mobiler Roboter und ein Semiroboter mit Kameras ausgestattet, um in zwei verschiedenen Jahren - 2017 und 2018 - die Umgebung im Freien zu erkunden. Während der Erkundung wird das Bild erfasst und automatisch vom System die Spezies der im Bild sichtbaren Pflanzen bestimmt. Die Endergebnisse zeigen, dass das mobile Echtzeitpflanzenerkennungssystem in der Lage ist, natürliche Pflanzenarten unabhängig von der verwendeten Kamera, der Entfernung, der Tageszeit und anderen Umwelt- und Nicht-Umweltfaktoren in unkontrollierten Umgebungen robust zu identifizieren.

Natürliches Pflanzenerkennungssystem, Deep Learning, Dynamische Umgebung, Kontrollierte Umgebung, Feldroboter

Acknowledgments

First and foremost, I would like to convey my deep gratitude and respect to my supervisor, Prof. Dr.-Ing. Klaus-Dieter Kuhnert, for his continuous support, advice, guidance and encouragement throughout the project. He provided an opportunity to join his research group at the Institute of Real-time Learning Systems (EZLS), University of Siegen, and to start an amazing journey. I deeply appreciate all his contributions of time and ideas to make the PhD more enjoyable while encountering challenges.

I would like to express my heartfelt thanks to my co-advisor, Prof. Dr. rer. nat. Volker Blanz, and I am particularly grateful for his interest and time. I will never forget how welcome he made me feel at the first meeting. My sincere thanks also go to Prof. Dr.-Ing Madjid Fathi and Prof. Dr. Roland Wismüller for their time, interest, helpful comments and practical points on my work. I would like to thank the members of the PhD committee again for giving me the opportunity to learn from them.

The members of the EZLS Institute have contributed immensely to my personal and professional time at the University of Siegen. The colleagues at the EZLS provided a great environment with a source of friendships as well as good atmosphere and collaboration. I am also grateful to all members and staffs at the University of Siegen who were like my family and always ready to help and contribute to overcoming the challenges.

Although my journey was a long one, I was not traveling alone, and my family and friends were always beside me with love and encouragement. Despite the completion of this chapter of my life, my memorable moments in this journey are unforgettable. The end of this chapter is the beginning of the next chapter of my life which will be enriched by all experiences gained in the PhD study and all my pursuits.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The real age of the Earth is still an unsolved question for scientists. However, the timeline of the Earth's ages provides an approximate planetary age, while the start of life is accordingly obtained and is estimated to have begun roughly 0.7 billion years later than the Earth itself. However, scientists approximate the age of the Earth to be 4.5 billion-years old. Many researchers believe that life on Earth is a process which has been gradually developed and its start is not a sudden phenomenon in reality. Due to the continual progress of life through different ages, we are able to declare that life is certainly a constant occurrence. A clever question that might be asked is, "How can we explore the Earth and life?" The main sources of information to answer this question are fossils, and scientists can investigate the fossil records to achieve valuable information about the history of humans, life and the Earth. In addition to such records, it is useful to study microscopic ancient plants to discover more aspects of the Earth and the life on it during different ages. Due to the existence of plants from early ages, they have played a significant role in different aspects of the mentioned process throughout history, and their effects are definitely undeniable. If we go back to the Ordovician-Silurian extinction, we see tracks of plants on climate change which happened more than 425 million years ago, and the role of plants is absolutely bold over geological time. Although it was approximately more than 3000 million years ago that the first plant-like living-organism called blue-green algae was found, it might still be found in water. Early plants differ from today's plants and researchers' findings have proven that the structure of early plants was as complex as plants in our age. Changes, roles, effects and other related issues of plants encourage us to inquire into different aspects of the evolutionary history of plants. Moreover, the level of complexity has been varied in plants over time and such variations start from the earliest algal mats, through bryophytes, lycopods, and ferns, to the complex gymnosperms and angiosperms of today [1].

Let's look into the structure and components of atmosphere during the existence of early plants. During that age, the major component of air was carbon dioxide and the amount of oxygen was not suitable for today's plants. Over time, photosynthetic plants slowly came along and had effects on the composition of the atmosphere; the amount of oxygen consequently increased, and the atmosphere was filled with oxygen as the plants moderated the amount of carbon dioxide by consuming it during the photosynthesis process, a unique ability of plants for turning solar energy into chemical energy. Two important factors, perspiration and breath, had direct effects on the cooling of the atmosphere. Atmospheric cooling had indirect effects on climate and weather conditions over time. To summarize the mentioned points, we should acknowledge that plants are key regulators of nature and life on the Earth, and they function as a significant and fundamental basis of the Earth's ecologies, either on land or under water.

Like other organisms, the history of plants is full of ups and downs over time. A common origin story describes the early humans who rubbed two sticks together and made fire for the first time. This story has frequently been proposed in elementary school courses, and some students may have even tried to conduct such an experiment. Perhaps it seems to be a simple fact for us today, but we cannot ignore the role of plants and the ingenuity of our early ancestors. Throughout history, human civilization has owed its stability, progress, and continuity of existence to plants, and no one can deny this truth. Two important connected factors, water and plants, are the basis of human life, and the human is able to consume water for agriculture activities regardless of irrigation method.

Let's look at the Earth and the distribution of the population on the planet. We find people are living all around the world and there is also a huge diversity of plants. Plants can be found in any place, such as deserts, oceans, etc., and the diversity and distribution of plants affect how the world's population is distributed. Furthermore, many plants grow in places that are not habitable by humans.

With regard to human life, an important issue is the development of human civilization. By studying different sources concerning human civilization, the emergence of civilization near rivers is the common point of explanation. The first human civilizations that appeared close to rivers can be considered the origins of modern life. In addition, another basis for human civilization is agriculture, including different plants cultivation and other activities in agriculture. Besides, the relationship between plants and human civilization has undoubtedly had various impacts on, for example, human history, sociology, economics, culture, literature, etc. Through the investigation of many cultural masterpieces, we find the undeniable effects of plants; for instance, Hafiz [2], the beloved Iranian poet, created a masterful style by using plants and flowers in various themes in his novels, providing evidence of the significance of plants in literature. If we have a wide look into the realms of human activity, we find the presence of plants in many different fields that we might not think to associate them, for instance, religion, social behavior, politics, and customs. Ultimately, plants and plant products have been essential components and driving forces in exploration, religion, wars, slavery, and innovation.

Domestication is a process which has usually been neglected and its impacts have not been evaluated as they are in reality. Over many decades, the main sources for human survival were wild animals, fish, and wild plants. Let's go back to the Paleolithic ("Old Stone") and the Ice Age periods and investigate human life. Interestingly, all human groups were involved in nomadic life as hunters and gatherers. Domestication of wild plants and wild animals started at the time that people settled down in the Middle East. Accordingly, success of human civilization is debated to domestication.

To continue our study on the influences of plants, we would like to consider terrestrial ecosystems and life cycles to specify the role of plants. Obviously, plants are rich sources of nutrients and they have the first rank of primary producers of nutrients, especially proteins. The presence of plants in the water cycle and biogeochemistry has made plant species the gold component. In addition, they are considered as the gold component in the energy cycle due to their role in producing energetic molecules.

Although many types of plants have been become extinct over time, the applications for plants have increased, and there has been no limit to these functions over the past years. The collection of information about extinct plants is a contemporary demand as advances in biotechnology and genetics contribute to the capacity to produce similar lab products in the future. Considering the possibility of the extinction of plant species, it is essential to have a comprehensive database of different plants [3] [4] [5] [6].

Since prehistoric times, plants have been used for their healing properties. From the beginning of the 20th century, a part of the pharmacy curriculum was growing, which involved the study of medicinal herbs [7], and continued for several decades. More attention to plants has been the result of the commercial growth of medicinal plants and the public interest to benefit from products and soothe and improve discomfort and different health problems. Furthermore, people tend to care for their ailments by using natural treatments. Rich resources of ingredients can be utilized in medicine and industry. Such resources are medicinal herbs, which can be utilized in drug development and synthesis, and many compounds of plants are widely usable in modern medicine. Consequently, the effects of plants on the advancement of drug development cannot be neglected as there is a very close relationship between plants and the drug industry. Furthermore, it should be pointed out that there is an estimation showing that more than two thirds of the plant species on the Earth have medicinal value. Nowadays, many scientists are targeting profitable and sustainable agrifood, fibre and horticultural industries as they attempt to develop new plant products with unique properties and manage natural resources as well.

These days, traditional agriculture in many countries has been replaced by modern methods, and many new methods and techniques have been developed to increase the efficiency of agricultural activities and avoid destroying the outdoor environment. In addition, biotic functions of ecosystems depend on plants including soil fertility and stability, water availability and pest control for sustainable agriculture, rangeland management and restoration. Plants also have effects on climate change and, ultimately, the health of ecosystems.

Before stepping into the next phase, we would like to have a general glance at the relationship between plants and humans and other creatures. If we consider conflicts and interactions between plants and other creatures like humans and animals, we find many positive and negative impacts on different resources. Any change of behavior in a creature, results in new changes in the natural environment. People want to govern the Earth and enjoy a set of ecological, economical, scientific and recreational benefits from different components of the environment, and we do believe that plants are one of the best resources for such benefits.

An investigation of the agricultural applications shows a new horizon of plants which is brightening in the field of plant genetics. New research areas have been developed and progressed to overcome new challenges for human life and to fulfill new necessities. If we would like to secure global food, we need to consider plant genetics as a key component which should be applied in agriculture. It is notable that the history of agricultural science is very rich, and it goes back to the Roman era. Despite being traditionally huge sources of energy, the role of plants has become more colorful in modern life, and the focus of many researchers is on plants either directly or indirectly. Furthermore, many human needs, such as shelter, clothing, medicine, fuel and raw materials, can be fulfilled by plants.

Due to the importance of plants in different fields, new roads have been constructed in other fields with regard to plants and their various applications. Two important fields, computer vision and robotics, are tied to plants, and new goals have been defined with regard to plants, their roles and human use. As an example of this interaction, let's consider a farm with many uneducated workers who are not familiar with nor can they recognize the many types of weeds which grow in the field. If the farm is providing raw materials for a pharmaceutical industry, correct recognition of plants is critical and this can be performed by workers or automatic systems. Recognition of plant species is a new door which has been opened in modern farming and agricultural applications. Hence, there is a demand to do research and work in this field in parallel with considering economic benefits.

Plant recognition involves the task of identifying plant species correctly and has differently related aspects. A beginner might ask some questions about this task. The first most frequently asked question is "Is it possible for human to identify plant species?" The second question asked is "Who is responsible for plant recognition and how is he/she able to recognize different plant species?" The third question is "Can we develop a system for recognizing plant species automatically?" It is hard to answer all questions and explain all aspects of them completely, thus, in this research, we introduce some major points. However, the answers to the proposed questions and many new questions will be answered in this work.

Botanists are specialists in botany and plant sciences. They are commonly involved in plant recognition, which is often a challenge even for them despite their expertise and knowledge of different plants. Ultimately, botanists are not capable of recognizing all plant species easily, and they have to spend quite a long time identifying species through the use of reference books. Furthermore, it should be pointed out that the status of plants is very important in biological applications, and accurate identification contributes to correct and quick diagnose of plant diseases. Automatic supervision of plants is also possible if we can identify species correctly and protect them from different diseases and pests. Other people who are closely involved in plant recognition are farmers and farm workers; however, as their knowledge is usually limited in this field, they often do not have broad information about many plant species and are sometimes unable to distinguish harmful weeds in the fields. Nowadays, a novel intention is to produce plants with new pre-defined characteristics and properties, and scientists in the field of biotechnology are working in labs across the globe to develop plants with unique advantages.

To summarize, different aspects of plants and their roles prove the importance of developing a plant recognition system that will let all people, even non-professionals, obtain the scientific knowledge of botanists concerning plants. Moreover, a real demand of contemporary life is to decrease the operation time of most man-made systems and accelerate recognition tasks. Other advantages of designing a plant recognition system are fast classification, understanding and management of plant species. Hence, the goal is to investigate, design, and implement a plant recognition system to meet the aforementioned advantages.

1.2 Problem Description

1.2.1 Classification of Plants

According to the literature on plant recognition research, plant classification is generally based on leaves, and recognition systems complete this task by considering only one leaf of each plant. If we go further and target plants in the natural environment, challenges will be changed in comparison to recognizing only one leaf of a plant, and we discuss this condition later.

Plant recognition systems can be divided into three different groups which include user-based systems, semi-automatic systems and fully-automatic systems. If the system is based on a user, the individual influences how the system is working and their decisions affect the performance of the system. For instance, there are some systems wherein a user is able to select different features and adjust parameters, and the system then starts doing the task by using the selected features and defined parameters. Furthermore, we should keep in mind that the knowledge of the user is also important, and we cannot ignore his/her familiarity with plants and the recognition process. The second category of plant recognition is the semi-automatic one. In such a system, the user has less influence on the performance of the system and the user's decisions have less effect on the final results. However, this type of system does not perform the task without human interaction. The third type of the systems is the fully-automatic system which fulfills the task of plant recognition automatically. In such systems, there is no interaction between the user and the system as the whole task is automatically performed. Some systems have been proposed as fully-automatic, however, in reality, they provide some different plant species as a result of a test but the user is required to find the exact plant species visually and only then the specific result will be provided as the output of the test. Hence, such systems are finally human vision-based, not machine-based. In general, all types of systems can potentially be used in innumerable domains of plant recognition tasks. Existence of human action is an important challenge in itself, and the decrease of external interference is a goal while finding a good solution is a desire.

We are able to group the current plant recognition systems based on another concept. A good hypothesis is to model machines on the basis of human vision and to try to model the human eye to carry out plant recognition tasks. If a system is based on human vision, it might also be dependent on human action and decision. Hence, the system is based on two different concepts, vision and action, where the action component can be performed physically or mentally. For instance, the user has to decide on the possible proposed plant species whereby the final result depends on his/her decision, and this decision is directly connected to the human vision and the brain. Plant recognition is a challenging problem in any case, but it is more challenging if we have several leaves in an image. Such cases will become very complicated if the background of the image is not constant, and the background varies from one image to another. Consequently, we have complex images with many different parameters if we take pictures of plants in outdoor environments. In such cases, we do not only have a plant, there might be different types of challenging objects, such as soil, mud, concrete, rocks, fences, signs, etc., within the scene. Therefore, there are many different factors and parameters that we have to struggle with to solve the problem of plant recognition.

The most important and crucial aspect is to find how we can classify different plant species in the natural environment. Firstly, we have to investigate the natural environment and the possible factors that might affect images. Subsequently, an optimal path should be found to help us achieve our goals. In this work, we are going to solve the problems of plant recognition step-by-step and develop different systems in each step. Moreover, we usually rely on features of images and types of features are not the same in different systems. However, the extracted features are typically local ones. We detect the features from the images and then engage in description to get rich information from the image. However, the starting point of feature engineering is very different from the last phase of the work. The point is that we change the method of feature extraction in the last phase and trust this to very new neural network modeling. Meanwhile, there will also be some changes to the learning part of the systems while we are "swimming in the river of plant recognition", so to speak, and approaching the finish line of the work. From a learning point of view, it becomes very difficult and sophisticated to compare the first steps of the work using traditional learning techniques.

It is worthwhile to mention that we introduce different existing plant datasets and start implementing plant recognition systems by using classic datasets. Particularly using the classic dataset, one might face many problems due to the effects of the number of plants, changes of color, changes of shapes, etc. A lack of information causes misclassifications of plant species as the number of plants of the dataset are rather large. Furthermore, using different feature extraction techniques and implementing various recognition systems contributes to having final systems with different runtimes and applicability, and the consumption of time varies from one system to the other one which can be considered as a strength for each system.

On the other hand, even though implemented systems provide both good accuracy and runtime, it is still necessary to improve proposed approaches in order to obtain better performance. Afterwards, we start working on natural images of plants, and our focus is to develop efficient systems for the recognition of plants in uncontrolled outdoor environments. Light changes have huge effects on images which are taken in outdoor environments, thus, we have prepared a modern dataset due to different lighting conditions. Let's imagine that the weather is windy, and we would like to identify plant species. New harmful effects will be added to the images taken in windy weather. In such cases, the human eye is not able to recognize the shapes of leaves, and plant recognition proves difficult through the use of a machine. Other factors, like the distance between the camera and plant, create new challenges, and we have to struggle with them as well. In the first phase of the natural plant recognition, local features are still trustable, and we have a depth investigation of this important issue for building systems based on modern description techniques. At this stage, we will stop describing the remainder of the problems and complete this explanation in the next section.

To summarize, this dissertation answers the question of if it is feasible to develop systems with good performance and speed for classifying different plant species so that it can be used for real-time applications and in uncontrolled natural environments. Also, we would like to clarify if it is worth combining modern feature detection and description approaches in the sense of producing better feature components to build classifiers with high performances in different aspects.

1.2.2 Real-time Natural Plant Recognition System

Till now, we have worked on classic and modern plant images by using local features and modern approaches. In this stage, we start exploring the new problems of the work and propose new systems with regard to real-time applications in uncontrolled environments. The first problem concerns how we will be able to design a real-time system for natural plant recognition. To solve this problem, we focus on deep neural networks and our investigation leads to designing and building a new system based on deep neural networks. As our main goal is to use the developed system as a mobile real-time system, we need to consider all the possible challenges that we might face in an uncontrolled environment. The main challenges for plant recognition in outdoor environments are the large variation in the shapes of plants and leaves as well as the lighting and background. The first challenge is to shape the variation of the leaf in an outdoor environment. A leaf of one specific natural plant species appears very different from various viewpoints and angles. Moreover, the age of the leaf affects its shape whereby the fresh leaf and the old leaf do not appear the same. On the other hand, there might exist dying and dried leaves amongst a bunch of leaves within the natural images of plants which makes the problem of classification extra challenging. An important question to address is why would we like to change the direction of the proposed approaches and developed systems.

Let's consider the first challenge and investigate the local feature-based approaches. The main concept behind such approaches is feature-matching. However, shape variation makes feature-matching difficult between two natural images of one plant species of different shapes. Furthermore, it might be impossible to do this technique even if we segment only one leaf in each image and use these segmented leaves instead of using all leaves for matching. On the other hand, leaves can be more similar to each other in terms of pixel values than matching two single leaves of different plants; however, similarity of leaves is very low in reality. As mentioned, the second obstacle is the variation of light conditions. It is very hard to recognize objects, especially leaves with their complexity, under varying light conditions with many shadowy areas. From morning to night, the amount of light increases and decreases, and we cannot influence the natural amount of light; this is because we do not want to interfere with the natural environment, although we would be able to adjust light conditions by using additional appropriate equipment. Meanwhile, even two images from the same part of a plant or the same leaf under different light conditions can appear dramatically different from each other. Finally, a complicated and unexpected background is also an impeding factor and the recognition performance is reduced dramatically by a varying background. These mentioned factors are out of our hands, and we cannot easily control them. However, even if we were able to control these factors, implemented systems are far away from the final goals of this research.

One factor is distance between camera and plant. However, in our real-time application we aim to develop a system that works without any limit on distance and is capable of plant classification from long distances of, for instance, 100 cm and 200 cm. It should be noted that this factor has not been considered in other plant recognition systems till now. In this phase of the work, our focus is primarily on overcoming all challenges to improve the plant recognition performance in uncontrolled outdoor environments.

Our purpose is to create a mobile and real-time system for identifying natural plants, therefore, we attempt to build semi-robot and mobile robot systems to fulfill the desired task. It should be pointed out that the mobile robot system is an autonomous robot which can also be utilized as a controllable robot. One idea is to enable the system to perform the recognition task without depending on the use of a single camera. As a result, the systems can be used in any natural condition and at any distance by using different cameras. Furthermore, the mobile system can navigate or be navigated through outdoor environments like farms and forests. In order to solve the problems and overcome challenges, we implement a natural plant recognition system based on a deep learning concept. The performance of the system is higher than other proposed systems. We utilize the system as a real-time one and test it in a mobile system by using three cameras. More details of the systems, robots and cameras will be provided in the sequel chapters. Our last test, the real-time test, is not common in plant recognition tasks. We will not only test images at different times and on different days, but we will also capture testing images in different years and seasons, spring and summer.

1.2.3 Problems and Needs of Plant Recognition System

According to the literature, most of current systems are deficient in some aspects, and there is still a gap between our needs and developed systems for plant recognition. In this work, the lack of related factors has been addressed and tried to solve environmental and non-environmental challenges in order to approach the final desired goals. In order to obtain satisfactory results, the listed factors have been taken into account. Some factors have been considered as problems and listed below. Also, these factors will be explained in detail in the following chapters.

Efficiency

The meaning of efficiency is the ability to get desired results. In order to inject efficiency into a system, the method used should be a ubiquitous tool and technique. Efficiency is a significant aspect in determining that a recognition system is reliable to be used even in challenging conditions and environments. That being said, a system's efficiency can be determined through measurable and mathematical concepts. Therefore, it is possible to express the efficiency quantitatively and compare the obtained results of different implemented systems. This feature can be translated into a system's productivity in achieving correct results whereby the amount of incorrect results is at a minimum and the waste of the systems resources is very low.

Effectiveness

Effectiveness can be referred to as the capability of obtaining desired results. The quality of a system can be shown by its effectiveness. In terms of computer usage, effectiveness can be connected to the completion of desired tasks. Therefore, the research intends to investigate the effectiveness of the existing systems and to try to compensate for their weaknesses.

Generality

Ultimately, the system should be connected to different situations. The feature of generality can be interpreted by extracting different concepts and defining various situations. One goal of this research is to achieve the highest level of generalizability. This feature may have its price whereby other crucial features are ignored. Consequently, one challenge is to avoid sacrificing other features uncritically.

Accuracy

Accuracy plays a central role in different fields of science and has different meanings. For instance, its definition involves the nearness of a calculation to the true value in numerical analysis, where it can be defined as the measurement of tolerance in industrial instrumentation. In order to investigate a classification system, final obtained accuracy can be used to compare the system with other systems. However, this feature cannot represent the quality of a system individually. In addition to accuracy, other factors such as precision and recall are essential in a comparison between classification systems.

Stability

In order to design a useful and powerful system, it is necessary to investigate the stability of algorithms. For instance, it is essential to use stable algorithms to detect corners or key-points in one image. If the image is rotated, the result of corner detection should remain correct. This is one of the features that should be taken into account.

Being Automatic

In order to classify plants, correct selection of metadata is one important step. There are three types of metadata selection, manual, semi-automatic and automatic. Manual selection relies on users and the individual decides which piece of information should be selected. In this type, the action of a user is mandatory. The second mentioned type is semi-automatic, and it is often used in place of the traditional selection type. The last type is automatic which can be achieved by using technology instead of a user. This type of selection intends to utilize modern methods to precisely, consistently and efficiently apply the metadata. As a classifier system consists of different components, it is very important to make the whole system automatic. All parts should be connected to each other to build a fully-automatic system without any user interference. In automatic systems, there is no request for user interaction and decision-making in a classification task. Automatic systems can be applied in industrial and robotic fields.

Responsiveness and Usability in Real World

Responsiveness is a concept in computer science which can be referred to as the ability of a system to complete the desired tasks in the real world. There is also another important factor called usability which has different definitions and relevant concepts. The first important concept is the ease of use. Another concept is the degree to which software or systems can be utilized by specified consumers to achieve quantified objectives with effectiveness, efficiency and satisfaction in a quantified context of use [8]. The designed system should not only be usable by experts in a lab but should be usable in the real world by non-experts too.

Robustness

Robustness can be referred to as a system's persistence despite noise and perturbation. Different factors and parameters can be assumed as unwanted noise or perturbation in one system. In order to have a robust system, it is necessary to investigate the robustness of the used methods and algorithms developed. In addition, apart from the robustness of the methods and algorithms, it is also crucial to check the robustness of the whole system. If a system is robust to changes of the environment and related parameters, this system can be trusted in different situations. For instance, a classification system can be designed and built due to its robustness against illumination variations. This feature should be taken into account due to mentioned points.

Availability

Various concepts of availability can be assumed for one implemented system. If availability is considered as a factor of reliability, reliability increases and so does availability. If the considered issue is data, availability of data in real world and labs can be an important feature for consideration. In addition, this feature can also be described as the ability to quickly adapt to changes in circumstances. Ultimately, consideration of this feature is inevitable.

Adaptability

Due to the rapid development of systems and applications, one important factor is the real adaptability of the target purpose in a heterogeneous environment. It is very important to create a system which has adaptability to light changes. Therefore, the implemented system should be adaptable.

Complexity

Complexity is one important problem that should be taken into account as a feature. It describes the behavior of an implemented system which consists of different parts. However, there is no specific meaning for complexity among scientists in the real world and the meaning can be defined in relation to systems and phenomena. If the system is not complex, the problems of the system are tractable, and it is possible to address the problems accurately. Different components of the system should be connected in a manner that they can be investigated simply. In order to be able to find and solve problems, it is necessary to pinpoint the cause and also know which components of the system fulfill which goals. Moreover, computational complexity is another concept which can be focused on. This concept is also helpful to compare different implemented systems according to the utilized algorithms and methods.

Flexibility

The high flexibility of an image processing system usually facilitates adaptation of the system to technical modifications. Hence, flexibility is an important factor to cope with the challenges of a useful system, and this feature should be taken into account. In addition, it is also possible to investigate flexibility of the algorithms used. Furthermore, flexibility can be used as a measure of the ability of a modeling technique.

The Robot's Future

In order to make agriculture a high-tech profession, robots should be entered into this field as a necessity. There are plenty of ripe tasks which require the aid of robots. Robotic technology is able to push any section of the field towards precision agriculture. The future of agriculture is connected to the future of robotics and its application in this context. Furthermore, a lot of challenges are ahead and many new issues should be taken into account.

Modern Farms and Intelligent Farmers

Farmers are a part of nature and we cannot imagine natural environments without farmers. Due to the increase in the world population, modern farming methods are necessary to increase and enhance production in every farming sector. Thus, technology has swept into soil development, soil management, pruning, seeding, harvesting, light management, pest control, etc. In order to monitor products, it is essential to know all the species which have been grown on the farm. During harvesting, it is of great importance to recognize plant species as well. The automatic recognition of plants is a demand of modern farms where time is a vital factor. To attain peak efficiency, farmers need to use advanced equipment and they tend to have remote access to the farms during day and night. Farmers always juggle a set of parameters such as weather, level of soil moisture and nutrient content. Ultimately, modern farming is a revolution in agriculture, and it is business. One important aspect of this farming business is to increase income sources, increase the earned money from the farm and also maximize production and profit. Moreover, intelligent tools are promising to have intelligent farmers on modern farms. Many robots can play the role of intelligent farmers and provide the tools and facilities that are needed to have modern farms.

1.3 Goals of the Dissertation

The main goal of this dissertation is to precisely address the challenging problems of automatic plant recognition systems in both controlled and uncontrolled environments as a means of challenging laboratory and natural environments and to present new ideas and approaches in this domain and related issues. We intend to focus not only on the current issues but also those of the future. The dissertation surveys the state-of-the-art, discusses various related challenging aspects, focuses on required systems in detail and addresses upcoming demands and technologies in this field. The aim of the dissertation is to establish new foundations for developing efficient and automatic plant recognition systems for both laboratory and natural environments by considering neglected factors and parameters. Moreover, the main topics located on the cutting edge of the state-of-the-art are addressed, from both the theoretical and practical points, which include: connecting human vision and computer power to make computer-based systems smarter, recognition of the artificial plant images and natural plant images, developing mobile real-time plant recognition systems to be used on farms and in outdoor environments.

Connecting human vision and computer power in the plant recognition field is studied in the sense of interpreting the current status as deep as possible. The knowledge about human vision and machine power will be the key factor for robust and precise classification, especially under natural outdoor conditions and in light of unexpected occurrences in the natural environment. Concretely, one goal is to classify different plant species as accurately as possible, and there should be a close relationship between human and computer vision to get the highest possible accuracy. However, it is not intended to create an exact simulation and model of the human vision system, and ultimately, this is out of the scope of this thesis.

To achieve the research aim, different modern systems and techniques are thoroughly researched. Additionally, it is important to cover gaps between human recognition systems and computer recognition systems. Obviously, the human recognition system is inherently probabilistic, therefore, it is intrinsically fallible. Although the chance of error can be low, there is no way to eliminate error completely even with human recognition as it depends on different factors such as level of health, perception, etc. Hence, our expectations are too great if we plan to design and operate systems with a zero occurrence of error. Although, the process of learning for a machine is still far away from this process for a human, more real-life applications of contemporary soft computing techniques have been recently proposed in different fields, and this dissertation is a part of the efforts to find efficient solutions to real-life problems.

Recognition of plants cannot be bound to artificial images, laboratory images or indoors images with specified backgrounds. Natural images are treated as special objects to be deeply studied in this thesis due to a huge need for plant recognition in complicated and cluttered natural outdoor environments. Up until now, there has been no plant identification and recognition system that can work properly under the presence of light intensity variations, changes in distances and weather conditions, changes of time, etc. because the environment is not fixed in most outdoor scenarios. The most reliable way to deal with plant recognition is to design a robust identification and classification module to be usable in different situations and conditions and build a system which provides suitable and correct prediction to cope with various changes and variations in the natural outdoor environment. Nevertheless, natural plants are really complex objects so that there is still no complete solution for such a recognition task. Therefore, the goal of this thesis is to get the problems of plant recognition solved in challenging environments.

Alternatively, there is a demand for this technology in many fields, such as agriculture, modern farming, the pharmaceutical industry, etc., and the lack of a natural plant recognition system is undeniable. We need to implement a real-time system which is capable of navigating through the natural environment for the classification of different plant species. The navigation can be done as an autonomous process or manual process by a user. Real-time systems usually have time constraints which are inevitably associated with hardware equipment and designed systems where the software part is so important. Navigating in harsh environments, like farms and forest, is also a challenge that we have to solve by building an appropriate robot. Finding a solution to this problem is recognized as necessary, and this will be considered as part of the thesis. The last focus of this thesis is the development of a semi-robot and robot for plant identification in the presence of many challenges.

1.4 Novel Contributions of the Dissertation

The dissertation provides eight different contributions with respect to the task of plant recognition, and we explain how to vanquish the main challenges for this task under different artificial and natural environments and build reliable systems. In addition, the contribution of a concept, which is the classification of complex images, is also shown. The main contributions are listed as follows.

1.4.1 Combined Feature Detection and Description in Plant Classification

First, it should be pointed out that the target of the thesis is to solve one important problem: plant classification; and this is fundamentally a recognition rather than detection task. However, many different approaches have been proposed for plant recognition, and matching techniques have been utilized as the basic concept of many approaches. With respect to matching processes, there are some additional considerations concerning the obtained amount of useful information, number of features, computational cost, timing and speed. In addition, the important point is to increase functionality and applicability of the current existing approaches. Regardless of the complexity of images, we would like to achieve better matching results, so we need to specify more important points and get richer information by considering the problem as a matching task. An increase in correctly matched points means a better performance of the matching approach. Each detection or description algorithm has its own specifications, properties, and characteristics. Some algorithms like the features from accelerated segment test (FAST) algorithm [9] and the HARRIS algorithm [10] can only be used as detection approaches. On the other hand, other approaches like the scale-invariant feature transform (SIFT) algorithm [11] and the speeded up robust features (SURF) algorithm [12] are useful for both detection and description purposes. Regardless of being artificial or natural images, we would like to use the potential of different detection and description algorithms as a part of the matching technique and overcome deficiencies and disadvantages of single algorithms by combining different detectors and descriptors. For instance, the FAST algorithm is usually considered as a fast detector, but it lacks in the description step. If we combine such a quick algorithm with a description algorithm, the final combined method benefits from high speed performance of the detection part, the FAST detector. Due to the aim of achieving the features fast, the combined method, the FAST-SIFT, is helpful. However we have to investigate other parameters and factors to know its performance compared with other methods. Furthermore, each detection algorithm uses a unique process to detect features, so detected features of different algorithms are different from each other and we are be able to explore new features and achieve completely new information from them. For instance, if we investigate two different algorithms, the FAST-SIFT as the combined method and the SIFT as the original one, we find out that the obtained information from one specific image is not the same as when using both methods and the image has individually been processed by each method.

Therefore, the work in this contribution deals with pure detection algorithms which cannot be used as description for recognition purposes and such algorithms are combined with description algorithms. Hence, the combined method can be used as the backbone of recognition systems and matching tasks by extracting information of the detected features. The combination of modern algorithms is not just applied to artificial images, the combined methods are also used for natural images, and we present this contribution to overcome challenges of recognition systems and we rely on the methods as a component of the systems. Recognizing the quality of the performance of the developed natural recognition systems using combined methods, they are able to cope with many different complex images even taken in harsh situations like windy weather. Consequently, the combined algorithms have been confirmed to be robust throughout the images in different and very challenging lighting conditions and in different scenarios based on used models in both laboratory and outdoor environments.

1.4.2 Classifying a Large Number of Plant Species

We would like to develop systems for plant recognition that are able to classify a large number of plant species. The plant recognition task is not only classifying plants or non-plants, it is ultimately classifying categories. One main goal is to cope with classification of different plant species. In this work, we develop systems for 32 different plant species. The implemented systems are based on different techniques, and we finally built six different systems for classifying the mentioned number of plants. Furthermore, we conduct various experiments and compare proposed systems from different aspects. Our experiments show that we obtain a good trade-off between different efficiency factors and that we would be able to choose one system according to our needs and purposes. It should be noted that all systems do the recognition task automatically.

1.4.3 Significant Improvement of Systems for Classification of a Large Number of Plants

After developing six systems for the recognition of a large number of plants, we would like to find a solution for improving the developed systems. Hence, we analyze the existing systems by investigating the used algorithms. In addition, we have a look into possible solutions to get impressive results. We propose a new foundation for modeling extracted data from input images and then do the training process. Two new systems are built based on the vector of locally aggregated descriptors (VLAD) technique [13]. We compare the new systems with similar systems of our state-of-the-art and observe higher accuracy for both new systems. The VLAD technique contributes to encoding images and extracted information in a more efficient way and results in an improvement of the systems.

1.4.4 General Natural Plant Recognition based on Modern Combined Algorithms

The final aim of this research is to achieve a plant classification mechanism which is not only meant to classify plants in controlled conditions like a laboratory, but also to advance the ability of natural plant classification beyond indoor and controlled environments. If a system is capable of identifying plants in abnormal areas, it brings many benefits to systems operating in outdoor environments for plant recognition. A lack of information is the main challenge in many recognition and detection tasks, especially the time that we are trying to extract information from natural images with the lowest amount of similarity and the highest amount of complexity. It should be pointed out that not all information obtained from natural images of plants is helpful. In such images, we usually find more useless information, and extracting information from a natural image is a critical part of the work.

In relation to natural images, human vision systems normally have the ability to differentiate between useful and useless information, but machine-based systems are largely not intelligent enough to decide whether data are useful or useless. Furthermore, we usually find large variations in different parameters and factors in natural images. For instance, a part of a natural scene might be dark in the presence of shadow and the other part might be affected by large amounts of light. As explained before, we apply combined modern detection and description algorithms for plant recognition. We can rely on such new methods and use them for developing and building systems to recognize plant species in natural environments. It was not predictable that modern combined algorithms would lead to good performance in natural plant recognition systems and that they could provide interesting and good performance under natural circumstances. If we divide our implemented systems into two groups, systems that are able to recognize plant images captured in controlled conditions and systems that can recognize natural plant images, we see that the good performance of modern combined methods is not limited to one of the groups and they show good performance for both types of images. This outcome is proof of the generality of the proposed combined methods.

To summarize, the classifiers generated by the proposed combined modern methods provide a range of accuracy between 90% and 94.94% for recognizing four different natural plant species, and the results are impressive with respect to complex natural images captured in outdoor environments.

1.4.5 Novel Natural Plant Recognition System Based on a Deep Learning Algorithm

One main challenge of a recognition task is to struggle with the features and types of them. In fact, we usually ask a question before starting the work and try to find the best answer for it. This question is "Which features can be used for fulfilling the recognition task?"

Finding a general algorithm that is able to detect features efficiently in any condition is very hard. In the natural environment, many different conditions such as changes of illumination, camera angles, background, crowded objects in scenes, etc. might occur, and feature extraction is impossible using only one generalized algorithm. Deep learning [14] is a new concept in today's neural network area which provides an opportunity to learn generic features, and it has changed the face of machine learning algorithms. In addition, it is a new generation for machine learning algorithms, and a multitude of researchers anticipate dealing with many unsolved challenges by using deep learning algorithms. Deep learning models are multi-layer neural networks with complex architecture for extracting very complicated information from input images. However, we could not design and develop deep neural networks without the advancement and availability of useful hardware like graphics processing units (GPUs) [15]. Our investigation proves that the most successful architecture in recognition tasks is convolutional neural networks (CNNs) [16] and we design a deep network with eight layers and plenty of parameters and our aim is to use its hidden potential and capabilities in attaining such an excellent performance in different aspects. Considering the process of the feature engineering in CNNs, less effort is needed for getting rich information. It leads to reduction of difficulties in some aspects like runtime, required domain expertise, etc. Hence, we would like to elaborate on plant recognition systems to apply advantages of deep convolutional neural network (CNN) which employs various filters. We develop a deep and powerful model to extract the general purpose features and perform the natural plant recognition task. In order to train the model, we use two different modes, central processing unit (CPU) [15] and graphics processing unit (GPU), and compare the training time as well. However, our priority was GPU-based training from the beginning.

Due to the uncontrolled conditions and difficult challenges in natural environments, we need to extract objective and rich information in any scene and in the presence of any harmful factor, therefore, better feature extraction is equal to more objective information. From our perspective, an increase or decrease of unwanted factors should not have any effect on the performance of the final system. Besides, we provide a system to visualize the deep model and final result of the testing input. Through various experiments, we demonstrate that the system has an accuracy of 99.5%, and there is a significant improvement in the recognition process and performance in comparison to previous systems for natural plant recognition.

1.4.6 Novelty of Dataset and Systems Implemented for Natural Plant Recognition

A key component of a computer vision task is the image, and the importance of the image is not limited to its digital information. Before entering the world of data, we need to have a deep look into images. Therefore, we investigate the role of the dataset for developing plant recognition systems and aim at the filling in of the missing gaps between the natural environment and artificial environment with regard to the desired recognition task. Our study has led to substantial progress in the research. Despite an increased interest in plant recognition, implemented systems are mostly able to recognize plants in controlled conditions, and they are based on datasets containing images taken in laboratories with only one single leaf or a homogeneous background with a single color.

Most of the plant datasets lack many natural factors and parameters, and we did not find a dataset consisting of all possible challenges in natural and uncontrolled environments. Lighting conditions and light intensity are important factors which affect the photographing process in natural environments. If we take two pictures of a plant in an outdoor environment, we are unable to get completely similar images because the light intensity and lighting conditions vary in the natural environment faster than our expectations. However, a professional photographer is probably able to take similar pictures by adjusting the camera's setting. Furthermore, illumination effects are undeniable in natural color images that are taken in outdoor environments. Although many approaches have been proposed for extracting illumination effects, we do not want to follow such approaches, and our aim is to use the whole capacity of an image as it is, without any additional pre-processing. In addition, a variety of image changes do not allow us to find a unique method for efficiently reducing illumination effects.

Let's imagine that we are going to walk through a farm in Germany on a sunny morning. We would like to test our plant recognition system and identify different plants and weeds on a farm. Someone might ask the question, "Can we do plant recognition if the weather becomes windy?" The demand is to develop such a system, which has the ability of plant recognition in different weather conditions. In the proceeding chapters we talk about the difficulties of weather conditions in detail. Our modern dataset consists of images taken in different weather conditions to achieve the goal of recognizing plants in these various circumstances.

One important challenge for plant recognition, which has been neglected is distance. The question is, "Is there any plant recognition system which performs the task at different distances?" Distance is a factor which affects recognition system performance, the distance being between camera and plant. Our goal is to build a system which is able to identify plant species at different distances like 25 cm, 50 cm, 150 cm and 200 cm. In fact, the system should not be dependent on the distance between camera and plant. This independence increases the efficiency of the plant recognition system. Hence, we take pictures from natural plants at 25 cm, 50 cm, 75 cm, 100 cm, 150 cm and 200 cm and this diversity of distances helps us to have a more useful system. It should be pointed out that a human is also not able to recognize the shape of leaf if he or she is far away from a plant, however, it is still very important to achieve a system with this ability. An increase in distance means that we do not have only one leaf within an image, there might even be a whole plant within the image captured. This goal is a significant challenge and finding a solution is a big jump for the next generation of plant recognition systems.

In random plant photography, it might be seen that there is no leaf within an image, and it is very hard to recognize a plant species without any specific shape of leaf. Meanwhile, it is extremely hard to distinguish plant species in a single image without any prior knowledge about the viewpoint and angle. Hence, the system should not have a viewpoint-dependent mechanism, and it should be completely viewpoint-invariant. In some cases, there is no visible shape of leaf within an image, and the human eye is also not able to find one single leaf in the scene. Furthermore, several leaves may be covered by each other, and there is no distinguishable view for a leaf to be identified. Due to important viewpoint characteristics of leaves, the modern dataset contains a variety of images with very different viewpoints, and each image has been taken at a random and unusual angle, so many different views can be observed among images of the modern dataset.

Complexity of images is not limited to mentioned factors, and the diversity of non-plant objects is often high in urban environments, hence, they bring more complications. In some non-urban environments, such as farms, the variety of objects is usually less than in urban environments. In such circumstances with plenty of objects, it is almost infeasible to interpret a complex scene with human knowledge and experience, and it is very hard to recognize plant species from far distances.

To address plant recognition problems, we identify the remaining points for the task and introduce a new modern dataset with many different challenges concerning different aspects, unique characteristics and large variations among images. To our best knowledge, there is no similar dataset available, and the modern dataset is unique. This dataset has been used for developing different natural plant recognition systems. Meanwhile, our experiments show good performances of the proposed systems, however, large variations exist.

1.4.7 Real-time, Mobile, Natural, Plant Recognition System

Although the natural plant recognition systems provide really a high accuracy in recognizing plant species, the applicability of deep natural plant recognition in real-time mobile systems, both semi-robot and robot, is a new goal and we would like to investigate such applicability in challenging tests. Although the deep system is limited due to its computational expensiveness, our purpose is to build a mobile system which is capable of plant recognition in outdoor environments in any challenging condition like lighting change, change of time, morning or evening, change of weather condition, etc. The fact is that the important limitations of mobile systems come mainly from hardware-related issues. Furthermore, navigating through different outdoor environments requires suitable and stable systems. Therefore, solutions have to be taken into account with regard to limitations. Although the training of the deep natural plant recognition system can be carried out by both CPU and GPU, it takes a very long time if we use the CPU mode. As we would like to have a mobile and real-time system, it is necessary to build two systems. The semi-robot one works and outputs the results in the GPU mode. The mobile robot, called Zephyr [17], uses the plant recognition system by using the CPU mode. This mobile robot can be used as an autonomous robot, though it is also controllable by using a joystick. In addition, the robot does not suffer from low speed while navigating in harsh environments. Additional details related to both systems are provided in Chapter 9.

In order to test the real-time mobile system, we chose to conduct the experiments in two different years, 2017 and 2018, days and times. Additionally, we used three different cameras including Samsung, iPhone 6s and Canon EOS 600D (details of the cameras will be provided later) therefore, new factors have been added to our previous challenge-factors and our investigation will be really close to reality in outdoor environments. Our final test shows an accuracy of 84.17% for 120 testing images [18].

1.4.8 Fully Automated Plant Recognition System

In this work, our attempt is to develop fully automated systems for the assigned classification task, plant species recognition, and the purpose is to eliminate human intervention. All systems involve classifying various plant species based on different feature extraction methods. Our goal is to automate the whole process and functions of plant recognition and replace humans by using appropriate hardware and software tools. In order to specify the type of such systems, we need to investigate the application of the systems and the nature of control. Due to the application area, the system can be applied industrially in agriculture, and it can be considered automated for plant recognition. For instance, in this research, we develop a system based on a deep learning model and the whole process from beginning to end is automatic. To increase its applicability, the plant recognition system and a mobile robot are connected, and an autonomous mobile robot obtains the ability of fully-automatic natural plant recognition.

1.5 Document Structure

In the section that follows, the organization of the remainder of the dissertation is outlined.

We begin Chapter 2 with a full discussion of the state-of-the-art and some fundamentals related to background knowledge for providing a better understanding of previous work. Particularly, we present the deficiencies of previous systems, and these guide us in creating new systems with different unique contributions.

Chapter 3 provides a survey of available plant datasets and introduces different types of datasets regarding the data within each. We investigate the images of each dataset and analyze it in terms of the shapes of leaves and photographing conditions. We present a study of information, which is provided in each dataset, and review the characteristics of the images of each set individually. In the presence of various dataset, we demonstrate their importance, which is elaborated upon in additional chapters as well.

Before starting the major parts of the work, an analysis of different types of images captured in various conditions, both controlled and uncontrolled, is undertaken. Chapter 4 covers the analysis of plant images, and these experiments are mostly histogram-based. Moreover, this analysis contributes to achieving a better understanding of the images that we are going to work on.

An investigation of the useful existing methods for classification tasks is the next step of the research. The idea is to use feature detection and description as the basis of the systems which we would like to implement for plant recognition. Chapter 5 studies different detection and description algorithms in detail. Afterwards, the focus turns to the problem of matching, as plant recognition can be inspired by this concept. In this chapter, novel modern combined detection and description methods are proposed, and they have been experimentally tested on several images. The outcomes of this chapter are completely analyzed to compare different possible approaches and to know if they are applicable for the classification of plant species.

Chapter 6 presents the work as a follow-up of using modern combined methods for developing plant recognition systems and classifying a large number of plants, totaling 32 different species. In this chapter, six systems are proposed which are actually based on modern combined detection and description algorithms and the bag of words (BoW) model [19]. The training phase of the classifiers is carried out by a support vector machine (SVM) [20] [21]. We show that traditional machine learning algorithms contribute to achieving good accuracy in the testing phase. To solve the plant recognition problem, one of the implemented systems can be selected with regard to the needs of, for instance, high accuracy, runtime, etc. The results show good system performance based on the modern methods of SURF, FAST-SURF and SIFT. The highest accuracy is obtained by doing detection and description with the SURF. The mechanisms of the proposed systems in Chapter 6 are comprehensively compared.

Chapter 7 considers the challenging problem of natural plant classification in outdoor environments. It begins with a discussion of the problems of natural environments and the novelty of working in outdoor environments where many external factors and parameters influence plant classification. If we consider natural plant recognition as a musical work, this chapter is actually a prologue. In this chapter, six different systems are proposed, and several experiments are conducted to compare the performance of the proposed systems.

Chapter 8 begins by introducing new machine learning algorithms and a deep study is provided. We propose a new and novel system based on deep neural networks for natural plant recognition. The final accuracy of the system is equal to 99.5% which is a large value without any doubt. This system can be used, for example, in different weather conditions, at different distances, during various times of the day or night, even if the shape of the plant's leaf is not clear to the human eye. The model is based on deep CNN, and a very useful system is provided for visualization of the system and its output during the test process.

An unsolved problem of plant recognition is to build a system which is mobile and operational in real-time. In order to design such a mobile system, we build two different systems, semi-robot and robot. Both systems are able to navigate through natural and uncontrolled environments for identifying plant species. Chapter 9 explains the entire process of building such systems and details different aspects of the problems encountered. The test shows that the accuracy of the experiments conducted in 2017 and 2018 is equal to 84.17%.

Chapter 10 concludes with a summary, a brief discussion of the whole work and its applications. Some unique ideas about the future directions for plant recognition systems are discussed.

Chapter 11, titled: "Appendices", adduces several concepts related to this work which may be helpful for readers to develop a better understanding of some of the relevant issues in the scope of this research.

1.6 Publications

Some parts of the research have been presented at a number of international conferences and published in international journals. The publications are listed below and grouped by the type of publication, international conference and international journal.

- Journal Articles

1. Fathi Kazerouni, Masoud and Mohammed Saeed, Nazeer T. and Kuhnert, Klaus-Dieter, Fully-automatic Natural Plant Recognition System Using Deep Neural Network for Dynamic Outdoor Environments, SN Applied Sciences, Springer Journal, 2019
2. Mohammed Saeed, Nazeer T. and Fathi Kazerouni, Masoud and Fathi, Madjid and Kuhnert, Klaus-Dieter, Robot Semantic Protocol (RoboSemProc) for Semantic Environment Description and Human-Robot Communication, International Journal of Social Robotics, Springer Journal, 2019
3. Fathi Kazerouni, Masoud and Schlemper, Jens and Kuhnert, Klaus-Dieter, Modern Detection and Description Methods for Natural Plants Recognition, International Journal of Computer, Electrical, Automation, Control and Information Engineering, Volume 10(8), 1497-1512 (2017)
4. Fathi Kazerouni, Masoud and Schlemper, Jens and Kuhnert, Klaus-Dieter, Efficient Modern Description Methods by Using SURF Algorithm for Recognition of Plant Species, Advances in Image and Video Processing, Volume 3(2), 10-24 (2015)
5. Fathi Kazerouni, Masoud and Schlemper, Jens and Kuhnert, Klaus-Dieter, Comparison of modern description methods for the recognition of 32 plant species, Signal & Image Processing, Volume 6(2), 1-13 (2015)

- Conference Papers

1. Fathi Kazerouni, Masoud and Mohammed Saeed, Nazeer T. and Kuhnert, Klaus-Dieter, Exploration of Autonomous Mobile Robots through Challenging Outdoor Environments for Natural Plant Recognition Using Deep Neural Network, 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP 2019), 2019
2. Fathi Kazerouni, Masoud and Schlemper, Jens and Kuhnert, Klaus-Dieter, Automatic Plant Recognition System for Challenging Natural Plant Species, Short Papers Proceedings of 25. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG 2017, 81-89 (2017)

Chapter 2

Literature Review and Fundamentals

As outlined in the first chapter, the investigation of plant recognition aims at identifying plant species in different environments, controlled and uncontrolled conditions, and developing mobile and autonomous systems to fully exploit these capabilities and accomplish challenging plant classification tasks. In order to present the context of the work in the domain of plant recognition and emphasize the state-of-the-art in this area, we would like to review the existing literature on plant classification and relevant work. Due to the diversity of proposed approaches and systems, we provide a brief literature overview and describe relevant and prominent state-of-the-art research related to the different parts of the work. In addition to explaining critical points, fundamentals are also reported briefly.

Over the last years various systems have been proposed for plant classification tasks which have been based on different techniques. In this chapter, we divide the existing systems into different categories based on the type of images, used components and type of systems as well as provide more details and information about them. In section 2.1, we first introduce the main types of plant species despite numerous types of plants all around the world. Section 2.2 describes the state-of-the-art in plant recognition and introduces various systems with different approaches and techniques for recognition and identification of plants. To provide initial knowledge about the literature on plant recognition systems based on neural networks, section 2.3 is provided. Furthermore, two examples of plant robots with recognition tasks are explained in section 2.3. This chapter is concluded by a summary in section 2.4.

2.1 Types of Plant Species

Although there are a large number of plant species all around the world, and they can survive climes from hot deserts in Africa to snowy mountains in different regions of the Himalayas, humans have attempted to categorize and separate plants according to particular characteristics, locations, usages, etc. As much as the plant world is familiar to us, we know surprisingly little about the plants around us and we lack some specific definitions for grouping plants. Many researchers consider simple characteristics like seed and flower as indicators for categorization. For instance, if we consider the existence of a flower as the discriminating factor, plants are divided into two different groups: flowering plants, such as the sunflower and orchid, and non-flowering plants, like mosses and ferns. The existence of a flower seems to be a simple concept for discriminating plants and categorizing them into two different groups. Another considerable factor for differentiating plants is seed. We can divide plant species into seed plants and seedless plants. Seedless plants are actually plant species that have no seeds. Seedless plants have some atypical properties, and they might lack in the ability to maintain and transport water. The existence of seed leads to sexual reproduction which contains two sets of chromosomes, but the reproduction of plants without a seed is based on spores and the process is usually done by one set of chromosomes. In fact, both methods are sexual reproduction, but reproduction of some seedless plants is asexual. As an example, reproduction can be done by leaves. In such plants, leaves fall off and new plants will be regenerated.

There are some other common types of plants and we try to group plant species according to these common factors, though it is hard to generalize the categorization process. Some common types are trees, vegetables and fruit bearing plants, grasses, shrubs and bushes, cacti, herbs, crops, annuals, biennials, perennials, creepers, climbers, bulbs, plants with tubers and tuberous roots, summer-flowering plants, spring-flowering plants, etc [22].

Let's consider trees, shrubs and bushes as common types of plants. We are able to divide plants into trees and shrubs where trees are taller than shrubs. In addition, trees can be divided into deciduous trees, ornamental trees, trees with flowers (angiosperms), trees without flowers (gymnosperms). Shrubs and bushes can also be grouped into flowering and non-flowering types. Most shrubs are the flowering type; and they are commonly used in home gardens and their ornamental value is high. It is worthwhile to mention that the importance of plants is not limited to material benefits and the green color of plants soothes the human soul, which can be considered as a spiritual benefit.

2.2 State-of-the-Art in Plant Recognition

Plant recognition is a huge task even for specialists and botanists. Botanists as experts for plants have developed a very precise system based on one hand perceivable features and on the other hand on evolutionary consideration that categorize all plants on the Earth. Herbarium is a traditional method of collecting different plant species in dried forms for further studies and research. In this way, plant parts, especially leaves, are mounted on exsiccatae to be used as reference. Furthermore, botanists have reference books which contain photos of the leaves of plants and information about scientific plant names, family, genus, etc. Two important books as references for plant identification are *Flora von Deutschland und angrenzender Länder: Ein Buch zum Bestimmen der wild wachsenden und häufig kultivierten Gefässpflanzen* [23] and *Computer-assisted storage and retrieval of the data of taxonomy and systematics* [24]. The first one is a proper destination book in the world of plants and corresponds to the requirements. The second one is also helpful by considering specimen data, ecological data and taxon-level information [24]. In addition, a review is provided by providing the details of seven projects to computer processing of taxonomic information [24]. Furthermore, a book was supposed to be the end of the road of plant identification which took 16 years to complete and was published in six volumes from 1984 till 2000 by Cambridge University Press. However, we cannot draw a border for plants with innumerable species. The title of the book is *The European Garden Flora* [25] and accurate information is provided for the manual identification of plants. Moreover, the subtitle of it is *manual for the identification of plants cultivated in Europe, both out-of-doors and under glass* [26]. This process is time-consuming, and there exists an undeniable demand to have a fast plant recognition process. Since the beginning of the 21st century, many approaches have been proposed to identify plants by using leaves to recognize plant species, and they are mostly based on images of leaves as botanists rely on these for plant identification. We would like to describe the progress of leaf and plant recognition in a timeline schematic.

In 2000, leaf image retrieval was proposed as the main key for plant identification [27]. The main focus was on the shape of leaves. The main problems are shape feature extraction and shape feature matching where the proposed solution is a two-step approach. The first step is to use centroid-contour distance curve which is a shape characterization function and utilize object eccentricity (or elongation) for leaf image retrieval. By doing proper normalizations, some useful features, scale, rotation and translation invariant will be achieved for the centroid-contour distance curve and the eccentricity of the image. In this step, leaf images are ranked by means of the eccentricity, and then another ranking process is performed by both centroid-contour distance curve and the eccentricity together. The second ranking part uses top scored images of the previous part. Reduction of the matching time is obtained by a thinning based method which locates start points for the matching process. It is worthy of note that there were plenty of techniques for shape representation at that time. Some important techniques, such as chain codes [28] [29], geometric moments [30], Fourier descriptors [31], shape signature [32] and matching method like [33] are proposed. In 2002, an interactive method was proposed for the recognition of flowers [34]. An important research project in this field was performed by Mokhtarian and Abbasi in [35] whereby they utilized curvature scale space images for representation of leaf shapes and then applied these to classify leaves with self-intersections. Saitoh et al. [36] worked on the recognition of blooming flowers and proposed a new method for extracting the boundary of blooming flowers. This method is based on selecting a route by minimizing a sum of the local cost divided by the route length [36]. The importance of this work is that they used digital pictures captured in the natural environment. However, the research has been neglected by many researchers in this field. After one year, a method based on shape features was proposed for leaf identification in [37]. An image segmentation process was performed to separate the leaf from the background, although the images of the leaves used were artificial images. The idea was to do classification of 20 different plant leaves by means of a moving center hypersphere (MCH) classifier [37]. Hence, eight geometric features and seven moment invariants were extracted and the final accuracy was about 92% [37].

Wu et al. [38] mixed up the plant recognition problem with artificial neural networks. They proposed a system based on neural network and domain-related visual features with three aspects of leaves which were shape, dent and vein. The shape features were actually slimness, roundness, solidity and moment invariants [30] features. Dent features consisted of cornerness, size, angle and sharpness features. The ramification and camber constituted the last group of features, vein features. As a result, the performance of the proposed system in [38] was satisfying after conducting experiments. By considering different size of training set, performance of the prototype system was examined and achieved accuracies were more than 92%. In [3], a new hypersphere classifier, called the moving median centers hypersphere (MMCH), was proposed for classification of 20 plant species. Digital morphology features, including geometrical features such as aspect ratio, rectangularity, area ratio of convexity, perimeter ratio of convexity, circularity, eccentricity, etc. [3] as well as invariable moment features, were extracted by the contours of plants' leaves. The findings showed that the storage space was saved and the classification accuracy was not sacrificed when the classification time was reduced.

Texture features have also been considered as useful information for implementing plant recognition systems. In 2010, Ehsanirad et al. [39] proposed a leaf recognition system for classification of plant species by extracting texture features using a gray-level co-occurrence matrix (GLCM) [40] [41] [42] and principal component analysis (PCA) [43] algorithms. The used dataset of this work was prepared by plucking fresh leaves from plants in fields and different rotations were applied in captured color images with simple and white background. They used 455 images for both training and testing dataset. Using the GLCM method and the PCA method led to the classification accuracy of 78% and 98% respectively. In 2010, two other works were published for the identification of plants. In order to classify medicinal plants, an approach was proposed by combining different features in [44] whereby color, edge and texture features were extracted and used for training two types of classifiers, SVM and the radial basis exact fit neural network (RBENN) [44]. This approach was used to classify three different classes: herbs, shrubs and trees. The accuracy of the proposed work with combined color and texture features was 90%, although the classification accuracy using just color features was 74% and it was 80% if edge texture features were used. In December 2010, a new work, entitled "Leaf Shape Identification Based Plant Biometrics" [45], was published and proposed a method which was applicable for the plants with broad flat leaves whereby the user should select some points including a base point of the sample leaf image and a few reference points on the blades of the sample leaf images. In fact, the user helps to extract the shape of the leaf. Afterwards, some extracted morphological features, such as eccentricity, area, perimeter, major axis, minor axis, equivalent diameter, convex area and extent [45] built a set of features. After the normalization of feature points by taking the ratio of slice lengths and leaf lengths (major axis), a probabilistic neural network was designed for the recognition task. In order to test the proposed work in [45], ten-fold cross-validation technique was used and the average accuracy was about 91.5%.

Up until 2011, proposed approaches for leaf classification were mostly based on global shape features. In [46], the most commonly used approach was based on local features and one modern algorithm, the SIFT algorithm, was used for this purpose. Furthermore, the shape features, which are global features, were added to local features and a weighted k-nearest neighbor (KNN) algorithm [47] was implemented for the classification task. In the area of plant classification, this work was a starting point for utilizing local detectors and descriptors. The final accuracy of the proposed approach was 91.30%. In 2011, two other works were proposed for plant recognition. In [48], a preferential image segmentation method was proposed for the automatic classification of leaves and flowers. In fact, this method used prior information and encoded it for preferential segmentation as a tree of shapes. This method's importance is related to it being invariant to translation, rotation and scale transformations. In the last work, Chaki et. al [49] implemented a plant recognition system based on two shape modeling techniques including the moments-invariant (M-I) model [30] and the centroid-radii (C-R) model [50]. Furthermore, an improved result was obtained by using a hybrid set of features involving both the M-I and C-R models. To demonstrate the work, a data set with 180 images for three classes was used. The final experiments showed a range of accuracies for the proposed models and techniques. This range was equal to [90%, 100%].

As an active area of research, others focused on the plant recognition in the year followed. In [51], the work was based on 12 digital morphological features (DMFs) [51] which were derived from 5 basic features. The minimization of the dimension of the input vector of the training model was carried out by the PCA method. Two different training methods, the SVM and KNN, were applied to two different datasets and the results were compared as well. It is worth considering that the proposed systems were mostly based on the extraction of shape features and relevant features. The system based on the SVM outperformed the system based on the KNN in both accuracy and execution time. When the used dataset is Flavia, the difference between the classification accuracies of the systems was 16.5% where the difference between the execution time of the systems was 2.9 seconds. Another proposed system for plant identification was based on a set of different information: color, shape volume and cell features. This was a semi-automatic system composed of three stages with regard to color index features, comparing shape features, cell features and volume fraction features [52]. The system was tested and evaluated on 1000 leaf and flower images. The final result showed that the recognition rate was up to 85% [52]. Due to the importance of plant recognition in medicine, a new system was proposed for the identification of medicinal plants [53], and the approach was also based on leaf features which were area, color histogram and edge histogram. An interesting work was proposed in [54] and the classification process was based in a random forest [55]. The used dataset consisted of scan photos, scan-like photos and natural images which made this work different from the previous systems. The procedure was to first categorize different types of images. Secondly, preprocessing approaches were implemented to correct shadow and background, removing petiole, and segmenting leaflet automatically. Then, an approach was created through the combination of shape, morphological and tooth (pixel on contour of leaf that has a high curvature [54]) features, and the extracted features were applied to a random forest classifier [54].

In 2013, two important systems for plant recognition were proposed in [56] and [57]. The first mentioned work used the potential of leaf contours and proposed a method for extracting features of leaf contours. The method relied on the lines between the centroid and each contour point on an image, and the distribution of distances in the leaf contour was shown by a length histogram. Furthermore, the used leaf images were taken in controlled conditions with white backgrounds. Different experiments were conducted for comparing the results. For instance, a scale invariance test was performed. In this test, the minimal value was equal to 0.98611 and the maximal value was 0.99992 where they considered 45 correlation coefficients. In [57], they had a look into different implemented systems for plant recognition and the approaches which were utilized for feature extraction. In addition, the types of extracted features were listed for some previous works.

In 2014, new works were proposed, and this research would like to investigate two important projects of that year. The focus of the first one [58] was on the automatic identification of medicinal plants and the second one [59] was based on using combined viewpoints for the purpose of plant classification. In [58], it was proven that extracted leaf features, such as leaf area, roundness, rectangularity, etc., were Gaussian distributed, and a weighted averaging technique was proposed to obtain an identity number for each plant. In the other work [59], a viewpoints combined classification method was proposed and a dense SIFT was applied to do detection and description steps. To represent the images with a high level descriptor, a Gaussian mixture model (GMM) [60] was utilized and the process was followed by a variation of the SVM. To show the results, they trained 7 classifiers for each viewpoint. The evaluation step was based on tow metrics, precision and runtime. The precision results were in the range of [0.314, 0.965] and the range of the runtime was [0.95, 1.82].

Before investigating proposed approaches and systems in 2015, we would like to introduce a paper which conducted experiments on Malaysian medicinal herbal plants and tried to answer an important question in the area of plant recognition: "Is Shape the Key Feature?" [61]. The extracted features were a fusion of shape, color, and texture which were based on the SIFT algorithm, color moments, and segmentation-based fractal texture analysis (SFTA) [62], respectively. It was proven that such a fusion of features outperformed the color or shape feature identification rate. In other words, the highest average identification rate was equal to 94% by using shape, color and texture features.

In [63], the authors proposed an automatic system and the first target was to segment the region of interest (ROI) [64] before extracting a set of shape features. To perform the classification task, weighted feature normalization [65], reduction of dimension by PCA, and SVM were used, and the final accuracy of the system on Flavia dataset [66] [67] was 87.40%.

Since 2017, more research has been proposed in the area of plant recognition, and different approaches have been utilized for classifying plant species. One of the first projects in 2017 was proposed by [68], where the used dataset was a classical one. Furthermore, a circle-based radii model [68] was proposed which was a new shape descriptor. The basis of the work was to consider the contour of the leaf and the center point and border of a circle inside in the contour. The goal was to extract 44 features for the training step which was performed by SVM. The accuracy of the proposed model was 93.33% for the shape descriptor [68]. In order to differentiate between types of leaves (lobed and unlobed) and classify simple and lobed simple leaves, a new research project [69] proposed a rotation and scaling invariant method by detecting changes between background and leaf (and vice versa) in binary images and used unlobed simple and lobed simple leaf features. Lee et al. [70] introduced a hybrid generic-organ convolutional neural network (HGO-CNN) [71], and it was used as a model for training in different mixtures of plant datasets. Interestingly, the authors mentioned that their model was not generalized enough for testing images.

Novel research in the area of plant recognition is yearly proposed and this is still a steady and growing trend. In order to identify the herbal plants, the texture features were extracted from the leaf images and the SVM classifier was experimented on using a set of herb plants which contained leaf images in a controlled environment. In January 2018, a general survey was provided in [72] and different references and approaches were compared by considering different aspects and parameters. Before starting a research on plants disease identification, it would be useful to have a look into the work in [72].

2.3 Short Literature Review for Plant Recognition Systems Based on Neural Networks and Plant Robots

Due to the importance of plant recognition systems based on neural networks and robots used for relevant agricultural tasks, we would like to have a short study on related literature in this section. It should be pointed out that more details concerning the previous research will also be provided in each chapter separately. Furthermore, the current research area is active as opposed to stagnant. Firstly, we look into several neural networks used for plant recognition and then we have a short glance at plant robots.

In 2007, one early neural network system for plant recognition was proposed [66] and a probabilistic neural network (PNN) [73] was employed to carry out the classification task. The extracted features in this work were 12 digital morphological leaf features orthogonalized into 5 principal variables to form the input of the PNN. In [74], a 3CCD camera was utilized for capturing color images from weeds and crops. Segmentation and image analysis operations were conducted on color images, and the process was then followed by a radial basis function neural network [74]. The final accuracy was approximately 80%. Rankothge et al. [75] developed a plant recognition system called the advanced plant identification system (APIS) [75], and the starting step of the approach consisted of removing noise, normalizing the leaf area, reducing the white background, and scaling the image of the leaf. They proposed an extraction step which consisted of color, shape and vein pattern extractions. It is worth mentioning that a rotational invariant was added by using a 2D-fast Fourier transform (2D-FFT) method [75]. However, the system was dependent on the quality of the images taken, and it was declared that the system needed less time compared to the manual identification of plants by experts. In 2013, an additional research project proposed the use of an artificial neural network (ANN) [76] for plant recognition and compared the results of two different classifiers, the KNN and ANN classifiers. Two different types of features, color and shape features, were extracted and utilized as inputs of the classifiers. In terms of execution time, the results showed that ANN was slower for smaller datasets and the performance of KNN was not good for a scaled dataset. In addition, they worked on a classical leaf dataset, Flavia dataset, and applied the proposed system to such a dataset.

Through the advances in control theory, the tendency toward using of automation in industry has been increased. Agricultural activities are mostly repetitive and dull tasks. A demand is to connect the robotic concepts to the agricultural needs and find high-tech solutions for the traditional activities. It contributes to reducing dependency on workers and performing tasks more precisely. One important aspect of automation is boosting productivity. In agriculture, productivity is one of the main components and no one can disconnect it from the related activities on farms. Nowadays, many investors and companies in different fields have been convinced to work on agricultural robots, although many projects are still in the prototype phase. Despite success in agricultural activities, the mentioned points have motivated researchers and owners of companies to develop new agricultural robots and benefit from advances in both mobile robots and computer vision systems.

In the practice of modern farming, digital farming plays an important role by covering different parts from sensors and data analysis to robots. The main purpose is to automate processes, especially in weed control, field scouting and harvesting. Concerning agricultural robots, several issues like human-robot collaboration and presence of robots in dynamic environments are highlighted. However, optimization of sensors, digitalization of information and applicability of multi-robots are also important for building the future of digital farms. Tractor is an important equipment in farms. Nowadays, autonomous tractors are needed for modern farms. Claas autonomous navigation [77] is an example of autonomous tractors which employs Cam Pilot steering and 3D computer vision [77]. It is featured with the Global Positioning System (GPS)-based control to follow on the ground [77]. The next example is John Deere iTEC Pro that applies global navigation satellite system to control steering and operate in rows of crops [77].

Furthermore, farming operations can be divided into some fundamental areas such as spraying, crop scouting, harvesting, sorting, pruning and weed control. The importance of weed control goes back to the economical benefits and increase of crops. One of the new developed machines is the Vibro Crop Intelli series presented by Kongskilde industries [78]. In addition to precision and ease-of-use, the system provides increase of profitability and efficiency of mechanical weed control in row crops. Hence, it is possible to recognize weeds and non-weeds easily.

Another important operation is to detect plant diseases. In [79], a mobile robot was built to overcome the existing challenges and there is a small, portable and reliable platform to check farms automatically, detect plant diseases and spray pesticides. The experiment was carried out on cotton as well as groundnut fields using different image sizes, 640×480 pixels and 1024×768 pixels. The name of the autonomous field robot in this work was eAGROBOT [79]. The authors deployed a disease detection approach which was proposed by Al-Hiary et al [80]. In 2016, a plant recognition system was developed by combining an oriented FAST and rotated BRIEF (ORB) [81] algorithm, a fast library for approximate nearest neighbours (FLANN) matcher [82] and a neural network, and the system was used by a robot [83]. In order to perform agricultural and gardening tasks, the robot was capable of measuring some main parameters for characteristics such as temperature, humidity, heat level, wind speed, wind direction and soil moisture [83]. Data acquisition was carried out by getting data from the on-board sensors of the gardening rover and the data was then sent to a cloud storage platform where it was prepared for future predictions in the garden. To have a remote control and monitor for the rover, a website and an android application were built and the internet of things (IoT) [83] was used for precision agricultural activities.

In addition to detection of plant diseases and recognition of weeds, there is another area that mobile robot technology can enter to increase the productivity of fields. This section is actually the plant recognition task on fields. It is an extension of weed control task and enables us to recognize various plant species on fields, not only recognizing weeds from non-weeds. Developing a mobile robot with this applicability is useful in gardens with various types of trees and fruits. In addition to performing the plant recognition task, such system contributes to accurate sorting of products and precise management of gardens for next crop year. The mentioned points motivate us to develop a mobile robot which is able to identify plant species in dynamic and outdoor environments.

2.4 Summary

Despite this chapter having introduced a range of different systems and approaches, the field of plant recognition still lacks a system which can be utilized in uncontrolled natural environments. In image processing, lighting variation is a challenge in many tasks such as object recognition, object detection, segmentation, etc. In addition, our final plant recognition story is not from a single object and we see different objects and a bunch of leaves within an image captured in a natural outdoor environment. A bunch might contain various types of leaves such as fresh leaf, old leaf, dried leaf, deformed leaf, etc. Furthermore, there is no routine for capturing images from natural plants, and this means that the images may be taken at different times (morning, noon, and evening) and on different days. Meanwhile, the position of the sun is not fixed, and the angle of the sun's radiation varies. In an outdoor environment, the weather condition changes daily, especially in Europe. For instance, the weather is cloudy in the morning and it becomes sunny at noon. The weather changes once again and it becomes windy. Changes of weather have effects on lighting conditions and might deform the shapes of leaves. These are only some challenges in the natural environment, and many proposed plant recognition systems try to segment leaves within the image or they need to have the whole shape of the leaf in the image. Our goal is to be able to classify plant species in difficult conditions, even if the image is blurry.

Within the literature reviewed, one important point is the dependency on the camera in the proposed systems. It is highly important to solve this problem and build a natural plant recognition system which can perform classification tasks without any consideration about the camera used. Such a system would be able to deploy ordinary cameras, such as mobile cameras. One important advantage of such a feature is the reduction of the overall cost of the whole system. Another important point is the distance between camera and plant. If the distance between camera and plant species in the natural environment increases, it is hard for both human and machine to recognize the shape of a leaf and the type of plant. It is also necessary to build a mobile system which is capable of classifying the plants as a real-time and fully automatic system. Therefore, for plant recognition, this research develops two different systems, semi-robot and robot, and both are able to navigate through the natural environment, including farms, to execute the desired task.

Although we have introduced different aspects of plant recognition systems found in the literature, various algorithms and approaches are proposed in this thesis, and the related state-of-the-art for each chapter is provided separately. Our purpose is to provide more details about each step of the proposed systems. It is worth mentioning that each chapter also introduces previous and related works as examples of the state-of-the-art.

Chapter 3

Datasets and Availability

In order to study plants and investigate plant recognition systems, it is mandatory to perform an accurate study and analysis of collected data in this field. There are some public plant and leaf datasets, and the availability of these datasets is helpful to consider desired goals at different levels and steps where each dataset has its own properties and characteristics. Similar to other classification tasks, it is important to select suitable datasets to obtain targeted results. Before introducing the available datasets, we have a look into plants by considering related issues in botany.

Botanists usually divide the plants into two main groups which are non-vascular (bryophytes) and vascular (tracheophytes). In general, early plants lacking in vascular tissues are members of the first group, which include liverworts, hornworts and mosses. Plants with vascular systems, such as Phylum Pteridophyta, Angiosperms and Gymnosperms, form the second group. It is worth mentioning that the main task of the vascular system is to transfer water and nutrients. Botanists introduce a plant species by establishing six different basic hierarchical levels: phyla, class, order, family, genus and species. However, this research does not focus on such information in plant recognition but rather the goal is to identify the class of the plant without providing additional information about its sub-levels.

In this chapter, some datasets will be introduced and explained briefly, furthermore the used dataset will be described in detail. Challenges of the used datasets will also be addressed in more detail. This chapter is grouped into three sections. The first section describes the classic datasets, the second section represents the semi-modern datasets and the third introduces the modern dataset in detail.

3.1 Classic Datasets

Most commonly, a dataset corresponds to the contents of a single type of data. Several classic datasets are publicly available for leaves of plants, and they have been used widely in the related literature.

3.1.1 One-hundred Species Plants Leaf Data Set

This dataset has been introduced in [84], and it comprises one hundred different minor/major species of the plant leaves. In fact, it is a planar binary shapes dataset and it has 1600 images in total. The dataset is available in the online UCI Machine Learning Repository [85], and each class has 16 images for its labeled species. Due to the number of images and classes, it is considered as a dataset with low example images in each class, although the number of classes is rather large. This dataset is useful for multi-class classification tasks when the training dataset is small. Figure 3.1 shows some samples of this dataset.

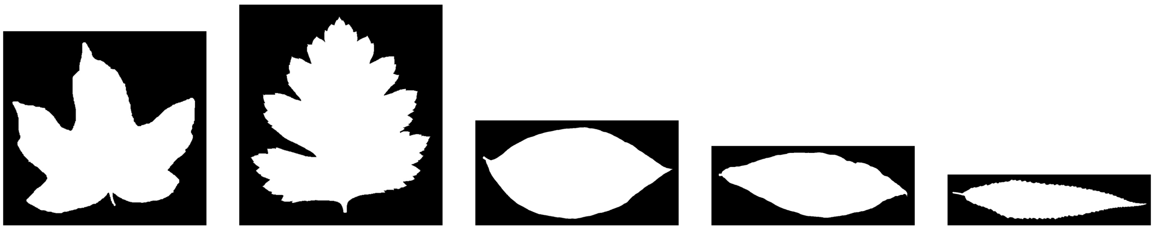


Figure 3.1: Several samples of the dataset from left to right, (a) Acer Campestre, (b) Crataegus Monogyna, (c) Magnolia Salicifolia, (d) Quercus Rhysophylla and (e) Salix Fragilis

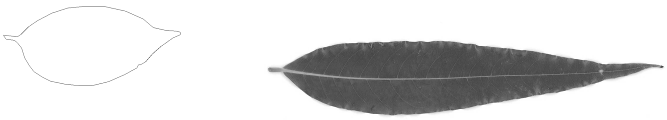


Figure 3.2: Two sample images of the Leaf Shapes database, Ground truth image- class C (Left), Sample image- class a (Right)

3.1.2 Leaf Shapes Database

This dataset has been developed for academic and research purposes, and the developer of the dataset is [86]. Leaves of 18 different plant species have been collected to create this dataset. It is worth noting that the number of images is not fixed for all classes, and it varies from one class to another one. Moreover, the name of each class has become unique by defining a special folder format. For instance, if the name of the folder is "cg1-cg10", it represents that the class is "c", the images are grayscale according to the "g" and the 10 shows that this folder contains 10 images of the class. Additionally, the format of the images is "tif" in this dataset. The current dataset can be applied in different areas such as shape analysis, shape feature representation, texture feature, contour analysis, contour-based image retrieval, leaf recognition, etc. One important point of this dataset is the availability of ground truth images. In Figure 3.2, two samples of this dataset have been shown.

3.1.3 Flavia Dataset

The Flavia leaf dataset is one of the most famous leaf datasets. In this dataset, there are only leaves of plants without stems. It contains leaf images of 32 different plant species. The number of leaves varies from one species to another, and it ranges from 50 to 77 images per species because of the difficulty of finding samples varies for each plant. The Flavia dataset is a collection of 1800 red-green-blue (RGB) images. The images have been recorded against a white background, and the resolution of the images is 1600 × 1200. Furthermore, no restriction has been taken into account while photographing. The sampling of the leaves has been done on the campus of Ninjing University and

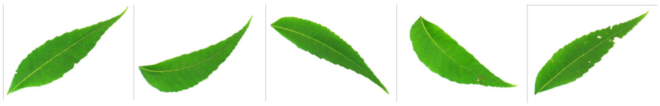


Figure 3.3: Five samples randomly selected from one class of the dataset

Sun Yat-Sen arboretum, Nanking, China. This dataset has several characteristics, for instance, the leaves in the images are not aligned and some of them have some rotations. Therefore, this dataset is closer to reality. Figure 3.3 represents several samples of one plant species which have been selected from the Flavia dataset. Figure 3.4 represents 32 different samples of all plant species where one leaf sample per species is shown.

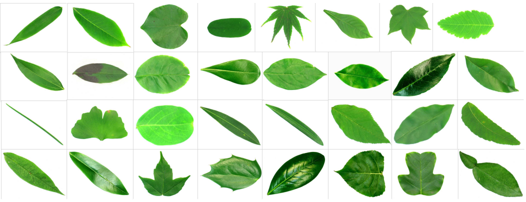


Figure 3.4: 32 different samples of all plant species of the dataset

3.1.4 Swedish Leaf Dataset

The Swedish leaf dataset [87] has 1125 images taken of 15 different plant species such as Ulmus carpinifolia, Acer, Salix sinerea, Betula pubescens, etc. There exist 75 images per each species. The leaves have been sampled from Swedish trees. In the images of the dataset, some of the leaves images have some parts of footstalks. In some cases, existence of footstalks has been considered as unsuitability, and some pre-processing methods like morphological operations have been carried out to remove the undesired parts. In order to have a robust leaf shape recognition [88], removing the footstalks has been performed as the pre-processing step. This dataset has also several characteristics. The leaves in the images are aligned very well to the background for taking pictures, and the alignment has manually been done. In addition to this alignment, the rotation is very small and it can be passed up. Another point associates with the way that the images have been taken. Actually, the images have been captured of one side of the leaves, not two sides of the leaves. The third characteristic is the good quality of the leaves in this dataset. There has been no serious partial loss in the leaves when the images have been taken. Figure 3.5 shows several sample images of the Swedish leaf dataset. They are three different species, and two sample images of each species are shown.

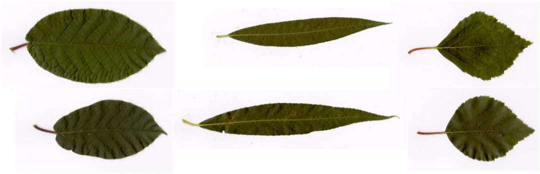


Figure 3.5: Several sample images of the Swedish leaf dataset

3.1.5 Smithsonian Leaf Dataset

In [89], 343 leaf images have been captured of 93 different species. Each image shows one isolated leaf, and the leaves have not been made flat during photographing of them. The number of leaves changes from one species to another.

3.2 Semi-Modern Datasets

Although identification of plant species might seem to be a simple task, it is a very difficult task even for botanists, scientists and professionals such as farmers, naturalists, foresters and nature exploiters. Semi-Modern datasets do not contain fully-natural plants images; however, it has been attempted to create datasets which are closer to the real world.

3.2.1 Pl@netleaf dataset

The Pl@netleaf dataset is a known dataset for plant recognition based on leaf images [90]. This dataset has been created in 2011, and it can be utilized for content-based plant identification tasks. Moreover, the dataset was used for the plant identification of Image Combined Lab Evaluation Forum (ImageCLEF) 2011. Seventyone tree species from the French Mediterranean area have been used to capture 5436 images and create the dataset. It contains three different types of images which are leaf scan images, leaf scan-like pictures with a uniform white background and free natural pictures, where they have 3070, 897 and 2469 images, respectively. The purpose of the third type of the images is to have natural conditions of different plant species which can be considered as a significant point of the dataset.

In addition, meta-data is created for each image, and each meta-data, xml file, stores the following information [90]:
- Date

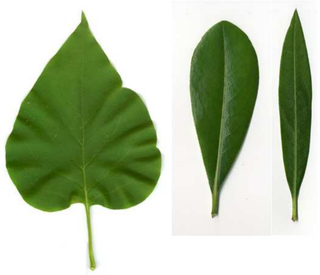


Figure 3.6: Three sample images with the scan type

- Acquisition type: scan, pseudoscan or photograph
- Content type: single leaf, single dead leaf or foliage (several leaves on the tree visible in the picture)
- Full taxon name (species, genus, family, etc.)
- French or English vernacular names (the common names)
- Name of the author of the picture
- Name of the organization of the author
- Locality name (a district or a country division or region)
- GPS coordinates of the observation

Figure 3.6, Figure 3.7 and Figure 3.8 represent three scanned images, three scan-like images and three free natural images, respectively.

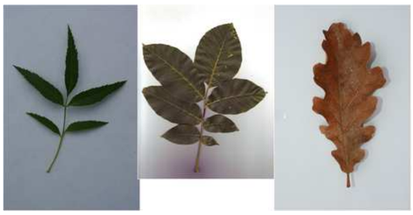


Figure 3.7: Three sample images with the scan-like type

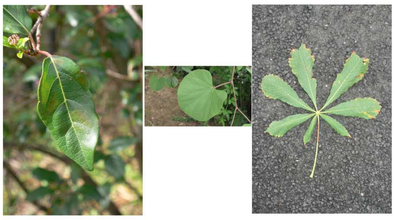


Figure 3.8: Three free natural pictures of the Pl@netleaf dataset

3.2.2 Pl@netleaf dataset II

The Pl@netleaf dataset II is an extension of the Pl@netleaf dataset described in section 3.2.1. The dataset contains images of 126 different tree species from French Mediterranean area. In comparison to the Pl@netleaf dataset [90], the number of images is increased to 11572 images. The images are divided into three groups, similarly as its previous version: leaf scan images (57% of total images), leaf scan-like pictures with a uniform white background (24% of total images) and free natural pictures (19% of total images). Meta-data is associated with each image. Furthermore, partial meta-data information can also be found in the image's EXIF, and it may include the following information:

- Model of the camera or the scanner
- Resolutions and dimensions of images
- Some optical parameter, the light measures, etc. for the photos

Besides, the images have been taken from distinct trees growing and living in distinct areas. Figure 3.9 represents localities of the plants included in the dataset.

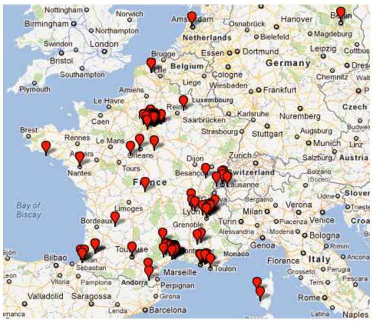


Figure 3.9: Locations of the plants included in the Pl@netleaf dataset II [90]

3.3 Modern Dataset

Ecosystems have attracted the attention of researchers and scientists. One noticeable dimension of the ecosystems is their high diversity. Accurate knowledge of ecosystems' evolutions results in the sustainable development of humanity and conservation of biodiversity. Therefore, it is necessary to have deep and accurate knowledge of ecosystems' components such as geographic distributions and living species. Plant species are one important member of the living species set. Moreover, identification of plant species plays a challenging role in ecological systems. One requirement is to have a natural dataset with outstanding records. This kind of datasets has been called modern datasets. A modern dataset is created as a part of this project, and it is called the natural plant dataset. It is also known as the modern plant dataset. This modern dataset contributes to making the pure computer vision knowledge closer to the real-world applications.

3.3.1 New Natural Plant Dataset

Despite the recent advances in the multimedia field, digital equipment, network bandwidth and information storage capacities, the absence of modern datasets for plants identification is beheld and it is needed to collect an outstanding set of records. It is vital to consider the requirements and necessities of the real world in terms of recognition systems. Due to the necessities of the modern life and the development of technology, modern data is needed to solve new problems. One modern dataset can evolve in terms of size, complexity, generality, etc. The proposed dataset, new natural plant dataset, is a modern one which is completely different from other available datasets due to its unique properties. It is also called the modern natural plant dataset (MNPD).

The investigation of the dataset shows that it contains color images taken of distinct plants with considerably different characteristics, percentage of homogeneous regions, details, etc [91]. Some points have been considered as general rules for preparing the dataset. In order to take pictures, similar protocols have not been used to acquire the images [91], and there is no special consideration about the camera selected. Consequently, there is no dependency on the used camera [91]. To have a useful natural dataset, different aspects or components of natural environments should be taken into account. Adding these continuum aspects mainly leads to creating and providing a logical and efficient collection of data to solve the problem and compensate for the lack of a modern natural dataset [91].

Distance

Distance is an important factor which has been neglected in other available datasets. To our best knowledge, it is the first time that this factor has been considered during the preparation of plant datasets. The distance is defined as the distance between the camera and the plant. This factor contributes to filling the gap between needs of the real-life and current systems for plant recognition. Images of the dataset have been captured at short and long distances of 25 cm, 50 cm, 75 cm, 100 cm, 150 cm and 200 cm.

It should be noted that the increase of the distance has effects on both the human eye and machine performance. For the human vision system, the increase of the distance between the human and the plant makes it difficult to correctly recognize the shape of the leaf with all details; therefore, the identification of the leaf shape will really be hard. In addition, the human eye is not able to distinguish the leaf shape in images captured at long distances, even if the human eye has already been trained for identifying the leaves of that plant species by observing previously different samples of the leaves of the plant species. Regardless of looking at a plant in natural environments at long distances or looking at a photo while the distance between the objects and the camera is long, it

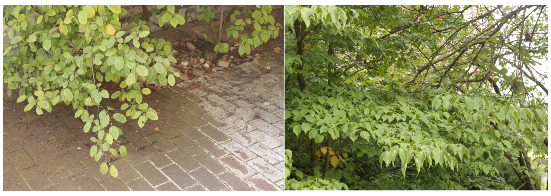


Figure 3.10: Two sample images of the natural plant dataset whereas the distance between the camera and the plant is 200 cm

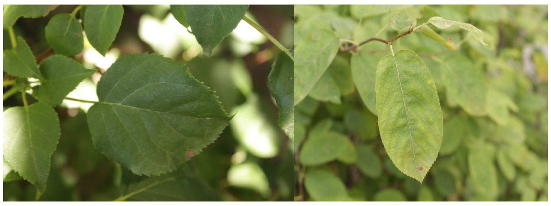


Figure 3.11: Change of the viewpoint at the distance of 25 cm

is really not easy to recognize the plant species and it is necessary to overcome this challenge in an appropriate way.

There is also another point about the distance. If the observer is looking at a scene with a bunch of leaves and several branches of the plant, it is relatively impossible to count the number of the leaves. Let's imagine that an image is taken at a long distance like 200 cm in the outdoor environment with different undesired objects which make the scene more complex, the difficulty is not bounded to the distance, and new challenges are added because of the environment and additional objects. In such case, the recognition process is not as simple as before, and a challenging factor is added to the whole task. As a result, adding this factor to the plant dataset is an apt change for making the dataset closer to natural outdoor environments. Figure 3.10 shows two images of the dataset taken at the distance of 200 cm.

This factor adds an important property to the future systems from a new aspect, and it helps to generalize through a fresh concept. It contributes to developing a plant recognition system independent of distance. This means that the system can be used at any distance in general. Furthermore, an efficient distance-independent system is more valuable for the real-life applications.

Change of Viewpoint

In order to evaluate the performance of a modern system, it is necessary to create a dataset containing images with significant amount of viewpoint variations. This modern dataset includes images taken at arbitrary viewpoints and different distances. At each defined distance, changes of the viewpoint among images are undeniable. Figure 3.11 represents two images with the change of the viewpoint at one distance. Although the size, color and clarity of different leaves are not the same and they vary from an image to another for one plant species in previous datasets, the leaves of the plant species are similar to each other. In addition, the lighting condition is usually kept fixed in many existing plant datasets and other factors like the point of view and the angle do not change during photographing.

As one of the goals is to develop a plant recognition system that can be utilized as a real-time system on farms, we cannot guarantee of having the same point of view or angle in all images. To fulfill this goal, it is necessary to increase the variety of the images by considering these two factors and take pictures at different points of view and angles. Our solution is to randomly capture images of the plants in natural environments and consider none of the mentioned factors. As a result, the final system will possess an angle-independent mechanism. Moreover, it will be able to recognize plant species correctly under naturally varying positional conditions.

Lighting Condition

The images have been taken under different illumination conditions. Some factors such as the position of the sun, positions and forms of clouds, overcast weather and rain cause illumination variations as they affect the other factors like shadows, location and position of shadows, color and light absorption. Illumination changes can be observed in this dataset and move it even more towards the real natural-world. Two of the first factors that usually come to photographer's mind are lighting condition and light intensity if the purpose is to take pictures in natural and outdoor environments.

Suppose that we take two pictures of a plant in the outdoor environment with the same other factors such as camera setting, angle, point of view, distance, etc. in two different sunny days, the pictures are not the same because both light intensity and lighting condition are different as the position of the sun, as the main source of light, affects the photographing process and the final pictures. One important effect of the position of the light source is the amount of shadow. Darkness in the uncontrolled outdoor environment and large variations of its amount affect the final natural color pictures as well. Figure 3.12 represents the images with different illuminations.

Basically, light is vital for taking pictures. Complex and natural plant images are taken with changing light intensity. Different types of natural light can produce a wide variety of appearances when an image is taken of a plant, even though the light source is the same. Light intensity may refer to the amount of available light for capturing photos. Light measurement has two main forms: the reflected light and the incident light. The direction of the light, such as side lighting, back-lighting and front-lighting, also affects the images. Due to lighting conditions of natural environments and the intention of generating different conditions, no special considerations about actively influencing the illumination have has been taken into account for capturing the images of the dataset to make it more natural. In addition, the attempt has been to strengthen the diversity of lighting conditions. In many datasets, we find that it has been tried to align and adjust the light source for obtaining uniform illumination, even with accurate calibration and proper alignment. Moreover, it is hard to keep light intensity fixed as light can vary in intensity by as much as 1000 times and the light intensity affects the quality of images. It is worth mentioning that the regulations in the human eye are more complex and include chemical processes in the retina.

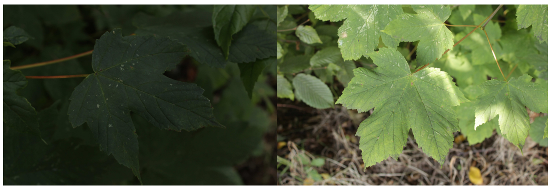


Figure 3.12: Two samples with different illuminations

Background

By considering different datasets, a factor that usually attracts our attention is the background. As previously explained, datasets usually contain images with homogeneous backgrounds (mostly white backgrounds) and the objects of the images are isolated leaves without petiole. In such laboratory conditions, leaf images are mainly taken with the same settings, parameters. Factors which might affect photographing are kept fixed and constant. It is worth mentioning that the diversity of images decreases in datasets with homogeneous backgrounds; however, the variety of the leaf shapes still remains.

Weather Condition

The images have been taken of complex scenes with various backgrounds in different weather types. In order to develop a general system, a new factor is added to this dataset, and it is actually weather condition (the type of the weather). More precisely, climate, weather, and wilderness can affect the performance of the plant recognition system. There are five different types of weather conditions listed as bad weather:

- Cloudy and unpredictable clouds
- Windy
- Rainy and drizzly weather
- Snowy
- Foggy

For instance, the contrast of the image is lower in the foggy weather. Small water droplets can cause light scattering and blocking, therefore other parameters such as the reaching light, contrast and visibility will be changed and reduced. In cloudy days, clouds absorb a part of the light and diffuse the rest; so there is usually no direct light on objects in natural environments, and it causes visual effects.

The mentioned weather conditions can be divided into two different classes according to their physical properties and visual effects. These classes are: steady (fog, mist and haze) and dynamic (rain and wind). Droplets of the steady class are too small, (1-10 μm), and they cannot be detected within the image if the distance between the camera and the plant is large, although they have effects on the recognition task. In comparison, the effects of the dynamic weather are much more complex. For example, wind can make leaves and their shapes indistinguishable. Furthermore, the rain produces sharp intensity edges and intensity variations in images, and it consists of small particles which are 1000 times larger in size, (0.1-10 mm), in comparison to the steady class. Besides, we may see dust on the leaves of the plants in natural uncontrolled environments, and the leaves might be covered by dust and leaf spot diseases affect the foliage of ornamentals and shade plants. These diseases cause damage the original appearance of leaves while they are clearer at short distances. This effective factor in the plant recognition process has been neglected in the existing datasets despite its importance. For instance, we would like to take pictures of plants when it is windy and there is no human intervention. In fact, the camera is in our hand and there is no unipod stabilizer available. In this weather condition, the camera may shake a lot while the leaves of the plants are moving too much. Consequently, the final images are blurry images in comparison to the sunny weather, and the clarity of leaves and plants reduces in captured images. In addition, we find that the number of deformed leaves increases in images.

Time of Photography

Time of photography is another factor which can be considered in developing a helpful and general dataset. As a consequence, the time of taking pictures has not been fixed to a certain time. The images have been taken at different times on different days. This factor certainly contributes to having a more realistic and natural dataset. Let's suppose that we keep the setting of the camera, distance and all other factors fixed and start taking pictures of one plant in the morning and evening; our purpose is to find the effect of the time on captured pictures. The investigation of the images taken of one specific plant while all factors and conditions have been kept fixed shows that the sun and the amount of the shadow have not been the same after a few hours. The final images are visually different, even for the human. Over time, the number of the dried and fresh leaves might be changed and the color of the leaves varies, even if the leaf shapes are not deformed. In fact, the dying leaves in yellow, brown or red colors contain low amount of chlorophyll. A fresh leaf absorbs most of the visible light and reflects a large amount of the near-infrared light, but more visible light and less near-infrared light are reflected by a non-fresh leaf. Therefore, photographing at different times adds additional challenges to the plant recognition process in uncontrolled natural conditions.

Selection of Camera

In order to take pictures for the dataset, there is no consideration about the model of the camera. The model of the camera is Canon EOS 600D.

Other Challenges, Random Photography and Environments

Random photography is the golden key for preparing a natural plant dataset. In the outdoor environment, many leaves of plants are covered by other leaves and the complete leaf shape cannot be distinguished easily. Even for the human eye, it is not easy to estimate and predict the hidden and invisible parts of the leaves. In addition, it is so hard for a computer-based machine to predict the leaf shapes and extract complete and enough information from the images taken in natural environments. A lack of information leads to wrong recognition of the plant species, and it is impossible to trust such plant recognition system. It is a desire to design a system that is able to identify types of plants in such complex uncontrolled environments. It is also necessary to have a look into different outdoor environments. Two main categories of outdoor environments are urban environments and non-urban environments. The variety of non-plant objects are often high in urban environments. Therefore, complexity of the images captured of the plants in urban environments is higher than in non-urban environments. It should be pointed out that the presence of different objects in urban environments and variations of natural backgrounds lead to very complex images taken of the scenes. It is almost impossible to interpret such complex scene with the human knowledge and experience; hence, it is also very hard to recognize plant species by using machines [91]. If we check the objects in non-urban environments such as farm, garden, etc., we find that the variety of objects is typically less than in urban environments. However, the simplicity of these environments in comparison with urban environments does not decrease the complexity of the plant recognition process and natural backgrounds vary in different captured images.

An Overview on the Selected Plant Species of the Modern Dataset

The selected plant species are common plants in Germany, especially in Siegerland. If we walk through this region, we find the mentioned plants in different places in both natural and urban environments. Due to the pre-defined goals for the future work and the importance of the plant recognition for the future of agriculture, the plant species have been chosen, and it is a need for having a natural modern dataset containing common plant species of the region and the country. Furthermore, different plant species had been checked before selecting these plant species. The investigation of the common plants during a period of time, several months, showed that Cornus is very sensitive. For example, a daily color change (from green to yellow) occurs in some parts of most leaves, which eventually results in an inevitable deformation of leaf shapes in this plant species. Furthermore, the leaves of the Cornus are similar to Amelanchier Canadensis if the number of rainy days decreases for a while. In addition, Acer Pseudoplatanus is usually exposed to diseases, and variations among its leaves increase when there is no adequate amount of rain in a period of time. For Acer Pseudoplatanus and Hydrangea, dead leaves can usually be found, even in rainy days in summer. At long distances, it is very difficult to distinguish Hydrangea from Acer Pseudoplatanus. Over time, the variation of the appearance of the leaf in Amelanchier Canadensis is usually high. Investigating a dataset containing these types of plants is an interesting challenge. The selected plant species add more challenges to the task of the natural plant recognition by considering different common plants of the region and create a task closer to that one we face in natural environments.

3.3.2 Summary of Modern Dataset

Capturing images in nature is so popular among the professional and amateur photographers and the tendency of the plant photography is also increasing as well. In addition to the industrial applications for the plant recognition systems, the possibility of identifying the plants in nature images is a demand for the today's world. In the nature photography, direct shooting of the plants in day time leads to obtaining very green images, and the images are not so artistic in this case. Professional photographers try to solve this problem by adjusting the camera settings and changing the source of the light. Furthermore, they also try to take pictures in closer distances to make more meaningful photos. Indeed, they are also able to do some post photographing operations and use some tools to achieve their goals of the nature capturing instead ruining the natural scene.

The motion of the leaves is an important factor which might happen when taking pictures. In many cases, photographers attempt to take photos during times when there is no wind and consequent leaf motion, but we just captured the images without any additional equipment for blowing the plants and making wind artificially or stopping the motion of the leaves. In addition to the motion, there is also another important factor that influences the natural plants during the photography. This factor is the background and different additional objects that we may see in the images of the plants. As we would like to make the system closer to the human abilities, we do not mind any consideration about the background. Meanwhile, when someone is taking pictures, they might think to capture pure images by using highly advanced cameras with fantastic lenses.

A professional-grade camera is usually expensive and, with such equipment, a photographer is able to take high quality pictures of the plants with very tiny details. However, we are sure that it is not possible to access highly equipped cameras for this research, and the type of the camera and its accessories should not have any impacts on our deep system. Confidently, we intend to reduce the distance between the human vision and the computer vision where the human has eyes and the computer and the robot have cameras. In addition, we did not use any sensor to change the nature and outdoor condition during photographing, and there is no specific protocol for taking the images.

During the preparation of the modern dataset, we did not try to put the leaves of plants in the center of scenes. Instead of this, we were eager to take the photos from the plants as they exist around us without any attempt to centralize the objects in the scene. Hence, the viewer's eye does not focus on one specific object. Furthermore, there is no consideration of positioning to benefit from this fact while taking the pictures of the plants, and different points of view will be provided.

The mentioned points are requirements of new datasets to fill in the gap between existing plant recognition systems and desired plant recognition systems for real-life applications. As an important part of the work, real challenges and difficulties of the plant recognition in uncontrolled conditions have been investigated from different aspects before preparing the dataset. In this section, the factors and challenges have been identified and introduced, and they can be considered as the starting point in developing the natural plant recognition system. Moreover, one important property of the modern dataset are large variations among its images. In fact, we need a dataset with natural images that have been taken at different angles, views, illuminations, light intensities, weather conditions, distances, positions of leaves, etc. Moreover, generalization of the dataset helps to build a recognition system which is capable of working in different situations in various environments with different conditions such as windy and cloudy weather. To our best knowledge, there is no similar dataset available, and existence of scenes with different objects and details makes the dataset very challenging.

Dividing the dataset into two sub-datasets, the training dataset and the testing one, is randomly done. Some factors such as non-uniform illuminations (shadows, underexposure and overexposure), background clutter and pose vary significantly among the images of the dataset, and such large range of variations in both the training and testing datasets is suitable to explore various aspects of the problem and to find an appropriate solution to overcome the challenges of recognizing plant species in natural environments [91]. Deep investigation of images of the dataset proves that images are affected by several factors, and there is mainly no focus on the effects of only one factor as there is not any control over environmental factors [91]. The aspects that will be added to future systems are responsiveness and usability in the real world, stability and robustness in difficult weather conditions, availability, adaptability, etc.

Chapter 4

Image Analysis

Image analysis is usually defined as a process of extracting quantitative and meaningful information or measurements from images by using computer approaches. The start point of such a process is getting an input image and the end point is getting an output in the form of numerical data while it is also possible to obtain the output in the form of an image if needed. Nowadays, a new meaning has also been added to the image analysis and the ability of computer-based algorithms for identifying visual information in an image is also considered as image analysis. Image analysis consists of many different simple and non-simple techniques which can be used for performing a wide range of tasks automatically. Special requirements for the image analysis are a computer with suitable additional devices and affordable equipment. It would be a good idea to have a look at a particular image and analyze it from a human's point of view. A human is able to categorize the objects within the image according to their types. Figure 4.1 shows a sample image of the modern dataset and a human can specify the types of objects including leaves, branches and the background. Not only recent advances in social media are based on text analysis but also text analysis is even applied to visual contents. Moreover, we have also had interesting advances in image analysis and related fields. The purpose of this chapter is introducing main concepts mentioned in the literature and investigating images of some plants using some typical approaches.

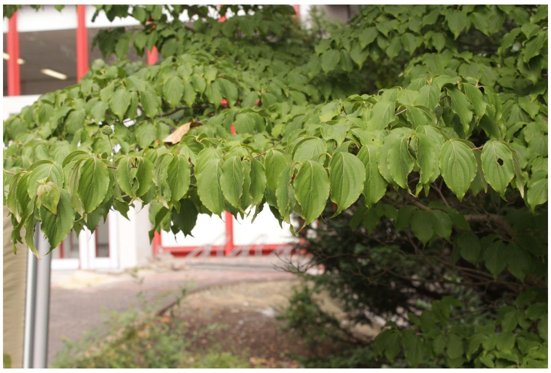


Figure 4.1: A sample image of the natural plant dataset (modern dataset) containing different objects with various types, leaves, branches and backgrounds

4.1 Investigation of Image Histograms

Image analysis contributes to getting basic details and extracting desired elements in a structured way. In this section, predominant intensities of some images by visualization through the histograms are investigated. An image histogram is usually a graph of the count of the number of pixels that are at a specific intensity. As we have different images of plants from the same species, we are able to compare them graphically by the histograms. It is worthwhile to mention that image histograms are nowadays available on many modern digital cameras [92]. They help photographers check the distribution of the captured images and find whether the details of the image have been lost to blown-out highlights and blacked-out shadows or not [92].

Let's consider two samples of one plant species and the plot histogram of these two samples, respectively. Figure 4.2 shows the first sample image and its histogram where the x-axis is the intensity value from 0 to 255 and the y-axis is the number of pixels with that intensity value.

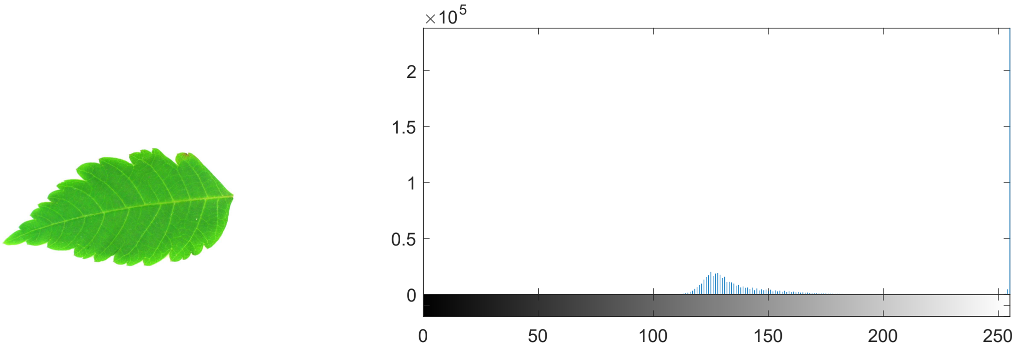


Figure 4.2: Left: First sample image Right: Histogram of the first sample image

Figure 4.3 represents the second sample image and its histogram. It can be noted that we firstly convert the RGB images into the grayscale with the following equation. Furthermore, converting color images into grayscale images can be performed using different methods which will be discussed in the next chapters.

$$Grayscale = 0.2989R + 0.5870G + 0.1140B \tag{4.1}$$

where R, G and B are red, green and blue components of one pixel.

As we see in the figures, both x-axis and y-axis vary in the histograms. Variation of values in y-axis depends on the number of the pixels in each image and how the intensities of the pixels are distributed.

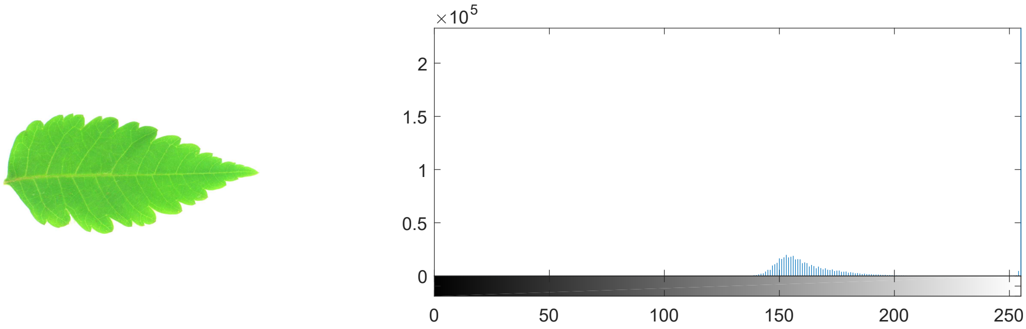


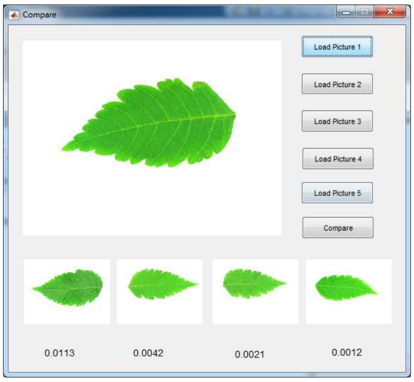
Figure 4.3: Left: Second sample image Right: Histogram of the second sample image

At this stage, a comparison is made between some samples of one species. We suppose one sample as the reference point for the comparison of four other samples. First of all, we change the range of the images and the new range is from 0 to 1 as a normalization step. Then, we convert our 5 images into grayscale. Afterwards, we create the normalized histograms. And finally, we calculate the histogram error between the first image and the second one using the following equation.

$$Error(1,2) = sum((NH1 - NH2)^2) \tag{4.2}$$

where normalized histograms of the first and the second samples are $NH1$ and $NH2$, respectively.

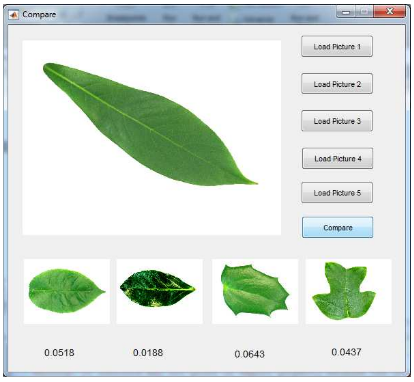
Figure 4.4 shows 5 sample images of the Flavia dataset and the obtained results after comparing the first sample with the other samples.



0.0113 0.0042 0.0021 0.0012

Figure 4.4: Five sample images of the Flavia dataset and the results obtained after comparing the first sample with the rest of the samples

Now, we select 5 images of various plant species randomly and compare the first selected plant species with the other four plant species. Figure 4.5 represents the results calculated by error. As one observes, the errors between the first plant species and other plant species are larger than the time when we investigated the errors among 5 samples of the same plant species. Unfortunately, we cannot find a general method for plant recognition by using the aforementioned comparing approach.



0.0518 0.0188 0.0643 0.0437

Figure 4.5: Images of 5 different plant species and the obtained results after comparing the first random selected plant species with the other plant species

4.2 Investigation of Histogram Equalization

In this section, the next step of the image analysis, the investigation of the histogram equalization, is discussed for some plant species. In order to automatically adjust image intensities, histogram equalization is a suitable technique and it enhances contrast within the original image. Furthermore, the histogram equalization transforms the intensity values and it results in matching of the histogram of the output image with a specified histogram. If the input image is shown as I and it is actually a matrix r by c with pixel intensities ranging from 0 to $L-1$, the normalized histogram of I with a bin for each possible intensity is the number of pixels with one intensity divided by the whole number of pixels. L usually equals 256 as we use images in grayscale.

$$p_x(i) = p(x = i) = \frac{n_i}{n} \tag{4.3}$$

n_i the number of occurrences of gray level where the gray level is equal to i.

Then the cumulative distribution function (CDF) [93] will be as follows and it will be the accumulated normalized histogram of the image, too.

$$CDF_x(i) = \sum_{j=0}^{i} p_x(j) \tag{4.4}$$

In order to produce a new image, called y, it is possible to create a function, called T, which transforms the image from x to the new image y and $y = T(x)$. Having a flat histogram, there is a linearized CDF across the value range. For instance, we have:

$$CDF_y(i) = iK \tag{4.5}$$

and K is constant and the properties of the CDF helps us do a transform process as below.

$$CDF_y(y') = CDF_y(T(K)) = CDF_y(K) \tag{4.6}$$

K lies in the range of [0,L] and T is responsible for mapping the levels into the range [0,L] where L is equal to 1 in our case as we utilized a normalized histogram of x. Figure 4.6 shows a sample image of the modern dataset.

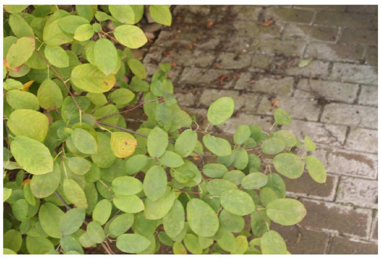


Figure 4.6: A sample image of the modern dataset

We calculate the histogram of the grayscale image where the number of bins used in the histogram are equal to 64 as the grayscale image is represented by 8 bits. Figure 4.7 represents the histogram of the grayscale image.

To adjust the contrast of the grayscale image, we use the histogram equalization technique and attempt to match a flat histogram with 64 bins. Figure 4.8 illustrates the contrast-adjusted image of the grayscale version and its new histogram. To see a bit better the details, it has been done as a

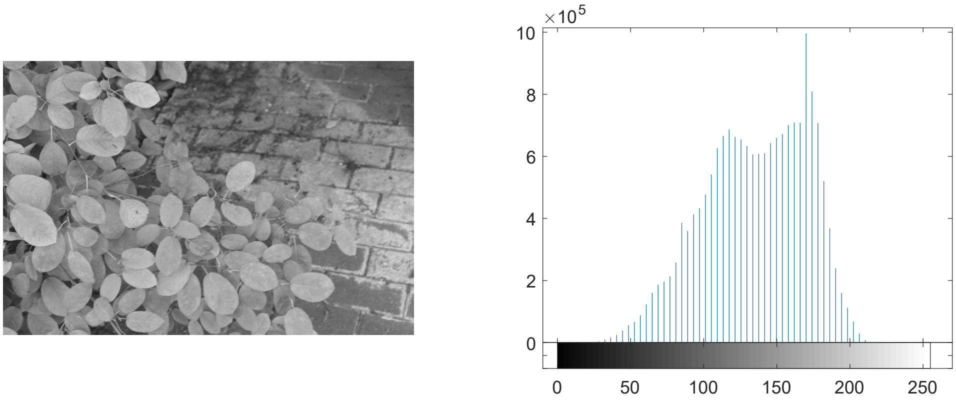


Figure 4.7: Grayscale image with its histogram

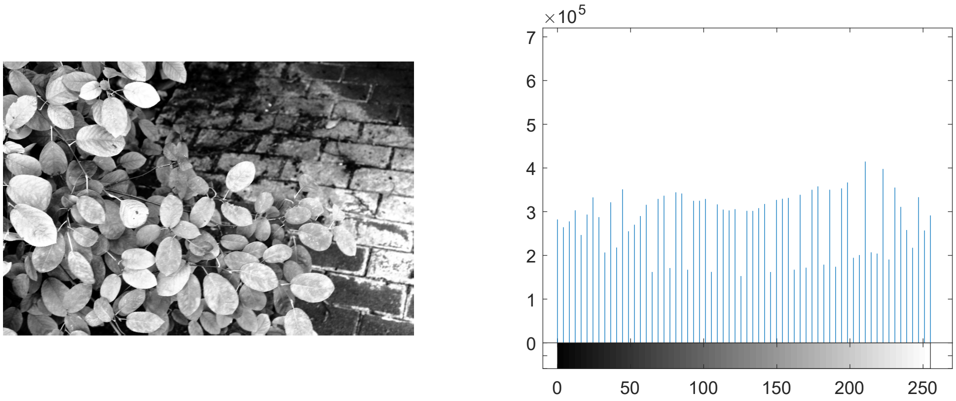


Figure 4.8: Contrast-adjusted image obtained from the grayscale version and its new histogram

first step.

As one can see, the histogram is completely changed and the distribution of the image is flatted in comparison to the normal histogram of the grayscale. The transformation return value is a vector that maps gray-levels in the intensity image, grayscale image, to gray levels in the histogram equalization. Figure 4.9 shows the plotted transformation curve. The input values are mostly in the range between 0.3 and 0.6, although the distribution of the output values is even in the range of [0,1].

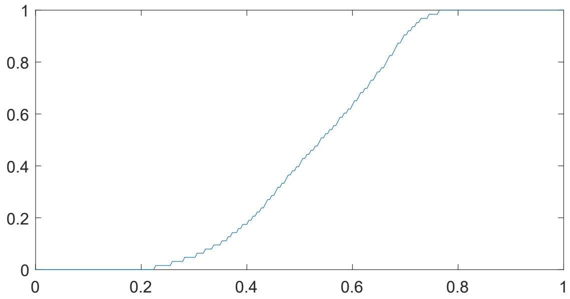


Figure 4.9: Plotted transformation curve

4.3 Channels of Image and Image Reconstruction

The present section is intended to have its focus on extracting channels of a color image and reconstructing the original image by means of the extracted channels. We use a repeatable and iterative procedure to get R, G, B channels from an input image and separate the results. Firstly, we split the color image to the green channel, the red channel and the blue one. Next, we create a black channel and then build a color version of each component individually. Figure 4.10 shows the input image which is a member of the modern dataset.

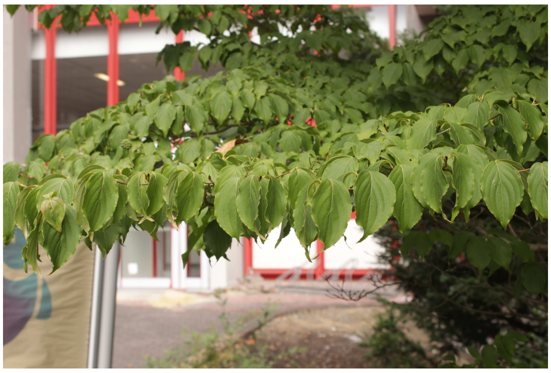


Figure 4.10: Original input image for extracting the channel

Figure 4.11 represents the histogram of red, green and blue channels while they have been extracted from the original input image.

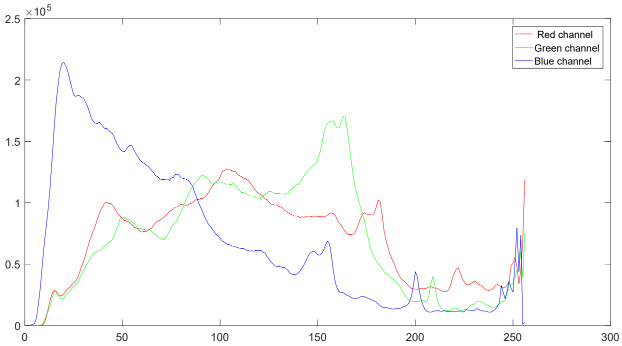


Figure 4.11: Histogram representation of the red, green and blue channels

After extracting the channels, we recombine the components in the sequence of red, green and blue and get the reconstructed version of the image. Figure 4.12 represents the red channel image, green channel image, blue channel image and reconstructed image.

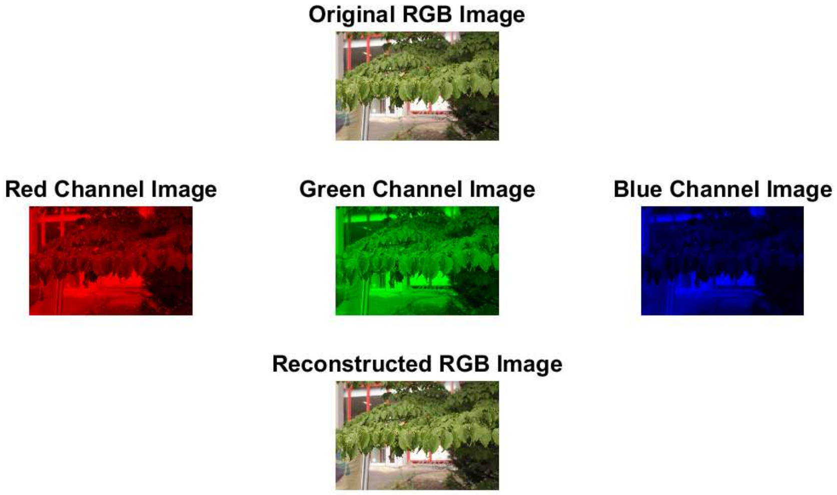


Figure 4.12: Representing the red channel image, green channel image, blue channel image and reconstructed image separately

4.4 Conclusion

An overview of the image analysis, comparison between some samples based on histogram and the information obtained from image analysis were presented in previous sections. We considered some samples of the modern dataset and other datasets used and took some simple and basic image analysis approaches to have a better understanding of plants images. Image analysis is not limited to getting information from different histogram-based approaches and it can also be used in other areas of computer vision. Importance of the image analysis has been extended to images with text contents as the number of images are increasing in social media and many users are interested in getting some specific information due to texts in images although the story is completely different in plant recognition in our point of view. One idea was to compare histograms of different samples of one plant species and compare plants of one specific family to know whether the extracted information of the histograms is useful for recognizing the plant species or not. Furthermore, we tried to find the relationships between one defined plant species and other plant species. But there was no way to discriminate between various plant species based on the information obtained merely from the histograms and generalized approaches. It is needed to do more operations so as to achieve rich suitable information for the plant recognition.

Chapter 5

Keypoint Detection, Feature Description and Matching

Feature is a basic concept with the same general sense in computer vision, image processing, pattern recognition and machine learning fields, although the complexity of this concept is undeniable, especially in image processing. In any particular case, this concept is highly dependent on a specific proposed problem. Therefore, collection and selection of features are important in computer vision systems. Each piece of image contains some pixels and information to solve the tasks related to the proposed problem, and subsets of features can be interpreted in different ways for the analysis of the image and its information. Features can be assumed as small particles of images. From this point of view, these particles can be utilized to represent images, and if efficiently detected and extracted, will be helpful for building useful systems and efficacious applications in the next steps.

Due to the rapid growth of using computer vision technology, computers are used to analyze images which are acquired by means of digital devices and image datasets. Basically, it is necessary to have a camera for building a simple computer vision system. However, such vision systems are not costly to develop through using personal computers with built-in camera and required interfaces. In order to analyze images in machine vision tasks and solve the basic computer vision problems, the matching process is particularly useful which can be done through various methods and algorithms. Being an important concept, matching is growingly used in different areas such as computer vision, computer graphics, photogrammetric and other applications of images. Matching does not merely focus on finding similarities in a group of images and includes some other important applications such as image alignment, 3D tracking, image registration, object recognition, motion tracking, robot navigation, etc. Furthermore, matching tasks can be accomplished by various algorithms proposed in [94] [95] [96].

The human eye can simply perform the matching based on some typical colors, textures, geometric distributions, characteristics of images, etc. Therefore, the idea of capturing such characteristics of images intuitively has been investigated by using different algorithms and techniques. In general, most of the available approaches in this regard are based on image processing or computer vision. The image processing-based detection exploits features of leaves by using different algorithms to obtain and examine a bunch of features. In this chapter, we are going to propose the combination of different algorithms and create combined methods for detection and extraction of features. A general review of the related works is presented in section 5.1. Firstly, we aim at investigating the capacity of each algorithm separately for our purposes (see more details in section 5.2). The nature of the image is an important factor which influences the performance of an implemented algorithm; for instance, illuminating conditions can significantly affect the results. As one important part of this chapter, we attempt to investigate the algorithms in the complex natural images, especially in the presence of dense edges and with variations in different parameters and factors. We need to use the whole potential of the algorithms in both detection and description tasks; hence, we propose the modern combined methods which can utilize the detection algorithms with different description algorithms (see section 5.3). Thus, we show the superior performance of the proposed combined methods in comparison with more conventional ones. First, a detection algorithm needs to be carried out in order to obtain the corresponding keypoints. Second, an algorithm should be used to complete the processes of extracting features and obtaining descriptors. Moreover, the proposed combined methods can be applied in different problems due to desired tasks as none of the approaches can separately lead to the desired results. A fundamental aspect of the object recognition task is matching, and it is possible to significantly improve it by choosing the best sets of features [97] [98]. Hence, a technique is studied and used experimentally in section 5.4. Moreover, the proposed combined methods for obtaining effective information for the computer vision tasks are described in section 5.5. Finally, section 5.6 draws the conclusion of the chapter.

5.1 Related Work

As mentioned above, keypoint detection and feature extraction are critical operations due to their undeniable roles and importance in computer vision tasks. The eye, as the main sensory organ of the visual system, is responsible to form an image in the human vision system. Despite the simplicity of the phenomenon at the first glance, the involvement of brain proves the complexity of the process. However, robot's eye and brain are not really comparable to the human systems beyond the abilities of the world's most powerful hardware and computers, the features are not trivial for the robots at all.

The variation of the viewpoints may lead to a complete change of objects and scenes, but the human brain and prior knowledge help to recognize objects correctly. Basically, it is impossible to assume one specific shape for leaves of one plant species. Moreover, leaves of one plant species do not possess only one color. Although the typical color of a leaf is green, it might be in other colors such as orange, red, reddish orange and yellow. These are two remarkable points at this stage.

In principle, a human is able to focus on the important parts of any image in the surrounding and natural environment. These parts could have unique perceptual significance or particular forms. The human eye is also capable of tracking changes in a natural scene which is a fundamental property of the human vision system. Due to these characteristics of the human eye, one question is posed; "Does the human eye detect and identify features in an image?" This question is the origin of other studies on biological nature of the human vision system. Hence, studies on both artificial and biological vision systems have been carried out to yield better understanding of features in an image. It is proven that any visual system, either artificial or biological, must simplify the image and record it in an economical 'token' form [99].

Therefore, it should be made clear that many publications regarding image processing and computer vision investigate plant/leaf detection by getting different information and using various methods [66] [100] [101] [102]. However, they usually define a set of features, use a limited number of features, and also show some specific species of plants but not plants in general.

As a consequence, these techniques are applied for specific and predefined species of plants and fixed photographing conditions, but in this work generality is always taken into account. Overall, all algorithms don't result in robust feature detection. Therefore, they cannot always be applied in the complex natural images to fulfill the goals. Moreover, there is no guarantee to detect desired information and obtain appropriate features. This leads researchers to come up with a solution rather than gathering only a limited set of features.

The literature on the plant recognition techniques reveals several studies using leaf as the main component in determining species of plants [103] [104] [105] [106]. In order to detect keypoints and extract features, there are various algorithms which can be applied to obtain results.

The SIFT algorithm is one important algorithm for the mentioned tasks. David G. Lowe introduced a more advanced approach leading to local image descriptors, invariant to translation, scaling and rotation. Furthermore, these extracted descriptors are partially invariant to illumination changes. Basic characteristics of this algorithm are useful in general. In [107], the SIFT algorithm, a powerful and modern algorithm, has been used in detecting objects under various imaging conditions.

Another modern and outstanding algorithm which is usable for detection of keypoints and extraction of features is the SURF algorithm. A real-time SURF-based system has been developed for traffic sign detection in [108]. These two main modern algorithms are the foundations of our work in this chapter. Each detection or extraction algorithm has its own characteristics and properties. Therefore, it is required to create new combined methods so that we can use altogether advantages of each separate algorithm in one new combined method. The idea of combining detection and extraction algorithms can be simply explained. Here, an image is passed through a detection algorithm to achieve keypoints. Then, an algorithm is performed to extract descriptors. The combined methods can be applied to different datasets, even natural datasets. It is also possible to compare different combined algorithms and compensate for their original drawbacks in the combination form in order to help for a better plant recognition system and enhance applicability of the algorithms for the final target. Some factors of the methods, such as efficiency, robustness and speed, are very important for plant recognition systems. The next phase of this chapter concerns the matching approaches. In [109], the motivation was to develop a shape detection scheme that can quickly assist vision systems on robot excavators to detect objects of interest based on shapes. Moreover, the aim was to develop a shape descriptor for a sampled boundary point of any shape. Object recognition can be acquired by matching features with a priori knowledge of the shape context of the boundary points in an object [109]. In order to identify the plants, a modified dynamic programming (MDP) algorithm [110] for shape matching is proposed in [110]. In [103], isolation of a leaf on a blank background can be performed by a user, and then the leaf shape is extracted by the system. Subsequently, the system matches the obtained leaf shape to the shape of leaves of known species. After several seconds, the top matching species along with textual descriptions and additional images, are represented. These are only some examples of using matching techniques for the plant recognition purposes. In order to find the constraints of possible methods and techniques and solve their drawbacks, it is intended to investigate the proposed combined methods and investigate the matching between different plant species.

In the present work, we are interested in approaches and methods which could be applied for detecting features of plants in both artificial and natural environments. In addition, we aim at comparing the similarity of plant species by means of matching techniques.

5.2 Keypoint Detection and Feature Extraction

As discussed, detecting keypoints and extracting features are extremely important issues for practical applications such as object recognition and 3D modeling. Even though many researchers attempted to find best detection and extraction algorithms for their systems, the results basically show that appropriate selection of algorithms is not enough and other parameters such as the cost of computation and running time are also important. In addition, the type of image and variations within the image are important to complete the detection and extraction tasks successfully by obtaining useful features. For instance, illumination changes cause a typical problem for precise feature matching tasks [111] [112] [113]. Another example refers to environmental conditions. If two images are taken of one plant species at different times and weather types, i.e. windy and cloudy, interest points

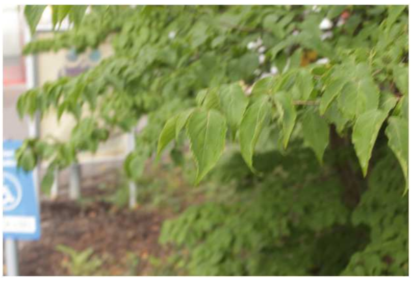


Figure 5.1: A sample image captured in windy condition

may be unstable due to vibrations of leaves as well as variations of unwanted illumination.

The human nervous system (see 11.2) systematically arranges the complex natural stimuli or filters them in order to extract behaviorally relevant cues. These cues have a high probability of being associated with the important objects or organisms in their environment, as opposed to irrelevant background or noise. In fact, the human nervous system helps the human eye concentrate and focus on interesting and significant parts of a photo or a scene. After a while, one remembers the significant parts of the photo or scene and not all the details. In computer vision and feature engineering, it is a demand to identify and store an image for any further process by means of a unique set of features. These features should be exclusive for each image and they should be able to distinguish between the original image and the other images. In order to obtain useful features, the focus is on the characterization of significant parts of the image as it can be an initial step for other parts and calculations [114] [115].

Feature detection algorithms could be divided into two different groups. In this way, we obtain different types of features. However, no clear-cut definition is available for the algorithms. These two main groups are the low-level feature detection algorithms and the high-level feature detection algorithms. Low-level features are the basic features that can automatically be extracted from an image without any shape information and they do not give any information about spatial relationships. Low-level feature detection algorithms are mostly concerned with finding corresponding points between images or finding interest points for classification as this concept is even remotely interesting at the lowest possible level like determining significant points of an image as well as finding edges, dots or lines in an image. Moreover, each pixel of the image has its own information, so some concepts like pixel intensities and colors can also be considered as low-level features. In the image processing domain, thresholding is a form of the low-level feature extraction performed as a point operation while the high-level feature detection algorithms are more in tune with how we classify objects in the real world. In the domain of machine learning, they are usually concerned with the interpretation or classification of the whole scene instead of finding only some significant parts or points, and they can be applied to body pose classification, classification of human's actions, object detection and recognition, etc.

It is worth mentioning that the low-level algorithms can be utilized in the high-level feature extraction approaches to do the desired tasks like large shapes detection in an image. In Figure 5.1, we attempt to add some details of the mentioned facts.

Observing the Figure 5.1, we identify it as "leaves with complex background consisting of a sign, a panel and soil on the surrounding." At the first glance, human identification is based on the whole image and scene without looking at the components separately or classifying the objects.

Algorithms such as the SIFT and the histogram of oriented gradients (HOG) [116] cannot perform similarly as does the human vision system, and only attempt to detect the local intensity variations such as keypoints, edges, etc. A meaningful classifier can be implemented to use the detected and extracted features. Obviously, there is a gap between two mentioned facts, and hence it is essential to minimize the distance between high level representations (interpreted by a human) and low-level features (performed by algorithms) to achieve a logical solution for object detection problems. As a result, a novel area, namely deep learning, has been created and affected the whole world of science, which will be discussed in detail later.

It should be alluded that the extracted information from the images is usually called features. Features are basically represented in terms of numerical values. Therefore, they cannot easily be understood or even correlated. Here, instead of using a bunch of images, the extracted information can be applied for further process. In other words, we reduce the data and attempt to have a new set of information in lower dimensions.

Features are sometimes called descriptors. Actually, features can be divided into two different types: global and local features, which can be applied based on the desired application. Global descriptors are usually used for image retrieval, object detection, and classification while local descriptors are used in the object recognition/identification tasks. Generalization of an entire object with a single vector is the main capability of global type, and therefore the knowledge of a shape is usable as a whole. Global features include contour representations, shape descriptors and texture features. Shape matrix, invariant moments [30] [117] [118], HOG and co-occurrence histograms of oriented gradients (CoHOG) [119] are only some examples of the global type descriptors. Hence, a rough segmentation of the desired object is possible by means of global features, and this advantage can be used for class discrimination [120] [121]. As the name of local features indicates, the bases of local feature are local properties like curvature. Computation of this type of features is performed in different parts and points of an image, and robustness is an important characteristic of local features. Most local features describe the small image patches (keypoints in the image). The SIFT, SURF, local binary patterns (LBP) [122] [123] [124], binary robust invariant scalable keypoints (BRISK) [125], maximally stable extremal regions (MSER) [126] [127] and fast retina keypoint (FREAK) [128] are some examples of the local descriptors.

Basic ideas and implementation steps behind some of the utilized feature detectors and descriptors will be explained in detail. A more extensive treatment of the algorithms, including comparison and usage guidelines, can be found in the following chapters while the algorithms are used as components of the implemented systems. Before moving to the next section, we attempt to clarify two important concepts which are detection and recognition. To explain the difference between these concepts, we assume to have an image containing different objects. In performing a detection task, we have to answer the question: is there an object of a certain class in the image present or not. Recognition means finding the identity of an object; for instance, the goal of a certain plant recognition is to find the species of the plant in the image.

5.3 Local Features: Detection and Description

In order to cope well with the changes and variations of the objects in images, it is not appropriate to use approaches which are based on consideration of the entire image. Partial occlusion, various changes of viewpoints and variations of distances in photographing are only three important properties of natural images in real outdoor environment. Hence, at first we need to pass up algorithms delivering the global features. One important point is to find the local invariant feature which allows performing an efficient matching between the local structures of images. In addition, translation, rotation, scaling and affine deformation are some examples of changes between different images. Therefore, it is necessary to find the invariant features which lead to representing them accurately and efficiently.

Efficient matching is one main goal of using local structures; hence, the aim is to obtain a sparse set of the local measurements identifying the images and capturing the right soul and nature of the

images. As a result, significant structures of images are encoded to use for further processes. The following points should be fulfilled to obtain the right features from the images:
1- The process of feature extraction should be repeatable, therefore the same features can be extracted from different images showing the same object.
2- The whole process of feature extraction should be precise and accurate.
3- The algorithm should lead to distinctive feature extraction. It means that the extracted features should be distinctive for different images; thus, a distinctive set of features can be used to represent a certain image.

As detection of keypoints and extraction of features are used like a chain to create a single algorithm, keypoints detection and features extraction are described in a specific manner instead of separating all components completely.

A general scheme of an algorithm can be explained as below:
- Finding a set of detected keypoints which are distinctive
- Defining a region around each detected keypoint in a scale- or affine-invariant way
- Extracting and normalizing the region content
- Computing a descriptor from the normalized region
- Matching the local descriptors

5.3.1 Localization of Keypoints

Keypoints detection is the first step in finding efficient features. Keypoints are reliable and applicable when it is possible to localize them under different varying conditions and situations such as viewpoint changes and presence of noise. It is obvious that transformations such as rotation and translation of an input image may occur. After any transformation, the same feature locations should be found in the extraction procedure. Although it is not easy to satisfy the criteria for all points in an image, increasing the probability of satisfying the criteria is helpful. For example, if a point is located in a uniform region, its motion cannot be determined exactly as it is not distinguishable from the neighbors. If we suppose a point is located on a line which is straight, its motion perpendicular to the line is measurable. Due to mentioned facts, the main motivation is to concentrate on points which are showing changes in two directions. Two important keypoints detectors which have been applied in the modern algorithms are discussed in the following. They are the Hessian affine detector and the HARRIS detector which are employed to find the desired regions.

Hessian Affine Region Detector

The Hessian affine region detector [129] belongs to a subclass of feature detectors which are known as affine-invariant detectors. Some other members of this subclass are MSER, Kadir-Brady saliency detector [130] [131], edge-based regions (EBR) [132] [133] and intensity-extrema-based regions (IBR) [134] [133]. The basis of this algorithm is the Hessian matrix which is the second derivative of the matrix. Therefore, this type of the detectors searches for the locations which show the strong derivatives in the image. In each image point, p, the Hessian matrix is as the following:

$$H(p) = \begin{bmatrix} I_{xx}(p,\sigma) & I_{xy}(p,\sigma) \\ I_{xy}(p,\sigma) & I_{yy}(p,\sigma) \end{bmatrix} \tag{5.1}$$

where σ is a scale. In addition, I_{xy} is, for instance, the second derivative in the x and y directions.

The procedure which is followed by the detector is as below:
1- Computing second derivative I_{xx}, I_{xy}, and I_{yy} for each image point
2- Finding the points where the following equation, the determinant of the Hessian, becomes maximal

in comparison with other points

$$det(H) = I_{xx}I_{yy} - I_{xy}^2 \tag{5.2}$$

The search is carried out over the whole image. A 3×3 window is used to estimate the derivatives. The used concept is non-maximum suppression. The pixel whose value is larger than the values of the other eight neighbors inside the window is preserved. Afterwards, the detector returned all the locations of the pixels whose values are higher than that of a predefined threshold which is called θ. In this procedure, the locations of the results are actually corners and textured image parts.

HARRIS Corner Detector

Human vision is able to recognize the edges as one of the low-level image features and obtain significant information, although there are other low-level features which can be detected in images to be applied in computer vision tasks.

The rate of change in the edge direction is considered curvature as an important feature. The rate of change gives new meanings to the points in a curve, and rapid variation of the edge direction is a proof that the points are corners. However, a small variation of the edge means that the points correspond to straight lines. As a result, it is feasible to reduce the data and use this significant information as an alternative in different tasks such as matching and shape description.

This is a mathematical approach which has been used in computer vision and image processing for detection of corners in images. This type of detectors has been named after Chris Harris. In [10], C. Harris and Mike Stephens proposed a combined corner and edge detector to improve one of the previous and old corner detection algorithms, e.g. Moravec's corner detection algorithm [135]. In 1977, Moravec defined the concept of interest points as distinct regions [136]. He was interested in finding distinct regions that could be used to register consecutive image frames [136]. This method is not invariant to rotation, and also has a low repeatability rate. The main concept of the Moravec's algorithm is to define points with low self-similarity as corners. The basic idea of the Moravec's corner detection operator is to measure curvature by considering the changes along a particular direction in the image. This algorithm checks each pixel of the image to find if there is a corner where it tests how similar a patch centered on the pixel is to nearby largely overlapping patches. In order to measure the similarity, the sum of squared differences (SSD) [137] has been utilized for the pixels of two patches. More similarity is shown when a lower number is achieved. This algorithm has its own problems, some of which are listed in [10]. When there is an edge whose direction differs from the direction of the neighbors, horizontally, vertically, or diagonally, the smallest SSD is large. Any edge is not basically an interest point and its selection as an interest point is not correct in this case. Therefore, the operator of the algorithm is non-isotropic. After 11 years, Harris and Stephens proposed a new method, the HARRIS detector, to improve limitations of the Moravec's algorithm.

In order to fulfill the disadvantages of the mentioned algorithm and solve its problems, Harris and Stephens implemented a new algorithm which is based on the differential of the corner score (local autocorrelation function [10]) with direct respect to direction. They proved its good consistency performance on natural imagery. As an intensity based method, the HARRIS detector is a good combined corner and edge detector which has roughly acceptable detection results and repetition rate. The repetition rate is an important factor for the detection algorithms.

In other words, this algorithm is based on intensity variation over all directions. Corners are the regions in the image with large variation in intensity in all directions. The simple idea behind the mathematical form is to find the difference in intensity for a displacement of (u, v) in all directions where the displacement in the x direction is u and the displacement in the y direction is v. The image intensities are denoted by I. In order to express the idea in mathematical form, the following equation is developed.

$$E(x,y) = \sum_{u,v} w(u,v)[I(x+u,y+v) - I(u,v)]^2 \tag{5.3}$$

The window function is $w(u,v)$ in the position (u,v) and works the same as a mask. This function is either a rectangular window or Gaussian window which gives weights to pixels underneath [138]. E is the difference between the original and the moved window and it is produced by a shift (x,y). $I(u,v)$ is the intensity of the original in the position (u,v) and $I(x+u,y+v)$ is the intensity of the moved window where the position changes to $(x+u,y+v)$. The purpose is to find the windows that are producing large E values. Therefore, it is needed to obtain high values in $w(u,v)[I(x+u,y+v) - I(u,v)]^2$. In order to maximize this term, it is possible to utilize Taylor series for expansion of the term as below.

$$I(x+u,y+v) \approx I(u,v) + I_x(u,v)x + I_y(u,v)y \tag{5.4}$$

Then:

$$E(x,y) \approx \sum_{u,v} w(u,v)[I_x(u,v)x + I_y(u,v)y]^2 \tag{5.5}$$

The term $I(x+u,y+v)$ changes to a new form by means of Taylor series which is infinite and contributes to approximating the equation. The approximated equation can be formed differently and tucked up into the matrix. The new generated form is as follows:

$$E(x,y) \approx \begin{bmatrix} x & y \end{bmatrix} \; M \begin{bmatrix} x \\ y \end{bmatrix} \tag{5.6}$$

The middle term in the above equation, which is called M, is equal to:

$$M = (\sum_u \sum_v w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}) \tag{5.7}$$

where Ix is the local image derivative in the x direction, Iy is local image derivative in the y direction, and $w(u,v)$ indicates a weighting window, rectangular or circular window, over the area (u,v). Moreover, the window function gives weights to the pixels. The state of response depends on the type of the used function. The response can be anisotropic or isotropic if the used function is either a box filter or a Gaussian filter. In the Moravec's algorithm, the easiest approximation is utilized for the window function. Inside and outside the window are defined as 1 (inside) and 0 (outside), respectively. In the isotropic case, the used window function is Gaussian and it delimits a circular window. The window function is valid for all directions and this specific function is as below:

$$w(x,y) = exp(-\frac{x^2+y^2}{2\sigma^2}) \tag{5.8}$$

M is the structure tensor which means the second moment matrix. Auto-correlation matrix is a very popular mathematical technique which is utilized in features detection methods. Thus this 2×2 symmetric auto-correlation matrix is used in the HARRIS method as above.

To apply a circular window, a Gaussian one should be used. In this case, the response is nearly isotropic and one of the problems of the Moravec's algorithm is solved, and the values are weighted more heavily near the center. Finding interest points is carried out by computation of eigenvalues of the mentioned matrix for each pixel. When both eigenvalues are large, it means that it is the location of a corner. In order to get a corner measure $C(x,y)$ for each pixel (x,y), the following equation,

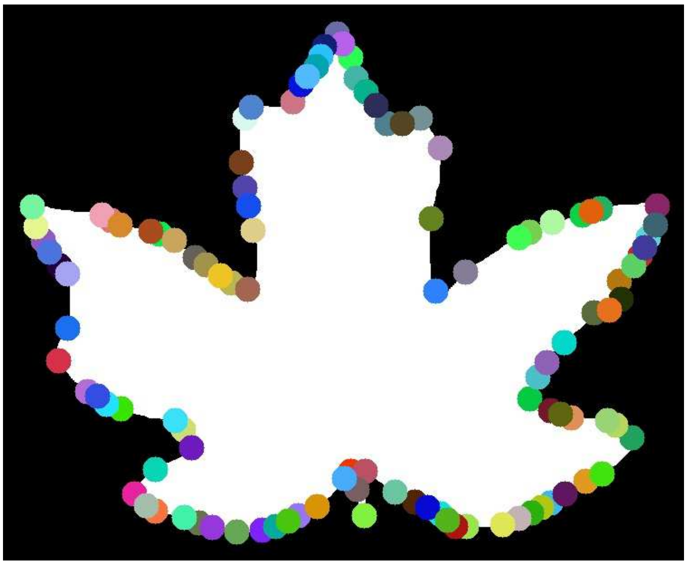


Figure 5.2: 110 detected keypoints in one sample leaf image using the HARRIS algorithm

namely the scoring function, is used:

$$C(x, y) = det(M) - K(trace(M))^2 \tag{5.9}$$

where

$$det(M) = \lambda_1 * \lambda_2, \quad \text{and} \quad trace(M) = \lambda_1 + \lambda_2 \tag{5.10}$$

K is used as an adjustment parameter and the eigenvalues of the auto-correlation matrix are λ_2, λ_2. Harris proposed to combine the eigenvalues in a single measure instead of two measures. Furthermore, the obtained eigenvalues make a decision on the status of the region. When λ_1 and λ_2 are small, $|C|$ is also small; then, the region is flat, and the windowed region has approximately a constant intensity. For example, the region is flat when there is only a slight change in C in any direction. When one eigenvalue is high and the other is low (λ_2» λ_1 or vice versa), C is less than zero and the region is an edge. In other words, the local auto correlation function is ridge-shaped and a slight change in C has been caused by local shifts in one direction along the ridge. Moreover, significant changes occur in the orthogonal direction. The last condition occurs when λ_1 and λ_2 are large. At these positions, the local auto correlation function peaks sharply and shifts in any direction causes an increase. In this condition, C is large and the region is one corner. The HARRIS detector is used as one of the detection algorithms in the implemented systems.

One important point of the algorithm is its wide usage for the corner detection in practice. Moreover, consistency, accuracy and speed are three important factors which should be taken into account for comparing performances of different algorithms. Further, it should be stated that the reduction of noise's impact has been obtained by using a Gaussian function $w(x, y)$, because the first-order directional differentials are sensitive to noise. Figure 5.2 represents a sample leaf image and its detected HARRIS features.

Furthermore, one important point is the sensitivity of the traditional HARRIS algorithm to noise and changes in image scale. Hence, the algorithm is not suitable for matching the images if the size of the images varies. In [139], it is explained that the HARRIS algorithm is rotationally invariant, partially invariant to affine change intensity and non-invariant to image scaling. Since the HARRIS algorithm is intrinsically an intensity based approach which attempts to find corners directly, it has been widely used in practice.

Shi-Tomasi Corner Detector

Someone might raise the question if such a detector, as the HARRIS corner detector, is really useful for many cases in detecting the corners in different species/types of plants and how the algorithm can be improved to detect the interest points in leaves of plants considerably. Intuitively, J. Shi and C. Tomasi proposed a new corner detector in June 1994 which was based on the HARRIS detector [140]. To answer the proposed question, a minor change in the HARRIS algorithm made it much better than before. The modified operator will run properly even when the HARRIS algorithm fails. The new algorithm is called good feature to track functions and the name reflects the concepts behind the algorithm. Certain assumptions can be taken into account to track corners due to their stability. Accordingly, direct computation of minimum between λ_1 and λ_2 can be performed. Therefore, the scoring function of the HARRIS algorithm is changed to the following equation in this algorithm:

$$C = min(\lambda_1, \lambda_2) \tag{5.11}$$

This value is compared to a threshold to find whether it is a corner or not. It means that if two eigenvalues are greater than a threshold, λ_{min}, a corner is found. This state has occurred in the green area of the next figure, Figure 5.3. The used procedure is shown in the next figure, where the x-axis is λ_1 and the y-axis is λ_2. In the blue and gray areas, one of the eigenvalues is less than the defined threshold and the other one satisfies the condition of being more than the threshold. These areas are edge parts of the image. In the red area, both eigenvalues are less than the defined threshold, making the area a flat one.

The performance of both algorithms, the HARRIS corner detector and Shi-Tomasi corner detector, can be compared for detecting keypoints in plants, especially for natural images where a part of an image can be wet soil and other materials with which reflection of the visible light changes and absorption of the reflected light from the sun varies in images.

This method has been performed on different images of the used datasets. The results open up new windows to select the appropriate algorithm for further steps of implementing the system. Figure 5.4 shows the detected features for two different images when the Shi-Tomasi method is applied and the threshold is fixed to 23. Consequently, the number of the detected features is 23.

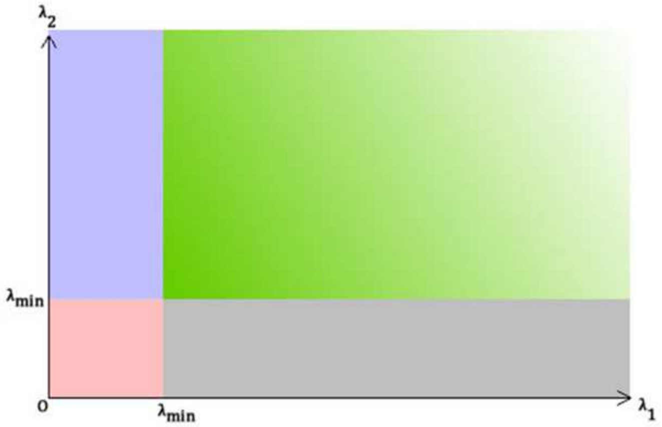


Figure 5.3: Comparing eigenvalues according to a threshold

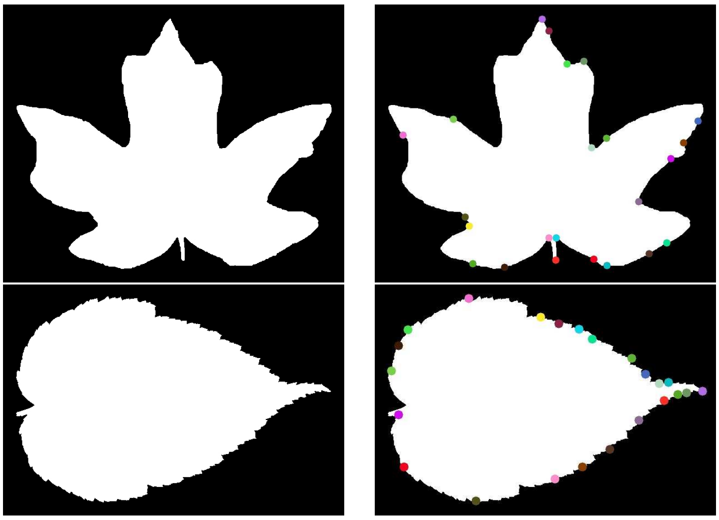


Figure 5.4: Original images (on the left side) and the results of the detected features (on the right side) when the Shi-Tomasi algorithm is used

FAST

The problem of speed is extremely important for building the practical real-time systems and applications. Even though a lot of researchers attempted to propose and implement efficient detection algorithms like the HARRIS, the results did not show good performance considering the algorithm's speed [141]. As it is an important task to identify correspondence of keypoints in images, one fast algorithm is very attractive for this very purpose to obtain keypoints at a high speed level.

By considering the deficiency of the HARRIS algorithm in speed, the FAST was proposed in 2006 by Rosten and Drummond [9]. It was intended to increase speed without sacrificing the quality of the detection procedure and to have high repeatable local information content. In principle, the machine learning concept has been added to change the flavor of corner detection to achieve a fast algorithm. One main advantage of this algorithm is its computational efficiency. To achieve a sufficient efficiency, the following main properties are required:

1- Fastness of the algorithm

2- Adequate repeatability

The FAST algorithm is an efficient corner detector based on comparing the pixels intensities. In this algorithm, a circle of 16 pixels surrounding the central pixel has been considered to identify corners. In fact, it compares pixels only on a circle of fixed radius (16 pixels) around the corner candidate point. Every circle's pixel is labeled from 1 to 16 clockwise. In this algorithm, all pixels are investigated and checked whether they can be desired and interest points. P is a pixel with Ip, which is assumed to be the intensity of the pixel. A threshold intensity value equals to 20% of the current pixel is set, which is called $Threshold$ in this study. Then, a circle of 16 pixels around P is considered for further procedure. This pixel is a corner if there exist n contiguous pixels in the surrounding circle which are lighter than $Ip+Threshold$ or darker than $Ip-Threshold$. n is defined as 12 in the original method. If the value is not defined less than 12, this method does not reject many candidates. A machine learning approach is applied to solve this weakness/failure. In the classification part, at least 12 continuous pixels must be darker or lighter than the central pixel, and then the central pixel is a corner. When the central pixel is classified as a corner pixel, it is not necessary to test all 16 pixels in cases of a non-corner pixel. Thus, the algorithm is quick and applicable when high speed is an essential factor. Consequently, it can be used in real-time applications.

To speed up the FAST algorithm, this procedure can test only four pixels at 1, 9, 5 and 13. Firstly, 1 and 9 are tested to examine whether they are highly bright or dark. Subsequently, we check the pixels at 5 and 13. If none of the pixels is the case, then, the pixels are not regarded as corners. This procedure will be continued for all the pixels of the image. Although this algorithm is faster than other corner detection methods, it is not robust to high levels of noise and there is a dependency on the mentioned threshold. Figure 5.5 shows the features detected in the two different images and the features are detected by using the FAST algorithm. The number of the detected points is also provided, as it is observable in the figure.

It may be true that the FAST algorithm is efficient due to its high speed, but it has one disadvantage. The main problem is the detection of a large number of corners. Natural images are complex and a large number of corners is detected which might include many noisy corners. This disadvantage arises from the basis of this algorithm because it is based on the intensity information of 16 surrounding pixels.

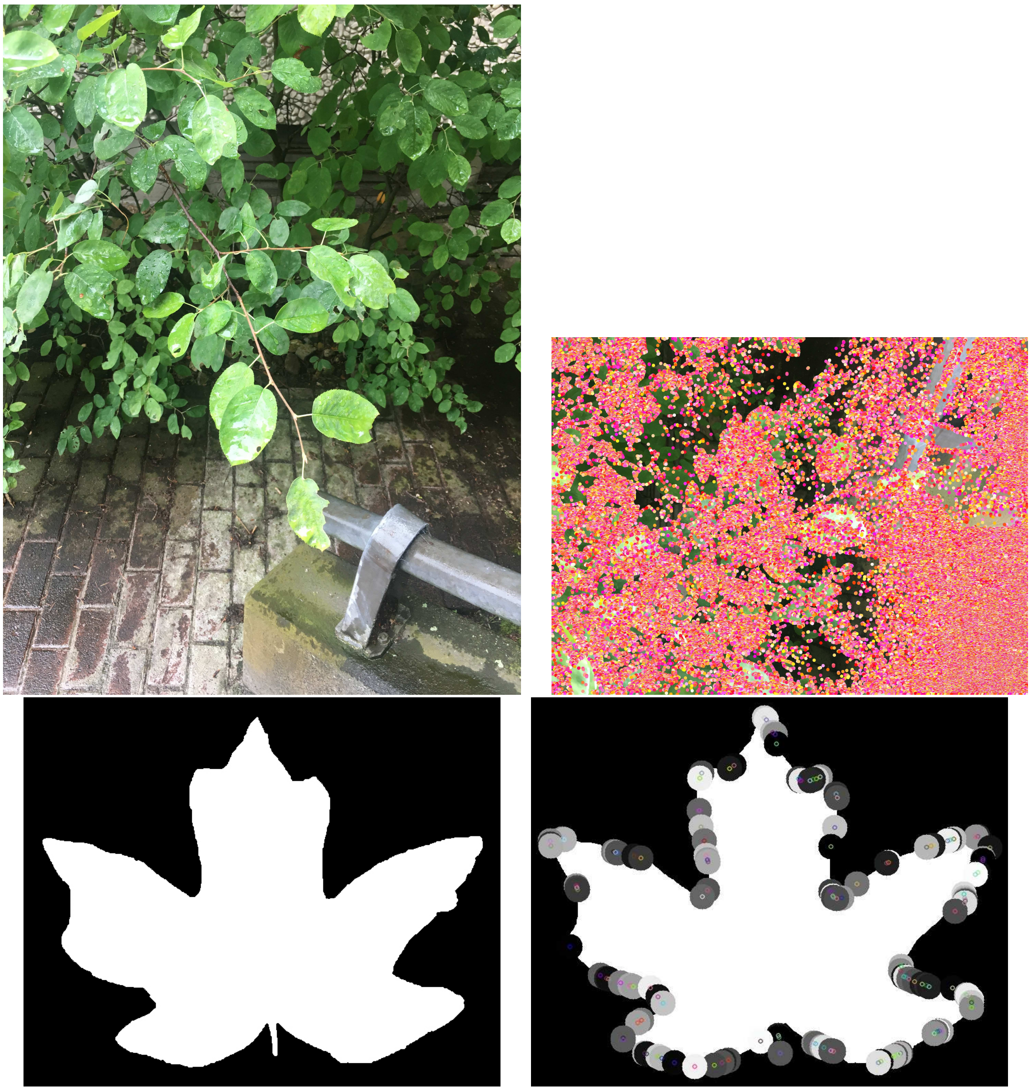


Figure 5.5: The features detected for two different images using the FAST algorithm (the number of the detected keypoints for the natural image: 114628, the number of the detected keypoints for the artificial image: 130)

Further Algorithms on Curvature

Many other important issues are available for the corner detection algorithms and each algorithm offers a different attribute with differing penalties. Förnster attempted to find the location of a corner with sub-pixel accuracy [142]. An ideal corner is assumed to be a single point that tangent lines cross. Hence, this algorithm uses a least-square solution to find the point closest to the tangent lines of the corner.

Although much more attention is paid to the edge detection algorithms in comparison to the corner detection algorithms, there are other popular works which have been devoted to corner detection algorithms such as smallest univalue segment assimilating nucleus (SUSAN) [143] and automatic synthesis of detectors [144].

In fact, SUSAN stands for the Smallest Univalue Segment Assimilating Nucleus with the same as the HARRIS corner detector which relies on the principle of intensity. It is a member of intensity based methods. Whereas other methods such as the proposed methods in [145] and [146] are contour based methods with different concepts. If the brightness of each pixel within a mask is compared with the brightness of that mask's nucleus, then, an area of the mask can be defined which has the same (or similar) brightness as the nucleus [147]. Jie Chen et al. [147] showed the overall superior performance from the HARRIS corner detector in comparison to the SUSAN corner detector on the whole.

In [148], Mokhtarian proposed a contour-based method for corner detection and indicated an extended curvature scale space corner detector. In order to represent shapes in different scales from coarse (low-level) to fine (detailed), another method was developed for curvature scale space by Mokhtarian [149]. In short, the orientation of each algorithm can affect the decision making to apply the proposed algorithm in the plant recognition system. To implement the desired systems with colorful characteristics, the investigation of algorithms is undeniable.

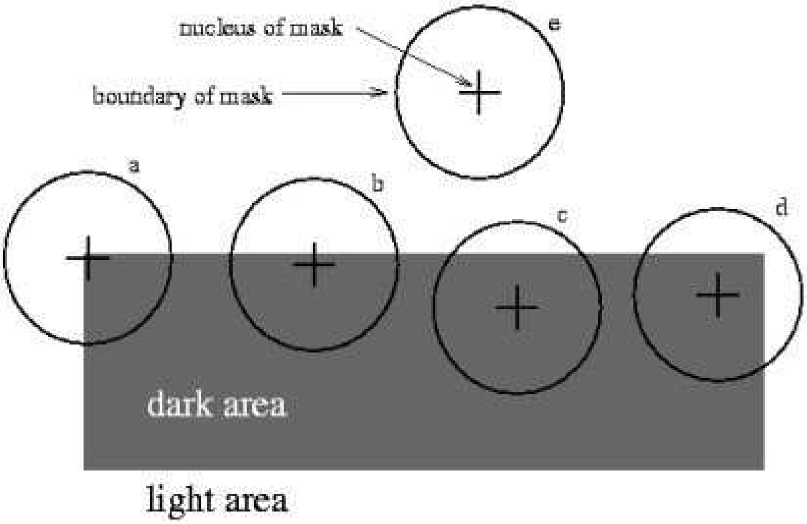


Figure 5.6: Four circular masks at different places on a simple image [147]

5.3.2 Modern Algorithms (Analysis of Region and Patch)

Perception of the human eyes contributes to recognizing different scenes and objects in images easily and storing some specific information of images' contents. In computer vision and image processing, it is needed to have algorithms which could work the same as this important visual element. After doing some computations for abstraction of image information and making local decisions on different points of the image, it is intended to encode the significant information of the collected features and interest points and gain meaningful feature vectors which differentiate one keypoint from another one in the image. In fact, information of an image is reduced to represent the image in a set of feature vectors and the obtained information can be availed for further desired tasks.

As discussed before, the importance of keypoints has some reasons behind it. Image transformations such as translation, rotation, illumination, change of scale, the variation of viewpoint, image blurring and added noise can cause practical effects on the original image. Selection of a proper technique leads to finding the same keypoints in both the original and modified images. The next step is the feature extraction where its obtained information depends on the feature detection algorithm. Traditionally, the term extraction refers to algorithms which extract local features and make them ready to pass to another processing step. Since feature detection is performed before feature extraction, feature description has the role of an intermediate stage between computer vision algorithms.

Descriptors are usable for summarizing some characteristics of keypoints. In order to have a successful description procedure, some considerations are required as below:
1- Position of keypoints should be ineffective. For instance, if translation occurs and the same keypoints are detected in different positions and pixels, the description algorithm should ensure the same outcome.
2- One important factor is the robustness against different image transformations and conditions. However, one description algorithm cannot be robust against all transformations. For instance, if two photos are taken of a leaf of the same plant in different time of day or at different weather conditions and types, sunny and cloudy, the leaf should be recognized similarly in the both cases.
3- Descriptors should be scale-independent. For instance, two images are supposed to be available and they show the same scene of a bunch of leaves. The image 1 is twice the size of the image 2. As previously discussed, similar keypoints should be extracted from two images, but the size of the keypoints in the image 2 is twice that of the first image. In the description step, similar descriptors should be assigned for the keypoints with different size. We consider prominent parts of the images. If the prominent part of one keypoint in the image 2 is a horizontal line of 20 pixels inside a circular area with a radius of 16 pixels and also the prominent part of the keypoint in the image 1 is a horizontal line of 10 pixels inside a circular area with radius of eight pixels, the same descriptor should be obtained by the description approach.

Someone may desire to know how to determine a technique by which the useful descriptors can be provided. We provide an answer for this question later when we explain and implement some modern description algorithms. Developing a solution that compensates for powerful algorithms is our main goal which we continuously consider in further steps to understand how to build desirable and efficient systems.

SIFT

In 2004, Lowe et al. [11] proposed a new algorithm to resolve relevant problems of practical applications at the low-level feature stage. This algorithm is useful and applicable for the both feature detection and feature extraction, and the nature of the algorithm classifies it in the group of local algorithms.

The SIFT algorithm, a popular, complicated and modern algorithm in computer vision applications, detects keypoints and extracts descriptors. Hence, the SIFT algorithm can be divided into two main stages which are feature detection and feature description. The SIFT detector extracts a collection of keypoints from an input image and then computes a histogram-based descriptor with 128 values for summarization of local image structures. For instance, the description stage can be connected to the use of the low-level features in object matching. Further, building Gaussian scale space, keypoint detection and localization, orientation assignment, and keypoint descriptor are the algorithm's steps [11].

It should be declared that low-level feature extraction within the SIFT approach selects salient features in a manner invariant to the image scale (feature size) and rotation, and with the partial invariance to change in illumination. Moreover, the formulation reduces the probability of poor extraction due to occlusion clutter and noise. It also shows how many of the techniques considered previously can be combined and capitalized on, to have a good effect.

Since we are not able to detect corners by applying the same windows for detection of keypoints with different scales, difference of Gaussian (DoG) [11] is used. The DoG is an approximation of Laplacian of Gaussian (LoG) [150]. By using this procedure, we actually perform a type of scale-space filtering. The DoG blurring of an image in various octaves is computed to obtain the DoG and the input image is repeatedly convolved with Gaussians for each octave of scale space to create a set of scale space images. Then, adjacent Gaussian images are subtracted to build the DoG images. Afterwards, a down-sampling by a factor of 2 is carried out on the Gaussian images, and the procedure is repeated. Finally, the convolved images are grouped by octave and we obtain a fixed number of DoG per octave.

The procedure can be reformulated using different words and mathematical forms. The method used for detecting the keypoints is based on a method proposed by Lindeberg in [152] for the scale-adaptive blob detection. In this systematic methodology, blobs with associated scale level can be detected from the scale space extrema of the scale-normalized Laplacian. In order to define a normalized Laplacian concerning the scale level in the scale space, the following calculation is performed:

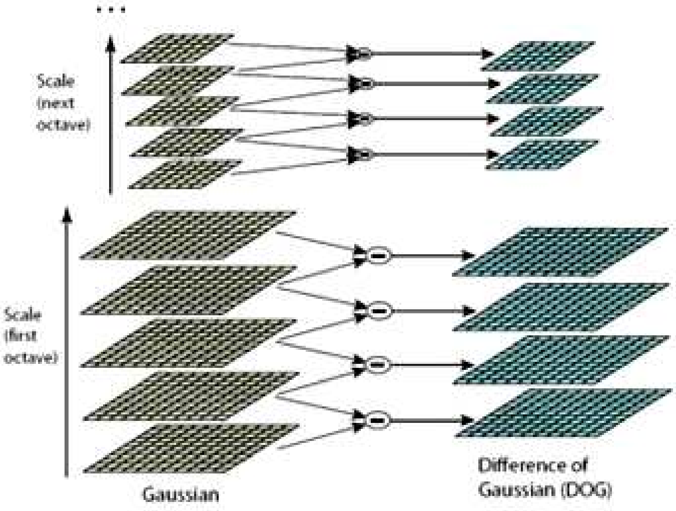


Figure 5.7: The octave of the scale space and the procedure [151]

$$\nabla^2_{norm}L(x,y;s) = s(L_{xx} + L_{yy}) = s(\frac{\partial^2 L}{\partial x^2} + \frac{\partial^2 L}{\partial y^2}) = s\nabla^2(G(x,y;s) * f(x,y)) \tag{5.12}$$

Calculation of the smoothed image values $L(x,y;s)$ is performed from the input image, $f(x,y)$, by convolving with Gaussian kernels of different widths $s = \sigma^2$. It should be pointed out that σ denotes the standard deviation and s is the variance of the Gaussian kernel; thus we have:

$$G(x,y;s) = \frac{1}{2\pi s} exp(\frac{-(x^2+y^2)}{2s}) \tag{5.13}$$

To find stable keypoint locations in the scale space, Lowe et al. [153] proposed a method based on scale space extrema in the DoG function convolved with the image, $DOG(x,y;s)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor.

Therefore, the scale space extrema are detected from the points $(x, y; s)$ in scale space, at which the scale-normalized Laplacian assumes local extrema with respect to space and scale. In a discrete setting, such comparisons are usually made in relation to all neighbors of a point in a $3 \times 3 \times 3$ neighborhood over space and scale, as it will be discussed later. The DoG operator forms an approximation of the Laplacian operator:

$$DoG(x, y; s) = L(x, y; s + \Delta s) - L(x, y; s) \approx \frac{\Delta s}{2} \nabla^2 L(x, y; s) \tag{5.14}$$

If we assume to have $\sigma_{i+1} = k\sigma^i$ as scale levels, then we have an approximation of the scale-normalized Laplacian with $\Delta s \nabla^2 L = (k^2 - 1)t\nabla^2 L = (k^2 - 1)\nabla^2_{norm} L$ which means:

$$DoG(x, y; s) \approx \frac{(k^2 - 1)}{2} \nabla^2_{norm} L(x, y; s) \tag{5.15}$$

It has been proven that the method leads to detecting scale invariance keypoints as:
- Preservation of keypoints under the scaling transformations [11].
- Amount of scaling affects the transformation, and the selected scale levels are transformed in correspondence with the amount of scaling [152].
- Since the Laplacian operation is rotationally invariant, keypoints are invariant if we consider rotation [154].

In order to localize the scale space extremum with a resolution higher than the sampling density over space and scale, both the DoG approach proposed by Lowe [11] and the Laplacian approach proposed by Lindeberg [152] involve the fitting of a quadratic polynomial to the magnitude values around each scale space extremum. Although it is a post-processing stage, it is highly important for increasing the accuracy of the scale estimates and fulfillment of the purpose of scale normalization.

Keypoint detection and localization constitute the next step of the SIFT algorithm. Without any doubt, candidates for keypoints are the local maxima or minima of DoG images. What is performed in this step is comparing the pixels in DoG images to neighbors; here, 26 neighbors in 3x3 regions in the current and adjacent scales of each pixel are taken into account and compared to the intended pixel. In the end, potential keypoints are the local maximum or minimum pixels and a filtering process is essential to obtain more accurate results. To solve the problem, a Taylor series expansion of scale space is utilized which leads to having a more accurate location of extrema. The process continues with comparing the intensity at the extremum to a contrast threshold (0.03). If this value is less than the threshold, it is automatically rejected for the next process. Since it is essential to remove edges as DoG has a higher response for them, a method similar to the HARRIS method is applied.

In fact, the Laplacian operator responds to the image structures that are like blobs and corners. By using the Laplacian operator, there might be another possibility and it also responds to edges. In this case, the operator is not suitable for the matching tasks. Therefore, Lowe found a solution by formulating a criterion in terms of the ratio between the eigenvalues of the Hessian matrix for suppression of these types of points [153] [11] where computation of the matrix is carried out at the position and the scale of the interest point.

$$H = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \tag{5.16}$$

In order to achieve efficient computations, it is feasible to reconsider the matrix in terms of the trace and the determinant; then, we have:

$$\frac{detH}{trace^2 H} = \frac{L_{xx}L_{yy} - L^2_{xy}}{(L_{xx} + L_{yy})^2} \geq \frac{r}{(r+1)^2} \tag{5.17}$$

Between the larger and the smaller eigenvalues, an upper limit on the ratio is permitted, being denoted by $r \geq 1$.

Basically, this 2×2 Hessian matrix (H) is applied to find curvature. In our implementation, an edge threshold is defined as a ratio. This threshold equals 10 in [11]. If it is greater than the threshold, the keypoint is removed and discarded. Therefore, strong keypoints are obtained.

In the next step of the method, the orientation of keypoints is determined by means of computing a gradient histogram in the neighborhood of the keypoints. This step contributes to achieving invariance to the image rotation. To this end, a 36-bin orientation histogram is produced, which is weighted by the gradient magnitude and a Gaussian window with a σ 1.5 times the scale of the keypoint. The highest peak of the histogram and other peaks with 80% of the highest peak are taken to compute the orientation. The outcome generates the keypoints with the similar location and scale, but in different directions.

The final step of the SIFT method is to compute the keypoint descriptors. For each keypoint, a 16×16 neighborhood is considered and divided into 16 sub-blocks. The size of each sub-block is also important and is defined to be 4×4. An 8-bin orientation histogram is built for each sub-block. Therefore, $4 \times 4 \times 8$ is the size vector which is equal to 128. The obtained vector represents the keypoint descriptor. Normalization is performed to enhance the invariance to changes in illumination and contrast invariance [155]. To avoid local high contrast measurements and reduce the influence of large gradient magnitudes, the normalization has been performed by thresholding the values in the unit feature vector so that each one is not larger than 0.2. Then, the values are renormalized to a unit length. This means that matching the magnitudes for large gradients is no longer as important, and that the distribution of orientations has greater emphasis. The value of 0.2 is determined experimentally using the images containing differing illuminations for the same 3D objects [11]. Figure 5.8 represents the formation of the keypoint descriptor.

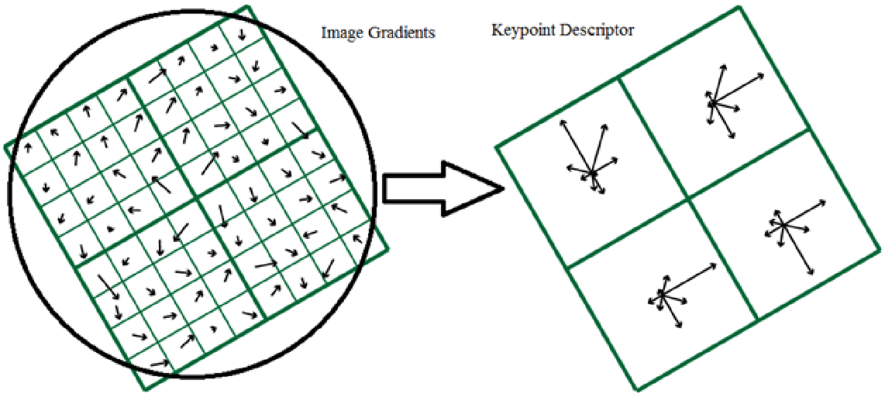


Figure 5.8: Formation of the keypoint descriptor. The black circle is utilized to indicate the presence of the Gaussian centered at the keypoint

Figure 5.9 represents the detected features for the two images when the SIFT algorithm is applied for detection of keypoints; here, the number of the detected features is also computed. The number of the detected features is equal to 2330 for the image on the top whereas the image at the bottom has 238 detected features.

SURF

An inspired algorithm from the SIFT algorithm is the SURF [12] which is based on the Hessian matrix and can be applied in both feature detection and extraction steps. The algorithm speeds up

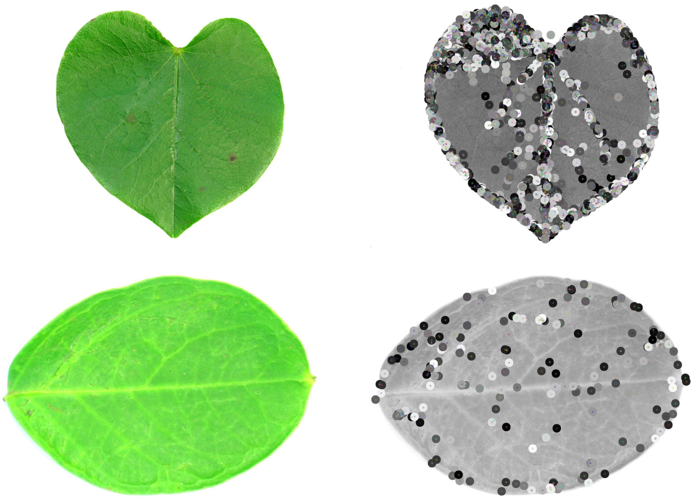


Figure 5.9: The detected features for two images when the SIFT algorithm is applied to detect keypoints

the SIFT algorithm without substantially sacrificing the quality of detected points. Thus, this method is widely used in different computer vision applications considering its efficiency, distinctiveness and robustness in invariant feature localization. Moreover, the algorithm is applied to extract features as a component of the combined methods which will be discussed later. In the SURF algorithm, an intermediate image representation, called the image integral [156], is used to increase the calculation speed of the algorithm. The integral image is obtained by computation of an input image. The input image is I and the integral image is IM where a point is (x, y).

$$IM(x, y) = \Sigma_{i=0}^{i \leq x} \Sigma_{j=0}^{j \leq y} I(x, y) \tag{5.18}$$

When the integral image is used, calculating the area of an upright rectangular leads to a reduction of four operations. Moreover, the change of size does not affect the computation time, and the algorithm is still efficient, even though large areas are required.

The SURF entails computation of the Hessian matrix and its detector is basically based on the determinant of this matrix. A two-dimensional Hessian matrix consists of a 2 × 2 matrix containing the second-order partial derivatives of a scalar-valued function (image pixel intensities) as shown below:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial xy} \\ \frac{\partial^2 f}{\partial xy} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \tag{5.19}$$

It is a symmetric matrix and its determinant is the product of eigenvalues.

Calculation of derivatives is performed by convolution with a suitable kernel. The determinant can be calculated in different scales. Gaussians are optimal for the scale space considerations and the SURF approximates LoG with a box filter. A parallel procedure is also possible due to two usages, box filters and integral images. The purpose is to increase the computational efficiency. Integral images help to do faster computation of the box convolutions and have a quick method to compute the intensities for any rectangle within the image, which is independent of the rectangle size. In addition, computation time is not sensitive to the filter's size. A scale space is divided into octaves which show a series of filter response maps obtained by convolving the same input with a filter of increasing size [157].

The construction of scale space begins with a 9 × 9 filter. It calculates the blob response of the image for the smallest scale. After that, the size of filters increases to 15 × 15, 21 × 21, 27 × 27, etc. to continue the procedure. Blob response is shown in the location (x, y, σ) as follows:

$$det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{5.20}$$

When the used filter is a 9 × 9 matrix, σ equals to 1.2. In general, the following equation exists:

$$\sigma = (currentfiltersize/basefiltersize) * (basefilterscale) \tag{5.21}$$

where the base filter size is 9 and the base filter scale is 1.2.

An important step is to localize and find the major keypoints in the scale space. To achieve this goal, a non-maximum suppression is applied in a 3 × 3 × 3 neighborhood. The maxima of the determinant of the Hessian matrix are then interpolated in the scale and image space with the method proposed by Brown in [157] [158]. One considerable point is the difference in scale between the first layers of every octave which is large, thus scale space interpolation is an important issue.

The next part is the feature description which should be robust and unique for a feature. The other points are the direct impact on computational complexity, robustness and accuracy. In the description phase, the bases are on Haar wavelet responses in horizontal and vertical directions, x and y. The integral images are used to do efficient calculations at any scale. Finding the orientation of interest points contributes to having a rotational invariance algorithm. Gaussian weights are applied to the interest point, obtaining robustness against deformations and translations. The SURF provides a functionality called upright-SURF or U-SURF which contributes to robustness up to ±15° [157]. This version of the SURF improves the speed of the algorithm as one of the advantages.

An interest area is defined by a window size of $20s \times 20s$. This area is divided into 4 × 4 square regions as subareas, and they contribute to keeping spatial information. Then, Haar wavelet responses are computed for each subarea in x and y directions. A vector is created after this procedure. In the following, the formed vector is shown:

$$v = (\Sigma d_x, \Sigma d_y, \Sigma|d_x|, \Sigma|d_y|) \tag{5.22}$$

In general, there are two different dimensions for the SURF feature descriptor, 64 and 128. The 64-dimension version has higher speed whereas the 128-dimension version provides better distinctiveness of features. This can be considered as another functionality of the SURF algorithm. The sign of the Laplacian contributes to speeding it up at the matching stage. It is another improvement of the algorithm. Here, the purpose is to distinguish bright blobs on dark backgrounds and vice versa:

$$\nabla^2 L = tr(H) = L_{xx}(x, \sigma) + L_{yy}(y, \sigma) \tag{5.23}$$

Laplacian is the trace of the Hessian matrix and the values are previously calculated for the determinant of the Hessian matrix. Moreover, it is possible to use the sign of Laplacian to have faster matching and it does not have any impact on the performance of the description and other stages.

Figure 5.10 represents the detected features for the two images when the SURF algorithm is applied for detection of keypoints and the number of the detected features is also computed. The number of the detected features is equal to 38717 for the natural image while the artificial image has 550 detected features.

ORB

The ORB has been proposed as an alternative to the SIFT or SURF in [81]. This algorithm can be explained in two steps, the detection and description parts. The basis of the detection part is

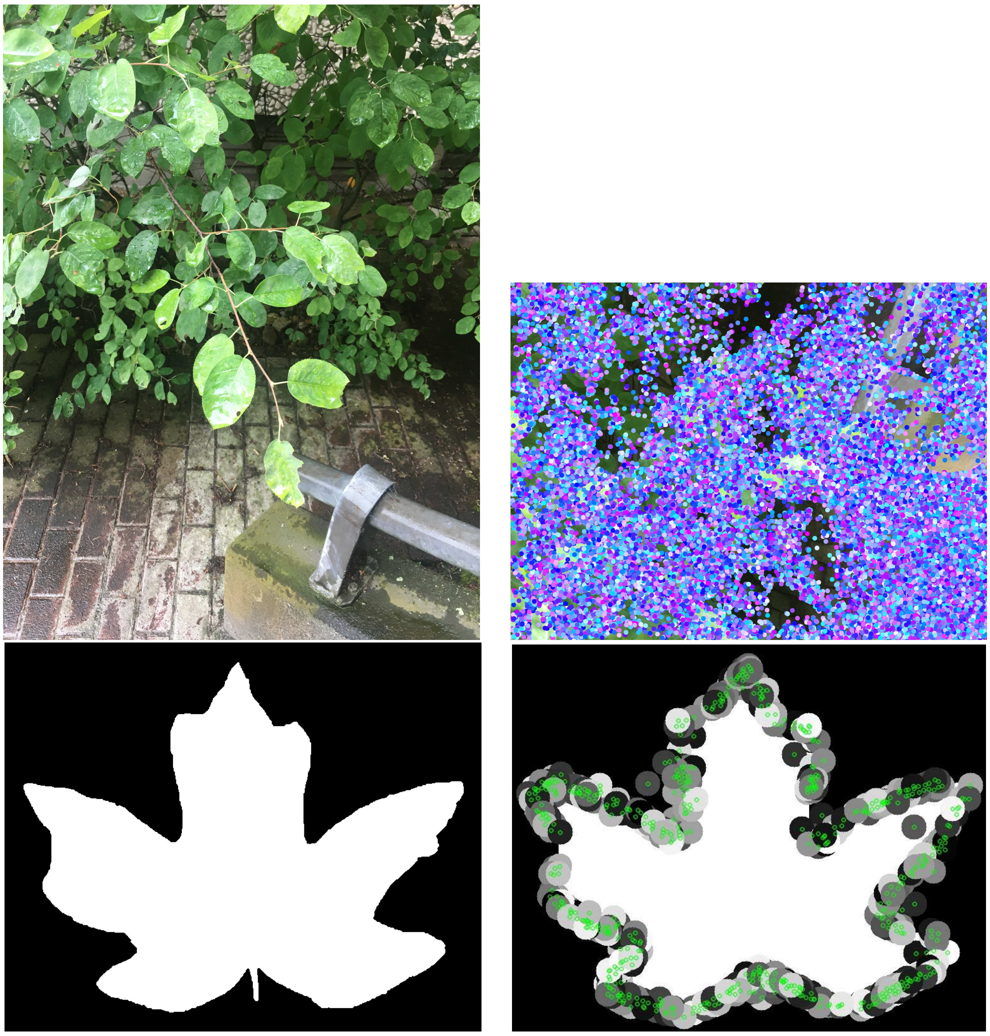


Figure 5.10: The detected features for two images when the SURF algorithm is applied to detect keypoints (the number of the detected keypoints for the natural image: 387171, the number of the detected keypoints for the artificial image: 550)

the FAST algorithm where the description part is based on the visual description algorithm, binary robust independent elementary features (BRIEF) [159]. Mixing detection and description algorithms is a good idea used in the next steps to achieve the purposes of the study.

The ORB algorithm applies the FAST algorithm and finds the keypoints first. Then, it uses the HARRIS corner measure to select top N points among the detected keypoints. In order to produce multiscale-features, it utilizes pyramid too. As we know, the orientation is not computed by the FAST algorithm. One question is how to solve the problem of the rotation invariance, and it leads the authors to proposing a modification. To this end, the intensity weighted centroid of the patch with a corner located at the center is calculated. The orientation is the direction of the vector from this corner point to the centroid. Moreover, moments contribute to improving the rotation invariance. If the size of the patch is equal to r, moments are computed with x and y in a circular region of this radius, r.

Secondly, the ORB algorithm applies the BRIEF algorithm for the description part. One question is how to use the BRIEF algorithm which has a poor performance with rotation. The ORB directs the BRIEF due to keypoints' orientations. If there are n binary tests at the location (x_i, y_i), a $2 \times n$ matrix is defined and named S. This matrix contains the coordinates of these pixels. The rotation matrix is obtained by means of the orientation of the patch called θ. It rotates the S matrix and the result matrix, S_θ.

The increment of the angle is performed by $2\pi/3$, and it enables the algorithm to approximate the angle and build a lookup table of pre-computed BRIEF patterns. As long as the keypoint orientation θ is consistent across views, the correct set of points S_θ is used to compute its descriptor [160].

In order to investigate the ORB algorithm, it is also necessary to consider the properties of the BRIEF algorithm. Each bit feature of the BRIEF algorithm has a large variance and its mean is roughly 0.5. This property is lost when it is oriented along the keypoint direction. By losing this property, it becomes more distributed and higher variance makes a feature more discriminative and its response differs from one input to another input.

One goal is to obtain the uncorrelated tests as each test helps the result. Due to the mentioned points, the ORB should find the uncorrelated binary tests with high variances and means near to 0.5. The solution is to run a greedy search among all possible binary tests. The result is called the rBRIEF.

As it is stated that the ORB is much faster than the SIFT and SURF algorithms and its descriptor has better performance in comparison to the SURF algorithm [81], the ORB algorithm can be utilized in low-power devices for panorama stitching [160]. Multiple-probe locality sensitive hashing (LSH) [160] is also applied for matching purposes and it is an improvement to the traditional LSH [161].

The ORB algorithm has been applied to one sample image, and the number of keypoints is equal to 218414.

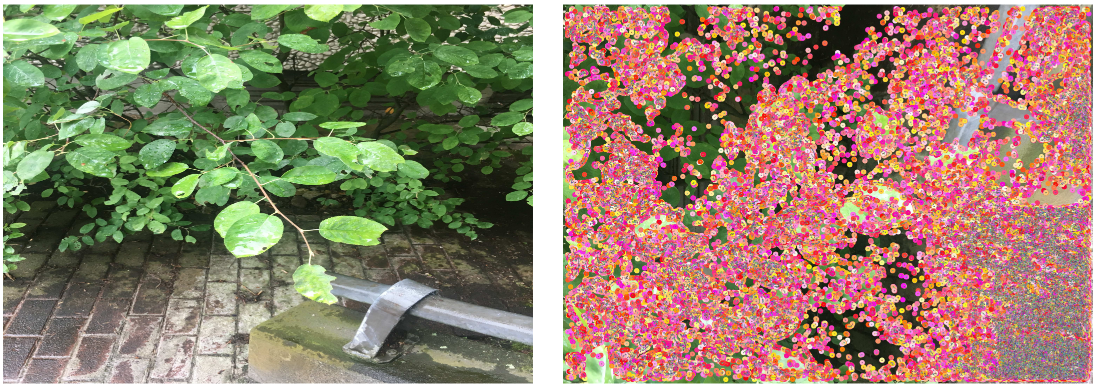


Figure 5.11: Representing the detected features using the ORB algorithm

5.4 Matching

First, it should be made clear that the current research project aims at exploring the different aspects of the plant recognition system. However, those aspects become very challenging even in the aspects that seem to be apparently simple. Therefore, many different approaches as well as various aspects at any stage have been investigated to discover possible solutions. At the current stage, the concepts of matching have been considered due to the aim of matching for next processes. Matching is a technique in image processing for finding parts of an image which match another image. By using this technique, it is possible to compare the quality of the implemented algorithms and find the corresponding features from different images based on a search distance between the feature vectors such as the Mahalanobis [162] and the Euclidean distance. With respect to our different datasets where images are not limited to only one scene for one specific plant species, we really need to use the matching technique and test out images. Unlike a common matching problem, we do not have only one leaf in an image and there is a bunch of leaves or a branch of a plant species. Accordingly, it should be mentioned that only the matching technique is not responsive for sophisticated tasks such as natural plant recognition. For the purpose of a facile implementation, the matching algorithm used is one of the standard matching approaches, which is not at that best performance for all different applications. The reason is that it is impossible to define only one general matching algorithm as the best algorithm for all purposes, although the implemented algorithm shows flexibility for various datasets.

5.4.1 General Overview of Matching Technique

The primary stage of the technique is to detect the keypoints. When keypoints are detected, it is possible to extract the descriptors in the right way and use them for further processes of matching. The most important stage of the matching algorithm is to define a criterion for comparing the extracted descriptors. Hence, a well-known computer vision method, called brute-force search [163], is applied as the core of the matching technique. The Brute-force matcher [164] takes one descriptor of the features in the first image and calculates the distance between the features of the image and the features of another one. Finally, the smallest distance is the corresponding match. In other words, all the possibilities are checked by using this technique.

According to the used method for feature detection and extraction, we define different distance types. For instance, the Euclidean distance is a good choice while the SIFT or SURF algorithm has been applied. If the given vector is X, the Euclidean distance is then computed as below:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{5.24}$$

$$|X| = \sqrt{\Sigma_{k=1}^{n} |x_k|^2} \tag{5.25}$$

Hamming distance [165] is another choice which can be used as a type of distance measurement. It is applicable for binary descriptors such as the ORB, BRIEF, BRISK, etc. Actually, as a metric distance, the Hamming norm is actually utilized. This type of distance measurement is usually used for binary descriptors, and the calculation of the distance is performed by counting the number of bits that are dissimilar. For instance if there are two vectors, $X_1 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$ and $X_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$, the Hamming distance is equal to 2.

Basically, the dependency of feature matching on similarity, complexity and quality of images is irrefutable and therefore better matching results can be obtained by more similar images. If a technique cuts out more uncertain matches, higher percentage of successful matching can be achieved.

5.4.2 Results for Matching and Explanation

In order to perform the matching task, one of the implemented systems consists of the SIFT detector and descriptor. In the case of the standard SIFT algorithm, it has a 128 dimensional description of a patch of pixels around the detected interest point. To perform the matching task, we use two images and carry out a process to match the features in these two images. The matching process consists of the following steps:

- Reading the two sample images
- Converting the images into grayscale
- Detecting and extracting the keypoints in both images
- Performing the Brute-force matcher by the use of the Euclidean distance (Norm L2, 4)

- Showing the keypoints in both images and drawing results by separate lines where each line represents the connection between matching keypoints in the sample images

In order to examine the proposed method, we utilize different samples and apply the proposed matching method on them. Before carrying out the experiments, we briefly examine the Brute-force matcher to know how it is actually working. The used matcher is based on the Euclidean distance which is the preferable choice for the matching process based on the SIFT algorithm [166]. Another important point is the behavior of this matcher and its goal is to find the k-nearest neighbors for each descriptor. It should be pointed out that the SIFT algorithm provides independent descriptors from scale and octave after detecting keypoints in any scale. We should not forget that descriptors are actually describing keypoints.

Figure 5.12 represents the input images and the final result of matching process. The number of the detected keypoints for the sample image on the left side of Figure 5.12 is 1160 while the number of the detected keypoints for the sample image on the right side of Figure 5.12 is 746.

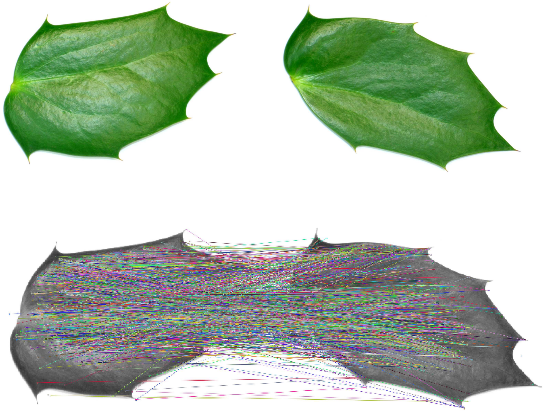


Figure 5.12: Matching between the two leaf images with the white background

Figure 5.13 shows two natural images, and the matching process has been conducted on these samples. The number of the detected keypoints for the sample image on the left side of Figure 5.13 is 3340 and the number of the detected keypoints for the sample image on the right side of Figure 5.13 is 2167.

The next experiment is to scale two samples of the previous matching experiment and test matching. It is notable that the new samples have approximately 31.25% of the original samples in the previous test. For both the original samples, the dimension is equal to 1600 × 1200. After scaling, the dimension of the new samples equals 500 × 375. Figure 5.14 shows the scaled images and the final matching result. The number of the detected keypoints for the sample image on the left side of Figure 5.14 is 253 and the number of the detected keypoints for the sample image on the right side of Figure 5.14 is 130. Meanwhile, it should be noted that the number of the detected keypoints for the image on the right side of Figure 5.14 is approximately 17% of the number of the detected keypoints for the original image without scaling.

Because the transformations are simple affine transformations the following is true:

If we plot the endpoints of the vectors in a coordinate system (d_x, d_y) we get a regular pattern; i.e. for translation one point. So, we can compare the real measurement patterns with the expected pattern.

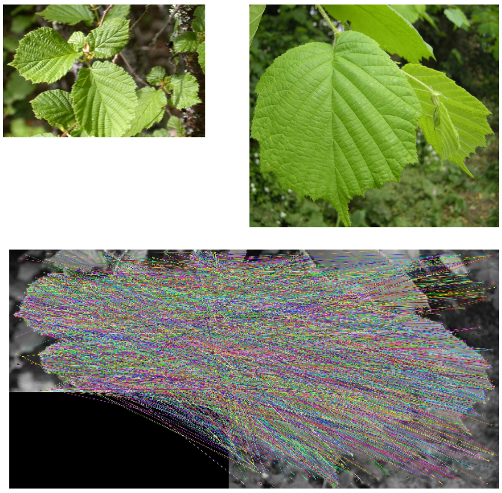


Figure 5.13: Matching between two natural plant images

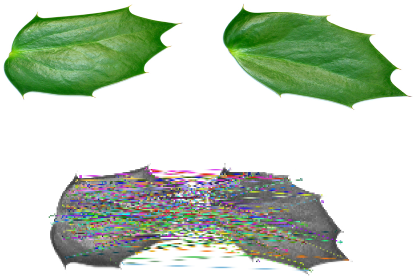


Figure 5.14: Matching between two scaled leaf images with the white background

5.5 Combined Detection and Description Methods

Our life is full of the memories of different events which have occurred in the past. When one thinks of a specific memory, he recalls some pieces of the memory depending on some factors such as the importance and the boldness of the memory. These pieces of the memory are the abstracted parts of it. If one decides to describe this memory to other people or attempts to write it down, he connects the pieces together and solves the puzzle of his specific memory even if many years have passed. The order of using these pieces depends on the sequence of occurrence time and the way he remembers the memories. Consequently, each piece contains different information although it is supposed as a point. All or some of the points might be used for explanation of the memory, and the selected points might affect the reality and veracity of the memory if we decided to compare and adapt it to the real occurrence after a while.

Suppose that some people took part in one special event on the same day and at the same time. After the event, one TV reporter asks this group of people to thoroughly explain the event for TV viewers. Although they explain one event, they give different information and details and describe it in their own words. It should be pointed out that vocabulary treasure may have influence on each one's explanation and story. Moreover, the charm of the event for TV viewers differs according to the explanation of each participant and the rest of any discussion on the event depends on explanations.

The mentioned examples are two important points which can be considered as the main key to our work and the idea of combining different algorithms. In the images, the initial attempt is to find the points which can be processed for further purposes and defined goals. Hence, different algorithms have been developed to detect interest points. Each algorithm can be assumed as one participant who has taken part in an event and is able to explain the finding of the event in a distinctive way. Therefore, each implemented algorithm can detect its own interest points of each image and the extracted information can be applied for the next steps.

As another example, if we ask two people to look at the same scene in one image, we cannot expect them to exactly identify the same features of the scene. However, if we pose a question on what they see in the image, they provide similar answers as the whole scene is almost the same and only some details of the scene are different. In order to computerize the process and find extracted details and information, the performance of each algorithm is unique for one specific image and the digitized information is different from one algorithm to another. Additionally, this is similar to the mentioned example.

Some characteristics, such as invariance, robustness to rotation changes, scaling variations and change of light, have impacts on the performances of the algorithms, and thus on the achieved results. Traditional and modern algorithms can be mixed up to add new characteristics and features like speeding up the whole process. Since the separate investigation of the algorithms has been carried out in previous sections successfully, the use of the combination in detection and extraction is investigated in this section. The use of the combined algorithms helps stabilize the process and reduce the undesired impacts of changes such as rotation and scaling. In this section, new algorithms, the so-called combination of different modern algorithms (methods), are derived to impressively detect and extract features. Generally, one purpose is to prevent failing of the algorithms when applied separately in different conditions such as space scaling and rotating. Moreover, we attempt to deal with images which have been taken at different conditions and it is important to have algorithms which are applicable.

In this section, some combined algorithms are introduced to detect and extract features in plants' images. Remarkably, unique potentials and high performance of a combined algorithm contribute to training a plant classifier and recognize plants accurately. However, many features need to be extracted and then trained, which increase the usability of the combined method for recognition applications, especially for an accurate and high-speed performance.

Therefore, this section attempts to provide a way to improve the robustness of the algorithms studied in the previous sections. This idea leads to new feature detection and extraction of plants using multi-algorithms; to this end, spreading different characteristics of the two algorithms helps spread out information based on different detected features. Thus, the aim of this section is to develop combined methods which perform new quantitatively accurate detection and extraction sufficiently fast for plant recognition applications. Indeed, rich features should be detected by detection algorithms and then should be considered as the points for description. For each point, a combination of detection and description is used to infer a feature vector. A combined approach takes into account both detection and description properties of utilized algorithms and benefits of the properties in different environments; however, the type of the environment affects the efficiency of the algorithms. This enables the combined algorithms to capture much more detailed information and features than does the prior art, and also to give a much richer experience in the representation of information, even for the scenes with significant and incomparable conditions such as various weather conditions and different photographing time as well as including the presence of shadow, wind and sunshine. In the experiments, we carry out our idea of combining the algorithms, yielding recognition accuracies of over 90% which outperforms the conventional systems.

5.5.1 HARRIS-SIFT and HARRIS-SURF

The HARRIS algorithm is a conventional corner detector which is suitable to apply in our work as it detects changes in image intensity and has some useful properties. The HARRIS algorithm is particularly applicable to a wide variety of images such as urban images, images of industrial products (cars and airplanes), etc. One property of this detection algorithm is being invariant to rotation. If a corner is found in one ellipse, the detected part remains as a corner even if we rotate the ellipse randomly. Therefore, rotation of images does not have any impact on the detected corner. Another property of the HARRIS algorithm is that it is partially invariant to affine intensity changes. For instance, if we shift the intensity from I to $I + b$, this operation is invariant as we only use derivatives in the HARRIS algorithm. Another possibility is to change the intensity from I to αI; it is not invariant in this case according to the calculation of derivatives. It is not invariant to spatial scaling. If an edge is detected by the algorithm, rescaling might lead to detecting it as a corner (see Figure 5.15).

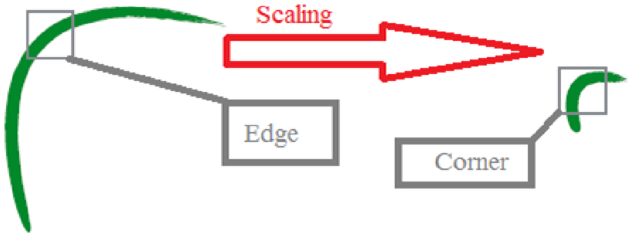


Figure 5.15: Edge, scaling and corner in the HARRIS algorithm

In other words, the algorithm is robust to the image translation, rotation and noise, high repeatable to luminance variation and rigid geometric transformation [167]. In [168], it is shown that the HARRIS detector is a good start point for the computation of affine invariant features and positions of scales, and a combination of the HARRIS detector with Laplacian-based scale selection is carried out. The HARRIS-Laplace detector is then extended to a new detector, called the HARRIS-affine detector, and it helps to solve the challenges of affine transformations. There is yet another important point to discuss. The computational cost of the proposed approaches is extremely high and therefore it is essential to compensate for this disadvantage. Our idea is to use the detected features of the HARRIS algorithm with description algorithms which benefit from other properties. We investigated our hypothesis by means of two modern algorithms, the SIFT and SURF, for the description step.

Being invariant and distinctive are two important factors and we intend to add them in the description step. Therefore, we decided to utilize the SIFT algorithm to achieve the descriptors. High reliability and robustness of the SIFT algorithm help us approach the goals. As discussed, the description part of the SIFT algorithm is carried out on detected keypoints of the implemented HARRIS algorithm. Now grids of 4 × 4 (sub-regions) are taken by making this feature point as the center in the scale of the image [11]. The gradient direction histograms in the eight directions of each sub-region are calculated. Then, a feature vector is generated and the dimension of it is 128. This feature vector is the desired descriptor that we need for the purposes of our study.

Definition

HARRIS-SIFT is a new term which consists of two terms. The first term shows the used algorithm, the HARRIS algorithm, for the detection step and the second one represents the applied algorithm, the SIFT algorithm, for the description step.

In order to improve the idea of image characterization, a new strategy, i.e. the SURF algorithm, is chosen as it has higher speed of computation in the description step. As previously explained, the

SURF algorithm applies wavelet responses in two different directions of x and y, i.e. horizontal and vertical directions, and neighborhood of size $20s \times 20s$ is taken around the keypoint where s is the size. It is divided into 4×4 grids. For each grid, the horizontal and vertical wavelet responses are taken and a vector is generated as below:

$$v = (\Sigma d_x, \Sigma d_y, \Sigma |d_x|, \Sigma |d_y|) \tag{5.26}$$

The length of the SURF feature vector is equal to 64 and the advantage of this lower dimension is its higher speed of computation compared to the SIFT algorithm.

These combined methods are significant because we can utilize a mere detection algorithm to obtain the keypoints and exploit these keypoints and related properties for the description step. One possibility is to use another detection algorithm instead of the HARRIS algorithm and combine this new algorithm with the SIFT and SURF algorithms. The new possibility is proposed in the next section.

5.5.2 FAST-SIFT and FAST-SURF

The HARRIS algorithm has two useful properties: stability and robustness. However, the algorithm suffers from one limitation which is related to the speed of computation. Similar to the previous section, we attempt to combine the algorithms, with the main purpose of proposing algorithms which perform faster. Pertaining to the mentioned properties and details of the FAST algorithm, it is used as a detection component of the combined algorithms. In the next chapters, the two methods are used for plant recognition systems. Similar to the previous section, the combined methods are explained in detail in the subsequent chapters.

5.6 Conclusion

We have introduced some new modern methods for feature detection and description using the combined approaches. Compared with the existing approaches, our method provides a new possibility to do the description for detection algorithms and achieve other properties as a result. For instance, the speed and computation time are two important factors and they affect our decision for selecting the appropriate algorithm. The final quantitative accuracy of a system should be also acceptable. Combination of the algorithms helps to catch the mentioned properties. For instance, it is possible to speed up the detection step of the SIFT algorithm by using the FAST algorithm. Overall, the combined algorithms enable fast and robust detection and description modules which can be used for automatic plant recognition systems. We acknowledge that the use of detection algorithms alone in the plant recognition system is not efficient, due to the lacking of the description module. A desired plant recognition system has two important parts, i.e. detection and description, so that it can find the applicable and useful features. Such a system is planned to be implemented and established.

Chapter 6

Implementation and Comparison of Efficient Modern Description Methods for Recognition of Classic Plant Species

The present chapter introduces six different systems for automatic recognition of plant species. Modern description methods are applied in the systems for automatic recognition for 32 different plant species (classic dataset). This classic dataset is actually the Flavia dataset explained in 3.1.3. The results will help us to compare the efficiency of the implemented systems, explore the goals and decide on selecting an appropriate system due to the pre-defined purposes of applications. In order to judge the systems, some experiments are conducted. The experiments contribute to drawing borders between the demands and the systems. As a result, we can make a decision about selecting various systems. We further illustrate which experiments have been carried out. The presented systems yield good recognition accuracies for classic datasets with a large number of plant species. The recognition accuracy of the systems differs from one to another, and the characteristics of each one contribute to using the correct system in appropriate situations.

The works have been published in Signal & Image Processing: An International Journal (SIPIJ), April 2015 [169] and in Advances in Image and Video Processing Journal (AIVP), 2015 [170].

6.1 Introduction

One of the important aspects of the biological evolution, known as a remarkable part of evolution, is plant evolution. It involves with other facts like adaptations, acclimations and modifications. Fossils are actually indirect evidences of the presence of plants around 3000 million years ago. For instance, oxygen-producing photosynthesis has been observed in geological records. Despite the low level of complexity of early plants, they have played an important role over successive generations and the passage of time. Early plants were responsible for cooling the climate, increasing the level of oxygen and lowering the amount of carbon dioxide in the Planet's atmosphere. It is also worth mentioning that fossil fuels like coal and oil have been made from plant material. One example is carbon that was taken out of the atmosphere and buried in swamps many years ago [171]. In other word, plants played the role of a bridge between the life's evolution and the chemical evolution of the atmosphere.

By looking into the evolution of plants we find a wide range of complexity from the early stages of existence of plants to current gymnosperms and angiosperms. In addition, they have diversified in aquatic and terrestrial environments over years. Four groups of land plants, mosses, ferns, conifers and flowering plants, reflect a sequence of the evolutionary history of land plants. Structural support is a factor affecting the life of plants in different environments. A plant is able to live in an environment if it has the essential components. In addition, other environmental factors have enormous effects on plants and relevant issues like adaption. Two of the environmental factors are buoyancy and gravitational force which are not the same in aquatic and terrestrial environments and vary from an environment to the other one.

Let's consider the mentioned factors in an environment with dense water. We know that gravity changes by height and the gravity also varies a little bit in oceans. In dense water, we have the feeling of being light. The reason is actually buoyancy which generates an upward force to objects with smaller density than water in aquatic environments. In addition, the effects of gravity reduction and structures with gas-filled vesicles allow them to float like kelps. Hence, an obstacle will be solved by structure and large forms of kleps and adaptation emerges.

Investigation of the role of plants is vital for better planning of the ecosystems' future. In any ecosystem such as forests, wetlands, etc., there is an important issue related to water called the water cycle. In the process of water cycle, plants play an invisible role which is transpiration. They absorb water through their roots from ground. Water ground is distributed in other components of plants like stems and leaves. A part of water is evaporated on the surface of leaves. Evaporation rate depends on the environment. In dry days, plants add more moisture to natural environments. Additionally more water will be returned to the atmosphere by the transpiration. For instance, leaves of an oak tree transpire 151416.47 liters of water per year [172].

By considering plants as producers, negative factors like diseases can affect their performance and interfere with their roles and tasks. In addition, plants are responsible for security of human life and the future world's food. These issues can be influenced by other factors such as urbanization, population growth, income levels, lifestyles and preferences over time. Let's have a look at two of mentioned factors, urbanization and income levels, and investigate the relationship between plants and them. The first factor, urbanization, is actually a very complex term because it can be defined differently and various interpretations can be derived from it. One definition for this concept refers to spread and strengthen of urban living, economic and behavioral patterns. Another definition is the population shift from rural to urban area. Urbanization has effects on patterns of food and dietary. Furthermore, food preferences are also changing because of new lifestyles in urban areas. Consequently, the diversity of people's diet is increasing. It is a demand to have plants suiting these whole new needs.

We investigate the next factor, income level, its effects and the importance of plants. Although poor people struggle to make a better life, sometimes they are not successful in providing decent food. In fact, they do not have enough income to buy their needs. On the other hand, economic growth does not necessarily mean an increase in their income. In many under-developed and developing countries, growth rates are negative or low. Therefore, income levels are very lower. As a result, agriculture and related activities are also influenced by negative and undesired rates. Hence, there is an invisible relationship between this point and plants. Furthermore, plants are prone to various injuries and diseases. Climate changes cause pest infestations and subsequent crops are being affected. The effects of pest might be continued in the years to come and we face marginal losses [173]. Perhaps the simplest solution for overcoming the diseases is the use of chemical pesticides. Nowadays, there are other options like pest management approaches and accurate use of pesticides to reduce harmful effects of chemical pesticides to human health and environmental safety. In addition, plant diseases are a threat to the world's food security and exacerbate deficit of the food supply. It should be pointed out that it is an aspect that has global effects.

Nowadays, one of the main concerns is global warming which deserves further discussion owing to its importance and effects on plants. In order to do the photosynthesis, plants take in carbon dioxide and give off water. Consequently, plants and the surrounding air will become cooler by the evapotranspiration process. In [174], it has been proven that plants can take the responsibility for offsetting greenhouse gas emissions. In addition, the findings show that plant leaves give rise to some methane in a very small amount and fears of forestry and agriculture contributions to global warming will be allayed [175]. As a result, we find that plants and their effects are not constrained to our today's world. In fact, they will affect our lives in the future too. Therefore, the study of plants in different aspects is necessary and none of the related fields can be neglected.

In botany and plant taxonomy, a challenging task is recognition of plants. It is even more difficult if it is needed to be done automatically. At this stage, the focus is on designing and implementing plant recognition systems due to the importance of plant recognition. Plant classification is also important for ecological purposes and discovery of the future of plant species. Taking medicinal and commercial applications of plants into account, precise identification of plants is a desire. We propose different approaches that can be used in plant recognition systems to meet the final needs.

To give computers a visual understanding of plants, specifying important component of plants is necessary. By looking at several plants, we may instantly find that stem, root, flower, fruit and leaf are common components of different plants. The question is, "Which components can be used for the plant recognition task by machines?" This step is very important as the development of systems depends on the nature of data and the final application.

Despite the presence of other components like fruits and flowers, leaf is usually considered as a reliable component for plant recognition in botany. This component usually grows after cold temperatures in winter. In other words, it is the response of plants to warmer days. Over time leaf grows and its color changes from light green to darker green. The shape of leaf remains somehow the same. The investigation of different plants shows that the shape of leaves differs from one plant species to another, but leaves of one plant species have mostly the same shape. However, they are rarely the same in all of their characteristics such as size, color, etc. If we compare this component to the others, we will find that it has two important properties namely generality and availability. In addition, different metrics can be derived from leaves for getting valuable information about the shapes of leaves for plant recognition. Another point is the simplicity of collecting leaves from various plant species. This property contributes to having useful plant datasets for the plant recognition.

Let us look at previous works in plant recognition and its related areas. Due to the importance of shapes and its related features, Takeshi Saitoh et al. [176] proposed an approach based on two components of wild flowers, flower and leaf, and obtained 95% recognition rate. They used two images (frontal flower and leaf images) for the recognition task. In this approach, 17 different features were extracted from two images and fed to the system. Considering flower and leaf images, eight color and shape features were extracted from the flower image while nine features were extracted from the leaf image. The set of leaf features composed of ratio of the average internal area connecting the valley points over the average external area, the ratio of the vertical length over the horizontal length, moment, roundness, a defined bias, opening angle at the stem, opening angle at the tip, structure index showing if a leaf structure is pinnate or ternate, and color [176]. In 2004, a computerized plant species recognition system under the name of CPSRS [4] was introduced as a web-based application consisting of two main parts, web client and web server. Text-based information retrieval and content-based leaf retrieval are two types of plant species retrieval methods that were proposed in this work. The experimental results showed a recall rate of about 71% by considering the top five images [4]. Another work proposed in 2006 [177] used shape features of leaf for the plant recognition with the maximum recognition rate of 92%. The extracted features were eight geometric features like rectangularity, circularity, eccentricity and seven moment invariants from contours of leaves. The features were applied to a new moving center hypersphere classifier to carry out the plant recognition.

In [178], new doors were opened to plant recognition by entering the support vector machine and extracting both color and texture features for SVM classifiers. Afterward, Zhang et al. [179] used a learning method called the locally linear embedding (LLE) [179] to benefit from this method for projecting the original samples into a low dimensional space. Another purpose was to preserve the least reconstructed weights among the neighbor points. Due to the sensitivity of LLE to noisy points and outliers, a weighted LLE (WLLE) algorithm [179] was proposed. In this approach, the score of each point, called the importance score, was obtained by the heat kernel function. They were added to the cost function of WLLE and the final recognition task was carried out. In 2011, another proposed work was based on combination of local descriptors and global features to recognize plant species [180]. In order to select most discriminant features, a linear discriminant analysis method was applied in this work to develop an automatic leaf recognition system and achieve acceptable results [180]. A modified locally linear discriminant embedding (MLLDE) algorithm [181] was proposed in [181] and it was based on LLE and modified maximizing margin criterion (MMMC) [182]. It was possible to map leaf images into leaf subspace for further analysis by using MLLDE. The benefit was full use of class information for improving discriminant power and developing an efficient plant recognition system. In another work [183], a new method called the multiscale distance matrix was proposed to get the geometry of the shape which has important properties such as translation invariant, rotation invariant, scaling invariant and bilateral symmetry. In order to improve the power of discrimination, descriptor and dimensionality reduction were combined. The method was extremely fast for real-time applications. Easy implementation was also mentioned as one of the advantages of the method in this work.

Since 2004, the SIFT algorithm has been used in different tasks, especially for detection and recognition purposes. In [184], SIFT features were assembled with a K-means matching method for face recognition. Final results showed the robustness of the SIFT algorithm in variations of expression, accessory and pose. In addition to the SIFT algorithm, the SURF algorithm has also been used widely in many classification tasks. For instance, SURF features were applied in [185] for detecting face components. To develop the system, a classifier checked the feature vectors firstly if they were from face images. The component labeling was then performed to specify nose, eye and mouth.

Using modern algorithms like the SURF and the SIFT usually leads to detecting many features which can be put in a bag of features. In such bag, accurate representation and use of features are so important for further applications. An effective solution is to use the bag of features approach [186] and represent features effectively for the training step of systems. The origin of this method is natural language processing tasks and information retrieval. This method has been introduced and applied widely; e.g. in [187] [188] [189] [190] [191] [192].

Before starting the design and implementation of an automatic plant recognition system, it is essential to investigate the availability of suitable leaf datasets to be selected from. There are some common datasets such as Flavia dataset, Leafsnap dataset and ImageCLEF dataset. We decided to select the Flavia dataset because it contains the leaf images of 32 different plant species. This number of classes encouraged us to design and implement the systems based on this dataset. The dataset consists of 32 different plant species and we divided it into two sub-datasets, i.e. training dataset and testing dataset. The training dataset consists of 1255 leaf images, and the testing dataset is made of 648 images taken of leaves. Before starting to work on this dataset, the investigation of the dataset was mandatory to find possible solutions of our main problem, plant recognition system. However, we just concentrated on considering the plant species of the dataset, not other additional information like location and distribution of the plants. Without any doubt, it is usually hard for a human to recognize the similarity of one plant species when different images of it have been taken. This fact motivates us to design and select algorithms which are able to extract discriminative and repeatable features of each plant species.

The structure of the chapter is organized as follows: Section 6.2 describes a general overview. Section 6.3 describes how to do pre-processing with respect to color and grayscale images. Section 6.4 introduces the backbone of our systems while an active way to do the training is illustrated in section 6.5. Experiments, results and performance analysis will be discussed in section 6.6 and applications of proposed systems will be provided in section 6.7 while section 6.8 provides the acknowledgment, and section 6.9 concludes the work.

6.2 General Overview

All over the world, there are plenty of plant species and subsequently a large volume of information of them in different types. Hence, the development of a fast, reliable and competent classification technique is necessary to handle the information and classify the data. Shapes of leaves are good solutions to plant recognition problems. They help botanists and biologist identify and recognize various plant species as they provide useful amount of information. In order to recognize plant species, botanists use their references books. They have to find the exact family and name of one plant species by searching several pages of books. This type of search is really time-consuming. In addition, the probability of wrong recognition is usually high. Recent advances in computer vision can help developing accurate and reliable systems for automatic plant recognition.

If we consider only one leaf, is it possible to use all information of the leaf image? How can we obtain useful information from the leaf image instead of extracting all the information? Do all pixels of the leaf image contain important and valuable information? The answers to these questions have influences on decisions for further steps and design of systems. With regard to this issue, we need to find effective methods for extracting information of leaf images to develop efficient and fast systems.

Detection of features is a basic and important part for many applications. Feature detection is the process where automatic examination of an image is done for finding unique features of an object. In a such manner, the object could be found based on its features in different images. Detection of local interest points plays an important role in different image processing tasks. For instance, it is the first step of the BoW model which will be discussed later. Automatic detection of features should be performed to detect unique points in images to construct a useful set of points. Using this process, any object will be detected based on its own features in each image. Several well-known region detectors have been mentioned in the literature [193] [194].

In computer vision and pattern recognition, feature extraction is one of the main processing blocks. The primitive objectives of feature extraction are reducing the computational complexity of the subsequent process and facilitating a reliable and accurate recognition. In other words, the goal of feature extraction is to yield a pattern representation that makes the classification trivial. Generally, feature extraction involves reducing the number of resources required to describe a large set of data. When performing analysis of complex data, one of the major problems stems from the number of variables involved. Analysis with a huge number of variables generally requires a large amount of memory and computation power. It may overfit the training samples and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of variables. Furthermore, it is needed to get around the problems while still describing the data with sufficient accuracy. To overcome the challenge of feature extraction in this work, the description part is performed by using SIFT and SURF algorithms. Comparing the algorithms, the SIFT creates 128-dimensional vectors when the SURF creates 64-dimensional vectors. By using the mentioned procedure, each image is a collection of vectors of the same dimension (128 for the SIFT and 64 for the SURF) where the number of components is of no significance.

Another important part of classification is implementing an effective method for representation of images. In classification, the bases are visual contents of images and the similarity of the image contents. The image contents are described via image features. Detecting keypoints with rich information should be done as a basic part. It can automatically be done by using different detection methods and represented by descriptors [168]. Then keypoints are grouped into a number of clusters with those having similar descriptors assigned into the same cluster. They would be shown by one single visual word, thus all keypoints will be mapped into a limited number of visual words.

A popular and widely used approach for feature representation is BoW which can be used to measure similarities between images. A definition of the BoW model can be the "histogram representation based on independent features" [19]. The BoW is inspired by models used in natural language processing (NLP) [195]. This model ignores or downplays the word arrangement (spatial information in the image). Then it does classification based on a histogram of the frequency of visual words. This method includes three steps:

- Feature detection, feature description and visual words generation.

It should be pointed out that this approach is used for document representation in information retrieval. This methodology was first proposed in the text retrieval domain problem for text document analysis [196]. It was further adapted for computer vision concepts and applications [196]. For image analysis, a visual analog of a word is used in the model. It is based on the vector quantization process by clustering low-level visual features of local regions or points, such as color, texture and so forth.

After detection and extraction of features, the next step is to convert vectors into visual words and create a codebook (dictionary) [197]. The visual word "vocabulary" is established by clustering a large corpus of local features. To achieve our purpose, K-means clustering is performed. The clustering algorithm contributes to finding the centers of the clusters of the feature descriptors. The visual words are collected to construct the visual vocabulary. For each feature descriptor in an image, the nearest visual word from the vocabulary is assigned. The distribution of visual words in the image is represented as a histogram. Every feature can be mapped onto a specific visual word through the clustering process. Hence, the image is represented by the histogram of the visual words. The details of the BoW are provided in section 6.4. After extraction of the BoW features from the images, they will be entered into a classifier for training or testing. In this stage, SVM is used to train the classifier, and the last step will be doing the testing part.

The whole system has four parts which are image pre-processing, feature detection and extraction and classification. We would like to follow two main goals, classification accuracy and low computational cost. Therefore, the following architecture is considered the initial scheme of the system. Figure 6.1 shows a general scheme of the plant recognition system.

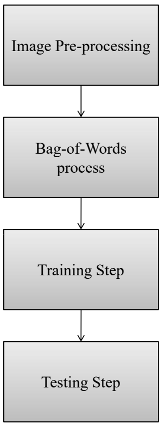


Figure 6.1: General scheme of the plant recognition system

6.3 Image Pre-processing

Definition of a computer image is a two-dimensional matrix of pixels like $N \times N$ m-bit pixels and each pixel has its own value. N is the number of points and m has the task of controlling the number of brightness values. The value is usually derived from the output of an analog-to-digital (A/D) converter. With regard to the value of each pixel, it is proportional to the brightness of the corresponding point in the scene. By using m bits, we will have a range of 2^m values and this range is $[0, 2^m - 1]$. Obviously, higher values of m give more available levels and increase the available contrast in an image.

Color images consist of three intensity components, which correspond to red, green and blue. They are commonly called the RGB images. The storage strategy of a color image is to be saved as a two-dimensional array of small integer triplets or three separate raster maps, one for each channel, although the second choice is rare nowadays.

Figure 6.2 shows one leaf which is represented in the RGB model. In this case, each component is in the range of [0, 255].

Monochrome usually refers to an image in black and white or even grayscale image, although it might be used to refer to other combinations containing single colors such as blue-white or blue-black. This means that all black and white images are monochrome images, but all monochrome images are not black and white images. Although the history of monochrome photography dates back to the middle of the 19th century, it is still popular among many professional photographers for some artistic purposes or imaging applications. Besides, monochrome photographing allows the photographer to focus on desired interpretation and forms instead of reproducing just reality without any emotional and additional impacts on the observers.

An image is taken of a plant with an iPhone 6s in an outdoor environment, and Figure 6.3 represents the original image (on the left side) and the converted grayscale version (on the right side).

6.3.1 Black and White Image

If we look at a chess board and consider each square as one pixel, the chess board is a black and white image with black pixels (black squares) and white pixels (white squares). In order to get a better understanding, we can think of the depth of an image. The depth of an image is defined as

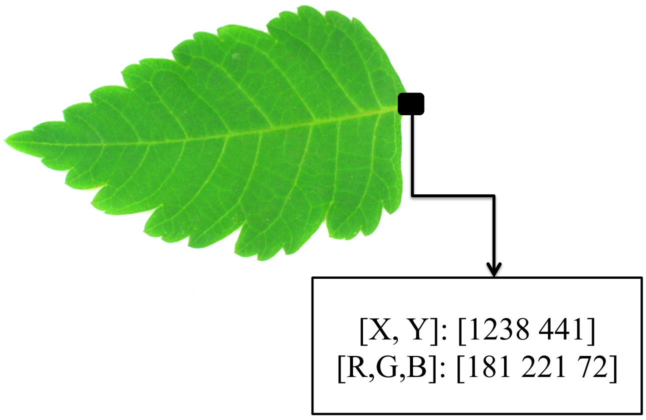


Figure 6.2: A sample leaf image in the RGB model

either the number of bits utilized to show the color of a single pixel in a bitmapped image or the number of bits used for each color component of a single pixel. For instance, if we have a 1-bit image, each pixel holds only one-bit number, either 0 or 1. A black pixel is 0 and a white pixel is 1. We suppose that there is a color image, we are able to convert it into a black and white image by using a threshold:
If p is greater than 150, then it is equal to 1.
If p is smaller than 150, then it is equal to 0.

In Figure 6.4, the assigned threshold is 150 and the color image is converted into a black and white image.

6.3.2 Grayscale Image

In this type of images, the color gradually varies from the weakest intensity, black to the strongest intensity which is white, and the range is made up of shades of gray between black and white. If we assign $m = 8$, we will obtain brightness levels ranging between 0 and 255, and we will be able to show the image in grayscale level.

Here, we can have a flashback to the human vision system. Cones play the role of wavelength-sensitive sensory cells in the human body. Cones are actually divided into three different types. Each type has its own sensitivity to light (electromagnetic radiation) of different wavelengths. In addition, each type is sensitive to one of red, green and blue lights. In fact, this is the reason behind color images (in RGB format) and their components.

Then new questions may be asked, "How does grayscale image work and from what it inspires?"

In a grayscale image, it is not important how much light is emitted of different colors and we have one value per pixel. The difference is the total amount of emitted light for each pixel where little light gives darker pixels and much light is perceived as bright pixels. The two following questions deserve an answer.
- "Why should we use grayscale images?"
- "How can we convert color image into grayscale image?"

When we want to answer the first question, we should consider different aspects of both grayscale

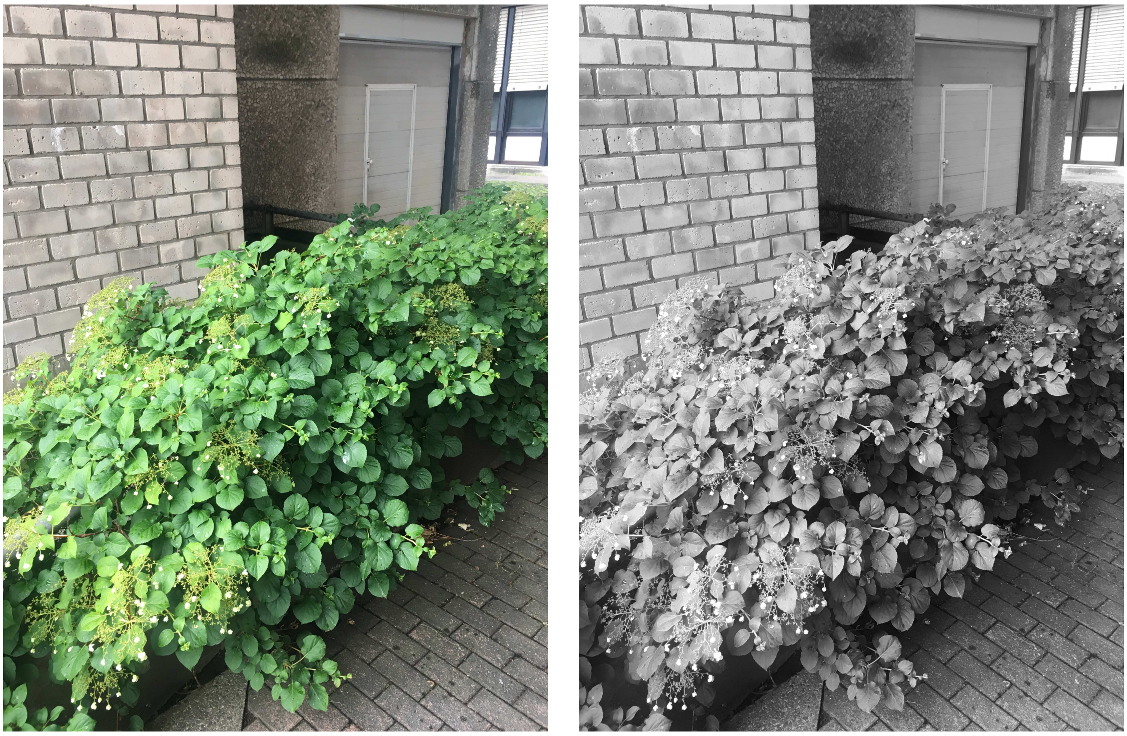


Figure 6.3: Converting the original image (on the left side) into the grayscale version (on the right side)

and color images. Although color images give us a lot of information, the whole information will not be necessarily useful. In many practical applications like corner detection and edge detection, we just need the information of edges or corners for further steps. Such information can be obtained by extracting features from a grayscale image. In this case, the information of the color image occupies more memory, too. Therefore, it is logical to investigate and reconsider the goals before making our decision about the type of the image. If we convert a color image into grayscale, it is similar to reducing noise from a signal as unwanted data is removed and the data becomes purely useful.

The answer to the second question is the key to the pre-processing step. In order to make our conversion closer to reality, we use a weighted average of three components, R, G and B. The role of the green component is unique in this conversion as brightness usually comes over by the green component. Hence, the coefficient of the green component has the largest value among all coefficients in our weighted equation. The equation used is as below:

$$Grayscale = 0.299 * R + 0.587 * G + 0.114 * B \tag{6.1}$$

This equation is applied to convert the input images of the system into grayscale images. The obtained grayscale images will be used in next steps to design and implement our desired systems.

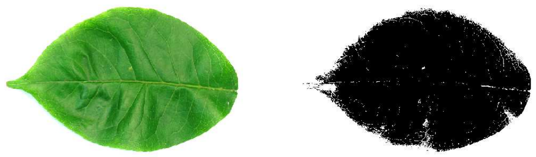


Figure 6.4: Converting the RGB image (on the left side) into the binary image (on the right side) using the threshold value of 150

6.4 Bag of Words

Local feature extraction involves interest point detection and computation of descriptors in the region surrounding those interest points. The first step of this stage is to detect features as an initial part for doing the feature extraction. An ideal detector detects the interest points even if some transformations such as scale and rotation have occurred. Interest points can be detected manually or, preferably, they can automatically be detected using some specific techniques. Features must be prominent, easily detectable and spread over the whole image. Good localization accuracy is an essential property that we expect from the feature detection technique in this step. Besides, feature detection should not be sensitive to the assumed image degradation. Furthermore, the used technique should be able to detect features regardless of image deformation and unwanted transformations such as scale and rotation in every situation.

The input of feature detection and extraction algorithms is a set of labeled training images. There are different techniques and algorithms to detect features.

In order to accomplish the task of plant recognition, one step towards the goals is to investigate useful and applicable algorithms for design and implementation of systems. We decide to consider a different procedure to detect keypoints and use the potential of each algorithm in the detection part. For the description part, we just use two modern description algorithms, the SIFT and the SURF. We combine these two algorithms with feature detection algorithms to obtain unique combined algorithms.

According to the description component, we divide the algorithms into two different subsets as below and compare the results in detail. For instance, if we would like to combine the HARRIS algorithm, the detection component, and the SIFT algorithm as the description algorithm, we use the term HARRIS-SIFT for our new algorithm. If the SURF algorithm is used to do the description part, the SIFT is replaced by the SURF. Therefore, the algorithm is called HARRIS-SURF.

- 1st subset: SIFT, HARRIS-SIFT and FAST-SIFT
- 2nd subset: SURF, HARRIS-SURF and FAST-SURF

As mentioned before, the combined term represents the names of the detection and description algorithms.

6.4.1 First Subset of Algorithms

At this stage, our first aim is to identify interest points which hold a high amount of local information. They are originally pixels with well-defined positions. These unique points can be used to describe the image. If we consider two different images of the same scene, useful interest points are those points which can be detected in both images. Additionally, higher repeatability means better robustness. As pointed out before, different available feature detection algorithms use various schemes. Since the 1980s, the HARRIS algorithm, a rotation-invariant method, has been used in different systems for detection of corners and edges because it is a combined technique of corner and edge detection. The basis of this algorithm are discrete image features, not continuum like the texture or edge pixels [10]. If the image is rotated, nearly the same corners will be found. When we have large-scale changes, the algorithm does not perform well [198]. Being invariant to rotation is helpful in classification tasks as any change in the position of objects in images influences recognition and classification results.

The SIFT algorithm is a rotation-invariant method which also owns other useful characteristics. This algorithm is invariant to image scale and highly distinctive. Therefore, other important characteristic could be added to the first algorithm. Moreover, it has no variance to variations of illumination, viewpoint and local affine distortions. To compute a 2D Gaussian function, the input image will be convolved by two passes of 1D Gaussian function in both horizontal and vertical directions. The SIFT algorithm is summarized as below:

- Scale-space extrema detection
- Keypoint localization
- Orientation assignment
- Keypoint descriptor

There are some differences between the first version of the SIFT algorithm [153] and its modified version [11] published in 2004. A modification was carried out to locate each keypoint at the location and scale of the candidate keypoint [153] for calculating the interpolated location of the maximum using quadratic Taylor expansion of DoG scale space function and best-bin-first (BBF) algorithm [199]. The modification leads to improvements of matching and stability of the selected image by approximating the closest neighbor with high probability.

The last used algorithm is FAST algorithm (revised in 2010) to do the detection part. One main reason behind the development of this algorithm was to struggle with limitations of detectors in real-time applications. For instance, vision simultaneous localization and mapping (VSLAM) [200] describes a mobile robot with limited computational resources [201]. Hence, the FAST algorithm is a weapon choice to find corners faster in real-time applications and tracking usages with limited computing resources. Obviously, one attempt towards real-time and industrial applications is to reduce the timing as much as it is possible without losing the needed information and essential contents of the image. This algorithm helps us to save the timing of the process. Another important characteristic of this algorithm is the large number of features that it finds in comparison to the SIFT and HARRIS algorithms. On the other hand, a difference between the FAST algorithm and the SIFT algorithm is that the FAST algorithm does not include any orientation operator.

Advantages of the SIFT algorithm and its performance against other algorithms made it the most used description algorithm [111]. Since detection algorithms like the HARRIS and the FAST suffer from lack of connectivity of interest points which show one limitation for obtaining descriptors, we use the SIFT to compensate for the defects of the algorithms in the description part. After finding keypoints, descriptors are generated according to the areas surrounding interest points for a set of labeled training images. For this part, the SIFT is used. A descriptor vector is computed for each keypoint and the dimension of the descriptor is 128. Although this value seems to be high, lower descriptors than it do not perform the task as well as it does. Also, computational cost is another aspect of the process. Undoubtedly, obtained descriptors should be rich enough to be usable at the category level.

6.4.2 Second Subset of Algorithms

The second subset of algorithms includes SURF, HARRIS-SURF and FAST-SURF algorithms. Results of the SIFT are invariant to changes of scale, rotation and variations of illumination, but its computational cost is high. One expectation is to reduce this factor in our systems. Hence, we decided to replace the SIFT algorithm with another modern algorithm called the SURF algorithm. The SURF algorithm is a sped-up version of the SIFT algorithm. The algorithm is rotation- and scale-invariant. In the SIFT, Lowe approximated LoG with DoG for finding scale-space. The SURF goes a little further and approximates the LoG with box filter. One big advantage of this approximation is that convolution with box filter can be easily calculated with the help of integral images. It can be done in parallel for different scales. Also, the SURF algorithm relies on the determinant of the Hessian matrix for both scale and location. As a robust local feature detection method, it works much faster than the SIFT algorithm.

Generally, the goal of a descriptor is to provide a unique and robust description of an image feature, e.g. by describing the intensity distribution of pixels within the neighborhood of the point of interest. Therefore, most descriptors are computed in a local manner. A description is obtained for every point of interest identified previously. The SURF descriptor is based on the similar properties of the SIFT, with even further stripped down complexities. The first step is fixing a reproducible orientation based on the information from a circular region around the interest point. The second step is constructing a square region aligned to the selected orientation and extracting the SURF descriptor from it. A descriptor vector is computed for each keypoint and the dimension of the descriptor is 64. This value is less than that of the SIFT algorithm with dimension of 128. The SURF algorithm has a lower dimension, higher speed of computation and better distinctiveness of features. Obtained descriptors will be used to find similarities among different images.

6.4.3 Bag of Words Model

BoW is one of the most important and competitive concepts in NLP [202] and information retrieval (IR) [203] such as text retrieval [204]. The first reference to the model is [205] which is in a linguistic context and combinations of elements are inspired by other algorithms. Recently, this unique strategy [206] [195] has been widely applied to computer vision tasks to solve complicated problems. A common point is existence of a large volume of obtained features and information. During the desired classification process, the importance of correct representation of the features is undeniable. Let us consider an example to clarify the concept of the BoW model in a text document, which contains two sentences as below:

1- One plant recognition system distinguishes different plant species.
2- We implement a plant recognition system.

Based on the sentences and their contents, we are able to create a list of structures (dictionary) as below:

["One":0
"plant":1
"recognition":2
"system":3
"distinguishes":4
"different":5
"species":6
"We":7
"implement":8
"a":9

]

Then, we can form a vector to represent each sentence according to the structure. Each element of the vector shows the frequency of each word in each sentence. For instance, if we check "plant" and its occurrence in the first sentence, we can see that it is used twice. Therefore, the frequency for this word is 2. The generated vectors are:
- Vector 1 for the first sentence: [1, 2, 1, 1, 1, 1, 1, 0, 0, 0]
- Vector 2 for the second sentence: [0, 1, 1, 1, 0, 0, 0, 1, 1, 1]

This approach is helpful because of two important factors, simplicity and flexibility. By using this approach, we will be able to obtain a dictionary of known words and a measure of occurrence and presence of each known word. As a result, we can build a histogram of words within a text. Additionally, we can consider the count of each word as a feature [207] to quantify and analyze documents and texts efficiently. One may observe that it is possible to ignore the location of words in this model. This very concept can also be applied to computer vision and image processing while using image patches instead of words.

Someone might ask, "Why has the BoW model become important in Computer Vision and Image Processing fields? And what are the strengths of the BoW model?" Recently, we have encountered two significant breakthroughs which led to a revolution in the fields mentioned above. The first strength of the BoW model is development of algorithms to extract discriminative low-level local features like the SIFT, SURF and HOG algorithms. As discussed before, the origin of the BoW model is representation of text data. The second strength of the BoW model is the possibility of using it to represent mid-level representations, increasing the level of representation of local features and providing an output with new vertical representations of images in a manner that can be used for potent statistical machine learning models. To model visual attention maps, outputs of BoW models can also be used [208] [209] [210].

Fei Fei Li et al. [211] proposed different models such as BoW model, Part-based model [212] [213] [214], etc. for object recognition. Apart from the model, one important point is the improvement of computational time; because the system is involved in lots of extracted information. There is a need for fast multi-label classification in addition to the robustness of the approach. Hence, the BoW model is chosen for our systems to make use of its advantages.

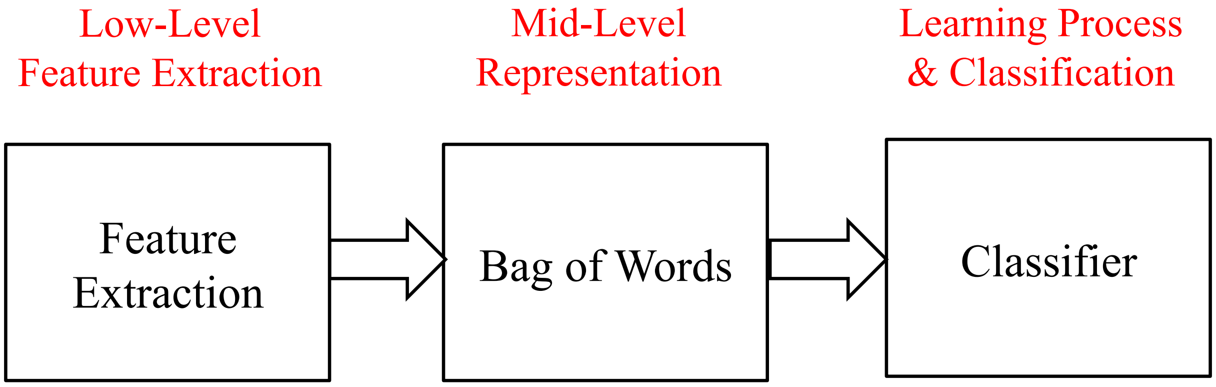


Figure 6.5: Low-level and mid-level information

It should be pointed out that part-based models are a wide range of models for object categorization problems. The origin of this type of models dates back to the work of Fischler and Elschlager in [215]. Part-based on manual labeling of parts and the constellation model [216] are two examples of the part-based model.

We study the pipeline of the BoW model in this section. Figure 6.5 illustrates representation levels of the model where the input extracted features are encoded into a visual vocabulary (dictionary of words). As we observe, the ultimate desired representation is obtained which is a mid-level one. The whole BoW model contributes to generating and preparing the features for the training step.

Classification Systems Based on Bag of Words Model

In this section, we provide a short review of using BoW models in classification tasks. Several existing examples in the literature on classification systems propose BoW as a basis of the system. In 2007 [217], a model was presented for human action categorization by proposing a hierarchical model. The idea was to make full use of both the geometric power of constellation model [216] and the richness of the BoW model. It is worth mentioning that the constellation model is a generative model [216] for representing the target categories. In this model, probability functions are used to show the objects of a class by considering the geometric relationship between different parts. Deselaers et al. [218] proposed an approach to classify and filter pornographic images from network traffic, and the approach was based on the BoW model. The system provided flexibility for the user, and the final result demonstrated good performance. In [219], the concept of the BoW was employed for semantic texton forests, and the results proved that the approach contributed to the state-of-the-art by reducing computation expenses and providing an image-level prior for segmentation.

In 2010, an interesting research was conducted by emphasizing the utilization of codebook discrimination information among various scene classes, and the authors proposed an improved approach based on the BoW model for scene recognition [220]. The purpose was to get a new weighted histogram and this histogram was obtained by incorporating information of a co-occurrence matrix [221] and a K-means algorithm [222] into the original BoW histogram. The BoW was used in [223] for describing complex objects in very high spatial resolution imagery and for classifying challenging objects in aerial images. As a result, the authors proved that the combination of spectral and texture features led to high classification accuracy in such images. To solve the problem of face and expression recognition, an active area of research, a system based on BoW was proposed in [224] by considering holistic and local features. Furthermore, an interesting part of the proposed system in [224] was simultaneous extraction of discriminative local facial features and maintenance of holistic spatial information.

As we know, we usually involve two serious challenges for classification of synthetic aperture radar (SAR) [225] images, and the goal is to solve the main problem by considering the challenges. In fact, the challenges are achieving an appropriate representation of features and finding a suitable pattern classification approach. In [226], the first challenge was solved by using BoW and an efficient representation of SAR images was achieved. A new extension of the BoW formalism was proposed in [227] and a flexible formalism was introduced. By dividing the process into BoW, dictionary coding and pooling, they applied a density function-based pooling strategy to improve the representation of the links between codewords of dictionary and descriptors in the resulting image signature [227] and tested the proposed approach in video and image classification tasks.

In 2015, a classification framework [228] was proposed for binary shapes with changes in scale, rotation and viewpoint. To classify animal shapes, invariant features and contextual information were incorporated in the BoW model and resulted in a significant performance of the implemented system compared to other projects in related literature. The application of the BoW model has not been restricted to the mentioned research, and it has also been used in satellite imagery. Yuan et al. [229]

proposed an automatic cloud extraction method. This method consisted of segmenting images into superpixels, computing dense SIFT descriptors from each superpixel, constructing compact feature vectors through the use of the BoW model and building a classifier. Interestingly, the authors proved that the proposed approach was not sensitive to the number of codewords in the codebook obtained by the BoW model.

As discussed, the BoW model has entered into many different areas and it has been used in various applications to fulfill desired goals. Peng et al. [230] provided a survey and comprehensive study of BoW and different fusions methods for recognition of actions. In addition, a simple and effective representation method, called hybrid supervector, was proposed and the obtained results on several datasets were impressive. Another work was proposed for scene classification and high spatial resolution (HSR) imagery [231] based on BoW. Different types of features, including global feature, local spectral and structural features, were employed to fuse local and global features at the histogram level and to create a local-global feature bag-of-visual-words (LGFBOVW) scene classifier [231]. The BoW has made an influence in social media resources, and a new and efficient approach was proposed in [232] to improve the original BoW model by designing a fuzzy membership function which was able to measure the similarity between the features and words.

Analysis of BoW's Approach

In order to embark on the BoW model used in our systems, we need to explain the process of building the visual model (visual model as the words are actually referred to images and their patches) and the vocabulary for the dataset. In the previous step, we attempted to get the set of descriptors in each image of the training dataset. To achieve our purpose, we used the SIFT and SURF algorithms to compute the image descriptors. If we perform the SIFT algorithm on a given image, the result is an $N \times 128$ dimension descriptor where N is the number of features. The next step to fulfill the BoW's approach is to construct a vocabulary by a clustering algorithm which consists of cluster centers. Our tendency is creating clusters of similar features and assigning words to them. The user is able to decide on the number of clusters. If we suppose that the number of features is equal to N and we select k number of clusters, there will be a model within k clusters which affects the size of the vocabulary. Consequently, the features will be distributed and separated due to the number of clusters. The vocabulary contains local patterns in images.

One important concern is similarity. In our approach, the similarity is determined by the Euclidean distance between descriptors without considering if the description step is done by the SIFT or SURF algorithms. The implemented Brute-force matcher in OpenCV is so helpful for computing similarity by using the Euclidean distance. Similar descriptors are clustered into K (where K equals 1000) number of groups. The range of K usually varies between 500 and 4000 in the literature. We use the K-means clustering algorithm which is basically a vector quantization method. Actually, descriptors are quantized. As a standard practice, each feature is put into the cluster which the feature has the minimum Euclidean distance from the cluster's center. Thereby, each image is finally grouped into its particular visual words. In other words, the clusters are named visual words and they represent the vocabulary collectively. Each cluster has a cluster center which can be thought of as the representative cluster of all the descriptors belonging to that class. Here, each cluster is a visual word and represents a special pattern by the keypoints in the cluster. The cluster centers are found and used to group input samples around the clusters. An equivalent histogram is formed and it contains bins that are equal to the size of the vocabulary. For each feature obtained from the SURF algorithm, the feature is assigned and quantized to its cluster optimally and then plotted in the histogram. Categorization of all features cools off the heat of the problem and we have to solve a multi-classification task.

Before explaining the next part, we would like to have a short glance at the K-means clustering algorithm with Euclidean distance. Suppose that we have 8 different observations ($i = 8$) consisting of two variables, x_i and y_i. If our final goal is to have two clusters, we then need to initialize two pairs which are actually the centroids of the clusters. After that, we have to compute the Euclidean distance between each observation and initial centroids of clusters 1 and 2. Then, each observation will be assigned to one of the clusters based on the minimum Euclidean distance.

It is considered that information of each detected keypoint in an image is mapped to a certain word through the clustering process. Hence, the image can be represented by the histogram of the visual words and each keypoint is encoded. After mapping the keypoints to visual words, each image can be represented as a bag of visual words. The obtained vocabulary should be large enough to distinguish relevant changes in image parts, but not so large to distinguish and recognize irrelevant variations such as noise. In doing so, novel image features can be translated into words. Translation of the extracted features into four words, G1, G2, G3 and G4 is shown in Figure 6.6.

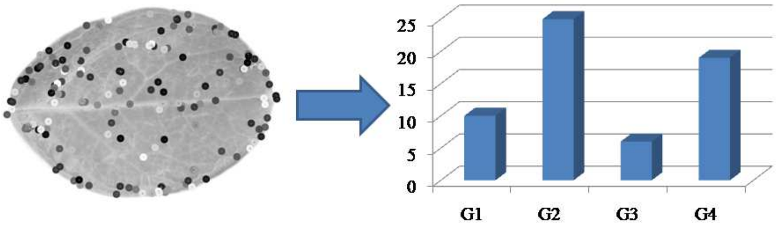


Figure 6.6: Translation of the extracted features into four words, G1, G2, G3 and G4

6.5 Classifier Training

Our approach has been sparked by feature detection and description algorithms and following by BoW model. Now we would like to answer the remaining questions to continue the approach. Some of the questions are as below and we are going to answer them to complete and finalize our approach properly.

1- What does the obtained result of the BoW model tell us?

2- How can we be capable of using the obtained result of the BoW for further processing?

3- As we have confronted 32 different plant species, how can we point out the optimum solution for the remaining step of the approach?

In order to answer to these questions, we did not confine ourselves to previous explanations since we aimed to classify a large number of plant species and get an automatic plant recognition system in the end. By generating vocabulary, every image is represented by a histogram of how often local features are assigned to each visual word. In such representation, the frequency, but not the position, of words is used to show text documents. Due to existence of 32 different plant species, we have to solve a multi-label classification task. In the literature of object recognition-based classification, one may find many different approaches and methodologies such as random forest, Naive Bayes classifier [233] [234], adaptive boosting (AdaBoost) [235] [236], expectation-maximization (EM) algorithm [237], etc. to build classifiers. Bag of Words is exactly the lost piece of the puzzle for connecting the description data to the training phase. It gives us isolated and meaningful data to organize the next steps. In fact, the BoW model prepares a large volume of data from 32 different classes in a useful form to help us in solving the classification problem. But, the question is "Which classifier is useful for our goal?" In this part, we investigate some common classifiers and machine learning algorithms.

6.5.1 Investigation of Machine Learning and Classifier Approaches

The training phase is surprisingly a significant component of classification tasks and helps us finalize a model and do the prediction task for new observations. A classification example would be determining if the given sample is a grass or not. Figure 6.7 shows a simple example of a classification task. In this case, the classifier does the classification of a given sample and determines whether it is a grass or not.

Machine learning is generally grouped into supervised learning and unsupervised learning. There are a lot of approaches which can be listed and used as machine learning algorithms. Therefore, brief descriptions of some machine learning algorithms are provided.

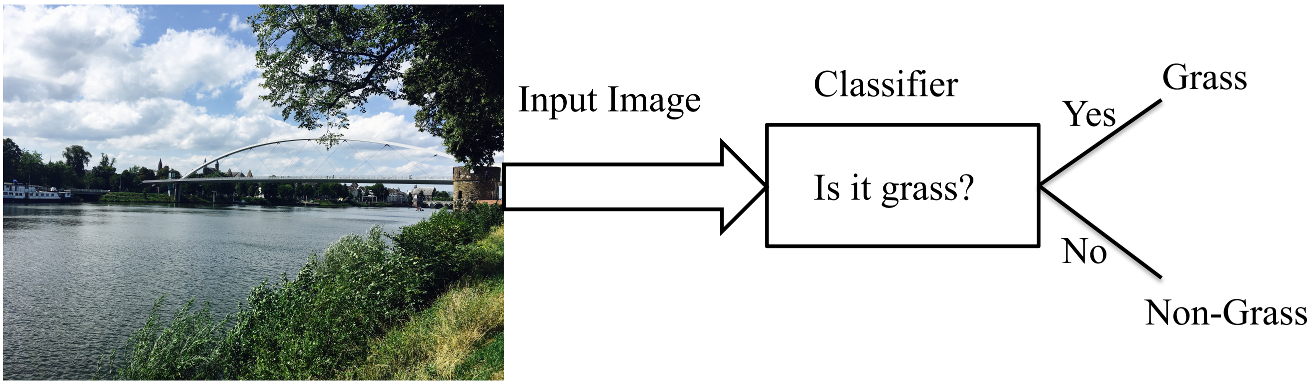


Figure 6.7: Classifying the input image whether there is grass or not

Random Forest Algorithm

The decision tree [238] is a data mining model similar to that of a flowchart, a tree structure for decision making, and the assignment of a class and a category of particular data. As its name implies, this tree is composed of a number of nodes and branches. The middle nodes are also used to make decisions according to specific attributes. In principle, a decision tree is a predictive model. It can be utilized for visual and unequivocal representation of decision and decision-making. Multiple decision trees can be merged together to build a useful concept in the machine learning field and it is the random forest.

The main concept of the random forest algorithm, an ensemble learning method, can be implied by its name. This algorithm can be applied in both classification and regression tasks and it lies in supervised learning. In the real world, each forest has a number of trees. In the world of machine learning, this algorithm also tries to create a forest with its essential elements, trees. In 1995, the first random decision forest [239] was proposed based on using the random subspace method [240]. The algorithm has been extended in [55] and [241] and it has become a trademark [242]. If we compare a real forest with the algorithm, we will see that a forest with more trees seems to be more robust in the real world. A forest with more trees leads to higher accuracy in the world of machine learning. In addition, more trees in this algorithm prevent overfitting of the model. The algorithm begins with decision tree learning [243]. The following Figure 6.8 explains how a decision tree algorithm works.

Naive Bayes Algorithm

Naive Bayes algorithm [244] is a simple technique to create efficient probabilistic classifiers based on Bayes' theorem [245]. This type of classifiers seems to be simple, but they are able to carry out

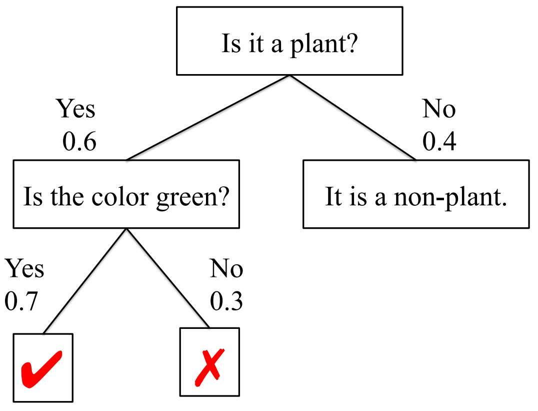


Figure 6.8: Representing how a decision tree algorithm works

sophisticated tasks for high dimensional inputs efficiently. Its performance is remarkably outstanding even for large datasets. One particular characteristic of the algorithm is the independent assumptions between predictors. Another important point is the ability of Naive Bayes classifiers to handle an arbitrary number of either continuous or non-continuous features. The foundation of the algorithm is posterior probability. We suppose that the data is $X = \{x_1, x_2, ..., x_k\}$ where it is representing n features. These features are actually independent variables. We would like to create a posterior probability for one of the events from the set of possible data (classes) $C = \{c_1, c_2, ..., c_k\}$. According to the Bayesian theorem, the following equation is applicable as the conditional probability:

$$p(C_k|x) = \frac{p(C_k)p(x|C_k)}{p(x)} \tag{6.2}$$

We can also rewrite the above equation in plain English, therefore:

$$posterior = \frac{perior \times likelihood}{evidence} \tag{6.3}$$

Support Vector Machine Algorithm

Support vector machines (SVMs), also called support vector networks [246], have opened new doors to efficiency and applicability with colorful views in machine learning in analyzing data for classification and regression problems, although they have been mostly used for classification problems to label different objects or observations. Reasonably, SVMs are supervised algorithms capable of dealing with highly complex tasks.

The basis of the SVMs is the concept of decision planes. It contributes to defining decision boundaries. What a decision plane does is to make an optimal separation between a set of objects belonging to various classes. The important element of the SVMs is support vector. In a simple example, a support vector is the coordinates of a single observation. The SVM model attempts to handle and segregate multiple data in the best possible way. By constructing support vectors (hyperplanes) in a multidimensional space, SVMs are able to carry out classification tasks and perform the estimation relationship between variables as a regression task. Novelty detection [247] and outliers detection [247] are also other prominence tasks which can be done by SVMs.

In classification tasks, two brilliant ideas of maximizing the margin and the kernel trick [248] [249] help SVMs work on unknown samples appropriately and gain optimal results accurately. In addition, high accuracy of classifiers can be guaranteed and the problem of the curse of dimensionality will be solved. The following questions are in focus with regard to SVMs as our goal is to implement an efficient plant recognition system with a number of different plant species.

1- Will the SVMs algorithms be helpful and reliable in creating a model for our desired goals in plant recognition?

2- Which type of SVMs is useful for our system?

3- What is the exact data that is needed to be fed into the SMV algorithm? Is the prepared data in the correct form?

4- What are the advantages and disadvantages of the SVMs?

To answer these questions and some other questions related to our system and the running step, we need to present a brief and high-level description of SVM algorithms.

In classification tasks, we are involved with two different datasets, the training dataset and the testing dataset, which consist of leaves' images of different plants. We apply some modern detection and description algorithms to obtain succinct information of the images. Then we model this information in the ways supposed to be useful for our classifiers. In addition, the label and features of each image in the training dataset are also known.

Firstly, we want to express a simple classification task with the SVM algorithm. In this case, if we have two plant species of our dataset as our two labeled classes, the main goal is to separate the available two plant species by using a function. This function is derived from the available data and information from the plant species of both classes. If the samples are spread as represented in Figure 6.9, a linear classifier is able to separate them. However, there are different lines that can do the separation procedure. A good classifier is the one that works optimally and maximizes the distance between the data and the line performing the classification of the samples of the plant species. Thus, this line is the best choice as one of the expectations is being general for the classification task. It is the idea behind the SVM classifier to select the maximum margin. If we do not consider this maximization, the noise will have unwanted effects on the performance of the classifier. Therefore, new predictions will be badly destroyed.

The red line (hyperplane in general) fulfills the constraint of the maximum margin between the classifier and samples in the represented space.

One other important question is, "How can we deal with the data which is not separable linearly?"

The initial idea is to try to map the data into a higher dimensional space where one can separate the data linearly and achieve a linear classification. In order to achieve this goal and build a higher dimensional feature space, kernel functions are a good choice. They can be applied to the separation of the data. Although a lot of kernel functions have been developed, it is also possible to develop custom kernels. Some of the standard kernel functions are as below:

- Polynomial (homogeneous):

$$k(x,y) = (x^T y)^d \tag{6.4}$$

- Polynomial (inhomogeneous):

$$k(x,y) = (x^T y + 1)^d \tag{6.5}$$

- Gaussian radial basis function (RBF):

$$k(x,y) = exp(-\gamma ||x-y||^2) \quad \text{for} \quad \gamma > 0 \tag{6.6}$$

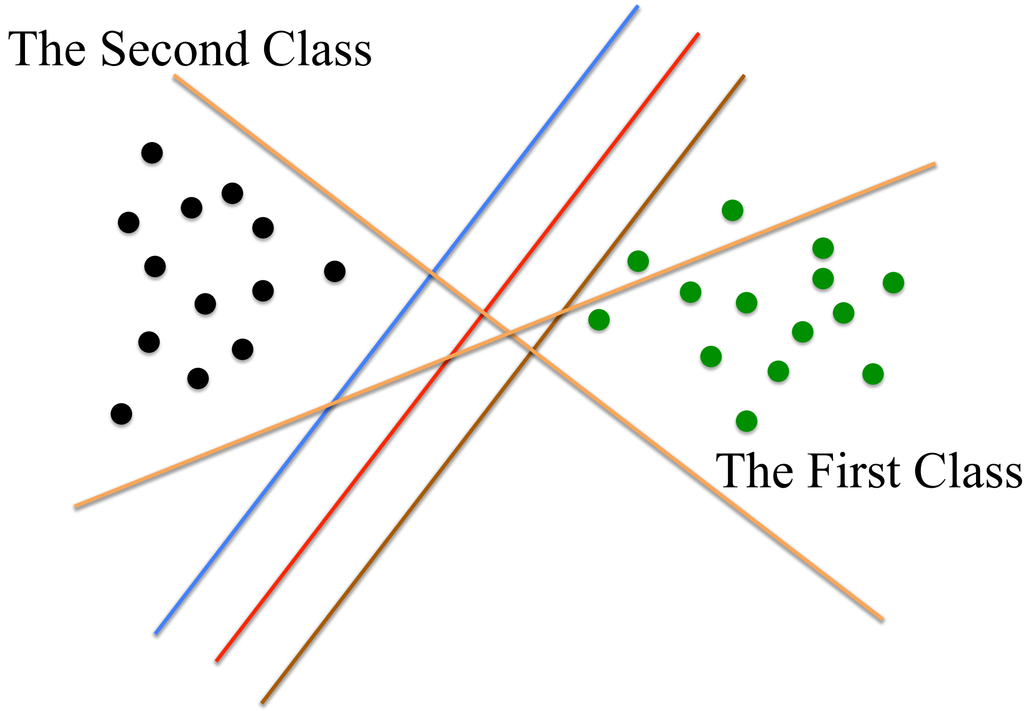


Figure 6.9: Representation for classifying two classes of plants

γ is equal to $\frac{1}{2\sigma^2}$
- Hyperbolic tangent:

$$k(x,y) = tanh(kx^T y + c) \quad \text{for some (not all)} \quad k > 0 \quad \text{and} \quad c < 0 \tag{6.7}$$

It has been proven in [250] that if kernel function $k(\vec{x_i}, \vec{x_j}) = \Phi(x_i) \cdot \Phi(x_j)$ satisfies the Mercer condition [251], in some transformation spaces, the function corresponds to the inner product and we have, $\Phi(x) \cdot \Phi(x_i) = k(\vec{x}, \vec{x_i})$.

The beauty of kernel functions is hidden in the transformation of nonlinear spaces into linear ones, providing proper inputs for classifiers and sending back the results into the original spaces. The flavor of flexibility can be added by SVM classifiers. It has been observed that SVM plays an important role in an efficient classification with high accuracy and complete theory. In addition, it is possible to have nice theoretical guarantees with regard to overfitting if an appropriate kernel is used. Simple structure, high adaptability, global optimization, short training time and good generalization performance are other advantages of SVMs [252]. The mentioned points make these types of classifiers interesting for us to explore. SVM algorithms are similar to pruning machines for cutting branches and creating pure models which are not only theoretically applicable but also practically assuring good final results in the plant recognition.

Although there are advantages to SVMs, there are still some drawbacks in SVMs from different standpoints. Generalization of SVMs is an important characteristic of these algorithms, but they are not fast enough in the testing phase [253] [254]. Since there are different options to select as kernel function, we have still the possibility of selecting a function freely. It can be considered the positive side of SVMs. Also, it can be considered a disadvantage of SVMs because the entire performance of the algorithm lies in the selected kernel function. Therefore, it is essential to find and choose the best kernel type according to desired goals.

In addition, each kernel function has some parameters. Appropriate selection of the parameters' values is also another side of this issue. We can assign different values to one specific parameter. Then we can examine how the algorithm is working with different assigned values. It results in comparing the implemented systems with different values. But, it is really difficult to determine the optimum value for each parameter and achieve the best possible performance. Hence, some advantages of SVMs wear other clothes unconsciously and become disadvantages in some cases. We are able to consider this fact as an opportunity for enhancing the final performance of the implemented system. Another point is tracking the behavior of kernel parameters and the final performance while initialization of one parameter has changed. If we incorporate the weights of advantages and disadvantages of SVMs, we will grasp the usefulness of SVMs in building efficient models for plant recognition systems.

Another point remaining is how to feed the data into the SVM algorithm. To this stage, the extracted features are quantized by using the K-means clustering algorithm. After making a decision on the size of vocabulary, the chosen size is used as the number of clusters. Then the center of each cluster is found for next steps. For instance, if we create an 800-word visual vocabulary, the number of clusters is equal to 800. We are able to create a histogram for visual word occurrences according to visual word indexes and frequency of occurrences. It results in a new representation of original images in an encoded format. This new format of data aims to provide a reasonable intelligence for classification of unknown data by feeding this data into SVMs.

There are different kernel functions which can be used for the training phase. Some of kernel functions such as linear, polynomial and sigmoid are functions of the inner product of the data. But, the RBF kernel function is a function of Euclidean distance between the points of the data. Based on the Euclidean distance, similarity depends on how close the points are. This concept helps to perform better in some cases. For instance, if there are two points close to the origin point while located on opposite sides, the Euclidean distance leads to higher values for these points whereas the kernel functions based on the inner product give lower values to points. In this case, the result is not correct.

One other important property of the RBF kernel is its smoothness that is also controllable. In signal processing, we usually use low pass filters to smooth signals. The RBF kernel function in image processing is also a low pass filter that selects out the smoothest solution. There is a direct relationship between the smoothest solutions and the fastest convergence of sum of high order derivatives. Moreover, by mathematical investigation of the RBF kernel function's formula, we can figure out that better performance of the kernel function happens when there is an infinite sum of high-order derivatives for fast convergence. However, the function is complex with an infinite sum of components. The outcome of this property is fitting of smooth solutions. It can contribute to producing more separating hyperplanes and finally achieving the goal of recognizing different plant species. In addition to being optimal, this type of kernel is aesthetically non-parametric as the model is basically infinite.

Our insight into different kernel functions proves that the treatment of the problem is mathematical and clarification of properties is essential to construct a useful model in the training phase and empower the designed systems. The emphasis on parameter tuning cannot be eliminated for the training step. It should be mentioned that the performance of the model can be improved dramatically by adjustment of SVM's parameters. A detailed explanation of the used SVM is provided in next sections.

6.6 Experiment, Discussion, Results and Performance Analysis

Plant recognition has proven to be problematic when the number of species increases [169] [170] since shapes of some plant species are similar and multi-classification of many plant species is not an easy task at all. Additionally, the color of a plant of the same species changes in different samples, and some parts of the samples might be destroyed randomly. In order to build an efficient and stable plant recognition system for 32 different plant species of the Flavia dataset, we used a machine with the following specifications: Intel® Core^{TM} i7-4790K, CPU @ 4.00 GHz and installed memory (random-access memory (RAM)) 16.0 GB [169] [170]. All designed systems are investigated, applied and evaluated by the same machine. We used a testing dataset consisting of 648 images of 32 different plant species of the Flavia dataset [169] [170] and six different systems were implemented and tested for obtaining empirical results. All in all, six different automatic systems with various modern detection and description algorithms for plant recognition were implemented and evaluated in the mentioned testing dataset. The modern detection and description algorithms are: (1) SIFT, (2) HARRIS-SIFT, (3) FAST-SIFT, (4) SURF, (5) HARRIS-SURF and (6) FAST-SURF.

By using novel combined approaches, we are going to extend the existing knowledge and techniques which will result in the development of new plant recognition systems as well as the improvement of the techniques which can be used in other different fields. We will also have a look at the experiments performed and we will attempt to find the best results, in terms of accuracy, confusion matrix, precision, recall, number of detected keypoints and the time needed. Moreover, the implemented systems will be compared in two different groups based on the description component. Comparison of results contributes to the search for the minimum error and the best reliability. In each group, we propose a possible combination of two different modern algorithms to have more robust results in the recognition system using a classifier. In the following, the investigation of the methods and the procedure of the experiments is illustrated.

6.6.1 Some Important Metrics for Measuring the Quality of Classifier Systems

In this section, some important metrics for measuring the quality of the classifier systems are introduced. These metrics are accuracy, precision and recall.

Accuracy

Accuracy is a metric measurement that shows the number of the correctly classified samples (images) of the dataset divided by the total number of the test samples in the testing dataset. The percentage of accuracy is obtained by the following equation. In other words, it is an analogy that depends on the correct prediction of unknown samples of the testing dataset and the total number of the unknown samples of the testing dataset multiplied by 100.

$$Accuracy \quad of \quad Classification = \frac{c}{n}.100 \tag{6.8}$$

c is the number of correct classified images of the testing dataset where n is the total number of the images in the testing dataset. This measurement is usually the start point of classification problems. Other measurements will help get more information about the implemented systems and their accurate comparison. Classification error is another metric that can be derived from the accuracy. This new metric is obtained by subtracting the accuracy (or percentage of accuracy) from one (or 100).

Precision and Recall

Precision and recall values are two metrics that can be measured by using the constructed confusion matrix.

$$Precision_i = \frac{M_{ii}}{\Sigma_j M_{ji}} \tag{6.9}$$

$$Recall_i = \frac{M_{ii}}{\Sigma_j M_{ij}} \tag{6.10}$$

Universally, precision is the fraction of events where we is correctly declared i out of all the instances where the algorithm declared i. Conversely, recall is the fraction of events where we correctly declared i out of all of the cases where the true of state of the world is i. In fact, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned.

6.6.2 Experiment and Discussion of the Systems by the SIFT Component

In this section, we used the SIFT algorithm as the main component of the implemented systems in extracting features. Three different combined methods have been used to do the detection and description parts of the systems and obtain the final result by each one, respectively. The methods used are SIFT, HARRIS-SIFT and FAST-SIFT. Table 6.1 shows the accuracy of the implemented systems with each method.

Since the difference between the systems lies in detection and description parts, it is essential to investigate and analyze the characteristics of these modern combined algorithms. The highest accuracy has been obtained by the implemented system with SIFT algorithm for detection and description. The reason lies in the properties of this modern algorithm, especially in the detection phase of the algorithm. In the SIFT algorithm, extracted features are scale, rotation and contrast invariant. Keypoints of the SIFT algorithm are actually the extremum of a DoG scale pyramid. Then a set of potential keypoints locations is found and many keypoints are produced. But, these keypoints should be refined and purged to achieve more accurate, more useful and stable results because some of the keypoints are not stable enough for further processing. The appropriate step, in this case, is to do a detailed fit close by data for precise location, scale and the ratio of principal curvatures. This step helps remove low contrast keypoints, which are sensitive to noise and poorly localized keypoints along an edge. A Taylor series expansion of scale space is used to get a more accurate location of extrema. If the intensity at this extrema is less than a threshold value (0.03 as per the original paper), it will be rejected [169]. Furthermore, the higher response for edges has been provided by DoG, so edges need to be removed too. To achieve this purpose, a concept similar to HARRIS corner detector is utilized and a 2×2 Hessian matrix (H) is applied to compute the principal curvature.

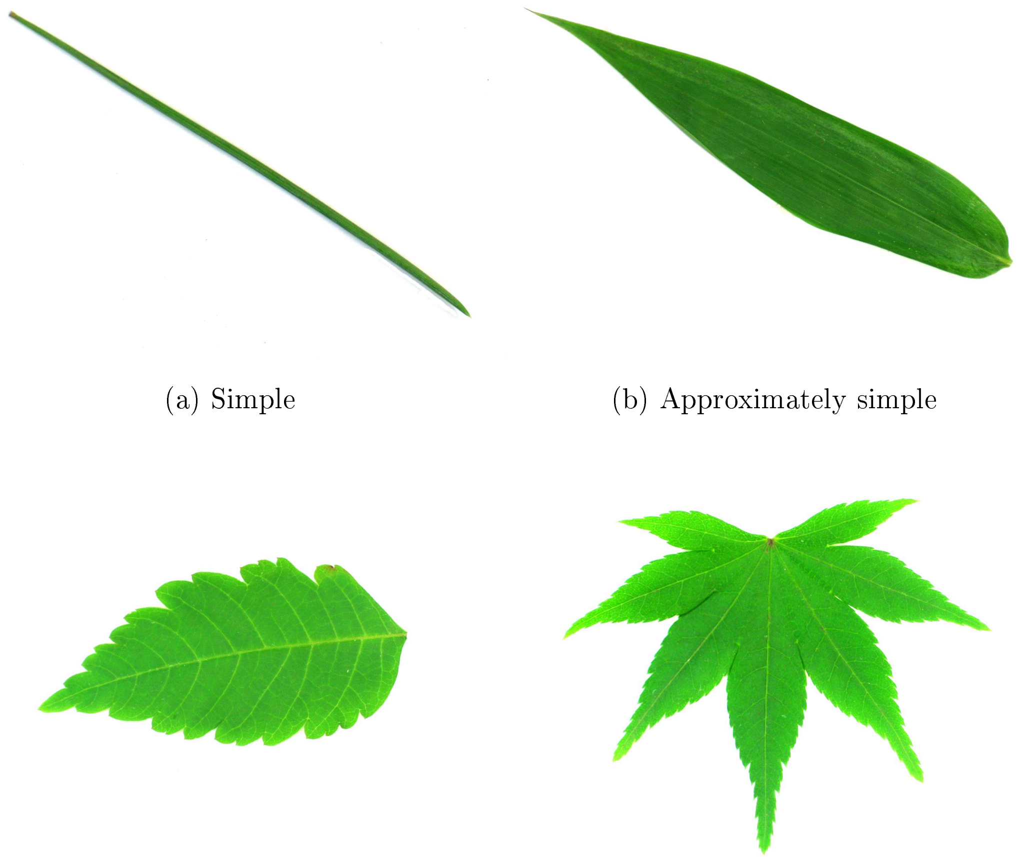
System	Percentage of Accuracy
Implemented System with the SIFT	89.35
Implemented System with the FAST-SIFT	81.94
Implemented System with the HARRIS-SIFT	80.4

Table 6.1: Accuracy of each implemented system [169]

For edges in the HARRIS corner detector, one eigenvalue is larger than the other. In order to achieve invariance to image rotation, an orientation is assigned to each keypoint. In addition, a neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and the direction are calculated in that region. The outcome is an orientation histogram with 36 bins covering 360 degrees. The highest peak in the histogram is considered as the criterion if the value of any peak is above 80% of the value of the highest peak. The orientation of that peak is also calculated. The final result of this step is a set of keypoints with same location and scale, but in different directions. Consequently, it contributes to achieving good stable results.

Now, the keypoint descriptor is created and a 16×16 neighborhood around the keypoint is taken. It is divided into 16 sub-blocks of 4×4 size. For each sub-block, an 8-bin orientation histogram is created. So, a total of 128 bin values are available. It is represented as a vector to form keypoint descriptor. The HARRIS detector is one of the components in combined modern algorithms. It is well-known to detect corners in images. In order to have reliable image matching, the level of the algorithm's invariance is of high importance. In reality, this detection algorithm does not provide the desired level of invariance, although it has been widely used for some computer vision applications. In addition, the accuracy of the implemented system with the HARRIS component is not the highest one.

The third explored algorithm was the FAST detector algorithm. This algorithm is used as the detection component of the combined algorithm, the FAST-SIFT. Basically, the FAST algorithm performs faster than other algorithms, SIFT and HARRIS, and enjoys a considerable improvement in computational speed. But the FAST algorithm is not very robust in the presence of noise. Less analysis of possible pixels leads to achieving high-speed property in the FAST algorithm, but the ability of the detector is reduced to average out the noise. There are many noisy features among the detected keypoints by the FAST algorithm and the number of keypoints increases totally. It should be pointed out that the noisy features are not appropriate for further tracking.



(a) Simple

(b) Approximately simple

(c) Approximately complicated

(d) Complicated

Figure 6.10: Sample images for calculating the number of the keypoints

Considering the discussed points with regard to the differences in detection algorithms, one im-

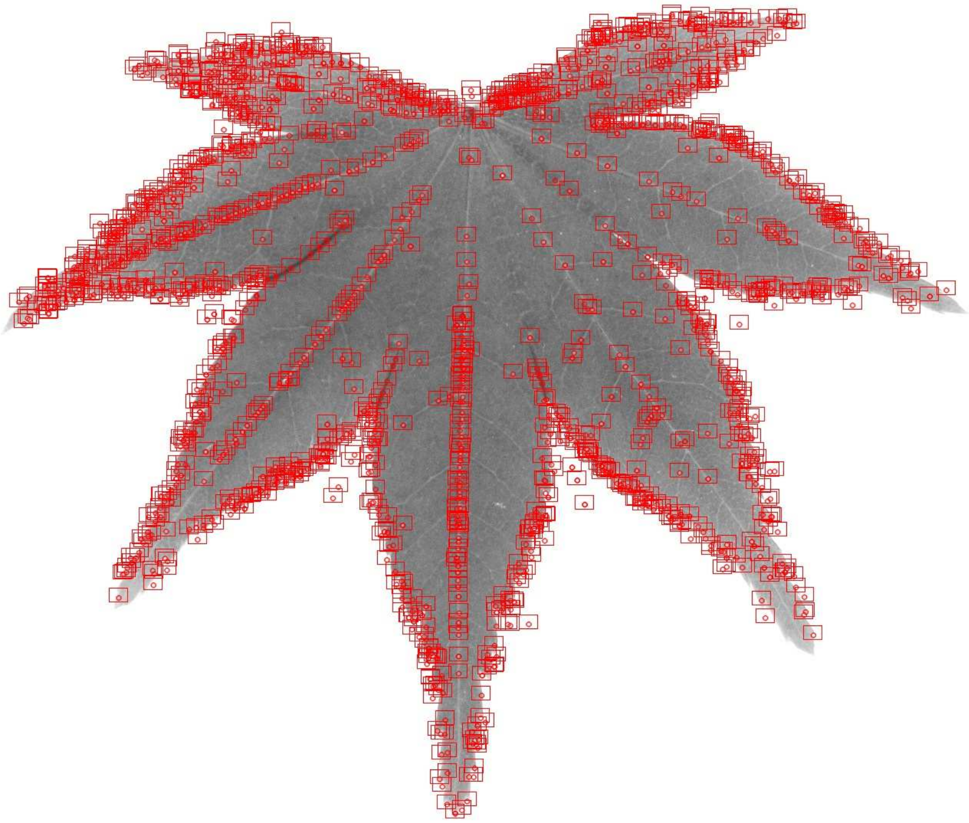


Figure 6.11: The detected keypoints using the SIFT algorithm [169]

portant issue is the number of the detected keypoints by means of different algorithms. We examined the number of the detected keypoints for four species of the dataset. The difference between the four species is the complexity pertaining to the human vision. Therefore, they are labeled as simple, approximately simple, approximately complicated and complicated. For the SIFT algorithm, the complicated one has the maximum number of keypoints while the obtained number of keypoints is minimum for the simple one.

For SIFT and FAST-SIFT algorithms, the number of keypoints is calculated and shown in Table 6.2 where Figure 6.10 also shows the images of the samples.

Number of keypoints in modern algorithm	Simple leaf	Approximately simple leaf	Approximately complicated leaf	Complicated leaf
SIFT	113	255	625	1656
FAST-SIFT	234	1040	1193	5894

Table 6.2: Number of the detected keypoints

Figure 6.11 and Figure 6.12 represent the detected keypoints for one leaf when the SIFT and FAST-SIFT algorithms have been applied to it, respectively.

The FAST algorithm finds thousands of keypoints, while the SIFT and HARRIS algorithms find only hundreds of keypoints. Detection with the FAST algorithm generates some noisy keypoints. A large number of keypoints mixed up with noisy keypoints cause decrement of classification accuracy of the implemented plant recognition system with the FAST-SIFT algorithm. By using the SIFT

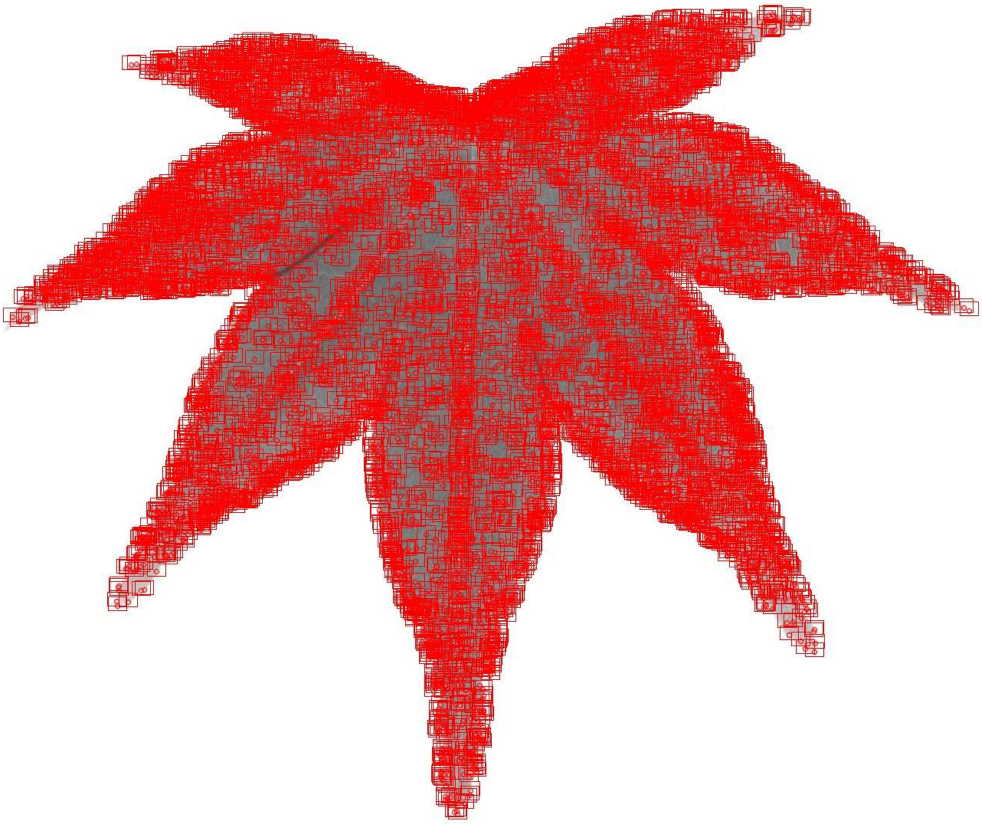


Figure 6.12: The detected keypoints using the FAST-SIFT algorithm [169]

algorithm, both the quantity and quality of the detected keypoints were adequate for our desired plant recognition system and these keypoints enrich the system for getting accurate and decent final result.

In order to compare the performance of the implemented plant recognition systems, we searched for a concept which could help us with a better evaluation of the systems. Two questions had to be answered, (1) "How long does the process of running the system on a new image take for finding the exact plant species?" and (2) "Is it useful to compare the required time of different recognition systems?" So, we measure and then compare the system from this point of view.

The system with the FAST-SIFT algorithm needs the lowest test time in comparison to other methods. The needed test time per image was measured for all of the three systems. Table 6.3 shows the needed test time for each system per image in milliseconds (ms).

Used Algorithm	Test time needed per image ms
SIFT	780.4300
FAST-SIFT	610.3900
HARRIS-SIFT	771.8700

Table 6.3: The test time per image [169]

So far, it has been discovered that performance of the FAST-SIFT algorithm performs better than

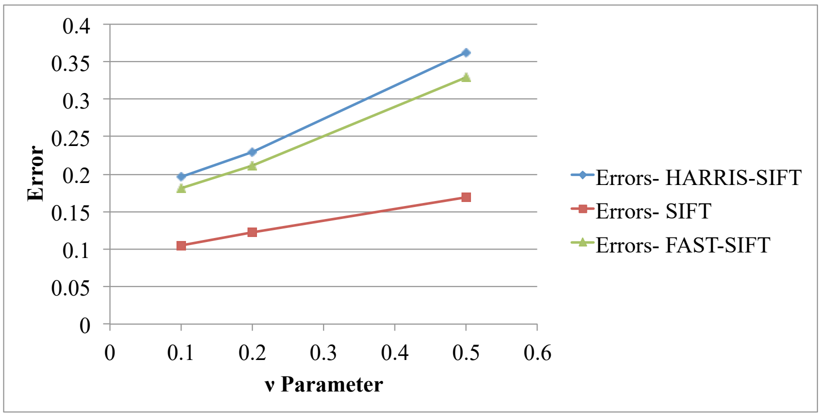


Figure 6.13: Variations of the ν parameter for the implemented systems respectively [169]

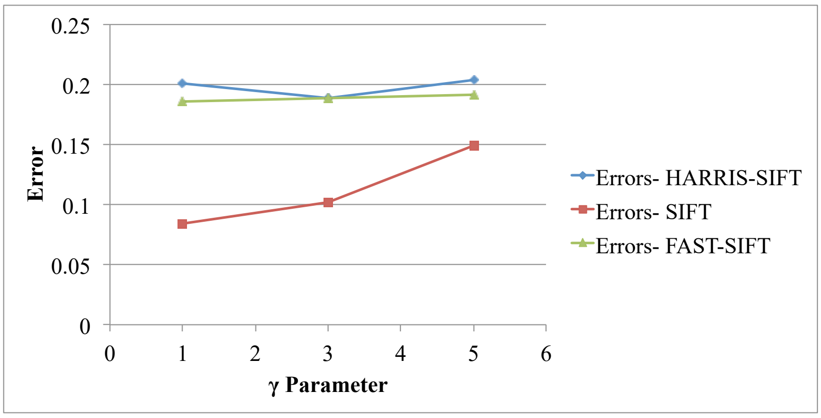


Figure 6.14: Variations of the γ parameter for the implemented systems respectively [169]

the HARRIS-SIFT algorithm if we consider both accuracy and the needed test time simultaneously. Probably, one reason is the large number of detected keypoints by means of the FAST algorithm. However, the increase in the number of keypoints leads to the detection of noisy keypoints. The reason behind a better performance in comparison to the HARRIS-SIFT lies in the increase of keypoints which speeds up the process. Accuracy of both systems is close, but the needed test time makes the system with the FAST-SIFT a better option in real-time applications for fairly good results.

Although very little information might be provided; in general, all detected pixels contain information. Descriptors use the relationship of the pixels and model them to have better information for further processing. Detected keypoints affect the results of plant recognition systems. The system with the SIFT algorithm has the best result among the implemented systems using other algorithms. High classification accuracy and acceptable time for finding the species of an unknown plant prove that the implemented system using the SIFT algorithm is the best choice from among the implemented systems in the first subset.

In the training phase, the RBF kernel has been utilized. The effects of varying some parameters on the final error for all three systems using the RBF kernel were investigated. The experiment was performed on ν and γ parameters. The ν parameter is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training examples. The value of the ν parameter is between 0 and 1. To do the experiment, the γ parameter is kept fixed equal to 1.0. Then the variation of the ν parameter was applied. As it is shown in Figure 6.13, error of each system increases when this parameter increases. By using the SIFT algorithm for the system, the increase in the ν parameter has less influence on the final result. It is the proof of the system's robustness against the varying ν parameter.

The effect of changing the γ parameter on the final error is also shown in Figure 6.14 while the ν parameter is held fixed at 1.0. There is a direct relationship between the increase of the γ parameter and the increase of the error. In comparison to the ν parameter, the impact of γ increase on the final results is less than the ν parameter for the implemented plant recognition systems.

An interesting matrix, named confusion matrix, is constructed. The confusion matrix is one $n \times n$ matrix ($n = 32$ in our case) containing information about the actual classification results (in its columns) and different category labels through the classification (in its rows). Confusion matrix of each implemented system is computed to get precision and recall values for each label of the systems.

In both Figure 6.15 and Figure 6.16, the minimum values among the systems belong to the implemented system with the combined HARRIS-SIFT algorithm. It was predictable for us as it had the least accuracy percentage between the three systems. The system with the SIFT algorithm has lees value variations than other systems in both figures. In comparison, the variation of the system with the FAST-SIFT is in the middle of the other systems and has the second rank as shown in the figures. The investigation of the precision and recall measurements illustrates that the sequence of efficiency is the SIFT, the FAST-SIFT and the HARRIS-SIFT.

The surrounded area is another concept to consider when comparing the systems. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate. High recall relates to a low false negative rate. In Figure 6.15 and Figure 6.16, larger areas belong to the system implemented with the SIFT algorithm. Therefore, this system performs better than other systems. Both high scores prove that the system is returning accurate results (high precision), as well as returning the majority of all positive results (high recall). It should be pointed out that the relationship between recall and precision can be observed for each system individually.

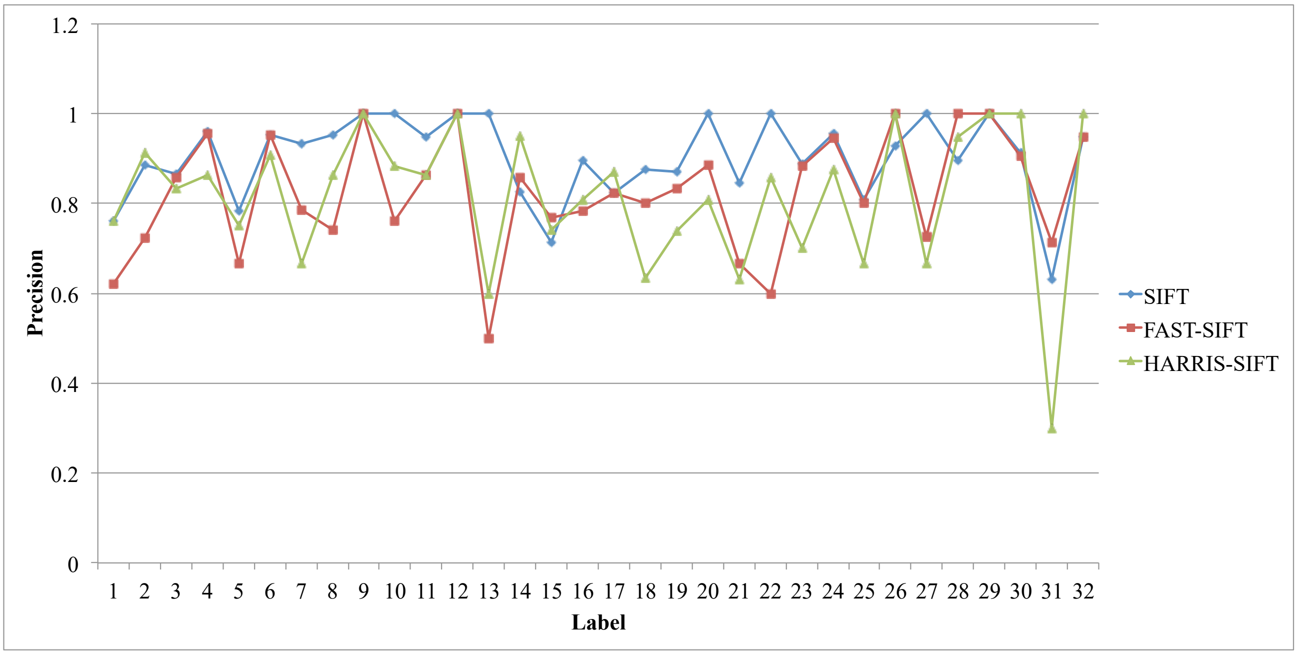


Figure 6.15: Precision measurements for the systems with different detection and description algorithms, SIFT, FAST-SIFT and HARRIS-SIFT [169]

If we consider the three implemented systems in the current set, we find out that the system using the SIFT algorithm plays the role of gold among other materials. It fascinates us to review more

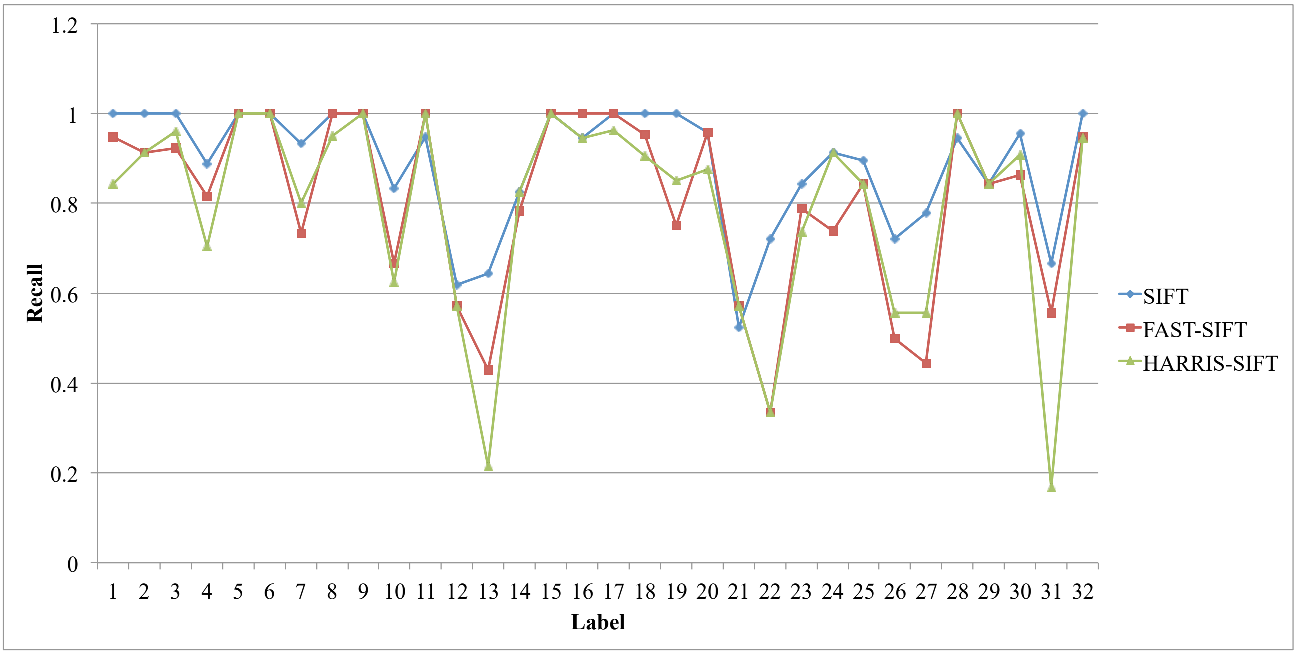


Figure 6.16: Recall measurements for the systems with different detection and description algorithms, SIFT, FAST-SIFT and HARRIS-SIFT [169]

details of the extracted information from the graphs for this system. As shown in the figures, the minimum value of recall is less than 0.6 for this system while the minimum value of precision is more than 0.6. Variation of values in precision is less than the variation in the recall and Figure 6.17 shows these variations. Also, more labels have values near the maximum and most of the values are in the highest interval that is [0.8, 1].

In Figure 6.18, the measurements of precision and recall are defined for the system with the

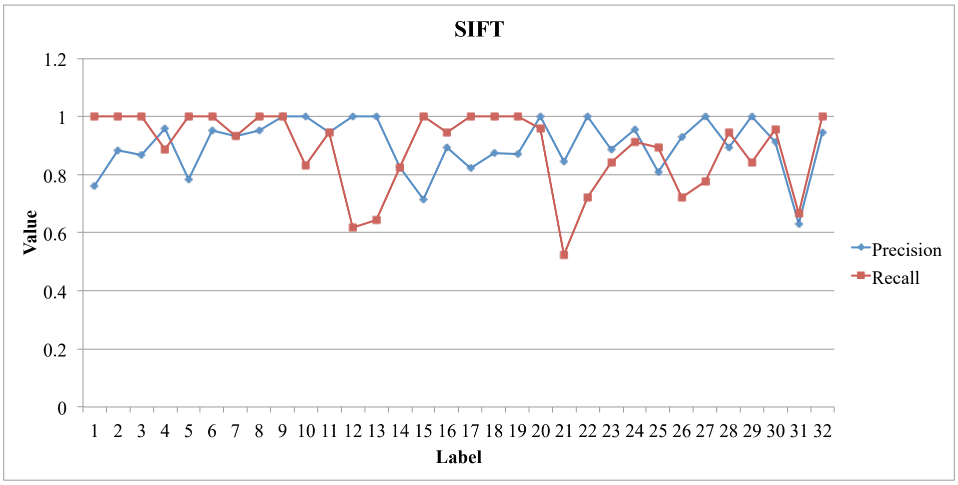


Figure 6.17: Measuring the precision and recall for the system with the SIFT algorithm [169]

FAST-SIFT algorithm. In this system, intervals of variation are larger than the system with the SIFT algorithm. In addition, the minimum value of the precision is 0.5. After an investigation into the minimum value of the recall, it is found that the value is less than 0.4 and equal to 0.3333.

The Figure 6.19 illustrates the precision and the recall of the system using the HARRIS-SIFT algorithm. The minimum values of the precision and recall are 0.3 and 0.1666. Concerning this system, the value intervals of precision and recall are larger than other systems. Additionally, this finding is

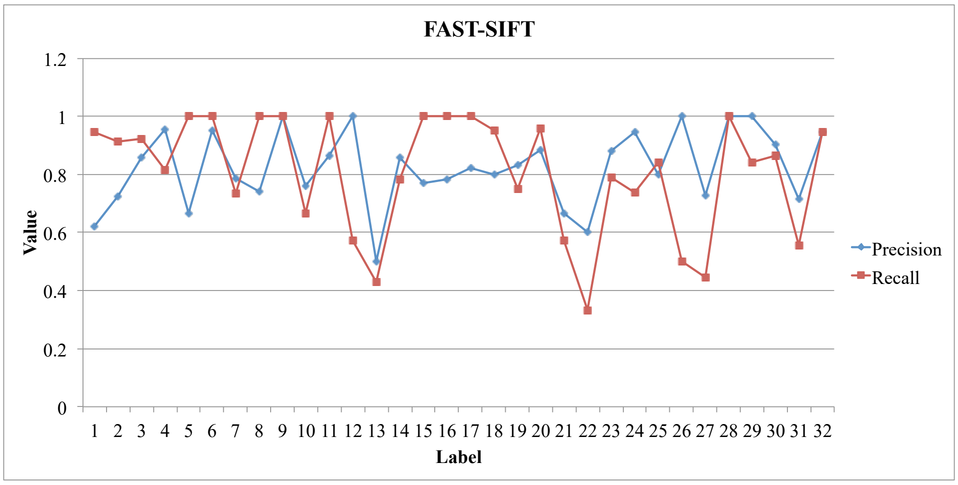


Figure 6.18: Measuring the precision and recall for the system with the FAST-SIFT algorithm [169]

expectable because the accuracy of this system is lower than other systems. Furthermore, a decrease in the measured values is also its clear evidence.

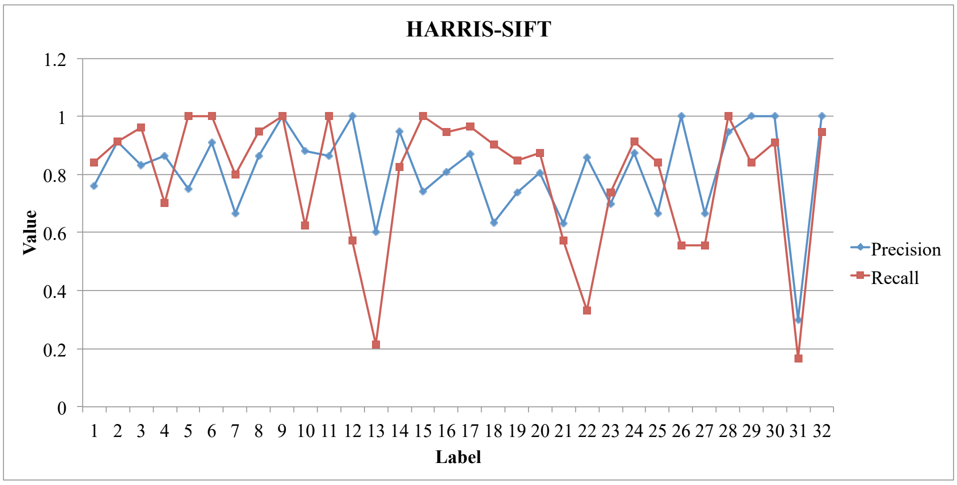


Figure 6.19: Measuring the precision and recall for the system with the HARRIS-SIFT algorithm [169]

6.6.3 Experiment and Discussion of the Systems by the SURF Component

In order to demonstrate the applicability and reliability of the proposed systems, 648 test images were tested by the systems using the SURF algorithm as the backbone of the combined used methods. The testing dataset was formed out of the images of 32 different plant species. They were used for the testing phase of the implemented plant identification systems. In fact, we were going to solve an inverse problem where we had just images without any additional information for the identification of plant species. So, disambiguation between potential systems seemed inevitable and we found the exact class (label) of the test image as the input of the system. Results of the recognition systems were evaluated by comparing the output of classifiers with different combined modern detection and description algorithms.

In this section, modern detection algorithms, the HARRIS and the FAST, are combined with the

SURF to form one part of the whole approach. In this part, we built three different plant recognition systems, and one of the modern algorithms was applied in each one. The used methods are actually SURF, HARRIS-SURF and FAST-SURF. One question might be asked, "Why do we use the SURF algorithm as the new component instead of the SIFT algorithm in [169]?" The purpose is to obtain better features leading to higher accuracy. Table 6.4 shows the accuracy of 32 plant species recognition results under different systems by the SVM classification algorithm with radial basic kernel [246] [255] and different combined modern and detection algorithms. The OpenCV library is available online at [256]. The accuracies shown in the table were calculated from running the proposed systems on a personal computer (PC) with Intel® Core^{TM} i7-4790K, CPU @ 4.00 GHz and installed memory (RAM) 16.0 GB [169] [170]. Of course, the performances of the proposed systems involve a trade-off between accuracy and other factors such as speed and needed time. Using the SURF algorithm leads us to obtaining the highest accuracy among other algorithms and the algorithms proposed in [169]. The system implemented using the FAST-SURF algorithm has a higher accuracy when it is compared to the proposed system using the SIFT algorithm in [169]. As mentioned before, all measurements of the experiments have been obtained by doing the desired experiments five times and averaging the measurements.

The maximum accuracy belongs to the system which was implemented by the SURF algorithm.

System	Percentage of Accuracy
Implemented System with the SURF	92.28
Implemented System with the FAST-SURF	89.66
Implemented System with the HARRIS-SURF	87.19

Table 6.4: Accuracy of each implemented system [170]

Since the SURF algorithm is a sped-up version of the SIFT algorithm, the proposed system with this algorithm is also speeded up version of the implemented recognition system with the SIFT algorithm where the other components are similar to each other. Additionally, the accuracy of the system with the SURF algorithm represents other aspects of its efficiency in comparison to other proposed novel systems. The SURF algorithm is clearly invariant with regard to scale, orientation and illumination. The SIFT algorithm leans on the DoG for finding scale-space, but the SURF algorithm goes further and approximates the LoG with a box filter. One may still ask what the main advantage of this approximation is.

The momentum for the proposed approximation is that convolution with a box filter can be simply calculated with the contribution of integral images. In addition, it can be performed in parallel for different scales. The SURF algorithm relies on the determinant of the Hessian matrix for both scale and location. For orientation assignment, the SURF algorithm utilizes wavelet responses in horizontal and vertical directions for a neighborhood of size $6s$. In order to give weight to the obtained responses, a Gaussian function centered at the point of interest is utilized. They are then plotted in a two-dimensional space. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of a 60 degrees angle. The interesting part is that wavelet response can be found out using integral images very easily at any scale.

In order to continue the SURF algorithm and perform feature description, the SURF uses Wavelet responses in the horizontal and vertical direction (again, use of integral images helps with the simplicity of the procedure). A neighborhood of size $20s \times 20s$ is taken around the keypoint where s is the size. It is divided into 4×4 sub-regions. For each sub-region, horizontal and vertical wavelet responses are computed and a vector is formed. The represented vector gives the SURF feature descriptor with total 64 dimensions. Ideally, it is good to know which aspects of the algorithm's performance this lower dimension affects. By investigating various aspects, lower dimension results in a higher speed of computation and matching in practice. It also provides a better distinctiveness of features.

The SURF algorithm has an option which can be considered as dimension extension and there is the possibility of achieving more distinctiveness by this extended 128 dimension version [157]. The sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sums of d_y and $|d_y|$ are split up according to the sign of d_x, thereby doubling the number of features. As an important point, complexity is not increased and the computational costs will not be much higher.

In order to underlie the interest point and create an improvement, trace of the Hessian matrix, sign of Laplacian, was utilized. This process does not add any additional computation cost because it has been already computed during the detection step and the improvement has been easily apprehended. The sign of the Laplacian is responsible for distinguishing bright blobs on dark backgrounds from the reverse situations. By getting the same type of contrast, we are able to compare features in the matching stage. The advantage is using minimal information which leads to faster matching without any reduction of the descriptor's performance. In addition, the SURF adds totally a lot of features to improve the speed in every step.

In the HARRIS-SIFT algorithm, the detection component is a corner detection algorithm, which is the HARRIS detector. Preferably we use the HARRIS points when looking for exact corners or the time that precise localization is required. It basically finds the difference in intensity for a displacement of (u, v) in all directions. In some cases, this algorithm is not reliable, because detected points of the algorithm do not have the required level of invariance for image matching. Although the accuracy of the system with this algorithm is less than other systems in our work, it has been used widely in different computer vision applications.

To have a faster detection, the FAST detector is chosen as one of the components. It is based on the accelerated segment test (AST) [257] which can be considered as a modification of the SUSAN corner detector. The FAST is not robust in the presence of noise and especially high-level noise as it might happen in plant recognition and moore importantly in the natural environment. A small damage in the leaf sample causes harmful effects on the used algorithm. Besides this disadvantage, it is many times faster than other existing corner detectors and provides high levels of repeatability under large aspect changes and for different kinds of features.

We are seeking other criteria for the investigating the performance of the implemented systems at this stage. Four different species of the dataset were randomly selected, and the second experiment was performed for calculation of the number of keypoints. The complexity of the species is specified and labeled as simple, approximately simple, approximately complicated and complicated according to the defined level of the complexity by the human vision. The number of the keypoints are calculated for the SURF and FAST-SURF algorithms (see Table 6.5). Figure 6.20 shows the images of the samples.

Number of keypoints in the modern algorithm	Simple leaf	Approximately simple leaf	Approximately complicated leaf	Complicated leaf
SURF	27	71	345	828
FAST-SURF	234	1040	1193	5894

Table 6.5: Number of the detected keypoints

Figure 6.20 shows the detected keypoints for one leaf when the SURF and FAST-SURF algorithms are applied.

If compared with the number of the detected keypoints by using the SIFT algorithm [169], the

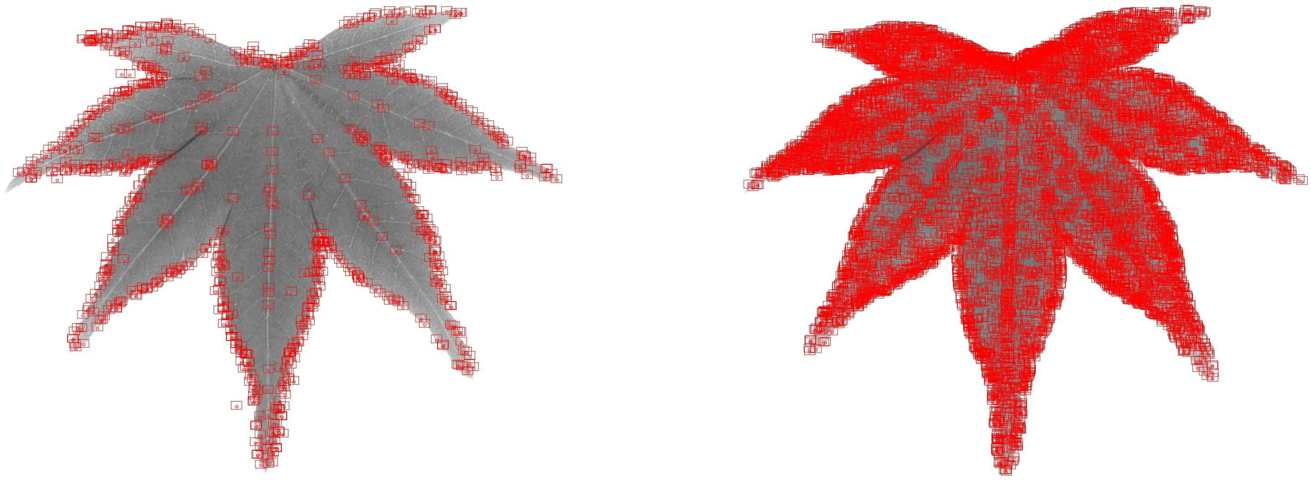


Figure 6.20: The detected keypoints using the SURF algorithm (Left) and the FAST-SURF algorithm (Right) [170]

SURF algorithm detects fewer keypoints and results in an adequate number of the detected keypoints with a high accuracy. The FAST detection algorithm finds a large number of keypoints mixed up with noisy and undesired keypoints which cause a decrement of accuracy.

Performance of an automatic recognition system can be evaluated by the running time for its recognition task. Due to the cost of operations, it is imperative that we should determine the required time for proposed systems. Without any doubt, there is a relationship between higher running time and costs of the whole system. We are looking for systems with less computation time. Therefore, we change the orientation of our investigation and compute the needed time of the implemented systems with different approaches.

Used Algorithm	Needed test time per image ms
SURF	445.2680
FAST-SURF	345.5120
HARRIS-SURF	528.7560

Table 6.6: The test time per image [170]

By using the SURF algorithm, faster computation was obtained without sacrificing the accuracy. In short, the SURF algorithm adds a lot of features which improve the speed in each step as well as the detection and description parts. The analysis shows that the automatic system with the SURF algorithm is faster than the system with the SIFT algorithm while the performance is somehow comparable to the SIFT algorithm. The SURF algorithm handles the images with the blurring and rotation well, but not changes of the viewpoint and illumination. This fact will have undesirable effects if we work on the classification of the natural images. It will be proven in next chapters both theoretically and practically.

The system with the FAST-SIFT algorithm has the minimum required computation time in comparison to the other proposed systems and the big problem of the speed of the system. The needed computation is measured for all three systems in ms (See Table 6.6). In comparison with the system using the SIFT algorithm [169], the system with the SURF algorithm needs less time for the same dataset. The needed time for the system using the HARRIS-SURF algorithm is lower than the system using the HARRIS-SIFT algorithm in [169].

We employed three different combined modern methods to deal with the possibility of the trade-off between the accuracy of the classification and the needed time for classifying the images. Although the accuracy of the system using the FAST-SURF algorithm is less than the system with the SURF, the system is still helpful because it reduces the risk of late prediction of the input test image.

The descriptor is a key element that contributes to modeling the relationship between the detected points with little information. Additionally, it helps to have useful and effective model with richer information. Since the FAST-SURF algorithm detects more keypoints than the HARRIS-SURF algorithm, more information is supplied. It results in a better performance of the system. One important characteristic of the FAST algorithm is its acceptable power for the repeatability which has a good influence on its performance in the recognition of 32 different plant species. Owing to the properties of the SURF algorithm as the main description algorithm, the performances of the detection algorithms, the HARRIS and the FAST, with the algorithm are acceptable and more than expected.

Now, we would like to focus on the influence of the changes in the SVM parameters for successful classification of the proposed systems. It also affects the answers to the questionnaire survey about the performance of the identification systems from another point of view. Fundamentally, when we evaluate the automatic plant recognition systems, it is essential to know what and how parameters might influence the systems. This type of evaluation has been skipped in some plant recognition systems such as [258], [259] and [260] and the capacity of the implemented systems has been ignored. However, even a small change in the parameters might affect the whole system and its behavior. ν and γ parameters, as two SVM parameters, are chosen to consider their variation effects on the final error of the systems. ν and γ parameters might produce unfavorable results.

Although both selected parameters have some impacts on the classification power of the systems, ν parameter has a more meaningful interpretation because the ν parameter presents an upper bound on the fraction of the training samples which are errors (badly predicted) and a lower bound on the fraction of the samples which are the support vectors. The value of the ν parameter cannot be out of [0, 1]. In order to start the experiment of this step, the ν parameter has been changed while the γ parameter was held constant at 1.0. By the increment of the ν parameter, the error of the systems is increased and it is shown in Figure 6.21. The increase of the ν parameter has the least influence on the system using the SURF algorithm in comparison to the other implemented systems. The robustness of the system with the SURF algorithm against the ν parameter variations is interesting. It proves the validity of the system without causing an inadmissible change in the system behavior when the allowable parameter changes.

The γ and its variations are used to consider the effects on the final error of each classification system. The γ parameter defines how far the influence of a single training example reaches, with the low and high values respectively meaning far and close [170]. For this experiment, the ν parameter is kept fixed at 0.1. The increase of the γ parameter causes the error increment as shown in Figure 6.22. The diagrams are ascending, but the values of the slopes are not large. The minimum slope value belongs to the system using the SURF algorithm which indicates the robustness of the system in this experiment. The increase of the γ parameter has less effects than the increase of the ν parameter.

The other experiment for the evaluation of the performance of the system is visualizing the results of the system by the constructing the confusion matrix and extracting the precision and recall measurements. Hence, the precision and recall are measured for each label of the implemented systems by using the information of the confusion matrix. In both Figure 6.23 and Figure 6.24, the system implemented by the HARRIS-SURF algorithm for the detection and description parts has minimum

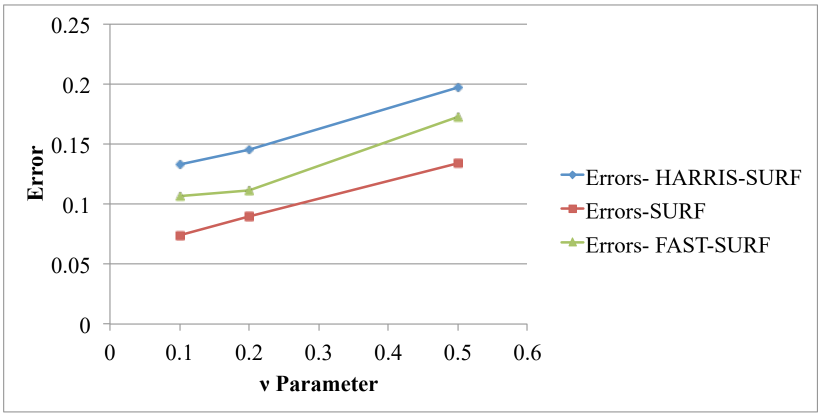


Figure 6.21: Variation of the ν parameter for the implemented systems respectively [170]

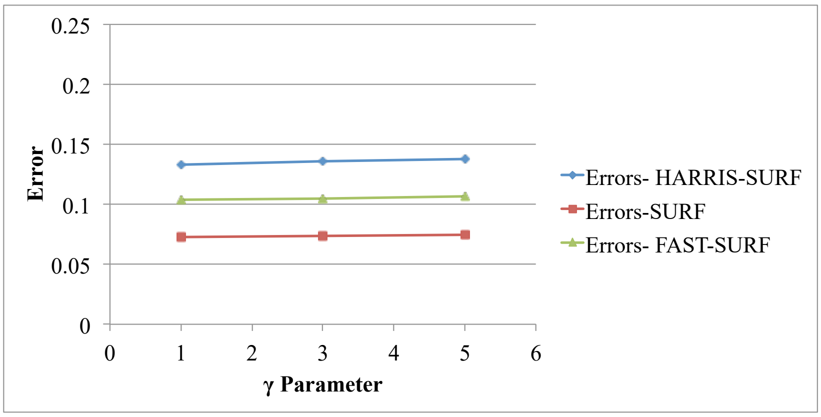


Figure 6.22: Variation of the γ parameter for the implemented systems respectively [170]

values. We were able to predict it because it has the least accuracy percentage among the other three systems. This experiment shows that variation of the obtained values of the system with the SURF algorithm is less than the other systems in both figures. The performance of the system with the SURF algorithm is exciting due to the results shown in plotted figures. The system implemented using the FAST-SURF algorithm obtained the second rank among our systems. Its performance is in the middle compared to the other systems of the current set of classifiers. According to the measurements, the sequence of the systems in this experiment is the SURF, the FAST-SURF and the HARRIS-SURF.

Another concept to be compared is the area under each plotted graph. The larger areas under the curve represent both higher precision and recall. These concepts as depicted in Figure 6.23 and Figure 6.24, the system with the SURF detection algorithm has better performance in the systems with the combined modern algorithms because the larger areas that belong to it. Both high scores show that the system is returning the accurate results (high precision) as well as returning a majority of the positive results (high recall).

For an effective investigation of the issue explained above, the relationship between the recall and the precision is shown in a figure for each system separately. For the system with the SURF detection component, the minimum value of the recall is 0.6190 and the minimum value of the precision is more than 0.7037 [170]. The variation of the values in this recognition system is less than the system with the SIFT detection component used in [169], and it is a notable result. Figure 6.25 shows how the system varies between different plants. It is worth mentioning that the value of most labels in Figure 6.25 lies between 0.9 and 1.0, and this range, [0.9,1.0], is undoubtedly the highest possible interval.

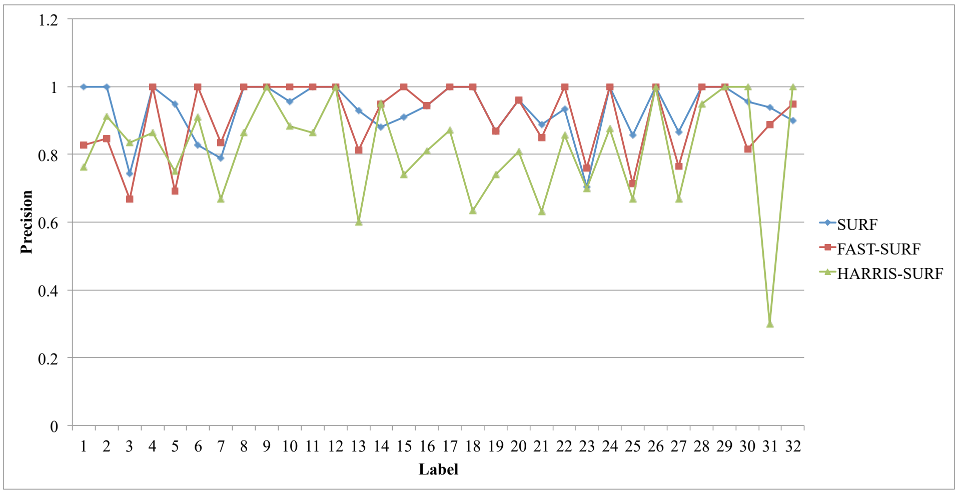


Figure 6.23: Precision measurement for the systems with different detection and description algorithms, SURF, FAST-SURF and HARRIS-SURF [170]

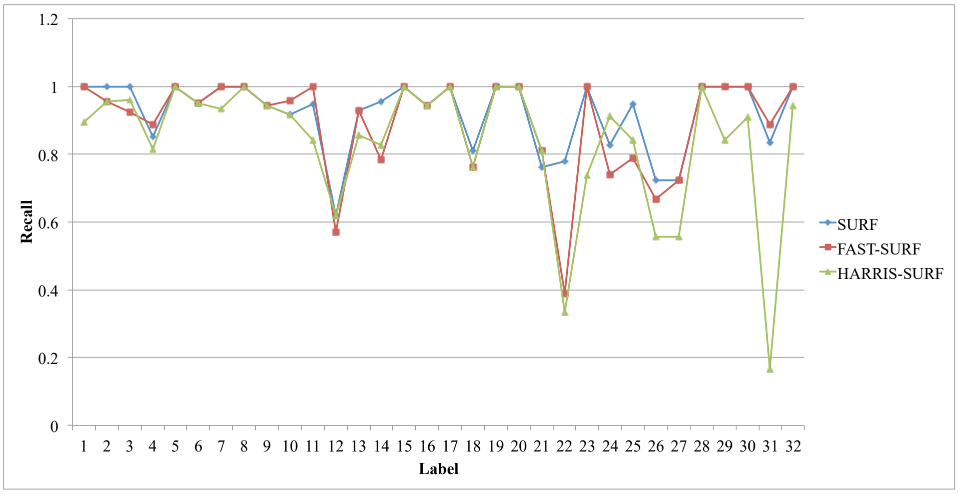


Figure 6.24: Recall measurement for the systems with different detection and description algorithms, SURF, FAST-SURF and HARRIS-SURF [170]

In Figure 6.26, the precision and recall values are represented for the system with the combined modern algorithm, the FAST-SURF algorithm. In this system, the minimum values of the precision and the recall are the same and equal to 0.67 [170]. In comparison to the implemented system with the SURF detection algorithm, there are more variations in the results of both precision and recall. After the investigation of the minimum recall value, we found that this value is less than 0.4 and equals 0.33. In comparison to the used FAST-SIFT algorithm in [169], this combined algorithm has a better performance.

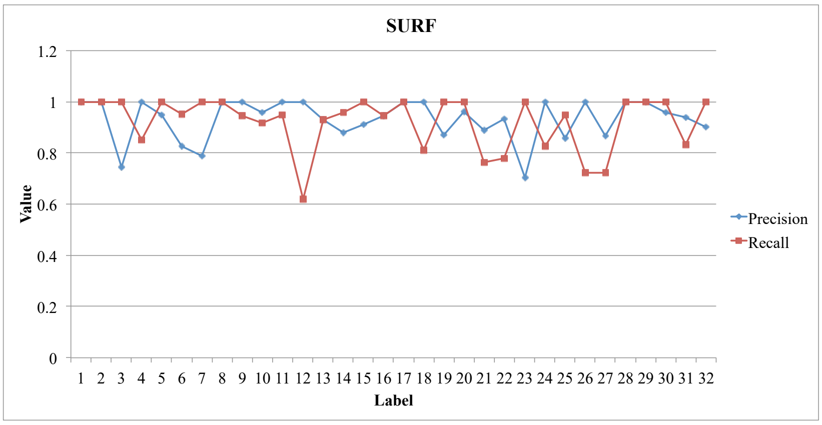


Figure 6.25: Measurement of precision and recall for the system with the SURF detection component [170]

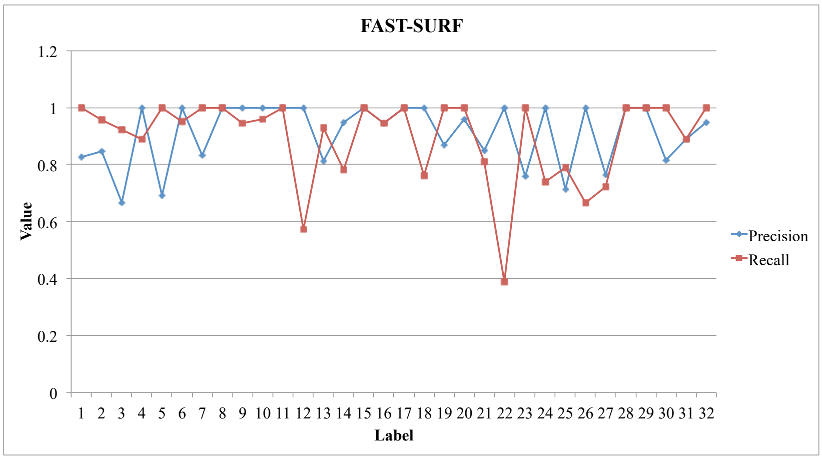


Figure 6.26: Measurement of precision and recall for the system with the FAST-SURF algorithm [170]

The precision and recall measurements of the system with the modern method, the HARRIS-SURF algorithm, are shown in Figure 6.27. The minimum value of the precision is 0.3 while the minimum value of the recall is 0.1666 and it is less than the minimum value of the precision. The difference between the maximum and the minimum is large in this system for both precision and recall values, and the range of the interval increases. In other words, the larger intervals are covered in the implemented system using the HARRIS-SURF algorithm for the obtained precision and recall values. Obviously, the main reason is the classification accuracy of this system that is lower than the other systems using the SURF and FAST-SURF algorithms.

If we consider all experiments including the presence of 32 plant species, the results of the current systems prove higher overall accuracies than that of the preliminary systems in pervious sections. We tested the systems under a large number of the plant species where the images were not exactly from the same plants in its natural environment. It is just the beginning and the further study and work should clarify how we can become capable of distinguishing the plant species in the outdoor environments and different challenging conditions.

The results of the implemented systems show a good achievement in the recognition systems using the modern combined algorithms, and also reflect somehow the level of difficulty of our dataset with huge diversity of the plant species. Overall, the systems using the SURF algorithm as their description base seems to be a decent choice to keep the balance between the accuracy and the computation time.

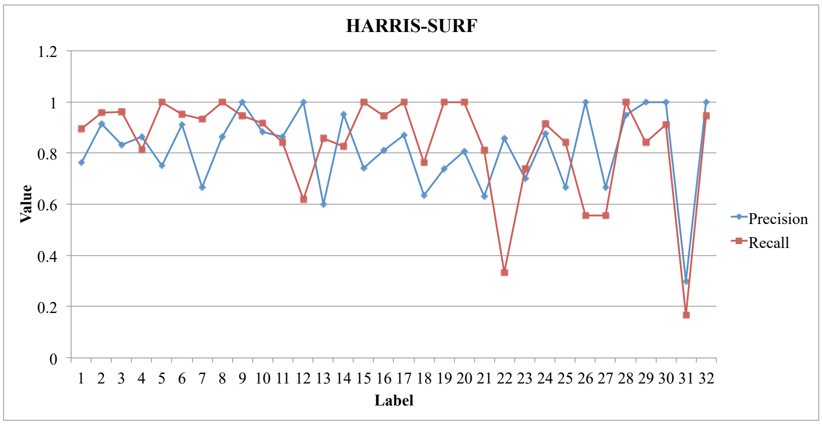


Figure 6.27: Measurement of precision and recall for the system with the HARRIS-SURF algorithm [170]

So, a high classification accuracy with good and suitable computation time can be achieved by the system used the SURF algorithm with an accuracy equal to 92.28%. In order to decide on choosing a useful system from the proposed systems, we have to know which plant recognition system favorably suits our specific case e.g. we can ignore the accuracy if the computational complexity and the speed are important for us. When the accuracy and the time are not critical issues, it is recommended to detect and extract more features by using the system with the SIFT algorithm instead of the system with the SURF algorithm.

In the end, we would like to consider one sample which was recognized differently by two different systems. Figure 6.28 shows the sample image (on the left side of the figure) of one class which we used as the input for two systems. One of the systems was based on the HARRIS-SURF, and the other was based on the FAST-SIFT. The former system recognizes the sample input plant image correctly and identifies its class while the latter wrongly recognizes the real class. A sample image of this class is shown in Figure 6.28 (on the right side of the figure).

6.7 Applications of the Proposed Systems

Botanists use the plant identification books like [261], [262] and [263] as their references for the plant recognition. Normally, botanists first consider the leaf of the seen plant and try to find its exact family. Then, they start to understand if the plant belongs to that family according to different factors like place. The botanists also check both sides of leaves and the veins. Considering all factors is a tough task and it might result in the wrong identification of the plants. In addition, this procedure usually takes a long time to distinguish the exact class of the plant. The proposed systems have extensive application in the botany and plant sciences. The botanists would be able to use the developed systems easily. Moreover, the ordinary people and non-specialists, especially high school and undergraduate students, can use the systems to identify the plants without any pre-knowledge if they have recently started learning and training in this field.

In addition, the volumes of the biological information of the plants are increasing daily. Specialists and non-specialists have access to this data gathered from all around the world and even museums. The developed systems contribute to sorting out the information of different plant species based on the desired factors. Furthermore, it can be used for reverse image searching that would be helpful in the research labs and biology departments. However, some functions and new features should be added to the current systems. The reverse image searching also allows the researchers and the botanists to discover related contents of a specific sample plant species [264].

E-commerce websites may be developed to sell different plant species from all around the world. The owners of such websites will be able to identify the plant species without getting the information from the people who want to offer their products through the websites. Additionally, the owners of the websites can examine the correctness and validity of the received information from the people offering those plant species.

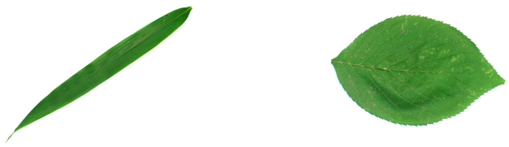


Figure 6.28: Sample input image for two plant recognition systems (on the left side), the system based on the HARRIS-SURF classifies it correctly, and the system based on the FAST-SIFT cannot recognize it correctly and identifies the input as another class. On the right side, there is a sample of the wrong identified image

6.8 Acknowledgment

We acknowledge the use of the OpenCV library [256] [265] for the implementation, development and building our desired automatic plant recognition systems.

6.9 Conclusions and Future Scope

We tried to introduce a newer and more efficient approach to the plant recognition using different modern detection and description methods. The benefit is to have the systems for the identification of a large number of various plant species, 32 different plant species in our case. The achievements of the technical research in [169] [170] are to provide the optimal and robust systems used to build several automatic plant recognition classifiers. In addition, the use of different algorithms resulted in extracting and providing efficient, high quality and repeatable features. In [169], the SIFT algorithm and two combined methods were taken into account for the plant recognition and classification. Moreover, the accuracy measurement and the efficiency of each method were described. The experimental results were also compared with some quantitative results and discussed according to the human vision for four different species. The experiments on the testing dataset, demonstrate that the system using the SIFT algorithm has the best performance among the proposed systems.

In [170], three methods, the SURF and two combined methods which were the HARRIS-SURF and the FAST-SURF, were taken into account for the plant recognition and identification on the Flavia dataset. For the implemented systems, the accuracy measurement and the efficiency of each method were explained in detail by performing the experiments. The obtained results are also compared with some quantitative criteria and explained according to the human vision for four different species as we did for the previous set of systems in [169] before. The experiments on the testing dataset demonstrate that the system with the SURF algorithm has the best performance among the proposed systems in [170]. In comparison with the methods used in [170] and [169], the systems using the SURF and the FAST-SURF have better performance and accuracy.

Although we stopped our experiment on the used dataset in [169] and [170], we are able to build systems based on the VLAD technique. In addition, it is possible to make some changes on this technique and create new systems. The first new idea is based on expanding the VLAD technique, using the squared residuals. It contributes to obtaining more compact representation of the images. Moreover, it would be feasible to search for the two nearest neighbor visual words for aggregating each descriptor. Improvement of the VLAD technique is also possible by combining the second-order information and using the vector of locally aggregated tensors (VLAT) [266]. This approach is an extension of the VLAD and two sub-terms representing respectively the first-order and the second-order information constitute the VLAT approach [267].

In sum, the presented work highlights the developments in the field of the plant classification using the bag of words model as one of its main components. Applying different detection and description methods proves that the initially-designed systems can be enhanced to provide a better performance by using the combination of the detection and description algorithms. We also proposed fast and robust systems that can be applied to the plant recognition as PC software. Our next goal can be to improve the implemented recognition systems and find a solution to get a more efficient representation of the extracted information.

Chapter 7

Automatic Plant Recognition Systems for Challenging Natural Plant Species using Modern Detection and Description Methods

In this chapter, we address an important unsolved problem of plant recognition systems firstly and scrutinize the problem from different aspects and points of view. In fact, we propose the first forward-thinking work to find a solution for the problem of the natural plant recognition. Here we start a new journey to secure the future of other related fields such as medicine, drugs industry, agriculture, etc. We introduce an effective scheme and design of an automatic system which will thoroughly be inspected with different possible methods. Several systems will accordingly be created. The challenges are not only restricted to the plant recognition task. Various environmental challenges such as wind, dust, shadows, etc. have undesired impacts on the recognition of plant species. Our main goal is to compensate the gap between current existing plant recognition systems and the real needed system. Hence, the modern dataset explained in 3.3.1 is utilized. Different detection and description techniques are applied to detect keypoints and extract features which are usually called modern methods. It should be pointed out that the other main components are BoW and SVM. The systems enable a reliable process for plant recognition which emphasizes on the purpose of natural plant identification. Since the systems are developed to fulfill the necessities of the real world, different experiments have been carried out. The system using the SIFT approach yields a high recognition accuracy of over 94%. We furthermore illustrate how each implemented system can be used to improve the ability of correct and accurate recognition.

This work has been published in conference [268] and journal [151].

7.1 Introduction

The Earth is known by various names like, green planet, terrestrial planet and as the fifth largest planet of the solar system. Exploring the history and existence of life on the Earth shows that our planet is approximately 4.5 billion years old [151]. The history of life is estimated to have originated approximately 0.7 billion years later [151]. However, life has developed on the Earth gradually, this phenomenon can be considered as a constant occurrence and happening, not like an exponential graph. To discover the Earth's history and to look back through the past years, the study of microscopic ancient plants and fossil records are essential for better understanding of the related issues. The distribution of plants is not steady and it varies from one place to another all around the world.

Considering one region, the distribution of one specific plant species varies in the unique nature and environment of the supposed region. The presence of plants is not only limited to botany labs. They are present in different places, such as deserts and jungles, and within various fields, such as literature and mythology, with useful and inestimable historical records.

Imagining the Earth without oxygen is impossible. The effects of plants become more manifest since no other living organism can exist on the Earth without plants because they form basic food staple and persist as the largest factory for oxygen production. This is not their only role, however, as plants are also responsible for the regulation of water cycles and they affect the environment and climate. Agricultural activities depend on plants. In addition, many countries benefit from these activities in both political and economic situations which have an essential influence on the future of countries.

In order to form the Earth and living organisms, photosynthesis was a turning point. Plants use this process to convert light energy into chemical energy. Some of the early microorganisms evolved a way to use the energy from sunlight to make sugar out of simpler molecules. However, unlike green plants today, the first photosynthesizing organisms did not release oxygen as a waste product, so there was no oxygen in the air [170]. The main part of these busy factories is the leaf which is the core of production.

Plants have contributed to the development of human civilization as they appeared close to rivers where they influenced the origins of modern life. They impacted the climate and its variations. From this aspect, their significance is also undeniable. Due to scientific findings, the perspiration and breath of plants leads to a cooling of the atmosphere. They consume and lead to a reduction in the amount of carbon dioxide through the process of photosynthesis. This reduction has an indirect cooling effect. Furthermore, climate change alters the life cycles of plants. It is an interesting point and approves the relationship between them. Additionally, plant species traits are the attributes that most directly affect the ecosystem processes. They contribute to the healthiness of an ecosystem.

Additionally, people have utilized plants with medicinal properties for many years to fight against diseases. Many patients insist on using herbal medicines and drugs to avoid chemical drugs and treatments which might have destructive effects over time. Besides being rich resources of ingredients, plants produce all food for living organisms, even their own food in order to survive and grow. Furthermore, many scientists are working in labs to help feed people all around the world and produce lab-grown plants to meet the new needs of human life. New generations of plants will be available as daily human food by genetic manipulations. The advancement of agriculture depends on this new paradise which could be helpful for reducing the waste of crops.

If we explore the history of plants, we find some plant species in the past years which have become extinct and we do not have any access to them now. Due to human activities, plant species are in danger of extinction. A complete information database of different plant species [3] [4] [5] [6] can be collected. Plants' role in change of the Earth's climate after the Ordovician extinction which happened more than 425 million years ago is irrefutable in addition to the activities affecting plants in the past years. Furthermore, plants have had different influences on the human life too.

Due to the importance of plants and their roles, their study is essential in various fields. Consideration of their different applications is a demand which leads research to focus on their details. The automatic recognition of plants is a novel field to contribute to research and future studies. A useful plant recognition system should be capable of the identification of different species in all places, even the natural environment. The connection between the computer vision and plants is undeniable. The change of the typical manner of plant recognition has become important nowadays where automatic plant recognition is also an exigency in the modern world. The typical plant recognition is impossible with either a glance or a blink. Experts and botanists use some books and references that contain plants information to identify the species which is a common and time-consuming way to identify plants. Accordingly, having an automatic recognition system helps scientists, researchers, managers and engineers in labs, offices and factories. Efficient plant recognition systems enable people who have not been trained in botany to participate in plant projects.

In the natural environment, we encounter plants grown in different regions, weather conditions and climates. The weather condition is one of the parameters which affects plant life and its existence in a particular area. Recognition of plants in different weather conditions is a new window of research in the field which can be considered for the generalization in a recognition task. In order to have a general system, the distance from the camera to plants is not only a problem. It should be considered as another factor to help us while we are going to implement an applicable recognition system. During day and night, the variation of light intensity in the environment is undeniable. Thus, it can be considered as an important factor as well. Adding these factors leads to a huge challenge to invent an accurate and secure system. Thus, recognition of species in different conditions is a real need as plants are ubiquitous to our life, and the development of an automatic plant recognition system is mandatory and will be effective for different aspects of life on the Earth.

There are many different plant species with various shapes, colors and textures in various illuminations and lighting conditions all around the world and particularly in Europe which is famous for being the Green Continent. During the last decades, many efforts have been used to make roles and activities of robots closer to real humans. In spite of many successful attempts and the conceptual philosophy, there is a big gap between the human visual system and those installed on robots. For instance, the human eye sees and recognizes different leaves of one plant species at very close distances to the whole plant even if there are shades of dark green, yellow, red or orange. The human brain can differentiate shades of colors under different lighting conditions and at different times of the day. But, this process is still not feasible for the current computer vision and robot systems. We prefer building a general system instead of just a specific system for modeling the leaves of plants. This is because one important factor of a model is the color, but leaf color is not always constant. We do not rely on one detection and description algorithm. Instead, we are going to work on a combination of detection and description algorithms to infer the visual differences that can be found by both humans and robots. In addition, we would like to create a system that helps humans recognize plant species completely without any pre-knowledge concerning plants.

If we compare the vegetation detection [269] to plant species recognition, we find that both vegetation and the leaves of plants can have different colors, but there is a big difference between them. If we consider the color in a small vegetation region, our expectation is to find the color homogeneous. Therefore, it is possible to find the first vegetation pixel. Then, it is possible to search for the other vegetation pixels among its neighbors and detect the vegetation region according to color similarities. In plant species recognition tasks in challenging cases, it is impossible to consider only the color similarity or dissimilarity because we face many different conditions and situations.

Finding the answers to the two important questions is necessary at this stage, "Which component of the plant is useful for the plant identification systems? Does it help us to achieve our goal?" We believe that finding the useful component of a plant is a long jump to having a recognition system. To find the answer of the proposed questions, we resile from the current stage and consider the factors which should be taken into account for the correct selection of the component. The intended factors are stability, consistency and independency for finding the desired component. The seasonal characteristics of plants surely affect the selection of the component. Therefore, we would like to have a robust component against these characteristics. One important component of the plant is the leaf. It can be investigated to build automatic systems for the plant recognition without any human interface and interaction.

The leaves of plants are the first characteristic to be selected as a trusted and important part. Therefore, we inspect these in detail. Some characteristics of leaves, like size and color, might vary in a single plant, but their typical shapes are generally the same as others. When we look at only one leaf, each side of the leaf might have a specific color. It is sometimes hard for the human eye to distinguish the difference of the colors. Furthermore, we presumably see one plant that its leaf has been transformed and it is going to be a dried leaf. In other case, the lower side of the leaf might be rolled to the upper side and the leaf loses its original shape. In addition, if we suppose that we are looking at an apple tree, the diversity of the leaf shapes in this specific tree is high. This diversity adds more challenges to easily identify the species of the tree. It is also hard for the human eye to determine whether the leaves are exactly similar to each other and whether they are coming from the same apple tree. Similarly, through the use of robots, it is really hard to classify natural plant species.

Although we think that the leaf shapes are commonly structured, we sometimes find turbulent and unstructured shapes among the leaves of one specific plant. Therefore, we cannot trust the texture of leaves in all outdoor environments. Hence, the texture orientation is not a good choice in challenging cases. In general, we are able to categorize the shapes of leaves in nature as below [270]:

- Ovate
- Obovate
- Lanceolate
- Oblanceolate
- Cordate
- Obcordate
- Elliptical
- Oblong
- Cuneate
- Linear
- Peltate
- Spatulate
- Reniform
- Hastate
- ...

Some factors, such as the smoothness of margins, form of curves, position of centers, condition of blades and the length and width of leaves, etc., differ in different plant species. The first effect is the increase in the changes of shapes between various leaves. The diversity of plant species and the various possible conditions in nature make the recognition of them harder than what we assume. Moreover, plants can survive in different places and regions by means of adaptation. Hence, adaptation is one important property of plants. It helps them in hard life situations. This concept might also result in changes to the shape of plant leaves. In addition, the shapes of the leaves change for some plants during day and night. This point is a golden key for opening the new land of plant recognition systems.

The investigation of plant recognition systems proves that the current plant recognition systems like [271], [272] and [49] are working under constrained conditions. In order to conquer these conditions, the first requirement is to consider some factors that should be added. It is necessary to find the factors carefully if the goal is to build a useful system for various real conditions. Moreover, the system should be robust enough to be used in different natural environments. Thus, different aspects or components of natural environments should be taken into account. More precisely, climate, weather and wilderness can affect the performance of a plant recognition system [268]. Therefore, we consider weather conditions as one of the added factors. Removing this factor would mean losing one of the pieces of the desired puzzle. In addition, different types of natural light can produce a wide variety of appearances when an image is taken of a plant. Even though the light source, the sun, is the same, this source provides a different light intensity every hour. Therefore, we have to create a system which can be utilized at different times of the day. The enhancement of the system depends on how it works under different variations of light intensity. Having a system that recognizes plant species in different light intensities is a significant goal. In other words, the system should work perfectly without any consideration of time of day, whether we are using the system in the morning, noon or evening.

Furthermore, in agricultural applications the distance between the observer and the plant species might be more than 1 meter or less than 50 cm. The observer can be a human or a system, machine or robot. The camera of the system plays the role of the human eye in this case. If we change the position and the distance of the leaves from the camera, we are simulating the mentioned fact that might happen in the real world. Furthermore, the point of view has an influence on what is seen of the plant species, too.

The subjective and objective abilities of the human eyes vary between one person and another. This also happens in plant recognition systems if we use a specific camera or a random camera. The independency of the identification system from the used camera is also an important point for obtaining a general system. This fact also will be considered in developing and testing a real-time system in the future of our work. At this stage, we choose our camera without any additional consideration. Therefore, the type of the camera will not have any effect on the proposed systems. It can contribute to us developing a general system.

A consideration of these continuum factors helps to develop and employ a reliable and general system for various applications with the purpose of plant recognition. Figure 7.1 depicts some sample images of the dataset which belong to one plant species [268]. These new factors will add unique and new features to systems that will be built. In addition, they make the systems more applicable and efficient. Although these are new challenges, the implemented systems should tackle all challenges to achieve the desired goals.

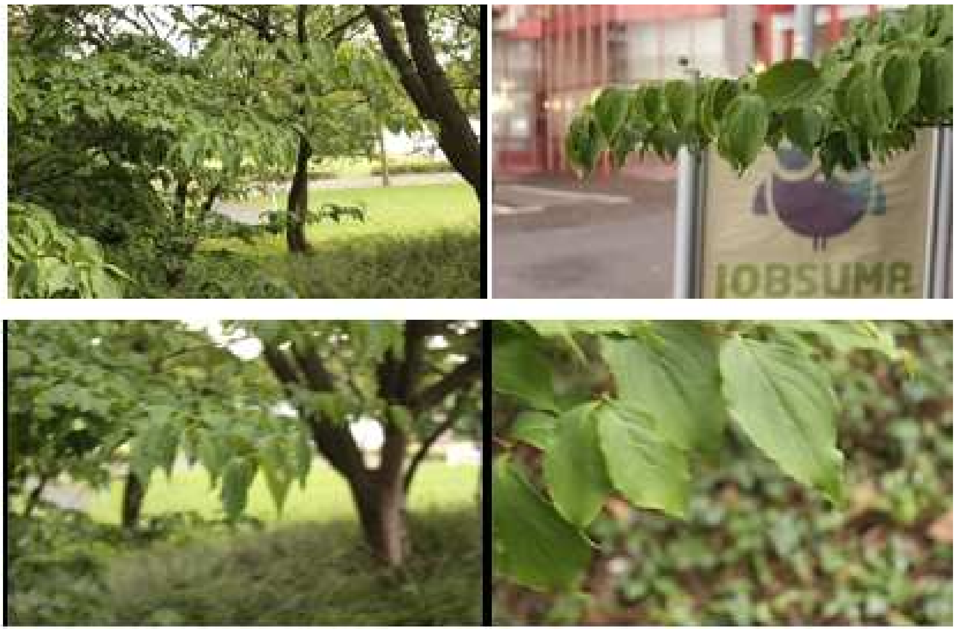


Figure 7.1: Four different sample images of the Cornus [268]

The result of modern life is modern needs. Therefore, we need to develop accurate systems to assist botanists and professionals. Here we would like to go deeper into the issue of the plant recognition and talk about it in another way. The first need of any computer vision system is to have visual information which can be images or frames of videos. To acquire images, we use a camera that has the role of the human eye. While a human is looking at a tree, the image of the scene and the tree will be processed in the brain. The related information will be stored as well. Then some important information of the scene will be highlighted in the memory. The individual will be able to remember this information after a while. With this in mind, we propose a system that is able to act like the human brain and vision systems for recognizing different plant species.

As we have discussed, the plant recognition system is based on the shape of the leaf. Due to added factors, the correct extraction of features and useful information is the critical point in the initial step of inventing the system. The appropriate analysis of the constituent shapes and using an accurate technique aims at the data processing. The automatic plant species recognition, which is a significant necessity, relies on computational methods to extract discriminative features from images like other image recognition tasks [273] [274] [275]. Due to the needs of modern life, the tendency to use automatic systems has increased recently, especially within the last two decades. Thus, a set of techniques that learn features automatically has become a priority. It is one of the main goals to transform raw data to correct representation which can be effectively exploited in machine learning and pattern recognition tasks. The two main advantages of feature learning are the automatic analysis of images and efficient use of features in classification and recognition tasks. Furthermore, real-world data is commonly very complex, redundant, variable and even noisy. For instance, images taken in outdoor environments are faced with natural factors, like light intensity and illumination. The adaptation of present strategies is very important to automate and generalize plant recognition systems.

In order to create an automatic system for plant identification, many methodologies have been proposed to analyze leaves of plants. Most of the proposed approaches have attempted to define contours of leaf shapes and apply the contours for their own purposes. Geometrical parameters, such as area, maximum length, maximum width and perimeter, are applied in [100]. These parameters are not effective enough to obtain a general recognition system in the natural environment because the distance between the camera and the plants changes as one of the assigned factors. In addition, the size and numbers of leaves in each image will not remain fixed. As a result, methodologies with similar concepts and ideas are useless in such environments. The reported work in [101] uses both color and geometrical features. As discussed before, the color is not a good choice in outdoor and challenging natural environments. Other methodologies have been introduced in [276], [277], [278], [279], [280] and [281] for shape representation. In [282], the proposed method is based on the curvature scale space (CSS) [282] approach and the classification of chrysanthemum leaves is based on the KNN method. In another work [283], region-based shape recognition techniques have been applied for doing leaf image classification. The final accuracy was 82.33% for the proposed method where the contour-based method had 37.6% classification accuracy [283]. The important point is that artificial images have been utilized in the mentioned literature. Despite the efficiency of these proposed approaches, they are a little far away from our work at this stage because the work is based on natural images. Some works like [176] and [284] can only be applied to the certain species. In [52], the implemented system is actually a semi-automatic system and the final accuracy is 85%.

Popular modern approaches, such as the SIFT and the SURF, are used by researchers and scientists to extract features of images. The fundamental characteristic of the mentioned methods are stable local feature detection and good representation of the original data. The SIFT, which is the resistant algorithm to the usual image deformations, includes both feature detection as well as description parts. It can be applied separately due to the desired purpose. In addition, a large amount of keypoints from one image will be obtained. The keypoints are originally the oriented disks attached to blob-like structures of the image. Being invariant to the image translation, scaling and rotation has made the algorithm popular and considered as an accurate feature detector and descriptor. In

[169], the basis of the methods used is the SIFT algorithm to recognize plants automatically. In order to classify flowers, SIFT features are utilized in [285]. The scope of using the SIFT algorithm is also extended to other fields such as object tracking [286], video matching [287] and even image retrieval [288].

The FAST algorithm is a popular approach to detect corners when real-time application is demanded. It is a fast algorithm from this point of view and it provides a lot of features, without the matter of the level of usefulness of features, in a short time. However, one important disadvantage of this algorithm is that it is not robust to high levels of noise. In [169] and [170], the FAST approach is a part of the combinational methods for automatic plant recognition.

Another interesting and popular approach for the feature detection is the HARRIS algorithm, proposed by Chris Harris and Mike Stephens. Since corners show a variation in the gradient of an image, this variation can be utilized to do a detection procedure perfectly [151]. The HARRIS method was used as a component of the combined methods used in [169] and [170] to do automatic plant classification. It should be pointed out that the used dataset in [169] and [170] is not natural, but it is practically a common dataset. There are several plant datasets such as the Flavia dataset, ImageCLEF dataset [90], Leafsnap dataset [289] and Intelengine dataset [76]. Each dataset contains different plant species. According to the need for a natural dataset for our final targets, we started the initial step and prepared a useful dataset to fulfill the demands of building a general, robust and accurate system. This new and unique natural dataset includes 1000 natural images of four different and common plant species of Siegerland, a region in Germany. The types of the plant species are Hydrangea, Amelanchier Canadensis, Acer Pseudoplatanus and Cornus. The used dataset has some unique characteristics and puts our work and research in the right direction.

The current chapter is devoted to principal directions of the effort of finding a right approach for implementing a useful natural plant recognition system. In this content, we are going to start solving the problem of the natural plant recognition and examining different methods to achieve our goal.

For an easier understanding of the case, let us start to do several segmentation algorithms to determine if they could be used as a part of the desired systems. With respect to the segmentation outcomes, it is easier to go through the next steps and make a correct decision. We are dealing with multiple branches of leaves, not only one leaf or a pseudo-scanned or scanned leaf with a white background. In natural environments, additional factors such as wind, angle of the received sunlight, etc. are making the original problem much harder. Overall, the procedure of implementing the classification systems will be explained in detail.

The structure of the remaining chapter is organized as follows: Section 7.2 describes how we examine pre-processing steps to know if it is useful for natural plant recognition. Section 7.3 provides a general overview of the plant recognition problem. Section 7.4 introduces the form and design of our system while the implemented systems are also illustrated in this section. All experiments and results will be discussed in section 7.5 while section 7.6 describes a short talk on the experiments, results and performances of the natural recognition systems. Section 7.7 describes the future work and section 7.8 provides the acknowledgment. The last section, section 7.9, concludes the work of this chapter.

7.2 Pre-processing Examination

One traditional way of designing a recognition system is to initialize the first step by using a pre-processing part. According to the state-of-the-art work in [290], the idea is to use a segmentation part or additional detection part. Finding a suitable pre-processing methodology contributes to enhancing the quality of the input images and preparing the raw data in a usable correct form for the next steps. The raw images might consist of artifacts and noise because we are navigating in natural and outdoor environments to capture photos from different plant species. In other words, we can apply pre-processing methodologies to reduce the input information and retrieve just an important set of useful data where de-noising would also be possible due to the nature of the used methodology.

In addition, the other goals might include detecting specific patterns in the input data, decomposing the input data into principal components and dealing with constraints and restrictions of the images taken. Intuitively, in the case of natural images, there is no dominant direction for creating a recognition system and soaring to the defined goals. Therefore, we are trying to test different approaches in the ocean of possible methodologies to find the effective approach which is a buried treasure, so to speak. Literature surveys reveal the availability of various segmentation, contour detection, corner detection, edge detection, filtering, etc. methods to remove desired parts of images for pre-processing analysis. Some questions behind the idea of the pre-processing are listed as follows:
- Should we remove the background of a natural image to obtain the desired region which is the leaf?
- If we have several leaves of a plant in an image, is it essential to identify the contour of each leaf?
- Do we get any benefit from the color segmentation of an image for the detection of a leaf of a plant?
- If we filter the background and remove the unwanted region around a leaf, is the obtained result good enough for the matching or the next steps?
- If we eliminate some important factors, such as weather condition and the time of day, distance, etc., and only take the pictures without any consideration of the light intensity and illumination, do we obtain acceptable results, even visually good results?

Although, there are several previous projects that have attempted to facilitate their tasks by doing additional pre-processing, our unsolved problem is still like a fresh and unknown fruit which has not yet been cut. And we are going to perform some algorithms to lower the unwanted information in each image. For instance, we may be able to remove the bad effect of edge reflectance and reduce the complexity of the original image. Let's consider possible scenes and objects that we might see in an image taken in the natural environment. In the case of a simple background and the existence of one leaf, we are usually able to detect the leaf from the background and segment the whole image into two parts, leaf and background. Conversely, if we have a complex background and several leaves in one image or a leaf with a complex background, this scene is even more complicated for the human eye. However, the human vision system works better than a computer-based system for identification and discrimination of the objects, background and foreground. The human vision system has also an inimitable ability. Hence, human can predict and estimate the shapes of leaves if a part of one leaf is not visible and it is under another leaf. In this case, the gap is related to the difference between understanding and the mentality of the human and machine as well as the feedback that might be stored in the human brain.

In April, 2015, we started to work on the possible pre-processing methods, and attempted to propose a comparative study in this section. An iPhone 5s has been used to take pictures from three different plants around the Campus Hölderlin of the University of Siegen, located Latitude: 50.90592 | Longitude: 8.02850. The final aim is to highlight the performance of such a pre-processing step for further processes and find out if we are able to extend this tool for our system design. The images are the Standard Red Green Blue (sRGB) that was created in 1996 by Microsoft and Hewlett-Packard (HP) [291]. The images are challenging in terms of colorimetry [292], illumination or defects, however, they are not as complex as the images of the modern dataset. Figure 7.2 depicts three sample images of the plants in the natural environment. This section first details the state-of-the-art and then refers to (11.1) concerning the implementations and available tools, and finally, various illustrations consisting of the aim of the study and the conclusion are addressed.

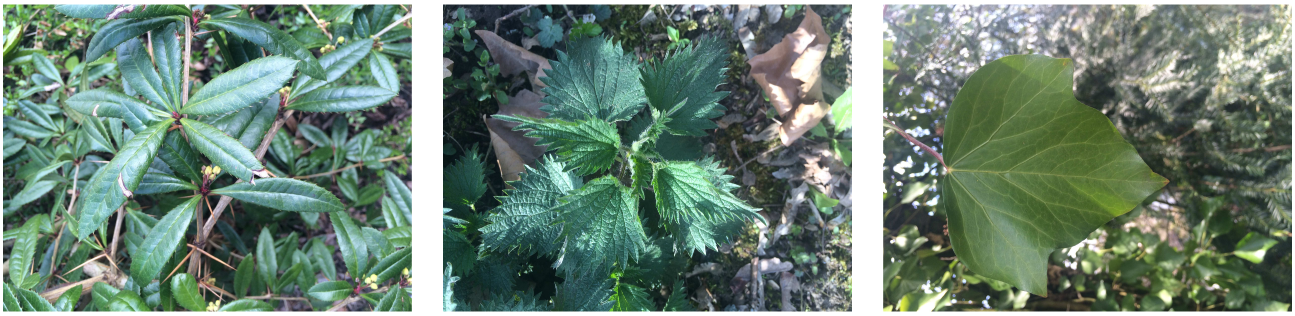


Figure 7.2: Sample natural image with defect (Left), sample natural image with colorimetry (Middle), sample natural image with illumination problem (Right)

7.2.1 State-of-the-Art

Many proposed methods are based on the analysis of a uniform and single-colored background, for instance a white background [293] [289]. Another typical solution is to use a pair of images and then do the removing process of the background [294]. In the case of our work, some segmentation methods have been proposed to analyze images with a natural background based on a single image [295] [296] [297]. The guided active contour (GAC) algorithm has been proposed in [298] to segment tree leaves on a natural background.

In this section, we use the iPhone 5s for photographing. Other cited works in this field are proposed in [299], [300] and [301]. Some techniques have been implemented and applied as shown in (11.1). The important factors are the capacity and reliability of the method and also the efficiency of the extracted parts of the image for the next steps.

7.2.2 Set-up and Study of Algorithms

In order to overcome the difficulties of the strong affection of the natural environment and the unwanted pseudo-noise in the captured images of the outdoor environment, it is necessary to set up and examine several available algorithms. In outdoor cases, the images are usually noisy. Contrast variation and changes of brightness throughout the images usually occur. The images taken represent palmately lobed leaves around the Campus Hölderlin of the University of Siegen. Obviously, for plant recognition systems, a good image gives a better performance recognition rate than a noisy image. With regard to the quality of the images, we do not consider the impacts of this factor in this phase of the research, although the poor quality of the images captured impacts the performance and final recognition rate. In natural images, it is also hard to find the best way for doing detection and description of high-quality features. To shorten the description and explanation of this section, the analysis of various pre-processing techniques, Canny algorithm (edge detector) [302], K-means color clustering, grabcut algorithm [303] and superpixel-based segmentation algorithm [304] for qualitative object segmentation and detection are provided in (11.1). Our purpose is to find out if these algorithms will provide superior performance in the matching and recognition tasks.

7.2.3 Aims of the Current Study

We present in this section and (11.1) a study of different methods of segmentation and edge detection applied to the problem of the pre-processing for the extraction of plant leaves in natural images taken in an outdoor environment. During this process, we first highlighted the problem and the performance obtained by using each method explained in (11.1). The quality of segmentation and detection tasks varies. Each method yields to new findings and properties of results. We analyzed the contributions of the algorithms if we used them as a pre-processing step. This analysis allowed us to know the efficiency of the choices and the different aspects of using them as the final choice of the pre-processing for plant recognition systems. This work is useful for designing and developing an online platform for both beginner or expert users to examine various algorithms for their own images, either artificial or natural.

Related parameters can be provided as a part of the tools for making better analyses and final validations. The application of this platform will not be limited to the plant images. Other images such as medical images, microscopic images, nuclear images, etc. can also be used. The user can give grades to each used approach for its own test image and obtain a ranking list of the approaches at the end. As an example, the user is able to specify the object position in GrabCut with the provided tools and then run the approach, although it can be applied as an automatic approach too. Some additional features such as morphological operations, thresholding in HSV space or thresholding in RGB space are also useful to add. They can provide extra characteristics to the platform. These features have also been implemented in our study, but further explanations of them is beyond the scope of this stage. One remaining point is the importance of the segmentation concept in different areas of image processing and computer vision. An image can be segmented based on the pixel [305], region [306], edge, as well as edge and region hybrid segmentation and clustering one [307].

7.2.4 Conclusion

After implementing and examining the described methods in (11.1), we find that a lot of additional costs might be incurred through the pre-processing of the entire plant recognition system. Given the importance of a real-time system and the high variety of images of the natural dataset, we decided to utilize local feature detection methods as the basis of the future systems instead of initializing the process by detecting and segmenting the leaves in each natural image. One point should not be forgotten and that is that we are not involved with man-labeled images and man-made features. Our "rocket", i.e., the natural plant recognition system, does not have ordinary fuel, and its "fuel", i.e., the images of the dataset, is very challenging. Consequently, it is critical to find the best "propulsion", in our case the most powerful features, and start with a great forward momentum. We rely on the drive of our previous experiences and start to develop our natural plant recognition systems without any pre-processing.

7.3 General Overview

Plants variety leads to a diversity of properties. Even for each specific species there are extremely unique characteristics. In the plant recognition, and stemming from diversity, one neglected area is the species classification and identification. Accordingly, we need to develop recognition systems which are accurate, reliable, and general and automatic, that work without needing botanists. One missed point here is the system's compatibility with different environmental and illumination conditions such as weather, distance, etc. In order to have a generalizable system, the first step is determination of an appropriate dataset with natural images and videos taken from natural environment. As inaccurate selection might lead to destructive effects on the whole system, finding an efficient method to extract the most useful data is vital to feed data appropriately into the classification step. It could be done through implementing an automatic system.

Inputs from raw images are mostly too large to be processed by algorithms and systems. Therefore, the first step is to detect features and output the significant locations and information of the natural images. This step is a low-level operation which has been used in many image analysis problems and computer vision applications. It helps to represent the initial natural data in a reduced format. For instance, corner detectors find locations of corners in one image where the detection part of the SIFT technique is responsible for information encoding about the local neighborhood image gradients the numbers of the feature vector. But the question is which parts of a natural image will be the interest points.

Historically, interest points can be corners, blobs or edges in an image. To expand the notion of the interest points, the following terms are determinative:
1- Well-founded and clear mathematical definition.
2- Well-defined position in the natural image space.
3- Richness of the local image structure around the interest point.
4- Simplifying the further processing in the vision system when it is detected.
5- Stability under local or global perturbations in the image domain. If illumination or brightness variations occur, the interest points should not lose the repeatability characteristic, and they can be reliably computed in any condition.
6- Correct behavior and high degree of robustness of the interest points when the scale of natural images changes.

Through a consideration of the richness of the detected points, the efficiency of the detection technique will be evaluated and determined. The concept of feature detection refers to the techniques that help to compute abstractions of image information. In addition, it contributes to investigating and making local decisions at every image point and pixel whether there is a feature of a given type at that point or not. There are different feature detection techniques. For instance, HARRIS and FAST are two common techniques to do feature detection which can be applied to relax the detection step and the complexity of the original natural images. Fathi Kazerouni et al. [169] and [170] used various detection algorithms, such as HARRIS and FAST, to detect the interest points which had the real concept of computing the abstractions of the image information.

Concerning the execution time of the system with the SIFT algorithm in [151] for plant recognition, our decision is to replace the mentioned algorithm by the SURF algorithm as the core of the description methods. A multi-resolution pyramid technique is utilized in the SURF algorithm to make a copy of the original image with a pyramid shape to obtain an image with the same size but with a reduced bandwidth [268]. Thus, a special blurring effect on the original image, called a scale-space, is achieved [268]. This technique ensures that the points of interest are scale invariant [268]. In [170], the SURF method was used to distinguish 32 different plant species, where the used dataset was a classic dataset and the images were captured against a homogeneous and white background. The final accuracy with the SURF method was 92.28% which is higher than the other proposed systems in [169] and [170].

A PNN is the proposed approach for the semi-automatic classification of the plant species in [45], and the obtained accuracy is 91.41%. In addition to being semi-automatic, this system was has been tested on only classic and artificial images. One detection method can be combined by the SIFT or SURF algorithms and a strong tool for the detection and description will be provided. For instance, this powerful tool was used in [169] and [170] and 32 different plant species were recognized automatically, and the highest obtained accuracy was 92.28%.

In order to organize the obtained information and form them in a useful way for the next steps, an intermediate step has been considered to connect the previously described step to classification. One efficient technique is the BoW which represents the descriptors in a compact format. It provides a brief summary of feature representation and contributes to showing an image in a feature vector. This vector is ready to be applied in a machine learning algorithm. Historically, this technique was often used in the NLP. Nowadays, it has been applied to images, and a word-like concept can be considered as the number of local features in image. In simple terms, this technique treats the document as a set of words. Therefore, the BoW can be applied to images as analogies. In an image, a word can be considered as the amount of local features. When images are expressed by vectors, it is possible to use the SVMs for the classification step.

The next step is to find and select a good learning technique. A learning technique may be either supervised or unsupervised. SVMs [186] [308] [309] and Bayesian classifiers [186] [310] [308] are the most popular techniques between the supervised learning methods, and we should not forget that the decision theory approaches are behind classification solutions.

To perform the training stage, we chose a robust, accurate and effective algorithm which can also be applied even when we have a small training dataset and the final goal is to handle the multiple classes. In 2014, an SVM was the backbone of a system which was proposed for the leaf classification by extracting 12 leaf features and orthogonalizing them into 5 variables where the variables were fed into the SVM [51]. In an SVM algorithm, a vector based machine learning method, each data item can be plotted as a point in the n-dimensional space with the value of each feature being the value of a particular coordinate. Then, the classification will be performed in the next part.

We have divided our natural plant dataset into two sub-datasets. The training sub-dataset consisted of 664 images and the testing one had 336 challenging images. A detailed explanation is provided with a step-by-step procedure for the construction of an automatic system for the natural plant species recognition.

7.4 Approach for Natural Plant Recognition Systems

Natural plant recognition is of high significance because it conveys the goodwill of different applications in medicine, drugs, etc. It is a significant issue among agricultural business undertakings. In addition, it is not only a concept for today's world. It will strongly influence the future success of humans on the Earth. Tremendous amount of plant species all around the world influences us to consider them deeply. One point is to try to use the plants in appropriate applications according to the demands of human life and the environment. Automatic plant recognition systems are required to operate at a high level of reliability and accuracy. The applicability of a proposed system in the natural environment and real-world scenarios has become more important in recent years.

The methodology presented here is an efficient approach for challenging plant recognition. Various systems have been developed by consideration of the approach. In general, it is comprised of two main phases which are actually training and testing. The first phase undergoes three stages: image pre-processing, feature detection and extraction, modeling and training. These sub-phases are debriefed in the three related sections. In order to recognize and classify plant species from a set of plant images, the natural plant images are read before undergoing the pre-processing. Then the grayscale images are computed from the real-world data. In other words, the beginning step is the pre-processing part and it deals with the conversion of RGB images. The second step will be done by consideration of the detection and description operations. The approach continues with the modeling and training as the third step. The final step is the detailed test procedure of the implemented systems. To summarize the overall process of the proposed approach, the block diagram of the proposed system is shown in Figure 7.3.

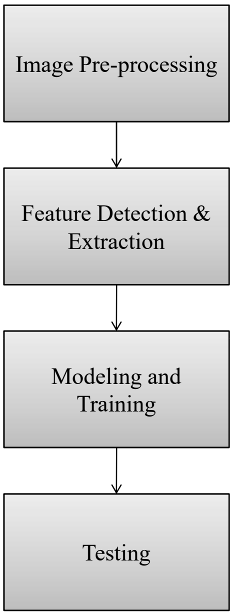


Figure 7.3: The overall process of the proposed natural plant recognition system

7.4.1 Image Pre-processing

The nature of the input images is complex, and the input images are the natural and challenging images in the RGB model. Each natural image of the dataset contains a scene of a plant and we observe different objects such as leaves, stems, challenging backgrounds, etc. An RGB image is composed of three channels: red, green and blue. These channels are usually abbreviated as R, G and B, respectively. By scrutinizing our images, we recognize that the color space is one of the most common color spaces, sRGB. As of 2007, the sRGB color space can be considered as the default color space for the RGB model. Almost all equipment for recording and displaying an image with the RGB color model supports the sRGB color space at a minimum.

Looking at the color quality diagram below in Figure 7.4, you can see that the triangle environment depicts the color range of the sRGB compared to the human vision range (CIE 1931 Color Space) [311]. A large part of the human color vision is outside the color space of the sRGB color space. These are not necessarily the colors we can see and they cannot be displayed in the sRGB color space. These colors are outside the scope of the color space of the sRGB. The fact that most human sight is outside the sRGB color space explains why this color space is at least minimal. It should be considered as a limited color space.

7.4.2 Feature Detection and Extraction

The second step of the approach is the keypoint detection and description parts which are similar to oxygen for the human respiratory system. When humans are looking at a scene, they usually focus on important and interesting details of the scene. Our decision on selecting the type of the feature detection and extraction is inspired by this fact. To represent images, there are two main methods, called global representation and local representation. In the mentioned scenario, the human behavior

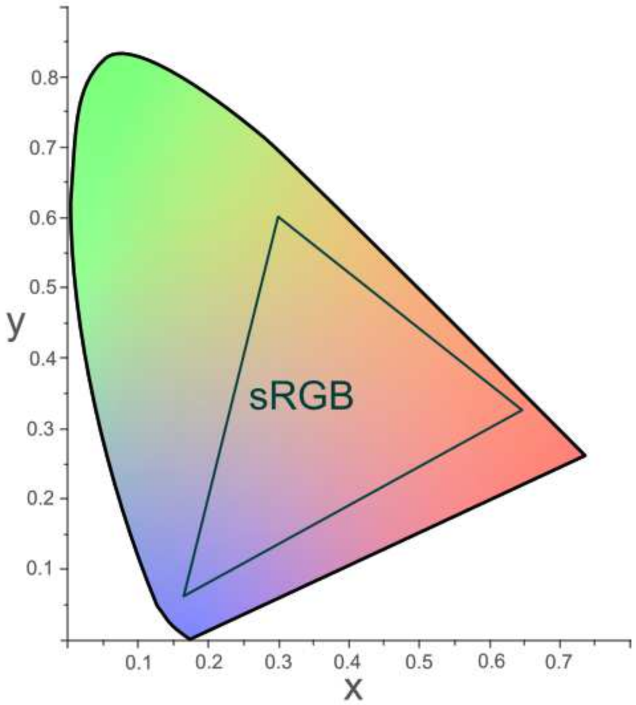


Figure 7.4: Color scheme for the sRGB color space compared to the human vision (CIE 1931)

while viewing a scene can be compared to the global representation and global features. It means looking at the whole image and the word global is actually the interpretation of this subject.

Human's attention to specific parts and details of the scene is comparable to extracting local features of the objects in the scene. In addition, the human eye is undoubtedly able to extract all information from a raw image. But, not all information has the same importance level for computer algorithms. Additionally, some information might not be useful for further applications. Useless information may also increase the computation costs. This point is one of the reasons that motivated us to choose local features instead of global features.

While photographing in outdoor environments, different factors, such as illumination, light intensity, viewpoints, angle, etc., undeniably change. Consequently, priority is given to a method that is resistant to changes. The stability and power of the performances of selected methods against the differing conditions of various scenes, including light intensities and illuminations, scaling, geometry, and shift transformations, are very important at this step. Since global methods are not invariant to transformations and they are mostly sensitive to different changes to the discussed factors, they are not appropriate for real natural images. However, local methods are rich enough to remain invariant to different changes, such as viewpoints and illumination. They are usually based on some salient regions. We will be able to obtain the relevant information from the natural data. Hence, we are able to represent any natural plant image based on its local structures by using a set of local feature descriptors extracted from a set from the regions of interest or keypoints. The structures of local features are usually helpful to be used in object recognition and classification applications as the local structures are more stable and distinctive than other structures in smooth regions. Meanwhile, they lead to achieving high accuracy. In Figure 7.5, the global feature representation and local feature representation are shown in one of the dataset's images, as an example.

Due to the explained facts and superior performance of the local features [312], the methods that produce the local features have been selected out for building our plant recognition systems. By utilizing such methods, large numbers of the local features, comprising hundreds of local features,

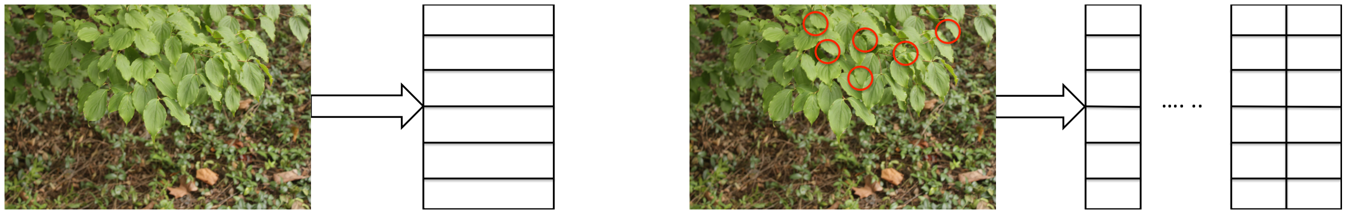


Figure 7.5: Left: Global image features representation, Right: Local image features representation

will be created and the amount of the memory increases in comparison to other types of methods. Therefore, a high amount of memory is a disadvantage of methods with the local representation. A good solution is to aggregate local image descriptors into compact vector representation [313].

In general, the detection method and detected features might greatly affect the desired applications and final goals. To utilize a feature detector, some properties, such as robustness, repeatability, accuracy, generality, efficiency and quantity, should be taken into account to follow the right path and reach the desired destination. In the next section, we investigate and look into some detection algorithms which will be used in our methodology and proposed approach.

Before continuing our explanation, some remaining points should be clarified about the process of feature detection and extraction. We seek invariance properties in the process of the feature detection. The goal is to have a feature extraction process that stays the same even if we involve different specified or non-specified conditions and undesired added phenomena. Consequently, we require a feature extraction algorithm which is able to find reliable and robust features despite the changes in time and appearance of plants, regardless of the reason. Furthermore, we need to have immunity to changes in the illumination level as we would like to identify the plant species from the images and the images might be dark or light in relation to the conditions.

In principle, the existence of the shape of a plant in an image can be approved if there exist contrasts between a plant and its background. The shape of a plant can be detected, as has been shown in the proposed pre-processing methods. As a result, invariance to illumination is an essential property. Clearly, any computer vision technique will fail in extreme lighting conditions. It is the same for the human eye as we cannot see anything when it is completely dark and there is no source of light.

Following the illumination, the next most important parameter is the position: we seek to find a plant (or leaf of a plant) wherever it appears in images captured in outdoor and natural environments. This factor is usually called position, location or translation invariance. Then, we do not want to find a plant species respective of its rotation (assuming that the leaf of a plant or even the camera has an unknown orientation and rotation is also not known). This is usually called rotation or orientation invariance. Then, we want to determine the leaf at whatever size it appears, which might be due to the physical change, or however close the plant or leaf might be placed to the camera. This requires an important property which is size or scale invariance.

The mentioned points are the main invariance properties we shall seek from our feature detection and extraction techniques. However, nature tends to show us that we have to struggle against challenges, and balls are usually rolling under our feet to prove that we are not so lucky: there is always noise in images and different environmental factors affect the photographing process. In addition, since we are concerned with the leaves of plant species, there may be more than a single leaf in the image. If one is on top of the other one, it will occlude, or hide the other, so all of the leaves will not be visible and there might be no leaf with a clear shape. So what should we do to overcome all visible and invisible challenges? What is the solution for clear and unclear sides of the plant recognition task? Before developing the recognition systems, we need powerful techniques to detect and extract the features. A higher complexity of the extraction step compared to the detection step is undeniable, since the extraction implies that we have a description of a feature.

Let's look into our expectation from a detection algorithm briefly. First of all, the detection algorithm should have the ability of detecting the same feature locations regardless of the other factors and parameters like scaling, rotating, shifting, deformations, artifacts, noise, etc. The feature detection algorithm should be a repetitive process which can find the same features of the same plant in any view. Another point is the ability of the algorithm in detecting efficient features where the quantity of features should not be either low or high. The quantity of the detected features should be the reflectance of the compressed information of the image. Another aspect of the detection algorithm is related to its efficiency in real-time applications. It is essential to consider if the algorithm is supported for such applications and it can be generalized as a part of the other systems.

By detecting the keypoints, we calculated the descriptors for all of them with the purpose to fulfill a procedure to use those keypoints in a correct way. Let's suppose having an image matching task. We would like to know if the objects, especially the leaves of the plants, are the same in two different images. We try to identify the similar parts in two different natural images or find differences between two different natural scenes. To solve the proposed problem, we have to compare every keypoint descriptor of the first natural plant image to every keypoint descriptor of the second natural plant image. Descriptors are actually the vectors of the numbers and we are able to compare them with a simple criterion like the Euclidian distance. However, more complex distances are also available to be used as similarity measures. When the distance between the descriptors is the lowest value, the related keypoints to the descriptors are the matches, for instance, the same leaf shapes or the same non-leaf objects in two different images. The following combined modern methods have been used for the feature detection and description:

- HARRIS-SIFT
- HARRIS-SURF
- FAST-SIFT
- FAST-SURF
- SIFT
- SURF

7.4.3 Modeling and Training

Up until this stage, we have gained the feature descriptors of the detected keypoints, but the main question is, "How can we connect the obtained descriptors to the classification stage?" To broach the answer of the proposed question, we should establish a bridge as a connection between the description algorithms and the training phase of the classification part. We have to use a manipulating approach for preparing the existing information in a way that can be applied in the next stage. One possible approach is to quantize local visual features. A manipulation approach, inspired by the BoW method, frequently used in the text domain and document classification [314] [315], is used for constructing a bridge between the previous and next steps. This concept has opened a new door into another domain, the visual domain of the image processing, as a means to describe images as a collection of words, to do object categorization and classification tasks and to achieve surprisingly promising results [206] [316] [186] [317] [318] [319].

Due to the main concepts of the BoW, we need to transfer the local features into the visual words, especially in the image representations [320] [321] [322] [323] [324]. Then, we build a codebook from the obtained local descriptors and try to create the final outputs, the image histograms. The pixels of an image can be represented as the letters in a text document. The structure of a pixel and its surrounding neighbors can consequently be assumed as words, the basic elements, in the text document.

The images of the dataset own unique characteristics because of very large variations among the images and scenes. Many clutters in the background and foreground can be observed in the images of the dataset. Scale, light intensity, time, distance and illumination are not kept constant and they are variant parameters. The idea of the BoW is to make compact packages of descriptors from the local features of the categories of different natural images. In addition, the intention is to obtain a finite number of clusters and create a visual vocabulary.

Splitting an image into small image patches helps us to represent them as numerical vectors by means of feature descriptors and steers towards having a set of words. Clustering algorithms, the key elements of machine learning, are helpful to convert these types of vectors into words and produce vocabulary. The words are then defined as the centers of the learned clusters. Moreover, each group can be considered as one specific word. The next step will be mapping each patch of an image to certain visual words through the clustering process, so the image can be represented by the codeword histogram. It should be noted that the number of the clusters is the vocabulary size. In [169] and [170], the BoW technique has been used to classify a large number of plant species because it is unaffected by the position and orientation of the object in an image. Hence, the BoW technique is a good choice and helpful for representing the images as we need them for the next steps.

To have a vector representation of the images, the quantized feature space contributes to indicating the frequency of the visual words which can be utilized in conjunction with some vector-based kernels or similarity measures for the matching or categorization of the image content. The question then arises, "What is the difference between the current approach and the one used for the previously implemented systems?" To answer this vital point, we have to reconsider the natural plant dataset and explore its properties. The changes in the distance between the camera and the plants in the dataset influence this part of the approach. In fact, we cannot build only a vocabulary for the whole process. Due to the change of distance, we need to construct a vocabulary for each distance separately.

In the BoW model, the image's features are denoted into words by specifying which visual words are actually nearest in the feature space based on the Euclidean distance between the cluster centers and the input descriptor. Moreover, each extracted region should be assigned to the corresponding visual word in the test phase. The model will be utilized for the new natural images in a certain procedure as well. Firstly, the keypoints of the new image will be detected. Then the descriptors will be extracted from them secondly. Thirdly, the nearest neighbor in the constructed vocabulary will be computed for each descriptor, and the histogram will be built in the final step. It should be considered that i_{th} value in the histogram is actually the frequency of the i_{th} vocabulary word. Finally, the histograms will be fed into a classifier to predict the labels and classes for images. Classifiers need fixed dimension feature vectors. The whole approach is really fast, robust and simple to understand. The properties and useful information of the detected keypoints have been captured and modeled to start the learning and classification steps.

Moreover, the BoW method can be replaced by neural networks. A unique part of our work, which will be explained in next chapters, is the use of deep neural networks to design and implement an excellent recognition system.

7.4.4 SVM Classification and Testing

After ending up with the BoW model, we need to sort all obtained results and design a concrete classifier to fight this challenge and formulate the desired tasks. In the previous steps, the primary substance of the classifiers has been prepared for each class. Now we come up with a new area, called SVM, to complete the necessities of the current step. The SVM, a powerful tool based in the statistical learning theory, is really useful in real world applications, especially in classification tasks, due to its good performance. In image recognition, supervised machine learning models have been considered as efficient methods in many cases. The SVM involves efficiently finding and separating the optimal hyperplane in higher dimensions, which maximizes the margin of the training data to guarantee the correct classification of the input pattern. When the number of dimensions is greater than the number of samples, the SVM stays efficient and it is one of its benefits in addition to the efficient memory. Therefore, concepts of decision planes have been applied to define decision boundaries.

As an example, it is feasible to separate the data from two categories by a hyperplane when an appropriate mapping is applied. Consequently, the hyperplane has the largest distance to the nearest training data point of any class, and it is then called the functional margin. In other words, from the given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes the new examples. Intrinsically, the SVM relies upon the data pre-processing to show the patterns in a higher dimension instead of the original feature space. Extension of the original SVM method [246] has been utilized in regression, classification and clustering problems.

The SVM method has some special benefits. One of the important benefits of this method is its ability to project the input data points on the high dimensions by means of the kernel functions to obtain separating hyperplane converting to lower dimensions. Furthermore, the method is more effective and efficient in high dimensional spaces. It does not lose its efficiency even if the number of dimensions is larger than the number of samples. For constructing an optimal hyperplane and minimizing error, the used method is an iterative training algorithm. To review the mathematical aspects of the SVM, we can refer to the explanations in [21], [325], [253], [326] and [327].

According to the explanations of Cortes and Vapnik in 1995 [246], we tried to use SVM as it demonstrates an acceptable performance and is more reliable than the other methods like Naive Bayes classifier. An ability of the SVM is to locate a separating hyperplane in the feature space and classify the points in the space without showing the space explicitly by utilizing a kernel function. It should be noted that the SVM can operate correctly even if the designer does not know how it is really working or completing the task. Therefore, it does not depend on the designer's knowledge. By using the kernel trick, it is also possible to build in the expert knowledge about the problem via the kernel engineering.

We want to train the SVM on a set of natural plant images and construct the training matrix. This process is followed by putting the histogram responses for each class and then setting up the labels for each training image. For instance, we have two different classes, leaf and non-leaf classes. One necessity is to define which row in the training matrix corresponds to a leaf and a non-leaf. In this special case, if the 1^{st} element of the label matrix is $+1$, it proves that the 1^{st} row of the training matrix falls into the leaf class. As a result, a $1D$ label matrix is defined and each element of this matrix corresponds to one row in a $2D$ matrix. It is noteworthy that different kernel types for the SVMs are available such as the linear kernel [328], polynomial kernel [328], RBF, sigmoid kernel [328], exponential chi2 kernel [329], and histogram intersection kernel [329]. The used SVM is explained in 7.5.

In a nutshell, the SVM technique is a part of the systems proposed and implemented in [169], [170], [268] and [151], since it has some key features, such as possibility of using different kernel functions, absence of local minima, the sparseness of the solution and the capacity control obtained by optimizing the margin. In fact, the SVM works differently, and it is a good and fast solution for many problems, especially plant recognition. One important side of the SVMs is that they own a regularization parameter which forces us to think about the important relevant issues, regularization and overfitting.

Furthermore, there are other methods such as random forests, probabilistic graphical models [330] or nonparametric Bayesian [331] methods with advantages and disadvantages. In a classic NN, the amount of parameters is enormously high. If there are the vectors of the length 100 and we want to classify into two classes, one hidden layer of the same size as an input layer will lead us to more than 100000 free parameters. We know how badly we can overfit and how easy it is to fall to the local minimum in such a space as well as how many training points we will need to prevent that and how much time will we need to train them.

7.5 Experiment, Discussion, Results and Performance Analysis

In our experimental activity, we have conducted different separate and connected experiments. Each experiment aimed at assessing the proposed scenarios and systems to recognize the plants in the challenging natural environment. The dataset of the images was acquired by using a Canon camera, Canon EOS 600D. The images were taken over different areas near the Hölderlin Campus of the University of Siegen (Siegen, Germany) and at different times, dates and weather conditions. The camera used is characterized by one 18-megapixel complementary metal-oxide semiconductor (CMOS) sensor, Digital Imaging Integrated Circuit (DIGIC) 4 processor. It is a shot-friendly and powerful camera in a small package for taking the pictures in any situation. It also has a scene intelligent auto mode and a 3-inch (3.2) vari-angle clear view LCD which give us the freedom of using various features to capture our images in different conditions with or without a flash. Figure 7.6 represents the camera used.

The International Organization of Standardization (ISO) speed is 400, but other settings of the camera changes across the different images taken. For instance, Figure 7.7 has the following camera details:

Figure 7.6: The camera used

F-stop: f/4.5
Exposure time: 1/30 sec
Focal length: 36 mm
Flash mode: No flash, compulsory
Dimension: 5184×3456

The following camera details belong to another sample of the dataset, and Figure 7.8 represents this sample.

F-stop: f/6.3

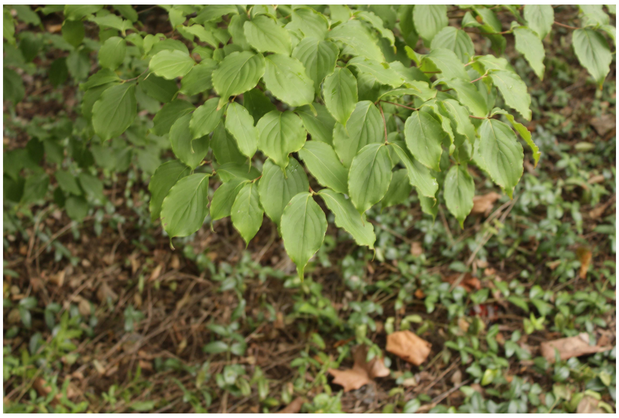


Figure 7.7: A sample of the dataset using the first setting

Exposure time: 1/100 sec
Focal length: 36 mm
Flash mode: No flash, compulsory

The acquired images have the following properties:

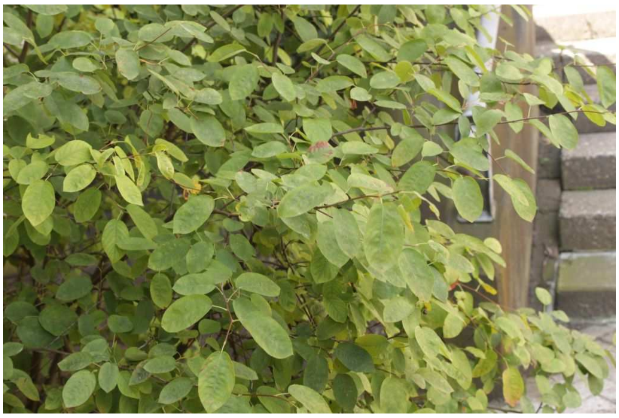


Figure 7.8: Another sample of the dataset using a different setting

Dimension: 5184×3456
Width: 5184 pixels
Height: 3456 pixels
Horizontal resolution: 72 dpi (dots per inch)
Vertical resolution: 72 dpi
Bit depth: 24
Resolution unit: 2 (this value means resolution should be interpreted as dots per inch, and if it was equal to 3, resolution should be interpreted as dots per centimeter)
Color representation: sRGB

7.5.1 Short Description of the Dataset and Setups

Recently, there has been a large amount of academic and non-academic research investigating different aspects of inventing the plant recognition systems such as [332], [333], [169], [170], etc. Most of the proposed systems are based on visual techniques and learning algorithms. Furthermore, there is a big obstacle which is the use of the artificial images and the images taken in defined conditions, for instance, where light intensity, illumination and distance are kept constant. Such parameters actually lead to a restriction of the implemented systems. In order to reduce the limitations of the current systems, it is essential to consider the huge impacts of the parameters that help us to build systems for recognizing natural plants, specifically plants in the natural environment. The variety among the plants in the natural environment is higher than when we set a specific condition for the environment or when the environment is artificial or under the human control. Despite the roughly similar appearance of a plant species' leaves in an outdoor environment, we see different colors and structures among the leaves as well as shapes. As mentioned previously, environmental factors, such as light intensity, illumination, time of day, etc., also affect plant recognition in outdoor environments. Furthermore, these factors influence the outcomes of systems. Our main goal is to drive through the challenges of natural plant recognition. Hence, we created a new dataset of natural images with a high diversity among our samples. According to our work up until this stage, we are going to follow our previous works of [169] and [170]. On the other hand, in regard to our stringent considerations, we face new challenges such as moving leaves under heavy wind, plants in cloudy weather, leaves in sunny weather, etc., instead of controlled weather conditions. These challenges can be clearly seen in the dataset of the images taken. We did not create any artificial or human-made conditions during the process of photographing and recording the images.

In particular in this dataset, the demands of generalized systems will be fulfilled, and the drawbacks of systems for the recognition of plants in hard situations will be remedied. The dataset is comprised of 1000 natural images of four different plant species, Hydrangea, Amelanchier Canadensis, Acer Pseudoplatanus, and Cornus, which are the common plants of Siegerland, a landscape that is a part of south Westphalia in Germany.

Similar to other classification tasks, we divided the original dataset into two different sub-datasets, called the training dataset and the test dataset. The number of images in the training dataset was 664, while the number of images in the test dataset was 336. Table 7.1 represents the number of images for each distance and subset. Someone might put forward the question, "Why did we take pictures at different distances?" Overall, plant recognition systems can be utilized by field robots which can be used for different purposes and tasks on farms and in its related industries. Imagine a robot is going through a farm, the distance between the robot and crops will not always be constant because of the ripeness of agricultural land and the robot's movements. Having samples at different distances between the camera and the plants compensates for the lack of this feature in current plant recognition systems. This unique characteristic helps us to obtain a freedom for recognizing the plants at various distances. It should be pointed out that we always measured the distance between the camera and the plant accurately when capturing the images of the dataset. We usually double-checked the measured distance.

Dataset	25 cm	50 cm	75 cm	100 cm	150 cm	200 cm
Number of images for training dataset	160	160	160	160	12	12
Number of images for test dataset	80	80	80	80	8	8

Table 7.1: The number of training and test images in each distance separately

By investigating the dataset, we found that the main configuration of the dataset which is large variations appearance such as scale, illumination, pose and background clutter in the natural training and test images, and the images have been taken at different distances, angles and views [151]. As explained, the other factors, like light intensities and illumination, light reflection, weather conditions and time of taking the images, have been changed during the preparation and finalizing of the dataset. Figure 7.9 shows two samples of the training dataset and two images of the test dataset.

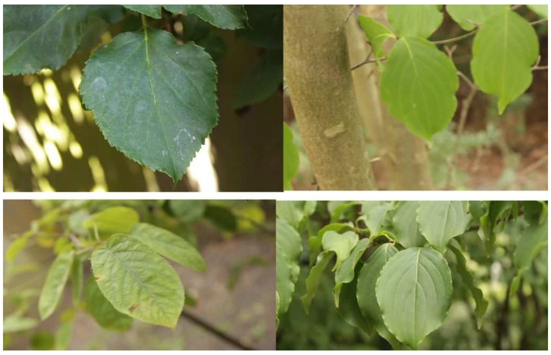


Figure 7.9: Representing four samples of the dataset, two training images on the top and two test images at the bottom

7.5.2 Details of Equipment

The object recognition abilities of the humans are completely different from the machine-based systems, and it is very tough and complex for a machine. To design and develop a useful and applicable plant recognition system, and fulfill the recognition task, the machine we used has the listed components:
- Intel® Core^{TM} i7-4790K
- CPU @ 4.00 GHz
- Installed memory (RAM) 16.0 GB

This machine is exactly the same one that has been previously utilized for the other systems, and we met our aims for the tasks of the plant recognition.

7.5.3 Visual Analysis of Natural Images

In the natural environment, the shapes of leaves are not predominantly well formed. It is very difficult to arrive merely at a visual analysis. In order to verify the shapes of leaves of natural plants, it is necessary to investigate several images of a specific plant species and find the shape and pattern of its leaves. The outgrowth of the dataset's investigation is the visual analysis of leaves which can lead to an exhaustive guide for our goals. It is also important to know whether the investigated samples of the plant species are healthy. For instance, if we look at one of the plant species called Cornus, we find out that some leaves are not completely green. The outliers of them have become yellow after a while. Turning yellow is not only a physical change, it has some effects on our work for plant recognition. There are a variety of nutritional deficiencies that can be addressed as we encounter a bunch of leaves in each scene. Though, each leaf of the plant might have its own properties in detail. In the following, some deficiencies are listed and described:
- Boron deficiency is a widespread deficiency of the plants all around the world, and a common result of this deficiency is a reduction of the crop production and quality. Boron is not a negligible element for the growth of plants, and a lack of it causes a deformity of the shapes of leaves, a loss of symmetry to the margins of leaves and a complete or partial absence of the apex.

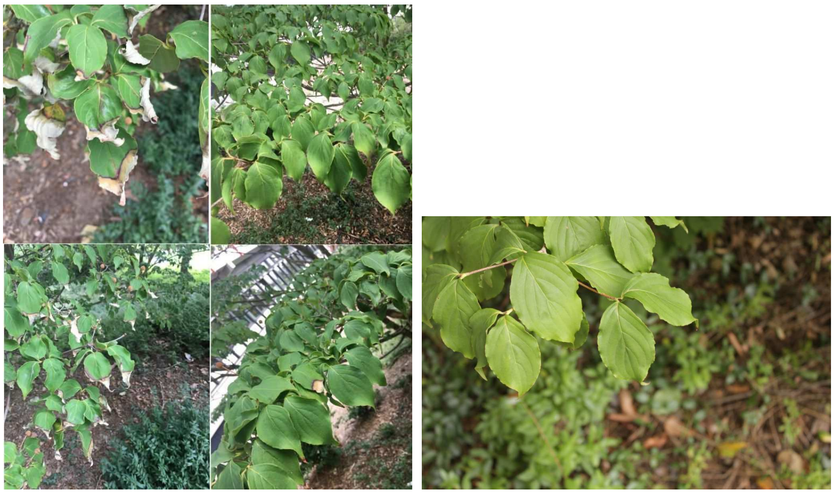


Figure 7.10: Four samples of one specific plant species with the deformed shapes of the leaves (Left), a sample which is not deformed (Right)

- The result of shape deformation and undulations might be calcium deficiency.
- The reason for the unusual shapes of leaves can be a deficiency of phosphorus. Another impact of phosphorus deficiency is the change of color to yellow or red in some parts of leaves.
- The reason of another change of leaves might be a lack of nitrogen. This deficiency effects chlorosis uniformity in the entire area of the leaves and the replacement of the color green by yellow from the base to the apex and the central vein to the leaf borders.
- Other deficiencies, such as iron deficiency, magnesium deficiency, manganese deficiency and potassium deficiency have usually visual effects on the leaves where color changes might appear.

Figure 7.10 shows some samples of shape deformation and color changes in the dataset contents.

7.5.4 Experiments and Measurements

Experimental assessments of the proposed systems have been prepared in different scenarios, measurements, evaluations, and comparisons. Each experiment has been organized into two groups of systems. The members of the first group are the systems with the SIFT algorithm as the descriptor. The second group is comprised of the systems with the SURF algorithm as the descriptor. In order to demonstrate the performance, effectiveness and applicability of the proposed natural plant recognition systems with different approaches and techniques, we investigated the experiments and the results [151] [268]. Whereby, 664 scenes were captured by the camera and used for the training, and the remaining scenes of 336 images were utilized for the testing. We present the experimental results on several measurements to answer the proposed problems. The results obtained are also evaluated by comparing the output of the different systems. The investigation of the results recalls Stephen Few's talk, "Numbers have an important story to tell. They rely on you to give them a voice [334]." It should be pointed out that all result-numbers reference to the test dataset.

Accuracy of Classification Using Different Proposed Systems

Since we have taken the images at different distances, we are able to define four different groups according to the distances. The first indicator of the systems' performance is the accuracy of the classification. Accuracy (6.8) is one of the performance metrics in classification problems. Using the accuracy helps us to check out the number of correct predictions made by the natural plant recognition system over all performed predictions, both correct predictions and wrong predictions together.

Table 7.2 shows the accuracy of the natural plant recognition results under different sets of the detection and description algorithms trained by the SVM for all distances in only one package without discriminating the systems due to the assigned distances [151] [268].

Proposed System with Detection and Description Approaches	Correct Predictions	Wrong Predictions	Percentage of Accuracy
SIFT	319	17	94.94
FAST-SIFT	309	27	91.96
HARRIS-SIFT	314	22	93.45
SURF	316	20	93.96
FAST-SURF	306	30	90.94
HARRIS-SURF	303	33	90

Table 7.2: The accuracy of the classification by applying each proposed system with its unique detection and description approaches [151] [268]

Calculation and Construction of the Confusion Matrix for Proposed Systems

A system that is generated at the classification and learning stage should be analyzed at an evaluation stage to determine its applicability. Subsequently, it is needed to identify the performance of the proposed algorithm and the details of the system. In the testing phase of system, there is a useful and popular concept for the classification which is called confusion matrix. The success of the systems may be evaluated by comparing the constructed confusion matrix of each system. Despite the simplicity of the confusion matrix, we find it a colorful world of information for our implemented systems. In [151], the confusion matrix has been called a visional tool to evaluate the performance of each proposed model or system in the classification and prediction tasks. It has been also named a predictive capability in the classification tasks.

To introduce the confusion matrix, we are going to explain some important properties of it firstly. This is a square matrix of order n, where n is the number of the target classes, in our case the number of the plant species, and the number of the rows and columns is equal to 4 in our case [151]. The trace of the confusion matrix as a square matrix is the sum of its main diagonal elements. The confusion matrix describes the performance and quality of an automatic plant recognition system. It contributes to easy understanding of the results obtained. Considering the structure of the confusion matrix, the columns represent the predicted class or label, and the rows correspond to the true and actual class or label [151] [335].

In 11.6, we build the confusion matrix for each developed system at different distances separately, and the related table is created. Each table, Table 11.1 – Table 11.24, illustrates one square matrix 4-by-4 for each system at the associated distance [151] [268] and helps us to identify and compare the inherent features of the classification errors.

In addition to all obtained information from the confusion matrix, this matrix conveys two other criteria: precision (6.9) and recall (6.10). The precision is "how many of the chosen samples are true" and the recall is "how many of the correct samples have been chosen." In the next section, we begin by going through a new experiment to provide these two criteria which are the statistical measures based on the evaluation of the confusion matrix. Hence, we examine the quality of the implemented systems and the status of the trained and tested models from the new aspects as well.

Evaluation of Proposed Systems by Using New Measurements Obtained from the Confusion Matrix

After accuracy calculations and building the confusion matrixes, we drive two common metrics, the precision and the recall, to complete our investigation and comparison. Apart from the information of confusion matrix, these two metrics help us to have a comparative evaluation of the proposed and implemented systems by different approaches. The computation of the precision is achieved by dividing the exact number of the correctly classified positive examples by the number of the examples labeled by the system as positive [151]. The definition of the next metric shows that the recall is obtained by means of dividing the number of the correctly classified positive examples by the number of the positive examples in the test data [151]. Before calculating the metrics separately, it should be noted that the behavior of the metrics provides different meanings in the final evaluations. If we consider the metrics of one implemented recognition system as the high value of recall and the low value of precision; in such values, the consequence would be a return of many results from automatic systems. However, a lack of the correct predictions is undeniable and we find many incorrect predictions compared to the actual training labels.

If we find the reverse conditions where the precision is high and the recall is low for a natural plant recognition system, the system returns considerably few results. But, most of the predictions are correct in comparison to the actual training labels. Let's think to an ideal system according to the precision and recall measurements and attempt to find what type of the system is ideal. Here, we would like to write a prescription for the quality of the automatic systems due to the mentioned measurements. If the recall and precision values are both high, the system is so close to an ideal and satisfactory recognizer of plant species. The highest score of either the precision or the recall is 1.0. For instance, if we obtain the precision score of 1, this would mean every result retrieved by the proposed plant recognition system was predicted correctly. Though, it does not give any information if all the relevant plants were correctly predicted. Another possibility is to obtain a perfect recall value, the highest value which is equal to 1.0. In this case, it means that all relevant predictions were retrieved from the natural plant recognition system. This metric does not provide any information about how many of the retrieved results were actually irrelevant.

As we have different distances, the precision and the recall for each proposed system have been derived from its own confusion matrix. The measurements and calculations are shown in different figures. In addition to the drawn graphs, a comparison of the results is also feasible if we investigate the area under the curves. A high area under a curve is interpreted as both high recall and high precision.

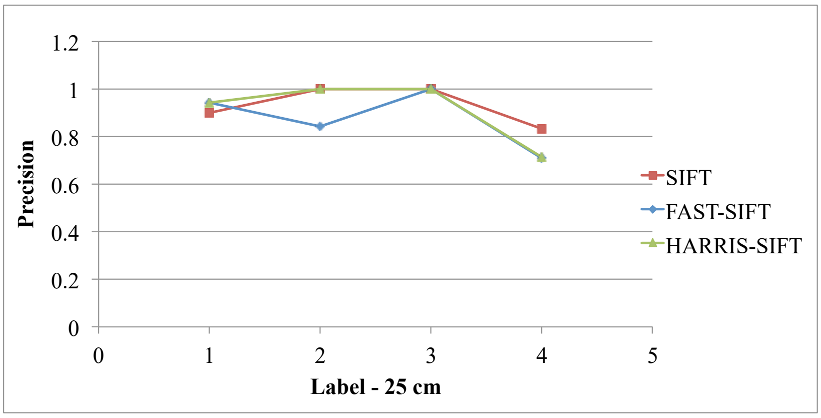


Figure 7.11: Precision measurements for three SIFT description-based systems for the natural plant recognition (distance 25 cm) [151]

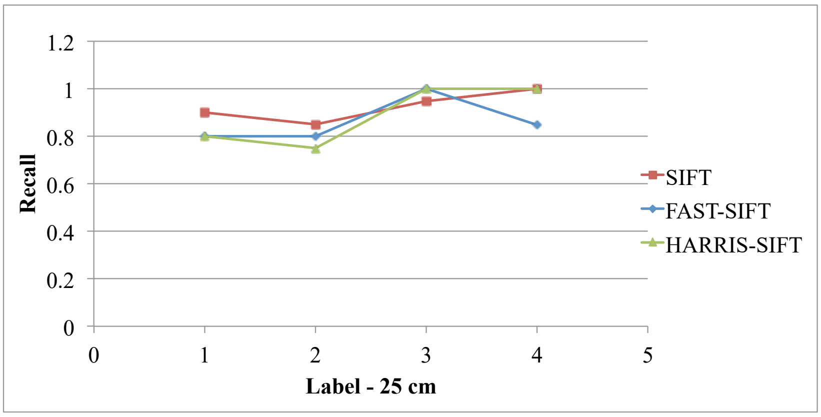


Figure 7.12: Recall measurements for three SIFT description-based systems for the natural plant recognition (distance 25 cm) [151]

The first distance equals 25 cm where the images of the natural plant dataset have been captured at this distance. Firstly, we calculated the precision and recall measurements for the first group of the proposed systems based on the SIFT description approach at this distance. As a reminder, we should note that precision is a measure that tells us what proportion of the plants we have recognized as the class of the plant species is actually the intended plant species. To accomplish the second experiment of this stage and fulfill the mission, the precision and recall measurements of the current distance have been computed for each proposed system in one figure individually. Figure 7.11 and Figure 7.12 show the precision and recall results of the proposed systems based on the SIFT description approach at the distance of 25 cm.

One important point is that the labels used, which are label 1, label 2, label 3 and label 4 represent Hydrangea, Amelanchier Canadensis, Acer Pseudoplatanus and Cornus, respectively.

To specify the outcome of the figures and curves, we are able to check the variation of the proposed systems in the precision and recall measurements. In addition, we can compare the recognition systems by means of this unique concept. For instance, the proposed system with the combined approach FAST-SIFT has more variation than the other proposed systems of this group in the precision and recall curves. Interestingly, the performance of this system is worse than other systems when the distance is 25 cm and we are trying to identify the natural plants. In addition to these metrics, it has the lowest accuracy among the other systems. If we gather all results, we conclude

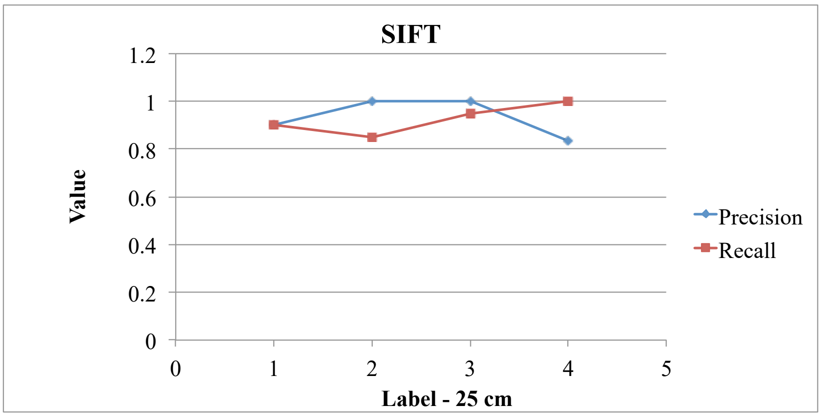


Figure 7.13: Measuring precision and recall metrics for the proposed natural plant recognition system based on the SIFT detection approach using the SIFT description approach (distance 25 cm) [151]

that it has been a predictable fact due to the obtained confusion matrix and extracted information. According to the obtained results of the precision and recall measurements, the proposed system using the SIFT detection approach has the least variations between all the proposed systems of the current group. The performance of the system with the HARRIS detection approach is less than the system with the SIFT detection approach, but it is more than the system with the FAST detection approach.

A new advantage of the plotted precision and recall measurements is the possibility of comparing the results of the proposed systems simultaneously and if considering each label as a dot, we are able to investigate and compare the proposed systems dot by dot in each figure. For example, the considered label is the second one, label 2. In Figure 7.11, both systems based on the HARRIS-SIFT and SIFT approaches have the highest possible precision value. They are equal to 1.0, whereas the system based on the FAST detection approach has a value between 0.80 and 0.90. In the above mentioned label, if we check the recall values, we find that the system based on the SIFT detection approach has the highest value among the proposed systems. According to the obtained values, the second rank of the recall measurements in the label 2 belongs to the system which is based on the FAST detector. The lowest value of the recall measurements in the label 2 belongs to the proposed system by the HARRIS detection approach.

Another investigation of the measured precision and recall is to compare the area under the

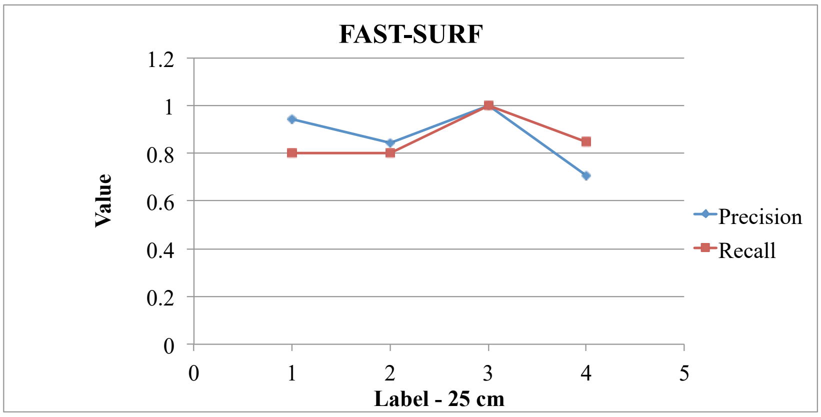


Figure 7.14: Measuring precision and recall metrics for the proposed natural plant recognition system based on the FAST detection approach using the SIFT description approach (distance 25 cm) [151]

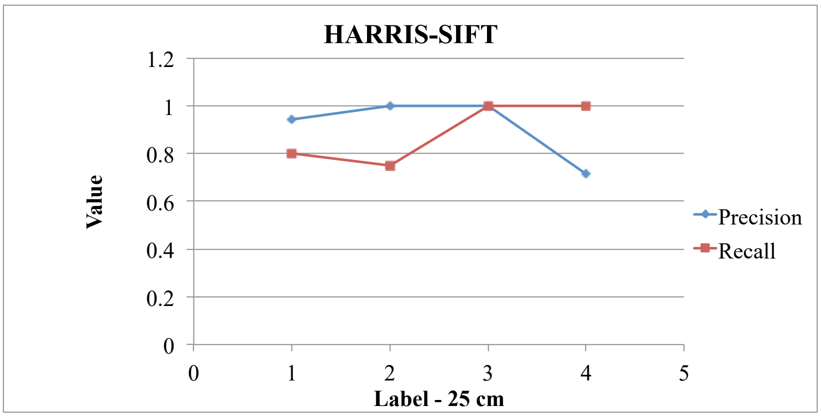


Figure 7.15: Measuring precision and recall metrics for the proposed natural plant recognition system based on the HARRIS detection approach using the SIFT description approach (distance 25 cm) [151]

curves of each system. If we consider the plotted precision curves of the systems based on the SIFT and FAST detectors, we find out that the area under the curve of the systems based on the SIFT detection approach is larger than the area under the curve of the systems based on the FAST detection approach. Combining the obtained results of both systems proves a better performance of the proposed system based on the SIFT detection approach than the proposed system based on the FAST detection.

To compare the precision and recall measurements for each proposed system individually, we have plotted these measurements of each system in a figure separately, and Figure 7.13, Figure 7.14 and Figure 7.15 represent the measurements of the precision and recall metrics for each system in one figure.

The next group of the proposed systems consists of three implemented systems based on the SURF description approach. First of all, the two figures, Figure 7.16 and Figure 7.17, are plotted and each one separately contains the precision measurements and recall measurements of the proposed systems.

At the distance 25 cm, the system that excites our attention is the proposed system based on the HARRIS detector. Its performance is comparable to the proposed system based on the SURF detection approach. The high precision and recall values prove this interesting fact which can be also double-checked through an investigation of the area under its precision and recall curves. In order to observe the relationship between the precision and recall measurements of the proposed system,

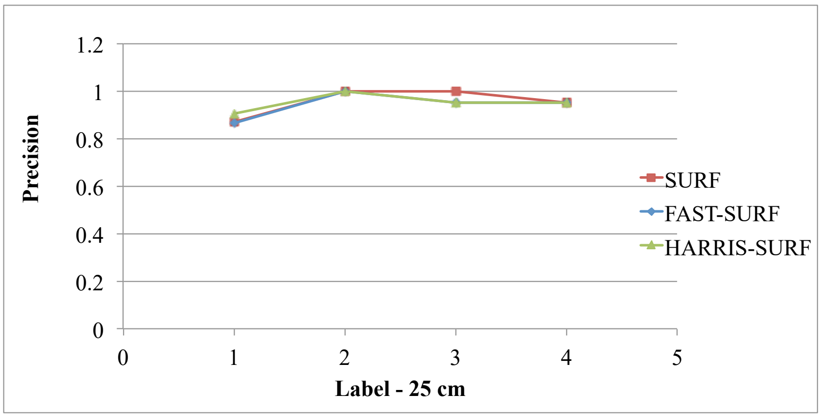


Figure 7.16: Precision measurements for three SURF description-based systems for the natural plant recognition (distance 25 cm) [268]

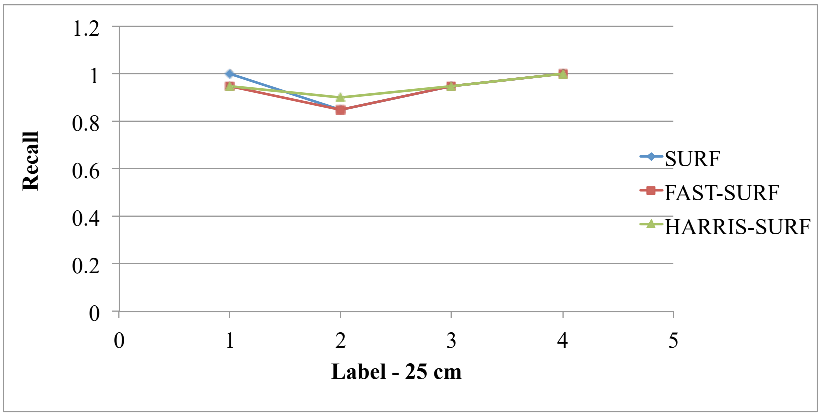


Figure 7.17: Recall measurements for three SURF description-based systems for the natural plant recognition (distance 25 cm) [268]

Figure 7.18, Figure 7.19 and Figure 7.20 have been plotted. Each figure indicates the obtained values of these two metrics for each system individually.

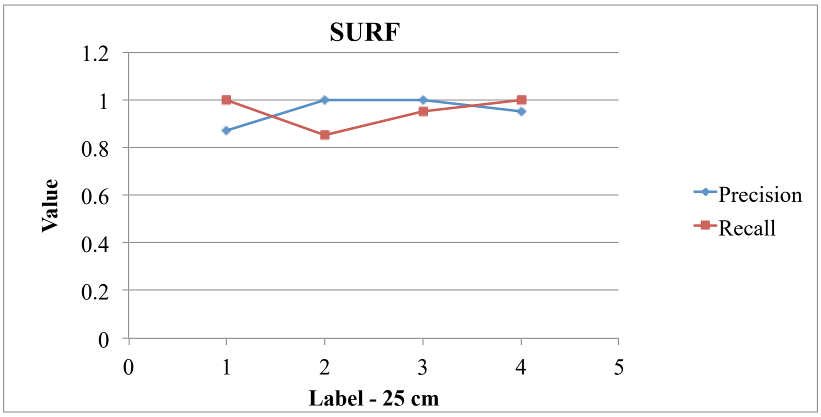


Figure 7.18: Measuring precision and recall metrics for the proposed natural plant recognition system based on the SURF detection approach using the SURF description approach (distance 25 cm) [268]

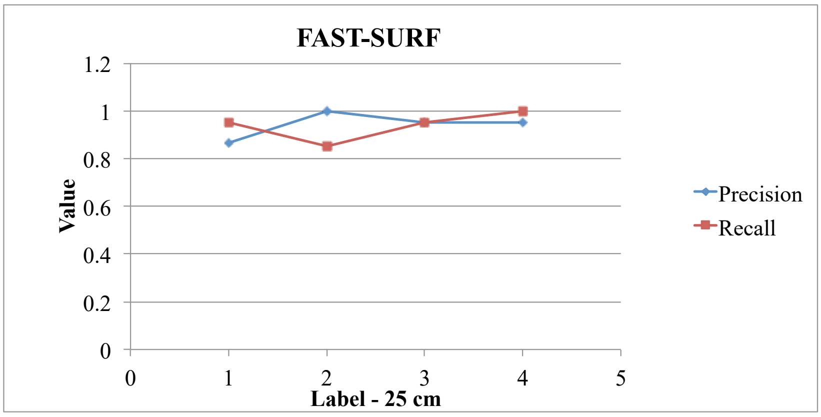


Figure 7.19: Measuring precision and recall metrics for the proposed natural plant recognition system based on the FAST detection approach using the SURF description approach (distance 25 cm) [268]

Clearly, the recall measurements provide the information about the performance of the system with respect to the missed classifications. The precision measurements provide the information about the performance of the systems by consideration the correct classifications. Hence, minimizing the false negatives leads to achieving a recall value of 1.0, and the result of minimizing false positives is to have a precision value of 1.0 as well.

If we consider one label, e.g. label 3, the precision value is at a maximum for the proposed system based on the SURF detection, and it is equal to 1.0. In this case, the values of the precision measurement are not maximum for the other proposed systems based on the FAST and HARRIS detection approaches.

The next experiment is to increase the distance between the plants and the camera. Now we are exactly at the distance of 50 cm. The distance change affects the outcomes of the precision and recall measurements.

At this distance, the proposed system based on the SIFT detection approach has the lowest range of the variations among all labels if we compare it to the other systems based on the SIFT description approach. An excellent performance from the system using the HARRIS-SIFT combined

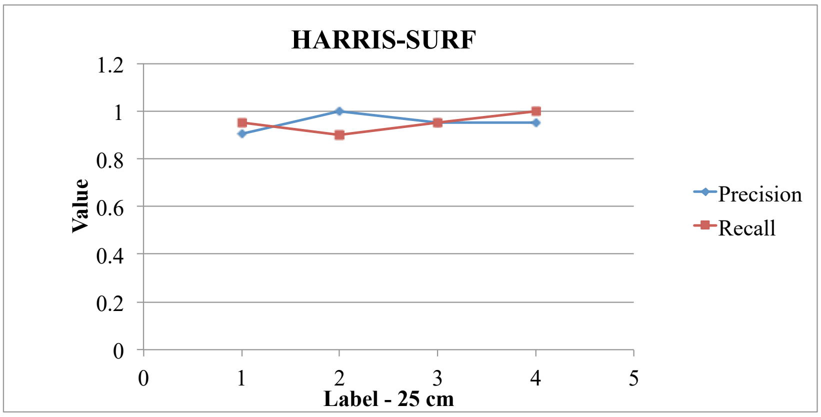


Figure 7.20: Measuring precision and recall metrics for the proposed natural plant recognition system based on the HARRIS detection approach using the SURF description approach (distance 25 cm) [268]

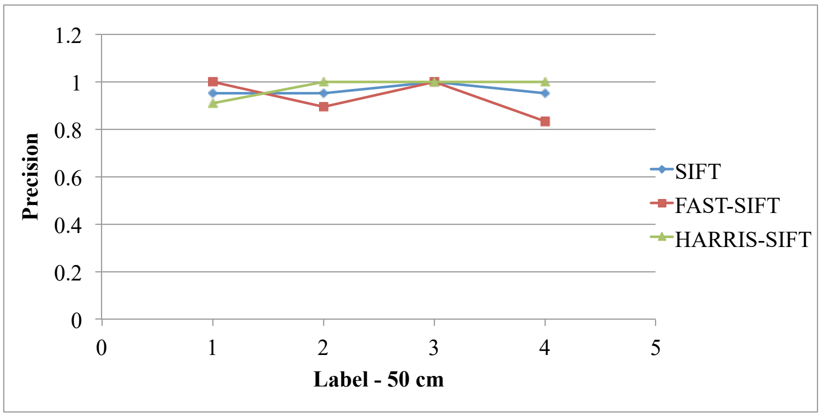


Figure 7.21: Precision measurements for three SIFT description-based systems for the natural plant recognition (distance 50 cm) [151]

approach was surprising. It implies that we are able to apply this system at this distance and use specific properties of this system in a correct form. If we compare the systems based on the combined detection and description approaches with the SIFT description basis at the distance 50 cm, a superior performance from the system with the HARRIS-SIFT approach is undeniable in the precision and recall measurements. Figure 7.21 represents the performed experiment for getting a precision measure of the three proposed systems.

Figure 7.22 shows the recall measure for the three SIFT description-based systems to recognize the natural plants at a distance of 50 cm. At this distance, the worst performance among the implemented systems with the SIFT description approach returns to the system using the FAST-SIFT approach.

Figure 7.23, Figure 7.24 and Figure 7.25 represent the recall and precision measurements for the three SIFT description-based systems to recognize the natural plants at the distance of 50 cm. Each figure is representative of both measurements of the system together. The area under the precision curve of the system with the SIFT detection approach is larger than the system with the FAST detection approach. This is evidence of a better performance from the implemented system with the SIFT detector in comparison to the proposed system with the FAST detector.

To complete the investigation of the remaining systems of the current system, we are going to present the precision and recall measurements in separate figures. The results of the metrics are

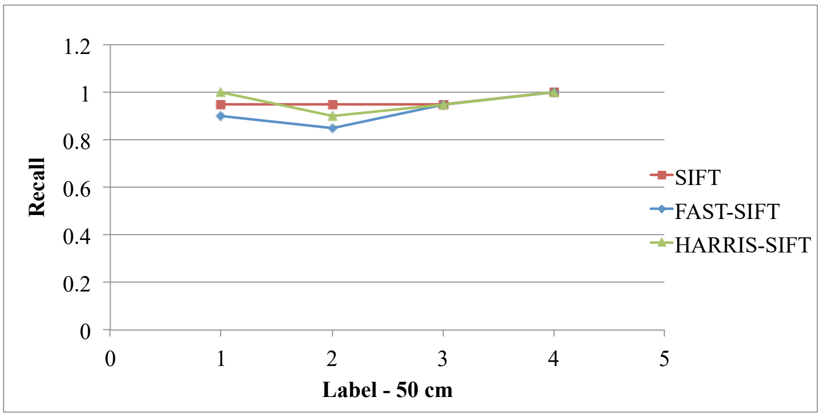


Figure 7.22: Recall measurements for three SIFT description-based systems for the natural plant recognition (distance 50 cm) [151]

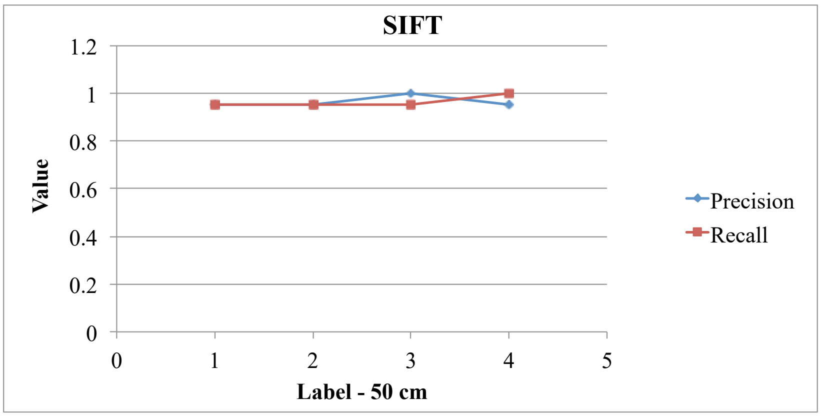


Figure 7.23: Measuring precision and recall metrics for the proposed natural plant recognition system based on the SIFT detection approach using the SIFT description approach (distance 50 cm) [151]

shown for all proposed systems of this group in its own figure.

In Figure 7.26 and Figure 7.27, the implemented system using the SURF detection approach has the largest areas under its curves, as shown in the plotted graphs. The system which has the smallest areas under its curves is the proposed system based on the HARRIS detection algorithm. In the recall measurement, the recognition systems based on the HARRIS algorithm has the highest range of variations if we consider the values of all four labels. If we consider the results of the accuracy of the classification, the recall and precision metrics, we find a superior performance from the system based on the SURF detector compared to the other systems based on the SURF descriptor.

Figure 7.28, Figure 7.29 and Figure 7.30 show precision and recall measurements. Instead of checking the area under the plotted curves, we are eager to investigate the results for one specific label. The second label is actually Amelanchier Canadensis. The measurements of the recall metric indicate that the system using the SURF detection algorithm has the highest value in this label where the system using the FAST detection algorithm has the second rank with respect to the measured recall values in this label. Furthermore, the system based on the HARRIS detector does not perform as well as the two other systems in this label. Its recall value is less than the others.

To follow up the further study of the experiments, we continue checking the precision and recall metrics for the first group of the proposed systems which are based on the SIFT as their description component. We increase the distance between the plants and the camera and the new distance is 75 cm. The precision and recall experiments will be carried out at this new distance to inspect the performance of the implemented systems. Figure 7.31 and Figure 7.32 show our results at the di-

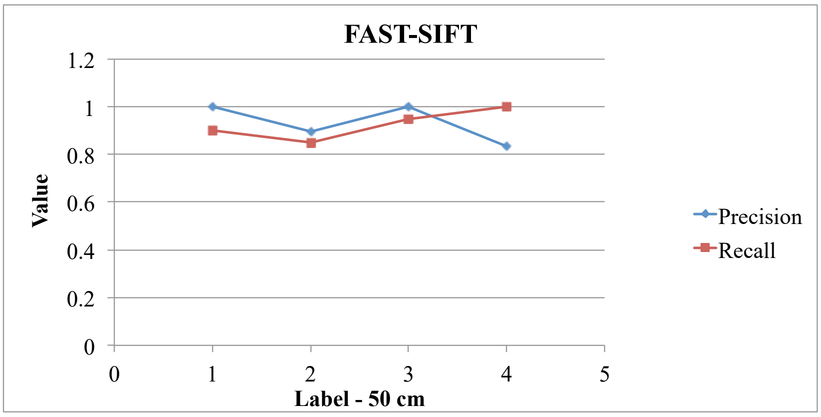


Figure 7.24: Measuring precision and recall metrics for the proposed natural plant recognition system based on the FAST detection approach using the SIFT description approach (distance 50 cm) [151]

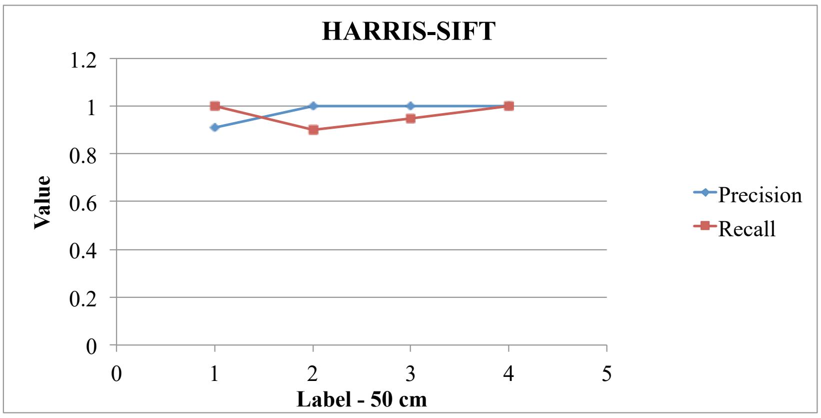


Figure 7.25: Measuring precision and recall metrics for the proposed natural plant recognition system based on the HARRIS detection approach using the SIFT description approach (distance 50 cm) [151]

stance of 75 cm. When the distance is 75 cm, the precision and recall results of the proposed system using the FAST-SIFT approach are comparable to the obtained results of the system with the other combined approach, the HARRIS-SIFT algorithm.

On the other hand, the accuracy of the classification is interestingly the same for these two mentioned recognition systems. If we consider the areas under the precision and recall curves of the system using the SIFT detection approach, we find that the areas are larger than other areas belonging to the other systems of this group. The higher precision and recall values mean a better performance of this recognition system. The obtained curve of the system using the FAST-SIFT proves that it is has a lower range of variation if it is compared to the changes of the system with the HARRIS-SIFT approach. However, the difference is insignificant, and it is equal to 0.006.

Our next attempt is to plot the precision and recall measurements of each of the proposed systems in just one figure. As a result, three different figures will be built. Figure 7.33 represents the precision and recall measurements of the proposed system based on the SIFT detection approach.

To evaluate each proposed system of the first group at the distance of 75 cm, the recall and precision measurements have been calculated for each system individually, though it is also possible to compare the results of different proposed systems. Amazingly, the obtained results of all proposed systems in this group, when the distance is 75 cm, are in the range of [0.8, 1.0], neither the precision nor the recall is less than 0.8.

For the second group of the proposed systems at the distance of 75 cm, we would like to observe

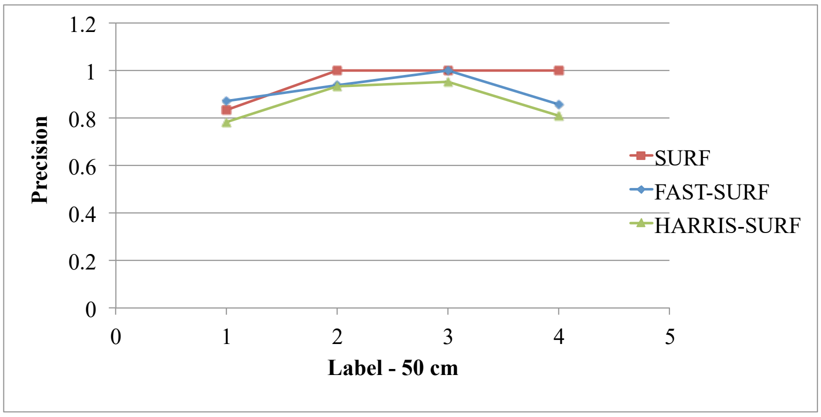


Figure 7.26: Precision measurements for three SURF description-based systems for the natural plant recognition (distance 50 cm) [268]

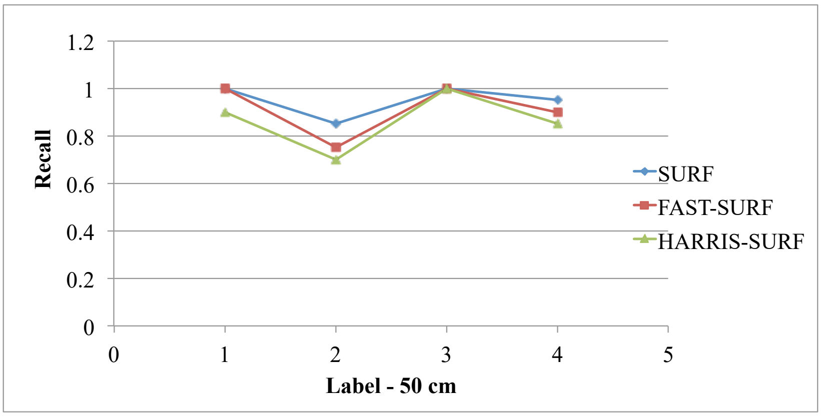


Figure 7.27: Recall measurements for three SURF description-based systems for the natural plant recognition (distance 50 cm) [268]

the differences between the precision and recall measurements of the systems in two figures, Figure 7.36 and Figure 7.37. The first figure shows the precision measurements and the second addresses the recall measurements.

In most labels, the performance of the system based on the SURF detector is better than the others when we investigate the precision results at 75 cm. This performance has been repeated in the recall results. If we choose one of the two systems based on the combined approaches, we are able to double-check the obtained results by considering three factors: accuracy, recall and precision. The same as the previous experiments, we represent the precision and recall results of each proposed system in separate figures. However, it is difficult to achieve high recall and high precision, and it sometimes turns out very difficult. Our observations, in Figure 7.38, Figure 7.39 and Figure 7.40, prove that the proposed systems have enabled us to recognize the plant species at a long distance, such as 75 cm.

In this recall experiment, the variation of the values is small when the used approach for detection is the SURF. The labels have large values if we compare them to the other proposed systems based on the SURF description approach.

The last experiment bears an important and completely unique part of our project. At this point, the distance is greater. A set of three different distances between the camera and the plant species are created: 100 cm, 150 cm and 200 cm. The first attempt is to build separate precision and recall figures, Figure 7.41 and Figure 7.42. The figures represent the results of these two metrics for the

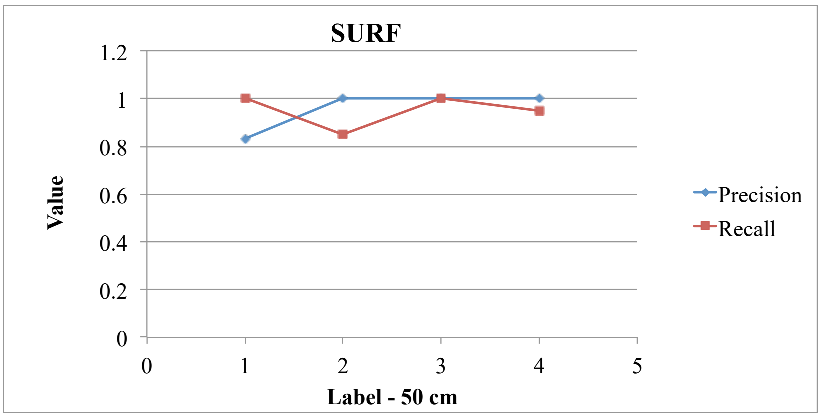


Figure 7.28: Measuring precision and recall metrics for the proposed natural plant recognition system based on the SURF detection approach using the SURF description approach (distance 50 cm) [268]

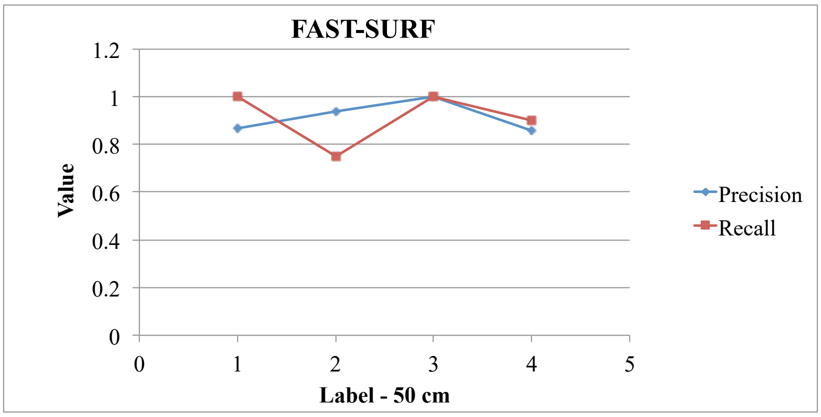


Figure 7.29: Measuring precision and recall metrics for the proposed natural plant recognition system based on the FAST detection approach using the SURF description approach (distance 50 cm) [268]

first group of the proposed recognition systems which are actually based on the SIFT description approach.

Without any doubt, we encounter good performances from the systems based on the SIFT detection approach. At this set of distances, the system based on the HARRIS detector has the three maximum values for the three labels: label 2, label 3 and label 4, if we investigate the recall results.

Let's start a new comparison due to recall values. If we compare this performance to the obtained recall results of the system based on the SIFT detection approach, we conclude that the system using the HARRIS detection approach overrides the better performance of the system based on the SIFT detection approach at this set of distances. The system using the SIFT detector is not solely our choice in all conditions and situations. Moreover, the investigation of the recall results asserts a good performance from the system using the HARRIS-SIFT approach in comparison to the other proposed system which has utilized the other combined approach. The area under the curve of the system using the FAST-SIFT is less than the others, and its performance is not comparable to the other systems if we consider the recall measurements.

To make our investigation more accurate and obtain more essential information from the experiment, we plot the recall and precision measurements of each system in one figure simultaneously, and it is a procedure that we have followed up on. Figure 7.43 shows that the system using the SIFT detector has low variations for both precision and recall metrics. The variations of the FAST-SIFT method are lower than the HARRIS-SIFT method. The results are shown in Figure 7.44 and Figure

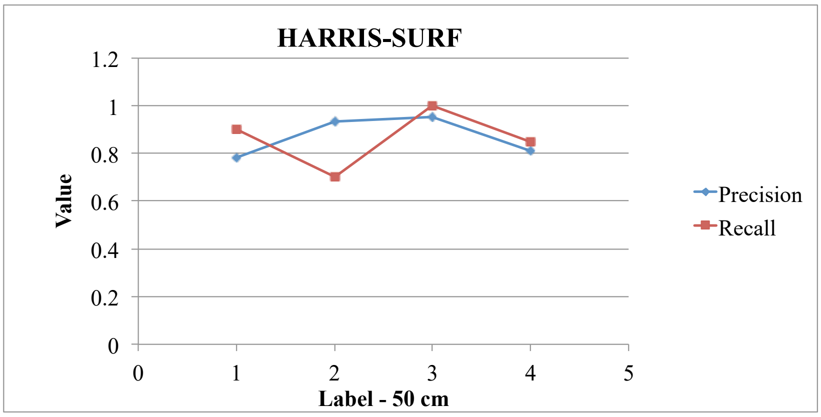


Figure 7.30: Measuring precision and recall metrics for the proposed natural plant recognition system based on the HARRIS detection approach using the SURF description approach (distance 50 cm) [268]

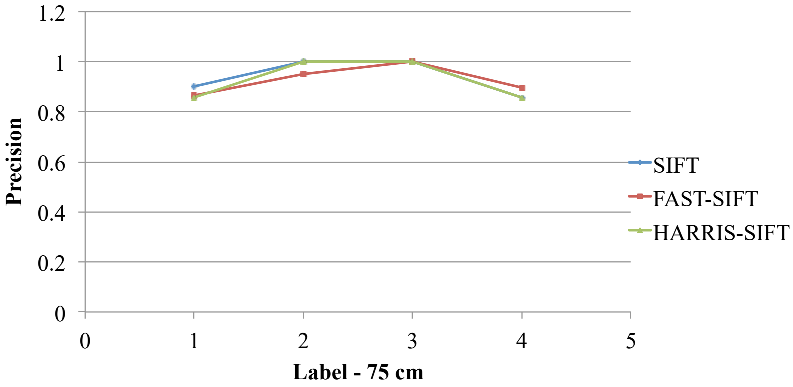


Figure 7.31: Precision measurements for three SIFT description-based systems for the natural plant recognition (distance 75 cm) [151]

7.45.

Figure 7.44 represents the performance of the system based on the FAST detector according to the precision and recall metrics. For the three labels, the recall values are greater or equal to the precision values.

Figure 7.45 shows that the recall curve has occupied more area than the precision curve when the basis of the proposed system is the HARRIS-SIFT approach.

The second group of the proposed recognition systems absorbs our attention in wanting to know their performance by examination of the precision and recall experiments and in considering the effectiveness of the proposed systems by the quantitative analysis. This investigation resembles the control of the parents during the early development of the children.

The distance has been increased from 75 cm to 100 cm, 150 cm and 200 cm. Obviously, the recall and precision results have been changed as Figure 7.46 and Figure 7.47 show related curves. The recall values of the system based on the SURF detection algorithm have been in the range of [0.91667, 1]. Its precision results have been in a smaller range, [0.92308, 1]. In addition to the high accuracy of this proposed system, the obtained ranges display the large values of the precision and recall measurements. This is a proof that the recognition system is able to return many correct labeled results during the classification task.

At this group of distances, the results of the system with the HARRIS-SURF approach are com-

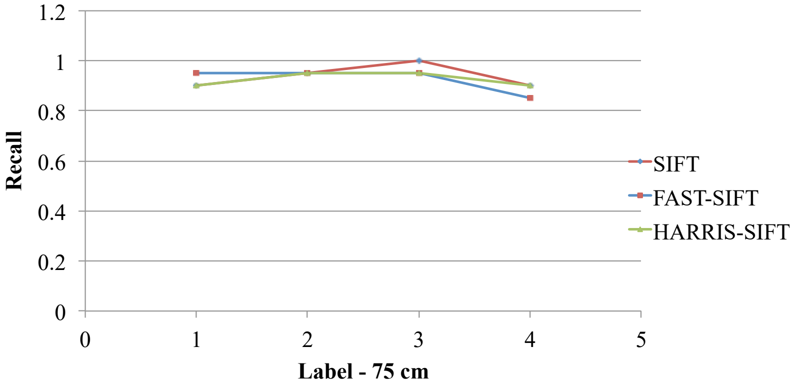


Figure 7.32: Recall measurements for three SIFT description-based systems for the natural plant recognition (distance 75 cm) [151]

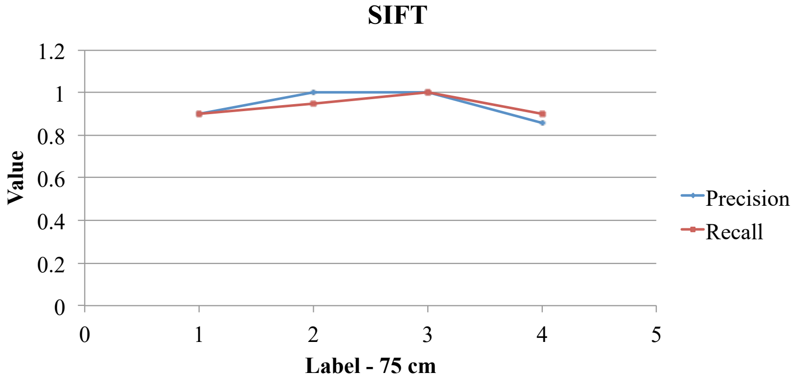


Figure 7.33: Measuring precision and recall metrics for the proposed natural plant recognition system based on the SIFT detection approach using the SIFT description approach (distance 75 cm) [151]

parable to the results of the system using the FAST-SURF approach; and the previous experiment, that of the classification accuracy. It also provides an evidence of this fact because of their equal accuracies at the mentioned distances. If we double-check the accuracy of the proposed systems using the SURF description approach, we find that the difference between the accuracy of the system based on the SURF detector, the HARRIS detector and the FAST detector is 2% which is roughly a small value. Therefore, these three different systems can be applied according to our goals and desired applications at the mentioned distances of this step. In Figure 7.48, Figure 7.49 and Figure 7.50, the precision and recall results of the three proposed systems are shown.

To complete our experiments, we consider the proposed systems according to the existing distances. In the lowest distance, 25 cm, the systems based on the SURF description method perform very well and have higher accuracy compared to the systems based on the SIFT description method. However, the performance of the system based on the SIFT is generally good. In longer distances, the system based on the SIFT outperforms the system based on the SURF. In general, the system based on the SIFT is more robust and it is one of its advantages.

To sum up the performed experiments and make a short conclusion of the precision and recall measurements for the first group of the proposed systems, the sequence of the best results and performances are the systems based on the SIFT detection approach, the system based on the HARRIS detection approach and the system based on the FAST detection approach. The areas under the curves are the evidence of this sequence [151].

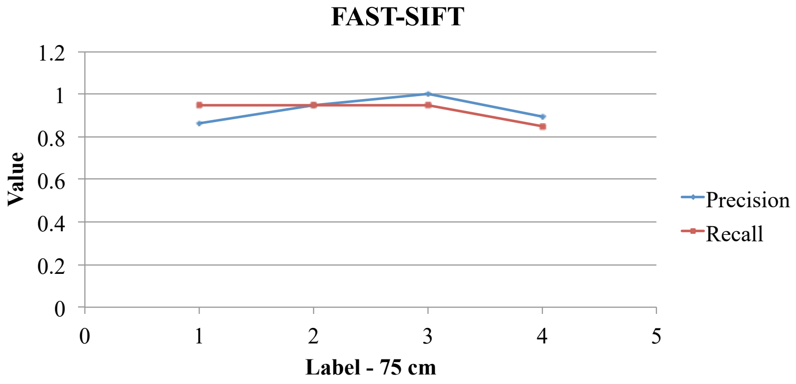


Figure 7.34: Measuring precision and recall metrics for proposed natural plant recognition system based on the FAST detection approach using the SIFT description approach (distance 75 cm) [151]

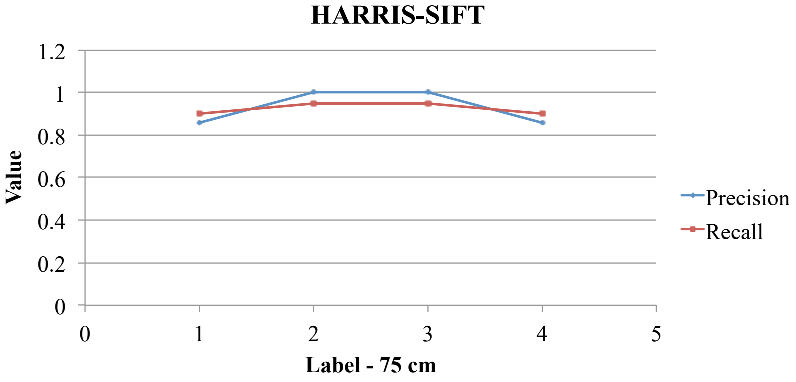


Figure 7.35: Measuring precision and recall metrics for proposed natural plant recognition system based on the HARRIS detection approach using the SIFT description approach (distance 75 cm) [151]

Number of Detected Keypoints

We are usually accustomed to performing some frequent experiments, but we are going to try out a new type of experiment. The goal is to follow up our curiosity in a new aspect instead of generating the regular tests. As a result, awesomeness and uniqueness will be added to our experiments and the previous experiments will be expanded differently.

In matching concepts and image processing, the keypoints correspond to image contents and similar parts of the images. One main contribution of keypoints is to be able to apply this universal tool for comparing different proposed approaches and indicate a new analysis. As we have used different approaches in our implemented systems, investigation of the number of detected keypoints has been shown in Table 7.3. The last scenario of our experiments is actually to detect the keypoints and to count the number of the keypoints at different distances if the detection part of the proposed system is changed. Figure 7.51 represents the used sample images for finding the keypoints and counting the number of the keypoints.

Since the test has been performed on the natural images at different distances, variation of the number of the detected keypoints depends on the captured scene of the plant species and the contents. In addition, it is worth mentioning that it is not easy to select the best detector and find a clear optimal way. For instance, the system based on the SURF detection method has higher accuracy at

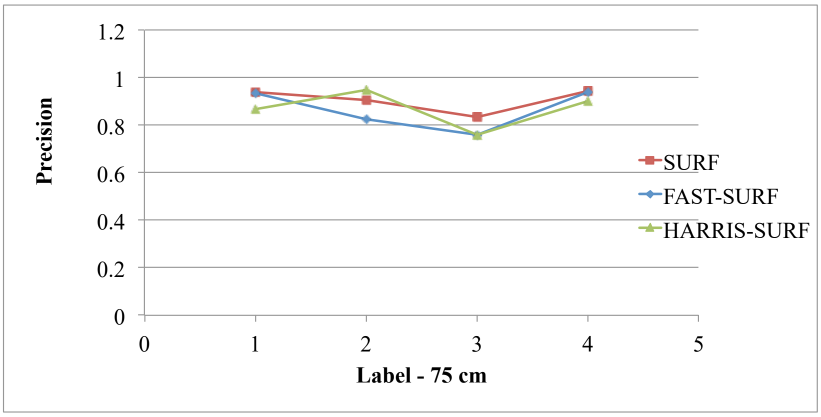


Figure 7.36: Precision measurements for three SURF description-based systems for the natural plant recognition (distance 75 cm) [268]

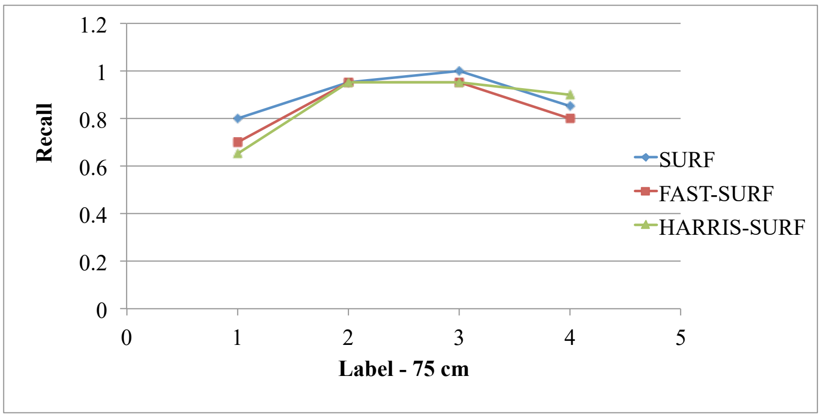


Figure 7.37: Recall measurements for three SURF description-based systems for the natural plant recognition (distance 75 cm) [268]

the distance 50 cm in comparison to the system based on the SIFT detection method. In this case, the number of detected keypoints is also larger when the SURF detection method is used. However, the performance of the system based on the SIFT detection method is acceptable. On the other hand, the system based on the SURF detection approach has still larger number of detected keypoints at the distance 100 cm, but the accuracy of this system is less than the system based on the SIFT detection approach which detects less number of keypoints at the mentioned distance.

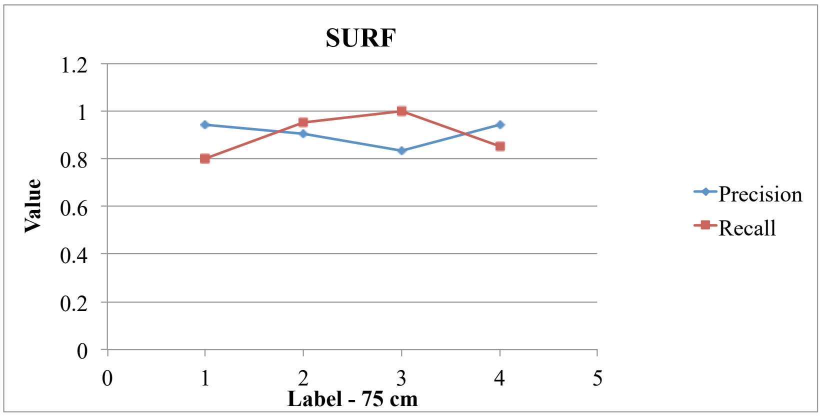


Figure 7.38: Measuring precision and recall metrics for the proposed natural plant recognition system based on the SURF detection approach using the SURF description approach (distance 75 cm) [268]

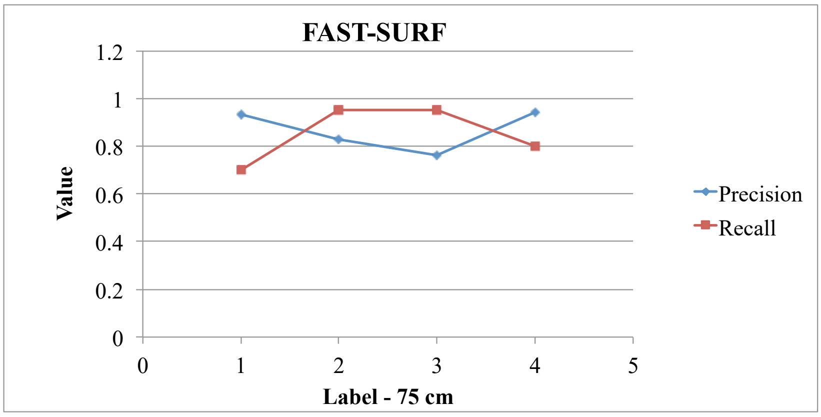


Figure 7.39: Measuring precision and recall metrics for the proposed natural plant recognition system based on the FAST detection approach using the SURF description approach (distance 75 cm) [268]

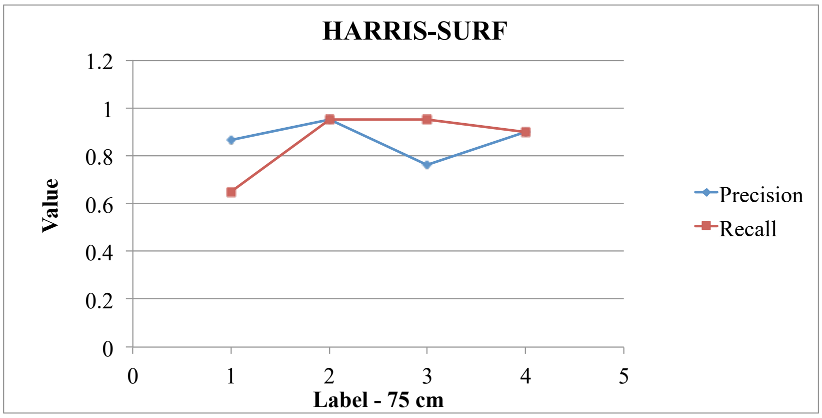


Figure 7.40: Measuring precision and recall metrics for the proposed natural plant recognition system based on the HARRIS detection approach using the SURF description approach (distance 75 cm) [268]

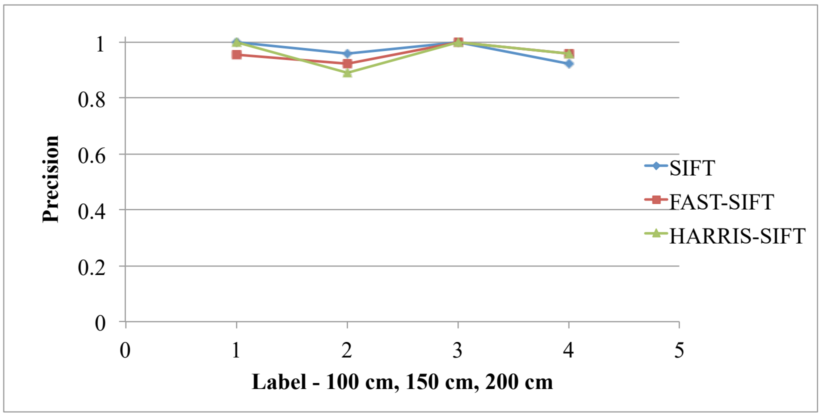


Figure 7.41: Precision measurements for three SIFT description-based systems for the natural plant recognition (distances 100 cm, 150 cm and 200 cm) [151]

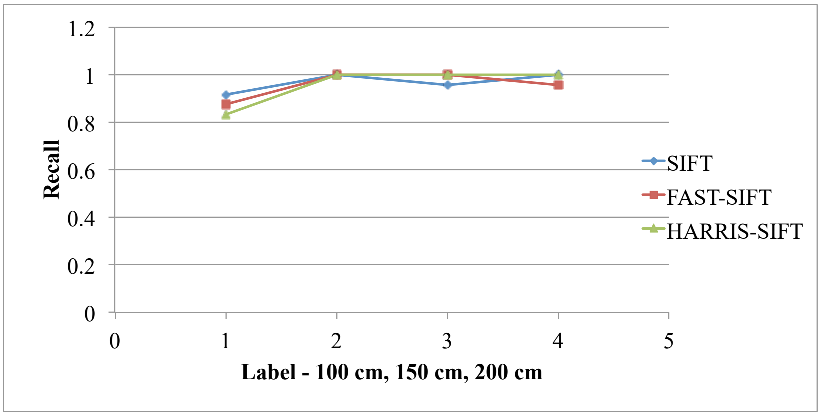


Figure 7.42: Recall measurements for three SIFT description-based systems for the natural plant recognition (distances 100 cm, 150 cm and 200 cm) [151]

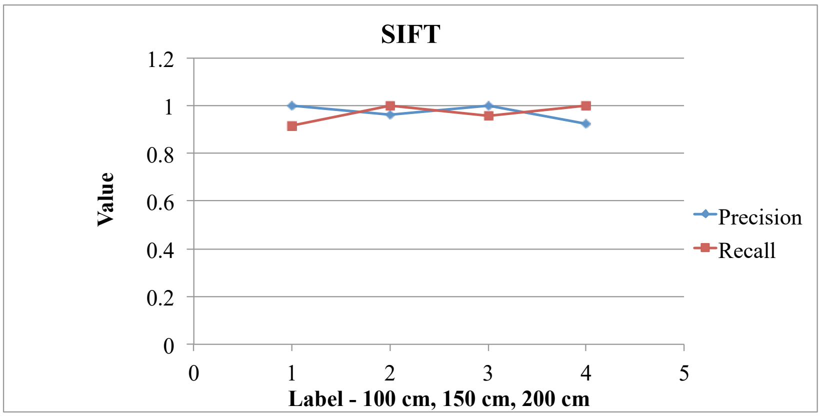


Figure 7.43: Measuring precision and recall metrics for proposed natural plant recognition system based on the SIFT detection approach using the SIFT description approach (distances 100 cm, 150 cm and 200 cm) [151]

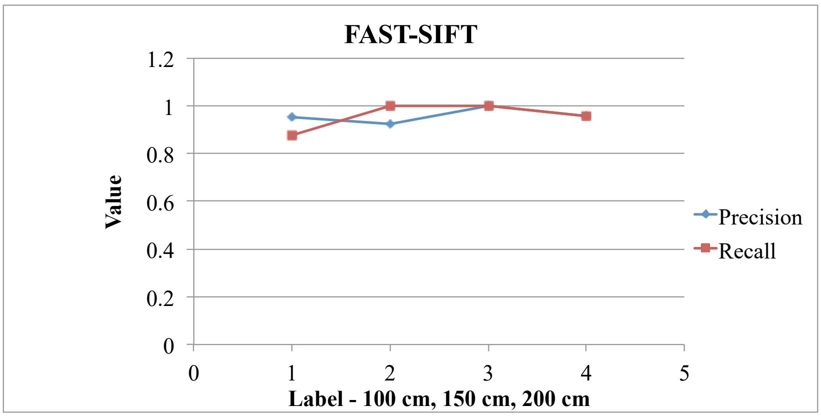


Figure 7.44: Measuring precision and recall metrics for proposed natural plant recognition system based on the FAST detection approach using the SIFT description approach (distances 100 cm, 150 cm and 200 cm) [151]

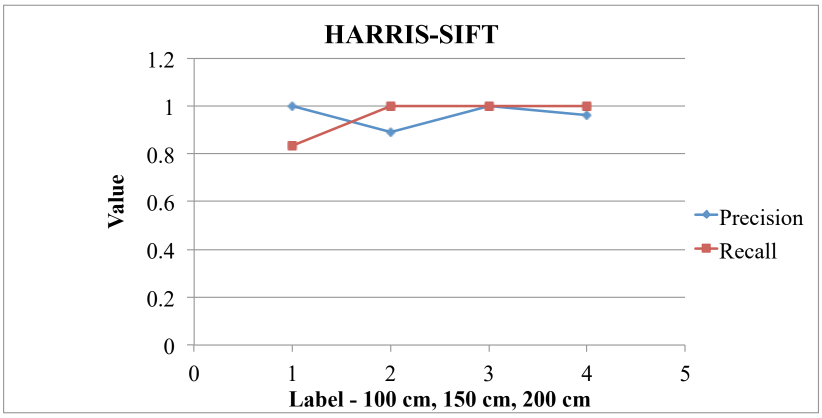


Figure 7.45: Measuring precision and recall metrics for proposed natural plant recognition system based on the HARRIS detection approach using the SIFT description approach (distances 100 cm, 150 cm and 200 cm) [151]

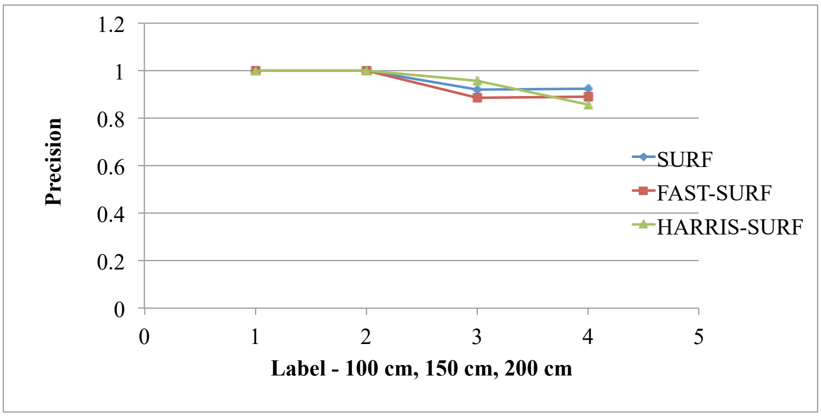


Figure 7.46: Precision measurements for three SURF description-based systems for the natural plant recognition (distances 100 cm, 150 cm and 200 cm) [268]

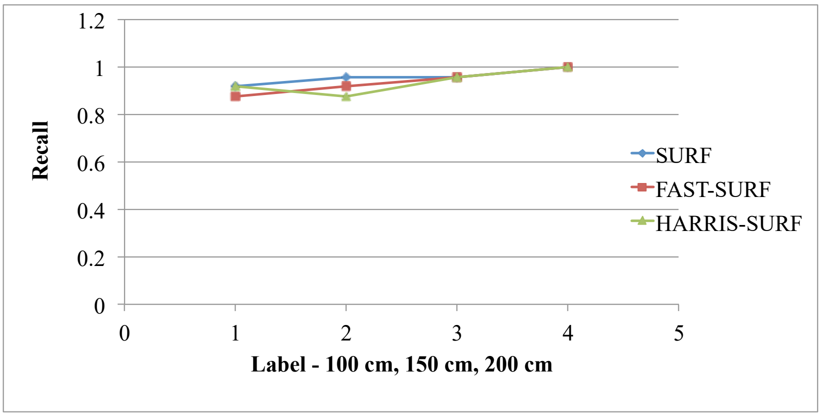


Figure 7.47: Recall measurements for three SURF description-based systems for the natural plant recognition (distances 100 cm, 150 cm and 200 cm) [268]

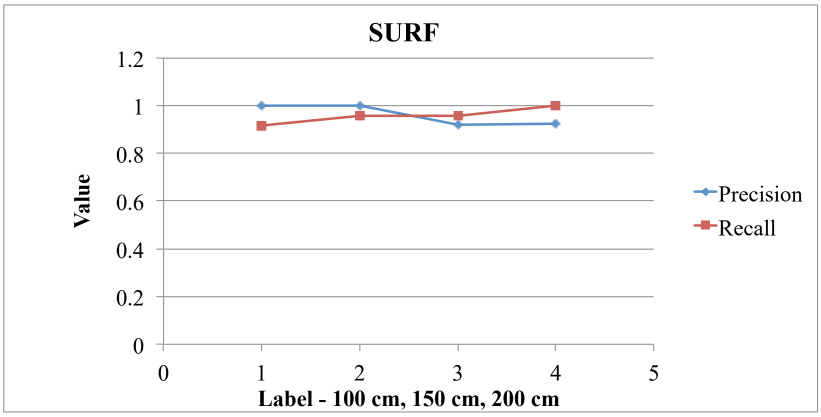


Figure 7.48: Measuring precision and recall metrics for the proposed natural plant recognition system based on the SURF detection approach using the SURF description approach (distances 100 cm, 150 cm and 200 cm) [268]

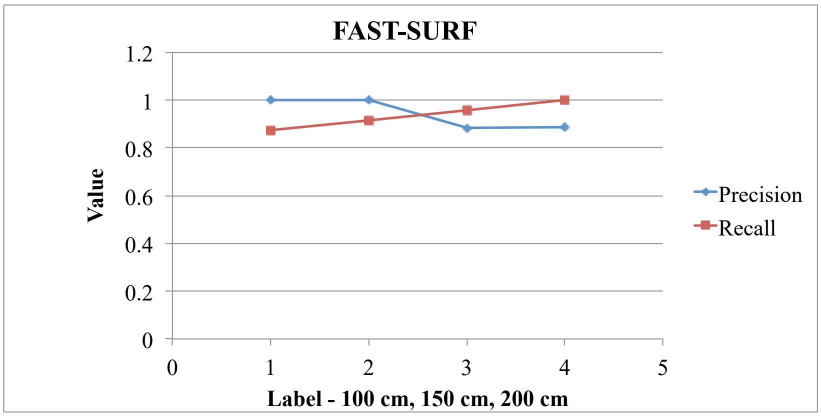


Figure 7.49: Measuring precision and recall metrics for the proposed natural plant recognition system based on the FAST detection approach using the SURF description approach (distances 100 cm, 150 cm and 200 cm) [268]

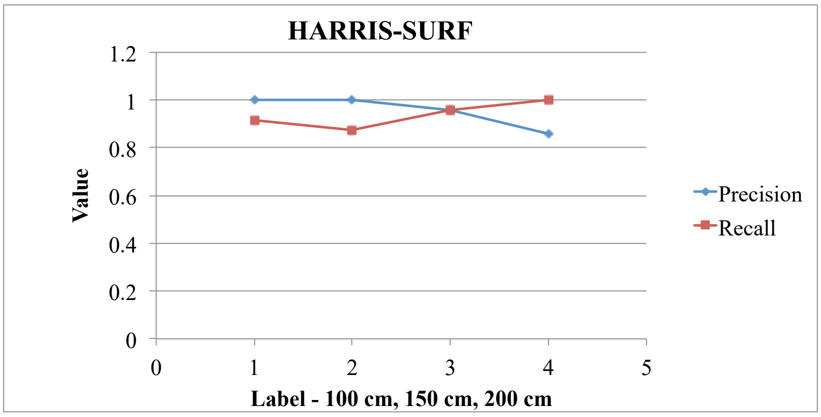


Figure 7.50: Measuring precision and recall metrics for the proposed natural plant recognition system based on the HARRIS detection approach using the SURF description approach (distances 100 cm, 150 cm and 200 cm) [268]

Number of keypoints according to the used approaches and distance	System based on the SIFT detector and SIFT descriptor	System based on the HARRIS detector and SIFT descriptor	System based on the SURF detector and SURF descriptor
50 cm	2385	314	9035
75 cm	1438	435	4894
100 cm	2251	1000	7099

Table 7.3: Number of the detected keypoints for some proposed systems [268] [151]

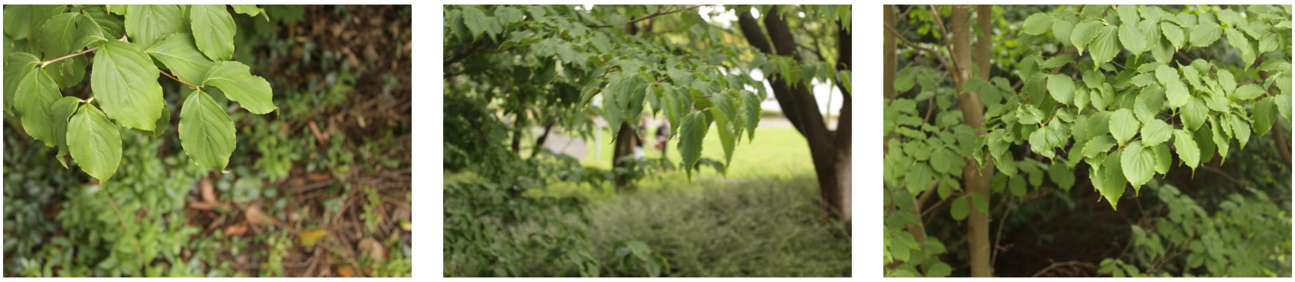


Figure 7.51: Sample image at the distance 50 cm (Left), sample image at the distance 75 cm (Middle), sample image at the distance 100 cm (Right)

7.6 A Short Talk on the Experiments, Results and Performances of the Natural Recognition Systems

Based on the observations during the design, implementation and test phases of the proposed systems, we would like to provide a view into the inner hidden layers of the proposed systems. To continue our study and open up the new aspects, we largely follow the performances of two highlighted systems and work on the reasons for our independent investigations. Exploring all proposed systems emboldens two implemented systems, the systems based on the SIFT and SURF detectors.

Due to the obtained results, we are going to especially compare two superior systems and find out the reason for the results. By investigating the system based on the SURF algorithm, this system is not completely affine invariant which is proven in [336]. Despite this shortcoming, using the SURF algorithm has not been limited. Its potential has been utilized widely in different systems. As an example of the shortcomings, the SURF algorithm does not work if we have a severe rotation or the view angle differs greatly. In addition, the SURF algorithm is sensitive to rotation and illumination changes that happen mostly in images captured from the natural scenes and objects in the outdoor environments without the human control of the mentioned factors.

On the other side, the description part of the SIFT algorithm is more appropriate and useful for describing the images affected by different translation, scale, rotation, and other deformations like changes of the illumination. The performances of these two algorithms are not the same in the face of noise and deformations. Here, noise means any variation and change which might affect the captured plant scene in the recognition task. We have a natural dataset of the plant species which includes various image capturing conditions containing the deformation of the leaves, the change of the viewpoint and the view angle, the distance variation, the change of the weather conditions, and the other changes such as background, illumination, etc. The images of the dataset are complex and highly different by consideration of these various aspects. The robustness of the SIFT algorithm is better than the SURF algorithm. The algorithm contributes to the developed system for a better resistance to undesirable factors and complicated changes with harmful effects in the natural plant recognition. Comparing the classification accuracy of the implemented system with the SIFT approach and the SURF approach shows a good robustness of the proposed system using the SIFT algorithm in a challenging natural environment. Due to the obtained results, the truth is that the system based on the SIFT detector has good robustness against the unexpected natural changes and various environmental variations.

The SIFT algorithm is computationally expensive and it is considered slow in its application in many systems. However, the acceleration of this system has been performed by means of the two other detection algorithms which are the HARRIS and FAST detection algorithms. The contribution to the systems based on the HARRIS and FAST detectors involves a trade-off between the accuracy of the classification and the speed of the implemented systems. The HARRIS and FAST detection algorithms play the role of an auxiliary algorithm to replace the detection part of the SIFT algorithm.

In addition to the SIFT description approach, the SURF description approach has also been applied in the proposed systems and the combined approaches have been created. While the basis of the description part is the SURF descriptor, HARRIS and FAST detectors are also applied as detection components of the systems. Using each proposed system depends on our expectations, limitations and the importance of some factors, like accuracy and runtime. Furthermore, it is possible to select out a system according to our real needs. For instance, we are able to use a system with shorter runtime and sacrifice the accuracy of the classification if our demand is a fast natural plant recognition system.

In a nutshell, the performance of each system is evaluated and compared to other implemented systems using different aspects. It is necessary to emphasize other remaining points. The dataset contains different imaging conditions including complex objects, challenging scenes, changes of rotation and scale, light intensity and illumination changes, various times of image capturing and changes of weather. The experimental assessments have been organized in different phases. The diversity of experiments helps us to evaluate the systems in different ways. Meanwhile, the images are captured at different distances which are also roughly large. Its contribution is to achieve recognition systems which are usable in farms and agriculture as people mostly do not care about the distance of the system for identifying the plant species. In addition, dividing the entire dataset into training and testing datasets has been performed automatically and randomly. Before finishing the discussion in this section, it is worthy of note that a good work in the plant recognition was proposed in [289]. Comparing the systems in this chapter to the proposed work in [289] proves that our systems are more robust and they work better. They are able to identify the plant species in the natural environment with many challenges. This represents significant progress in the area of the plant recognition.

7.7 Systems Potential for Future Use

We proposed six natural plant recognition systems based on the modern combination of the detection and description approaches in [151] and [268]. The implemented systems are markedly different to the common systems used in different fields. The current systems usually lack an important factor, the generalization of the system. They mostly cannot be used in different situations and various environments such as windy and cloudy weather, which are challenging situations and conditions. In addition to the generality of the systems, we have the taste of efficiency, reliability and high accuracy, as the experimental tests and quantitative results have proven in detail.

In all implemented systems, the construction of the visual vocabulary is done by clustering the data and representing the resulting data. Feature vectors play an important role in the whole process, and the used bag of words step. It is possible to use a potential model, GMM, and examine the influence of this model. The important point is to find how the GMM handles 128 dimensional feature vectors of the SIFT algorithm is used. Unfortunately, this model is computationally intensive and expensive. Although it can be a potential model, we did not use it due to the reason mentioned.

As explained, the implemented systems deal with robots and the related technologies. Drones can use the proposed systems on farms and recognize the plant species at different distances between the drones and the plants. The systems are also useful for the robots deployed in the agricultural fields. The story of recognizing plant species consists of two different conditions while we are using robots and the direction of the sight is a problem in many systems. The first condition is to set up a camera and quadcopter and look at the plants from above, therefore, the sight from the camera is vertical to the plants in the field as shown in Figure 7.52.

The second condition happens if we use a robot which is able to drive through the paths inside a farm. In this condition, the lens of the camera might be parallel to the plants inside the field. Hence, the robot takes pictures from the front scene of the plants, instead of taking the pictures from above. Figure 7.53 shows this special case and the direction of the sight parallel to the plants.

The proposed systems enable the experts and non-experts to recognize the plants in any direction of photographing and at different angles of view in the natural environment. It is one of the novel potentials of the proposed systems to be able to be used for any type of robot or drone. If we suppose having a robot with one of the proposed systems, the application is not limited to the classification of the plant species. The robot can also be used for crop monitoring in huge fields, therefore, the farmers always get close to crops and collect the data in a real-time system. In the traditional farming, the farmers consumed a lot of water to irrigate the crops. In the today's world, the primary focus is to reduce the amount of the water consumed and save more water for the future. Hence, the traditional farming methods are not useful and applicable. An efficient method is to target specific plant species

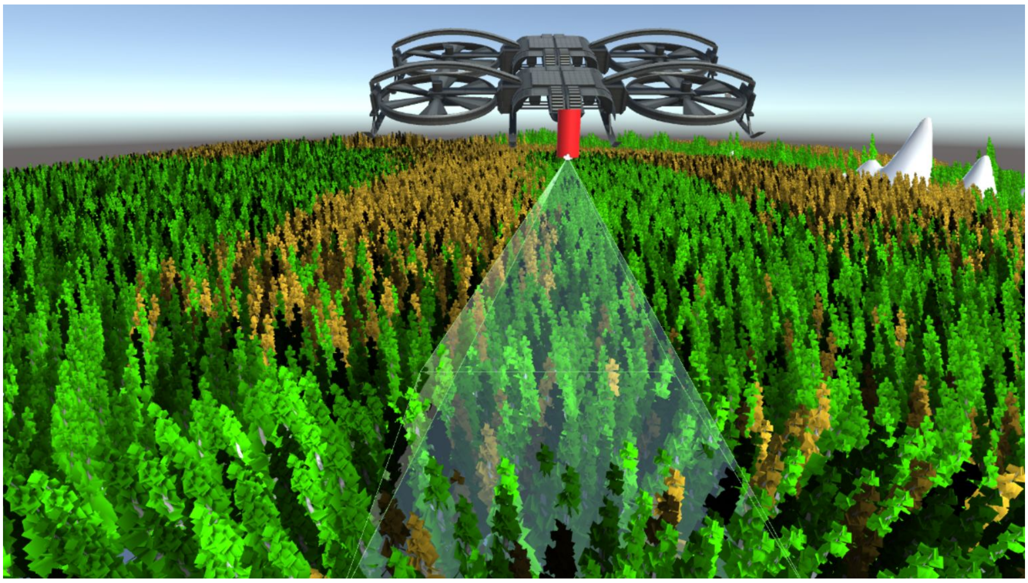


Figure 7.52: Vertical sight into plants from a camera mounted on a quadcopter

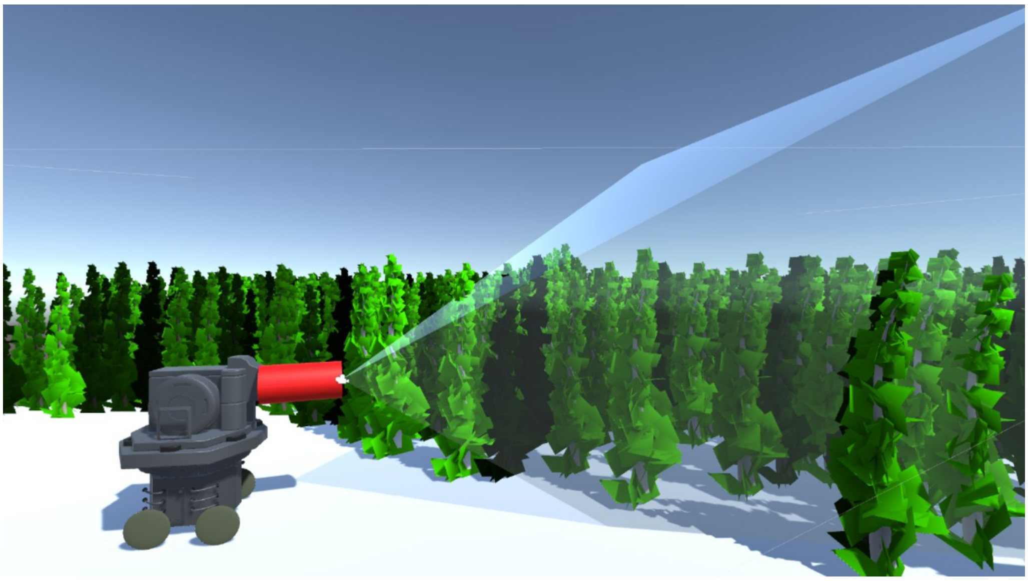


Figure 7.53: Taking images of the front side of plants

and irrigate them in the correct time. Furthermore, such a robot is also useful for removing the undesired weeds and increasing the efficiency of the growth of the desired plants. In this case, the robot would be able to decide on keeping or removing the plants. Farmers are also able to control the field easily and track the changes in the field to improve the final productivity. In addition, a remote control for different tasks on the farm can be provided. In addition to the mentioned applications, the experimental results also show the potential of the implemented systems in other fields, such as botany and the pharmaceutical industry, because we are able to use the systems as applications on a PC. Consequently, the botanists and other experts are able to profit from the natural plant recognition and gather the data for further scientific purposes. Ultimately, the systems also save time instead of using the traditional methods for the plant identification.

7.8 Acknowledgment

The natural image dataset for the plants has been created and prepared by Masoud Fathi Kazerouni. All copyrights are reserved and the dataset belongs to the Institute of Real-time Learning Systems, University of Siegen, Germany.

7.9 Conclusions and the Future Scope

At this stage of the work, a few natural plant recognition systems, 6 systems, have been introduced which propagate the classification of the plants in the natural environment through the entire pipeline of the proposed approaches. Each system has been trained on the output of the SIFT or SURF descriptor by applying the SVM algorithm. The proposed system, which utilized the SIFT detection and description method, has been able to obtain the highest recognition accuracy, 94.94%. Very few algorithms like the SIFT and SURF algorithms can be applied for both feature detection and feature description and effectively perform in different situations, such as rotation, scaling and changes of blurring and illumination. Our work proves that both algorithms also have a good performance in the natural situations such as various weather conditions, complex backgrounds, the time of photographing, large viewpoint changes, change of light intensity, etc. The combination of these description algorithms with the other detection algorithms, the HARRIS and FAST algorithms, leads to new types of the detection and description algorithms. It contributes to the implementation of the real-time recognition systems as well as efficiently overcoming the challenges in the real situations. Due to the results, the final recognition systems are effective enough to recognize the natural plants in such challenging situations, particularly at different distances and times of the day.

The proposed systems in both [268] and [151] are the most realistic references of the current state-of-the-art in the natural plant recognition systems. In other words, the proposed and implemented systems extract features and automatically classify images of the natural plants with a high accuracy and impressive results by the use of the machine learning methods. In past years, it was a dream to invent systems which are able to recognize plant species in the natural environment. We have considered natural factors and unique challenges. We accordingly developed new plant recognition systems. Consequently, this dream is a reality today. Some contributions of the implemented systems are that they are fully-automatic, they exploit the combined modern detection and description algorithms, they have utilized the natural images throughout the research and they can function in non-fixed environmental and non-environmental parameters such as light intensity, illumination, weather condition and distance.

In the experiment section, it has been feasible to use the GPU instead of the CPU based implementation and to speed up the systems. There is also an important point about the implemented systems related to creating different vocabularies for the distances. Since we have four different sets of distances, one vocabulary is constructed for each distance. It is impossible to have only one vocabulary for one system. Hence, the future work should be able to solve this problem and compensate this shortcoming correctly and accurately. An alternative is to find a solution for calculating the distance between the camera and the plant from a new test image. To measure this distance, there is an interesting point where the ratio between the focal length and the distance between the plant and the camera is equal to the ratio of the size of the leaf in the image and the height of the leaf in the real life. Therefore, it would be possible to add a pre-processing part to the system and calculate the distance between the plant and the camera. However, such a pre-processing step adds computational cost to the system. In addition, it is sometimes hard to get the size of the leaf within the natural image as there might be no complete single leaf in the image or a part of the leaf might be covered by the other branches or leaves. Another feasible solution is to ask the user to measure the distance between the plant and the camera during photographing. This distance can then be used as a pre-knowledge for the system. Furthermore, another option is to use an ultrasonic sensor for measuring the distance. Consequently, the future work brings a significant improvement for the natural plant recognition. The deep learning algorithm will be applied to design a new system in the next chapter. It will be explained why deep learning is used and how the model is implemented. More questions will also be answered in detail.

Chapter 8

Novel System: Deep Learning System for Recognition of Natural Plant Species

The use of natural plant species images has been investigated successfully in Chapter 7. Using various systems contributes to having access to different systems with their own properties which might be applied in desired situations. In addition, it helps to obtain systems with desired characteristics such as generality, reliability, stability, etc. in spite of changes in the natural environment. Although the obtained accuracies of the proposed systems are high, there is still one limit with regard to producing one vocabulary per distance for each implemented system. It might degrade the usability of the systems in real-time applications. Since we do not like to be involved with any limitation and failure, it is essential to find an effective and practical solution. Another goal is to increase the accuracy of the recognition system and predict more samples correctly in a challenging test. While most of recent approaches and systems are purely based on SVM algorithms, we are going to change the direction of our study and go to a new world to achieve our goals in a new foundation and framework.

This chapter studies a famous category of classification approaches to considerably improve the proposed systems in Chapter 7. The idea is to expand our knowledge of the neural network field and then use the neural network concepts for implementing a new plant recognition system. This study enables us to distinguish the plant species with higher accuracy than the prior systems and get firm experience for one of the final goals which is the real-time and mobile use of systems for plant recognition in outdoor environments. The presence of a longer distance between camera and plant, sunshine, wind or small drops of rain has an effect on the structure of the captured images from the natural plants. Hence, the performance will be significantly changed.

A system based on neural networks will be designed and implemented in the current chapter. in order to use natural images, the dataset explained in 3.3.1 is applied. Using a deep learning algorithm yields a very high accuracy of over 99%, where the basis of the system is a convolutional neural network, CNN. High accuracy is not the only sufficient factor, trade-off between energy consumption and building the final system is also important. Thus, the energy consumption will be explored and explained in one section separately. In this chapter, we carry out the experiments extensively. Furthermore, we provide a complementary demonstration for the analysis of the implemented system using the CNN and represent significant improvement in natural plant recognition performance in outdoor environments.

This work has been published in SN Applied Sciences, Springer journal [91].

8.1 Introduction

The curious nature of humans often places a question mark before anything that happens in the real world, and even in the virtual world. The power of perception, realization and thinking always accompanies human beings and makes distinctions in many issues and facts. It is rare to see a big tsunami in a research area, and we do not usually face such an experience. But such scenarios have taken place in the area of machine learning since a few years ago, and deep learning became the topic of the day. Interestingly, the fever of this topic has not subsided, and the number of fascinated researchers increases daily. Surprisingly deep learning is not limited to the machine learning field. It has affected other fields such as robotics, the car industry, data science and computer vision. It has had a progressive revolution that shows no signs of slowing down. In addition, many questions have been raised and many more will be forwarded in the future. However, the important point is the refulgence of one lamp among many machine learning lamps. Deep learning represents a unique beacon that is emitting its light regardless of any positive or negative comments.

Let's flashback to the past decades and refer to the previous machine learning algorithms which have been utilized for many years. It is almost fair to refer to machine learning algorithms, aside from deep learning algorithms, as traditional algorithms. For several decades algorithms such as SVM, Naive Bayes, ANN, etc., have been applied in various systems. People have paid attention to the mentioned traditional learning algorithms. Since the new generation of machine learning algorithms is still fresh, prodigious and new, it has become a battle, so to speak, and many researchers and experts are trying to be victorious.

During the last and current century, the desire has been to design systems that mirror human functions, such as the human speech system, auditory system, vision system, brain, etc. The belief is in the superiority of human systems in comparison to other living organisms in the world. The main purpose is to invent new systems similar to human capabilities in different aspects, such as efficiency, precision, repeatability, reproducibility, etc.

Deep learning is still in its infancy [337]. Although it has only recently taken it first steps, its achievements in different fields are still commendable. Many active areas and projects such as speech recognition, object detection and recognition, data mining, NLP, customer relationship management (CRM) [338], etc., are connected to deep learning algorithms. As a result, they are benefiting amazingly from this novel generation of neural networks. In addition, lots of forgotten projects and unsolved problems have been redefined. New goals have been assigned after the development of deep learning algorithms, although the goals seemed to be likely unachievable. Due to the mentioned points, we introduce the human nervous system (see 11.2) to become familiar with the main concepts and similarities between the human nervous system and the deep type of neural networks.

If we return to the middle of 1980s, we find a novel research carried out by LeCun et al. [339]. It is supposed to be the first version of a learning algorithm in its category. After more than a decade, a particular work [16] has been proposed. The contribution of this work is undeniable in the world of the CNNs, widely used in the object recognition and classification problems. The spirit of the human system is blown up and the architecture of CNNs has been built. Although the first exploration of neural networks began several decades ago, its current generation, deep learning, has attracted many eyes, and the scope of machine learning has been widened. If we venture to name previous neural networks, we can call them shallow learning in contrast to deep learning, whereby a shallow net is composed of input, output and, at the most, one or two hidden layers in between them. The fresh chapter of the neural networks was proposed in 2006 [340] and the main purpose was to have a high-level abstraction model of data. To achieve this purpose, a set of algorithms and complex structures with multiple non-linear transformations was performed.

Various theoretical and practical works have been performed in the field of deep learning and different types of algorithms and architectures have been proposed. Different deep learning architectures, for example, deep neural networks [341], convolutional deep neural networks [342], deep belief networks [343], recurrent neural networks (RNNs) [344], and auto-encoders [345] have been applied to the fields like computer vision, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks [91]. The selection of appropriate neural networks is a vital part of our work, thus a deep study has been performed. By considering different aspects of the work and the desired goals, we decided to use CNN. In [346] and [347], the CNNs have proven their huge power and capacity for the image recognition.

The diversity of deep neural networks is not limited to the architecture of different networks; there are different frameworks that might be used by researchers. There is also a big competition among current deep learning frameworks. On the other hand, the pressure of new desires and features is undeniable.

The presence of high CPU power, the new generation of the GPUs and access to higher amounts of data are some reasons for the creation of more efficient neural networks with higher numbers of layers and complex architectures. Plant recognition is a daunting challenge, especially in natural and outdoor environments. Therefore, we decided to design, develop, implement, and test a natural plant recognition system based on deep learning concepts. Furthermore, the intention is to use the potential of a deep learning algorithm to conquer the remaining land of accurate recognition and identification of plant species. Up until now, we have always grouped the samples of the dataset into training and testing samples. This procedure will be continued in this stage of the work to utilize a decision-making mechanism and predict the class of testing samples as well.

Some ideas behind the convolutional deep neural networks are local receptive fields (8.3.4), shared weights (8.3.4) and pooling (8.3.6). If we use the weight sharing of the convolutional layer with the scheme of pooling, the result is an enrichment of the properties of invariance. In addition, it is inadequate to have limited invariance and equi-variance if we attempt to solve complex pattern recognition tasks. As a result, it is necessary to utilize a wider range of invariance and systematic ways. Hence, the CNN with appropriate changes is efficient to be used in the computer vision and image recognition tasks for getting superior results.

Study and examination of different deep architectures highlight the reason for the attractiveness and popularity of CNNs. In comparison to other common neural networks, the training phase of CNNs is simpler. Hence, its industrialization is easier, and the chance of developing a user-friendly product is increased. Additionally, the contemporary demand for new technology and systems has rapidly increased, and new technology has daily penetrated different scientific and industry fields and human life. Similar to other fields, robot technology has entered the agricultural field. Farming robots play a fundamental role in different applications to achieve great improvements in farming tasks. As discussed previously, we would like to open a new window and use the potential of deep learning methods in the classification of plant species because of the shortcomings of those previous developed systems. This work is actually a new generation of plant recognition systems in outdoor environments with very challenging parameters and factors. As explained before, the dataset consists of four plant species, Hydrangea, Amelanchier Canadensis, Acer Pseudoplatanus and Cornus, belonging to the Siegerland region in Germany. It should be pointed out that the dataset is very complex, and our work cannot be directly compared with other studies using artificial datasets. In this work, the proposed deep model is the next step of previous works in [268] and [151]. One reason is to achieve higher accuracy and compensate for the remaining disadvantages. The final accuracy of the implemented system with deep neural network is 99.5%. In general, a fully-automatic plant recognition system is a helpful tool for plant specialists and botanists.

In this chapter, the proposed system uses a convolutional deep neural network architecture to do the plant recognition and classification task with high accuracy. The final results will be investigated and then evaluated. Furthermore, the contribution of the system will be reported and the final results of the experiments will also be compared to the results of previous proposed systems in [268] and [151] which have been implemented through the other pipelines.

The rest of this chapter is organized as follows: presentation of deep learning and neural networks' fundamentals are provided in section 8.2; describing the proposed approach and providing a report about the deep model and how it is working for recognizing the natural plant species is addressed in section 8.3; materials and equipment are represented in section 8.4; explanation of the experiments and results with additional details of the whole deep system and related evaluations is provided in section 8.5; and section 8.6, concludes the work of this chapter.

8.2 Deep Learning and Neural Networks' Fundamentals

Nowadays, we often hear a new term, deep learning, which conjures up mental images of another term, machine learning. Many may consider these terms interchangeable, but deep learning is a new frontier, even for experts in the field of artificial intelligence. It is undoubtedly a giant step forward for the study of neural networks and the related areas. One primary idea of artificial intelligence is to have the human capabilities relevant to the human brain system such as thinking, making a decision, etc. Over the last century, many people have been involved in this field and the wheels of progress are ceaselessly moving. However, this movement is sometimes fast and other times slow. Due to the mentioned idea, one important point is to create machine learning systems which are smarter than before.

Let's look at these systems from another point of view. Imagine, for instance, a child who is four years old. While the child is looking at a scene, it can distinguish if there is a ball. An educated person who is 31 years old is also looking at the scene. She realizes more information and details than identifying only a ball. In addition, if she makes a mistake with respect to the details of the scene, she is able to rectify and correct the mistake. The child might not be able to give more details than the presence of a ball. Learning from mistakes helps us to make systems smarter and more efficient in difficult situations. Before starting our discussion on traditional machine learning, neural networks and deep learning, we would like to explore the timeline of deep learning and then continue our explanation from there.

8.2.1 Deep Learning Timeline

Discussion concerning deep learning is increasing daily, and a lot of experts and non-experts are trying to understand this fascinating topic. Once a researcher starts using these advanced neural network models, he might think that it is not only research, it is also a hype in his scientific journey. Although many academic people still look at deep learning as an artistic masterpiece in the Louvre Museum. But, they do not want to face the challenges of this concept in their own research. In addition, they mostly admire the outcomes of deep learning without applying it to their own projects. Today, deep learning is usually considered as a solution in many new proposals, but this unique proposed solution is often not applied in the end. In reality, we have two groups of people who represent two sides of deep learning. The first side is comprised of the people who have fallen in love with deep learning, while the other side is made up of those who saying it is worthless to explore. They usually consider it only a bridge between the traditional learning algorithms and the next modern ones. This second group thinks that the deep learning is similar to the mayfly, a metaphorical comparison to describe it as having short life-span.

A common question raised is about the general role of deep learning algorithms in the future of the Artificial Intelligence. At the moment, deep learning tackles many problems which seemed to be impossible and hard to solve for many years. This area goes beyond a simple neural network. It has contributed to both theoretical and practical advancements. Meanwhile, it is a result of new hardware and our access to powerful equipment. Deep learning has been similar to polar bears in winter days if we investigate its timeline deeply as a showcase of its creation and existence. Despite its new and fresh appearance, deep learning was proposed for the first time a few decades ago. Current deep learning algorithms are the outcome of many years of efforts.

The uniqueness of deep learning is actually based on its contribution to new learning representations of raw data. Deep learning algorithms emphasize a high number of layers to represent the data meaningfully. The idea behind such algorithms are actually artificial neural networks.

In this section, we would like to have a realistic look at the history of the deep learning and its career as it seems to be a perplexing subject. Its major developments are an important part of gaining a better understanding. Figure 8.1 represents the deep learning timeline made by Favio Vázquez [348] which indicates a rich history behind deep learning as a modern neural network.

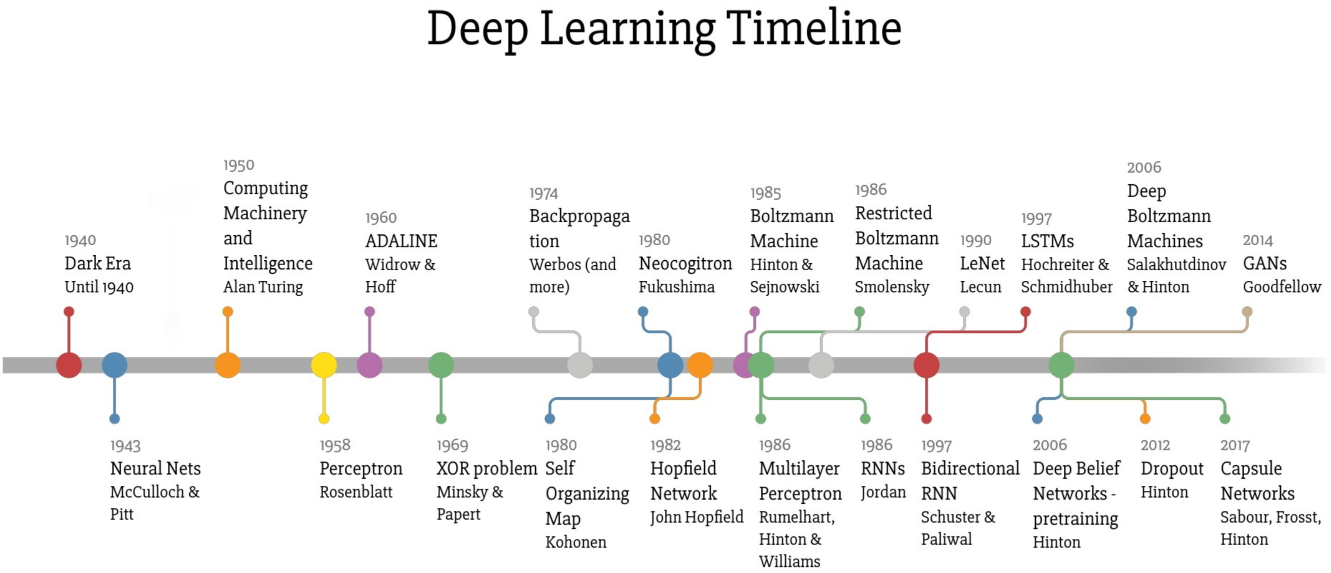

Deep Learning Timeline
1940 Dark Era Until 1940
1950 Computing Machinery and Intelligence Alan Turing
1960 ADALINE Widrow & Hoff
1974 Backpropagation Werbos (and more)
1980 Neocogitron Fukushima
1985 Boltzmann Machine Hinton & Sejnowski
1986 Restricted Boltzmann Machine Smolensky
1990 LeNet Lecun
1997 LSTMs Hochreiter & Schmidhuber
2006 Deep Boltzmann Machines Salakhutdinov & Hinton
2014 GANs Goodfellow
1943 Neural Nets McCulloch & Pitt
1958 Perceptron Rosenblatt
1969 XOR problem Minsky & Papert
1980 Self Organizing Map Kohonen
1982 Hopfield Network John Hopfield
1986 Multilayer Perceptron Rumelhart, Hinton & Williams
1986 RNNs Jordan
1997 Bidirectional RNN Schuster & Paliwal
2006 Deep Belief Networks - pretraining Hinton
2012 Dropout Hinton
2017 Capsule Networks Sabour, Frosst, Hinton

Figure 8.1: The timeline of the deep learning created by Favio Vázquez [348]

It was, in fact, during a dark era before 1943 that the first mathematical model of a neural network was proposed. Two scientists, Walter Pitts and Warren McCulloh, from different research fields, logic and neuroscience, cooperated and proposed the first neural network model [349]. Moreover, this work has become the foundation of neural network models and logical calculation for neural activity. The references of this work were limited to three works published in 1925, 1927, and 1938. This model is still alive and known as the McCulloch-Pitts neurons [350], although it has been gradually developed and changed over time.

In 1950, Alan Turing published a paper titled "Computing Machinery and Intelligence-AM Turing" [351] and proposed the inquiry of whether a computer can think. He was a mathematician and also widely known for his involvement in breaking code in World War II. For the first time, Turing predicted the development of machine learning in 1947 and believed that a machine with the ability to learn from experience is necessary.

Seven years later, Frank Rosenblatt prepared a paper which was submitted to Cornell Aeronautical Laboratory in January of 1957. The title of the paper was "The Perceptron: A Perceiving and Recognizing Automaton" [352]. His research was closer to the hardware aspect of the issue. He pointed out that it would be possible to build a system which could be an electronic or electromechanical one. Furthermore, this system would be able to learn to recognize similarities between different patterns and information in which it would be similar to the perceptual process of the brain. His work is also considered as the basis of deep neural network (DNN) development.

In 1959, an additional advancement in machine learning took place which led to a new discovery in the visual cortex found in animals, hence we can call this year, the year of discovery of two types of cells, simple cells and complex cells. This work was carried out by two neurophysiologists who won Nobel prizes, David H. Hubel and Torsten Wiesel. This discovery influenced artificial neural networks and played an inspirational role for plenty of neural network models.

The next novel work is titled "Gradient Theory of Optimal Flight Paths" and published in 1960 by Kelley [353]. Although it was a significant work in the field of control theory, it has been used directly and indirectly in artificial intelligence and artificial neural networking since 1960. For instance, the behavior of systems with inputs and the modification of systems by using feedback contributed to propose the basis of continuous backpropagation models which can be used in the training phase of neural networks.

Several years passed and a new work was published, "Neocognitron" [354], and a new concept by the same name was proposed. A neocognitron is a hierarchical, multilayered neural network capable of robust visual pattern recognition through learning. The model was utilized for recognition of visual patterns, handwritten character recognition and other pattern recognition tasks, recommender systems, and even NLP [14]. In addition, convolutional neural networks were inspired by this unique work.

In 1982, Hopfield made an excellent progress in this field of research [355] by creating "Hopfield Net" which is a recurrent neural network (RNN). Interestingly, its popularity has not decreased over the ensuing years. If we call this type of neural network a gift from the 20th century for modern deep learning, we have not exaggerated at all.

After 4 years, an important learning method, back propagation, was proposed in [356]. The application of this method in existing neural networks proved that it would be an effective method for improvement of many proposed problems and tasks at that time, like shape and word recognition. The importance of this work is not limited to the 1980s as it laid the groundwork of deep neural networks. Hence, Hinton is usually referred to as the godfather of the deep learning area. He deserves this title because of his abundant attempts and sincere contributions over the years.

In the late 1980s, another great work [357] was published. The backpropagation approach was combined with convolutional neural networks whereby it was applied to handwritten digit recognition. In addition, the implemented system was utilized over the 1990s and the beginning of the 2000s by many companies, especially in the United States. Without any doubt, LeCun is a king in the neural network research.

A new term, long short-term memory networks (LSTMs), was proposed by Hochreiter and Schmidhuber in 1997 [358]. It was a new generation of RNN which had been improved by the capability of learning long-term dependencies. In this new model, the problem of long-term dependency was solved. In addition, the network could practically remember the information for long periods of time without any additional challenge for learning. This work is still popular and is used frequently in many tasks. For instance, Google implemented it into its speech-recognition software for Android-powered smartphones [359].

After about 9 years, another effective research was proposed in [340]. The important point is the similarity of the paper to a relaxation beach for neural networks in the middle of 2000s. Hinton proposed also the use of complementary priors to get a fast and greedy layer-wise unsupervised learning algorithm for deep belief network (DBN). It is a generative model with many layers of hidden causal variables [340] [360].

Advances in neural networks and image classification are owed to Alex Krizhevsky [361] who came up with the idea of AlexNet [362]. He is the winner of several international competitions on machine learning and deep learning. The first attempt was to improve LeNet which was proposed in the 1990s [363] [357] [364] [365] [366] [367] [368] [369] [346] [370] [371] [16] and to build a new and improved one. His success has undoubtedly been a new renaissance in neural network research. In fact, deep learning was kicked into high gear towards further success.

In 2012, Hinton et. al [372] marked a bold signature and introduced a novel regularization technique to prevent overfitting, a serious problem for deep networks, in deep neural network models. Another paper was published in 2014 [373]. This unique work explains the Dropout technique [373] and its improvement on the performances of deep neural networks in supervised learning tasks such as speech recognition, image classification, etc.

Ian Goodfellow, the leader of a research team, et al. [374] proposed a new framework to estimate generative models. Additionally, a goal was to cope with unsupervised learning which is generally a goal in artificial intelligence. This framework is called the generative adversarial nets (GANs) [374]. The main idea is to use an adversarial process for simultaneous training of two models and make a competition between them. The first model, the generative one, takes in the data distribution. The second one, the discriminative model, estimates the probability that a sample came from the training data rather than the generative model. It is responsible to determine if the sample is real or generated. The importance of this work was seen in LeCun's talk [375] where he recognized the generative adversarial net (GAN) as the most important development in the last 10 years in the area of deep learning.

In 2017, an interesting paper was published [376] and entitled "Dynamic Routing Between Capsules." Many people were curious and excited to know more about this article as it was a work of Hinton's research group, "the godfather" of deep learning. By reading the paper, we find that it introduces a completely new concept, capsules in neural networks. To explain this new work, let's consider one example. Figure 8.2 shows a woman's face, and Figure 8.3 represents the components of the woman's face. If we apply convolutional neural networks, both pictures will be considered to represent a face. Although the second picture consists of components of the face, orientational and spatial relationships between parts of the face are not considered in the second picture, and it will be predicted as face by using a CNN model. In fact, a combination of two eyes, a nose, and lips is not a face at all. A CNN model uses max pooling or successive convolutional layers to reduce the size of flowing data through the model. Therefore, the field of view will be increased by neurons of higher layers, and the detection of higher order features will be provided, but the important point is that max pooling is the result of losing significant information. The idea of Hinton's capsule networks is to solve the shortcoming of CNNs and use important spatial hierarchies among objects, whether the objects are simple or complex.

A group of the neurons forms one capsule and the used mechanism is an iterative routing-by-agreement [376]. In fact, lower-level capsules opt to send their outputs to the higher-level capsules because the higher-level capsules have activity vectors with big scalar products with the predictions which are coming from the lower-level capsules.

The deep learning topic is still open and ordinary people use its applications every day, even if they know nothing about this hot topic. For instance, Google's voice recognition, Facebook's face recognition, Netflix's recommendation engine, Apple's Siri are only some daily applications of the deep learning concepts. In this section, our purpose was to provide a survey of the history behind the deep learning and its sequel.

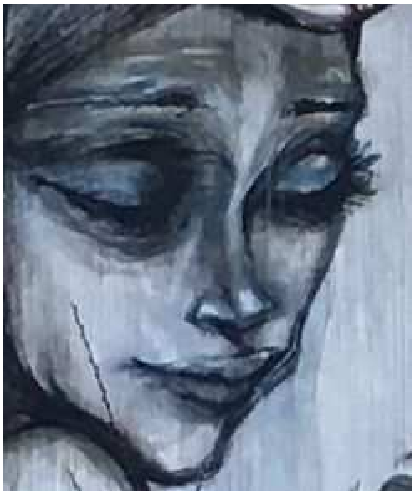


Figure 8.2: A Scene including a whole face

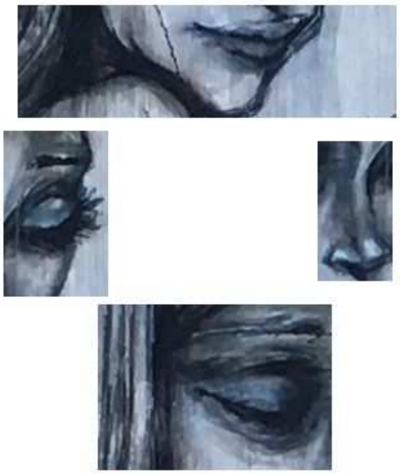


Figure 8.3: The components of the face seen in the scene

8.2.2 ANN

The guidance of the ANN is the human nervous system (see 11.2) with its unique components and functions. The creation of a new neural network is so similar to giving birth to a child. The child needs to be trained. His or her behavior and activities are undoubtedly dependent on the training procedure. The child's ability to learn is one of the most important properties of the human development. There is usually a focus on the ability to learn more sophisticated subjects and harder tasks over time.

Although artificial neural networks are inspired by biological neural networks, there are some basic differences between the two. The procedure of the learning and training is absolutely different. If we build an artificial neural network, we are going to achieve a final goal. This goal is pre-defined by the desired application of the network. In fact, we do not expect an additional demand when the network is ready to use. Consequently, the network is able to make decisions on unknown samples due to its practical learning process. When a child is born, there are still training and learning procedures, but we cannot specify a unique target. The passing of the time represents the final result. In addition to the training and learning, there are many factors, such as parents, economic situations, cultural effects, etc., that have impacts on a child's performance and development. Therefore, the result will be unpretentiously random and out of our hands.

Let's continue with artificial neural networks and try to find out how they have been modeled and what the basic structure is. The first aim is to emulate the human nervous system (see 11.2) in a mathematical and computational form. Hence, the aim is to achieve a simpler and smaller system which can be used by humans, robots and other computers.

Due to the presence of streaks of the biological nervous system, one important component of artificial neural networks (ANNs) is neuron and interconnection among the neurons to send and receive the information is based on the defined tasks. An ANN consists of different layers to simulate the biological model. Each layer typically consists of hundreds to thousands of neurons while its biological counter part has billions. The connectivity path between neurons is called topology. The outcome is actually a map of the neural network. Furthermore, each neuron has some inputs and a set of weights, and the point is a finite number of the inputs. Then, some mathematical computations are carried out on them by an activation function, and the output of the neuron is obtained. In comparison to biological neurons, the activation functions have the role of the synapses. Various layers and neurons are connected to each other to form the neural network. If we consider the learning algorithm as a component of the neural network, this component differs between two different networks. Figure 8.4 is an example of the neuron's structure in an artificial neural network.

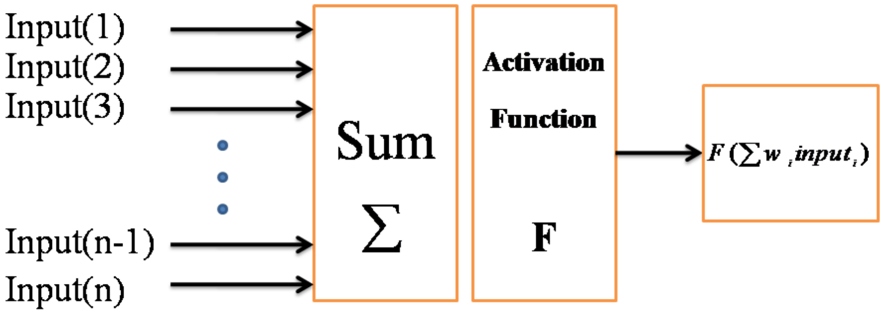


Figure 8.4: Artificial neuron's structure

It should be pointed out that the components of the biological neural networks of the human brain and the nervous system and the artificial neural networks are not exactly the same in structure and behavior as biological neurons, although recent developments have helped to increase the number of the neurons in the artificial neural networks and have made it closer to the biological one.

As previously mentioned, the learning process is so important in artificial neural networks, but it is unlike the learning process in biological neural networks. In artificial neural networks, we face a defined topology which is designed for solving a problem. It substantially follows specific goals and remains fixed over time. In addition, the learning is started from the scratch. To get weights, optimization algorithms can be applied. These will be adjusted randomly, then aggregations of input stimuli will be mapped to the desired output function.

There is also another method of doing the learning process which is called fine tuning [377]. In this type of learning, a pre-trained network topology is utilized, and adjustment of the weights is carried out by this pre-trained one. In order to fulfill the process successfully, the learning rate should be low. Therefore, the learning process is relatively slow. In both types of learning processes, the input data should be fed into the network and spread through the whole network. Continuously the outcome measurement and weight modification are carried out. The best possible weights will be finally put in the current direction for hitting the target of the task.

This learning process is comparable to a child learning to solve a puzzle. The child attempts to put the pieces of the puzzle in the correct positions. He/she tries to find whether the position suits the piece or not. If it is not the right position, he/she has received the feedback and knows that he/she has to find another place which should be smaller or larger than the previous position. Therefore, he/she tries again to solve the problem. Gradually the child puts all the pieces of the puzzle in the correct positions and achieves the final desire of the task. After the completion of the learning process, both the child and the neural network are able to do the desired tasks. However, the difference goes back to the needed time for doing the task. When the child learns, he/she is able to perform the task faster with the repetitive practice. Another difference is related to creativity. The child has a unique ability and he/she might have creativity to speed up the duration for finishing the task and tackling new problems as well. An artificial neural network responds to a new task without any creativity because the nature of the learning process is not the same as the one that the child obtained. A trained artificial neural network publishes the result of the new task as it has been learned. In fact, there is no creativity for solving new problems. To provide more information, a traditional and typical type of neural network, called feedforward neural network, is explained in (11.4).

8.2.3 Deep Learning Definitions and Classes

To define deep learning concepts and bases, we should not forget that deep learning is involved with many hard and unsolved problems of past and recent years. Most problems are interestingly complicated because of high dimensionality and lack of rules. In order to cope with the difficulties of such problems, it is very important to train the system for challenging circumstances and make it capable of doing the desired task in unforeseen and unexpected situations without pre-existing knowledge of the rules. Although deep learning is still fresh and young, we see its contributions in many existing systems. In addition to huge advances in deep learning algorithms, we cannot ignore the help of high CPU and GPU power and the availability of big and complex data. In the near future, we will surely hear more about developed systems based on the deep learning algorithms.

Deep Learning Definitions

Through the emergence of deep learning, a lot of definitions have been proposed for this field and high-tech descriptions have been created. In this part, we would like to investigate some proposed definitions.

Some researchers believe that the deep learning is a class of the machine learning algorithms that benefits from its large number of layers to perform the information processing nonlinearly and extract the features which might be supervised or unsupervised. Another definition of deep learning is based on another characteristic of such algorithms and a common target of the deep learning approaches. The purpose is to learn multiple levels of representation for modeling relationships of data where basically the complex and unsupervised learning of representations is the basis of many models. In fact, the features with lower levels are used to form features with higher levels, hence, a deep architecture is actually a hierarchy of features.

It has also been proposed that the goal of deep learning is to make better representations of data and learn them in the best possible way. The fourth definition of deep learning holds the hypothesis of being a part of machine learning and undertaking the learning process at multiple levels of abstraction. Use of the artificial neural networks and the statistical models contribute to building higher level concepts from lower level ones. In addition, if we consider the same lower level concepts, it will also be possible to create many high level concepts by using them. The last definition of deep learning focuses on the main goal of machine learning approaches. The aim is to create real artificial intelligence. The deep learning moves us towards this goal and makes more sense of data representation and abstraction.

Deep Learning Classes

Here, we would like to investigate the brain's working model and the classes of deep learning. Let's assume that we have a computer with many small processors. These processors can be compared to a massive number of neurons in the human brain system. The needed time for reacting to any input is a few milliseconds. If we design a very small electronic device for modeling a biological neuron, it is theoretically necessary to add a transfer function. This function plays the role of the neuron for responding to the inputs and commands. Connecting different neurons in neural networks makes it more similar to what we have in the human brain system. The final model will be closer to the working of the brain as well. Due to the complexity of deep learning algorithms, the used techniques and architectures, it is drastically hard to accurately define the borders among the deep learning approaches. Furthermore, we do not have a limited number of deep learning classes. As there is no scaling system for defining different classes of deep networks, we classify them into three major groups according to the architecture and techniques. Our categorization has three members, which are deep networks for unsupervised or generative learning, supervised deep networks and hybrid deep networks.

The first category, deep networks for the unsupervised or the generative learning, is a group of networks where there is no access to information about the final labels of the classes. This means that we do not use any specific supervision information in this group of deep networks. The point is that the deep networks of this group can be generative or non-generative. There is no obligation to be generative naturally, although they are mostly generative. Their intention is to get high order correlation of the available data for the desired tasks such as the analysis of pattern and synthesis usages. If the features representations and abstraction are unsupervised, then the network is absolutely a member of this group. Some example members of this group are DBN, deep Boltzmann machine (DBM) [378], restricted Boltzmann machine (RBM) [379], and generalized denoising autoencoders [380]. It should be noted that the most common members of this category are deep energy based models [381].

The second category belongs to deep discriminative networks for supervised learning. This type of network is usually utilized for classification tasks as they are powerful discriminative tools. In this group of networks, the labels of classes are in the hand directly or indirectly. The deep stacking network (DSN) [382], CNN and time delay neural network (TDNN) [383] are only some members of this group. In addition, some deep networks such as the RNN and the sum product network (SPN) [384] can be considered as supervised as well as unsupervised learning models.

The last defined category is the hybrid deep networks [385] where the term "hybrid" and "fusion" usually indicate the result of combined methods and algorithms. If a model is built based on the fusion of different approaches with the deep learning model, this developed model is a hybrid deep network. The approach used might be another deep learning model or another approach like feature detection and extraction algorithms. For instance, a hybrid deep model of convolutional and RBM models has been used in [386] for face verification.

8.2.4 Deep Learning and Traditional Machine Learning in Classification Tasks

Classification tasks can be divided into two main phases if machine learning algorithms are utilized. The first main phase is called the training phase. The second main phase is actually the testing phase where new and unseen observations should be tested. In this phase, the labels of the test images should be predicted. An important technical point is to know how the traditional machine learning algorithms and the deep learning approaches differentiate.

The training part of classification problems is usually performed by using the training images and the relevant image labels. Then, the obtained model of the training is evaluated by the use of testing images. Prediction of each testing image is carried out where the label of each testing image is determined. If the used algorithm is a traditional machine learning one, we usually have two important tasks which should be carefully completed in the training. The first one is to extract features which are useful for the traditional machine learning algorithms. As explained before, two important detection and extraction algorithms are the SIFT and SURF methods. The next step is to utilize the obtained features for building a training model. The feature extraction process will be repeated for the new images of the testing dataset, and the features will be applied to the trained model and the labels will be predicted.

Obviously, the important point is feature engineering, though the difference between traditional machine learning algorithms and deep learning algorithms goes back to the procedure of feature engineering. Feature engineering is a critical, difficult and time-consuming task. In addition, it involves a high level of knowledge. The workflow of the deep learning algorithms shows how it is different from the traditional machine learning algorithms. Figure 8.5 represents the workflow of each algorithm separately. Meanwhile, the traditional machine learning algorithms are not obsolete, although the advances of deep learning algorithms are faster than what we can imagine.

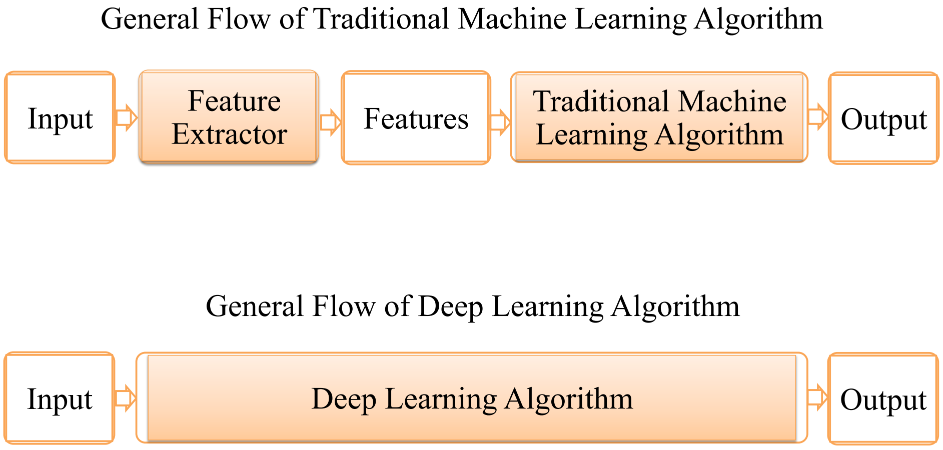

General Flow of Traditional Machine Learning Algorithm
Input
Feature Extractor
Features
Traditional Machine Learning Algorithm
Output
General Flow of Deep Learning Algorithm
Input
Deep Learning Algorithm
Output

Figure 8.5: Representing the workflow of the deep learning and the traditional machine learning separately

To wind up the explanation about the fundamentals of the deep learning and artificial neural networks, we compare some aspects of the human and machine learning in (11.3).

8.3 Proposed Approach

A modern method for analyzing and representing 2D images is to use deep neural networks and split data into smaller pieces, abstractions and levels, although many representation learning approaches were first proposed several decades ago [387]. Until 2006, there were two challenging subjects with respect to training multi-layer neural networks, overfitting and gradient scattering, and it was very hard to adjust parameters and obtain optimum performance in implemented neural systems. In [360], stacked autoencoders were proposed, and a hypothesis was confirmed that greedy unsupervised layer-wise training is helpful for the optimization of deep networks as the proposed approach leads to a better representation of relevant high level abstractions. It was a good start to step into a new generation of neural networks. Without any doubt, we owe to an increase in the volume of available data and advances in hardware components like the GPU and powerful processors. Our proposed approach is based on fully connected layers of a deep neural network for automatic natural plant recognition.

As there is a full connection between all components of the whole neural network, it is very similar to the human brain and nervous system. Due to our knowledge about the human body, we are more familiar with the architecture of such a neural network. Therefore, realization of the system is easier. Furthermore, a deep CNN, a hierarchy with many layers, is also known as a shift invariant or space invariant artificial neural network (SIANN) because of the shared-weights of the architecture and the translation invariance characteristics [388] [389]. Interestingly, this deep model is very dependent on nonlinear transformation functions. The input data and its scheme exceedingly depend on the desired task. In this section, the historical explanation of this type of deep network, the linked concepts and the topology of the proposed deep CNN model will be provided in detail.

8.3.1 CNN History and State-of-the-art

As you might know, the main task of plant species recognition is to classify an unseen input image of a plant and distinguish its class and species. If there is a tree in a scene, people are able to state that there is a tree in the picture because of the way that they learned and the visional ability of determining the image as a tree. Computers and machines do not have such a visional ability. If they look inside an image, they find quite different information from that of a human. A computer sees a matrix of pixels with different values between 0 and 255 which are the intensities of the pixels.

To solve the problem, the computer attempts to achieve important characteristics of the input image at different levels. For better understanding of the proposed approach, we first investigate the history behind the CNNs and describe different parts of its components.

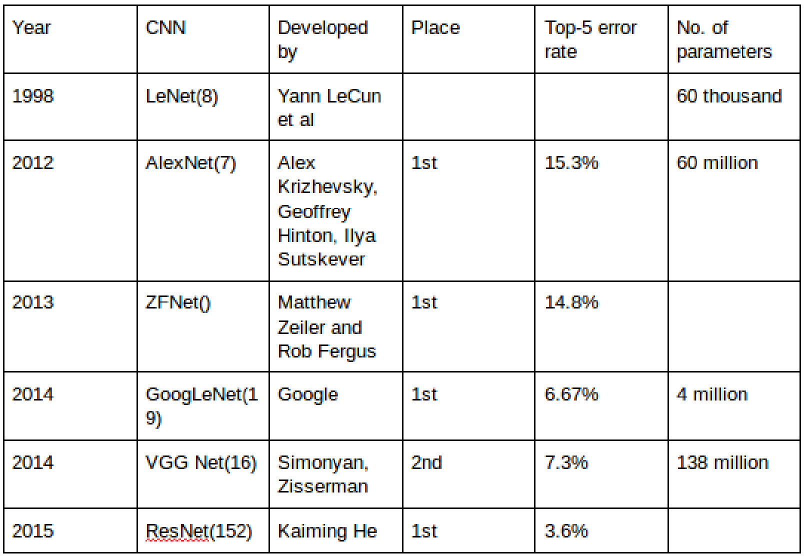
A CNN is not a deep network for today, it will be the future of image classification. The history behind it proves that its application will not be stopped in the near future. This class of neural networks was inspired by the visual cortex of two animals, cat and monkey, and simple and complex cells [390]. After several years, a new concept was introduced and it was called neocognitron [354]. This self-organizing neural network model has the ability of unsupervised learning which means there is no need to have a teacher during the learning process. In order to recognize stimulus patterns, this model worked based on the geometrical similarity of the shapes of patterns without being affected by the shape deformation of the input patterns such as the change of size, shift of position, etc. The model was tested on the handwritten numbers.

The CNN's evolution continued and a time-delay neural network was proposed in [391] whereby the results were remarkably good for the speech recognition of the phonemes and simple words. This type of network is a primary 1D convolutional network. In addition, the role of this network is also found in [392]. In 1998, a novel work was proposed in [16]. Its impact on deep neural networks was inevitable while a back-propagation algorithm and gradient based learning technique were utilized. This work also showed how a convolutional neural network can be combined with a search or inference mechanism to be used as document recognition tools. The application of convolutional neural networks entered a new phase when it was used for different systems in optical character recognition and handwriting recognition by Microsoft [393]. In addition, a convolutional neural network was proposed and used in the hand tracking [394], and another one was suggested for the face recognition in [347].

In the middle of the 2000s, a new suggestion was to use the GPU for machine learning purposes

[395]. Other works such as [396], [340], [360] and [397] also proposed more efficient applications of GPUs for training convolutional neural networks. The work in [360] was the initiator of the new advances in neural networks and many ideas have since been proposed in the field of deep learning. The advent of the deep neural networking goes back to the ImageNet Large Scale Visual Recognition Competition (ImageNet LSVRC) 2012 [398] where Krizhevsky, Sutskever and Hinton won the competition by building the "AlexNet" [362]. The network resembles LeNet which was proposed in [16]. With this network, the top-5 error was reduced from 26.2% to 15.3%. Furthermore, the used dataset consisted of roughly 1.2 million training images of 1000 classes, 50000 validation images and 150000 testing images [362]. Investigation of this success proves the contribution of Rectified Linear Units (ReLUs) [372] and Dropout [373] and speeding up the whole task by means of GPUs. Undoubtedly, we can call this success one of the main revolutions of the current century. Interestingly, the basis of the models with good performances in ImageNet LSVRC was deep CNNs from 2013 until 2016. Some examples are OverFeat in 2013 [399], ZFNet in 2014 [400], VGGNet in 2014 [401], GoogLeNet in 2014 [402], and ResNet in 2016 [403]. It is worth mentioning that Ke Jie, the GO world's champion, was defeated by Google's DeepMind AlphaGo artificial intelligence [404]. Figure 8.6 represents a brief overview of the CNN models and some details of each model separately [405].

Deep CNN models have been employed in many applications since 2012. In order to localize objects, CNNs have been used in [406] and the task has been completed efficiently. In [407], face recognition was performed by deep networks. The system is called the VGG-Face which led to obtaining impressive results in comparison with the state-of-the-art. Deep CNNs have also been proposed in the medical applications as we find in [408], [409], [410] and [411]. In addition, the CNN has been utilized for the circuit recognition in [412]. Furthermore, the CNNs have been examined in more areas such as estimation of pose [413], text spotting [414], visual saliency detection [415] and recognition of action [416]. These are only some examples of using deep CNNs for various domains.



Year	CNN	Developed by	Place	Top-5 error rate	No. of parameters
1998	LeNet(8)	Yann LeCun et al			60 thousand
2012	AlexNet(7)	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	1st	15.3%	60 million
2013	ZFNet()	Matthew Zeiler and Rob Fergus	1st	14.8%	
2014	GoogLeNet(19)	Google	1st	6.67%	4 million
2014	VGG Net(16)	Simonyan, Zisserman	2nd	7.3%	138 million
2015	ResNet(152)	Kaiming He	1st	3.6%	

Figure 8.6: Different CNN models [405]

We follow up our discussion with linked concepts of CNNs and provide a short explanation of the major steps and related concepts. Before starting the next section, it is worth noting that CNNs have been extended into many domains such as video analysis [417], the pharmaceutical industry [418], checkers [419], NLP [420], etc.

8.3.2 Linked Concepts

CNN has its own world with exclusive creatures, and the related concepts are arguably different from other domains. The effects of this unique world are undeniable in modern life. CNN is one of the advanced technologies that we are lucky to see and enjoy its advantages for many unsolved problems. In order to walk through this exciting world, we need to first know the relevant definitions and concepts. Then we can investigate the proposed deep model. Regardless of the difficulties, we would like to take a tour and jump into the major steps of deep CNN models.

Before starting the tour, we must mention some initial points in this part and (8.3.3). CNNs consist of different layers such as a convolutional layer, pooling layer, fully connected layer, etc (8.3.3). The architecture of this type of deep network is discriminative. The presence of the weight sharing in the convolutional layer and the possibility of selecting the pooling layer add invariance and semi-invariance properties, like translation invariance, to the CNN models. These characteristics endow the CNN models with effective solutions for image classification and pattern recognition tasks.

Main Steps of Deep CNN

For building a deep CNN model and doing the training phase, we usually involve four main steps and wrong processes during this phase might have disruptive effects. Preparing data is the first important step in creating a deep CNN model and all machine learning techniques. It is necessary to have a good dataset instead of good images of the dataset, but the question is, "What is the meaning of a good dataset instead of good images of the dataset?"

The dataset must be descriptive and representative. Our prepared dataset, a natural plant dataset, is exactly the one that we need. A considered factor of the dataset is diversity among the images. If we consider only one image, it might appear unusable while the whole dataset consists of images to help us reach our final goal. In addition, it is essential to handle and hoard the natural images of the dataset in an appropriate format which can be utilized in the system efficiently. It is also worth mentioning that one pre-processing step might be necessary. It depends on our definition of the model and the format of images. In our case, we have RGB images which are composed of three channels, R, G and B. Therefore, the input data is 3-dimensional and provides an extra depth.

The definition of the desired deep model is surely important as the second step. The architecture of the model will be defined in this step and the configuration of the model will be created. There are many parameters, such as the number of layers, type of layers, number of iterations, type of functions, filters, etc. It is necessary to create the first scheme of the model and compute the parameters that might be dependent on other parameters as well. The optimization process is the third step of deep CNN models. It plays an important role to reduce the loss of the model.

The last general step of preparing a deep CNN model is to do the training and get the final trained model. Due to the volume of the deep CNN model, one necessity is to use powerful hardware for the training step. Although it is possible to train the deep CNN model slowly by means of central processing units (CPUs), the priority is to utilize GPUs for this step and speed up the training process to get satisfactory parameters and the final model. The final trained model will be saved and can be applied in making new tests and predictions.

Deep Learning Frameworks

Deep learning is not only a huge area of machine learning, many researchers, developers and scientists are trying to create and develop new interfaces, frameworks, and toolboxes based on this concept. The final desire of the developers is to put their own framework in the highest place of deep learning's showcase and for passionate users to utilize their framework. In addition to researchers and scientists, companies also compete in the battle of building more efficient frameworks. No one is merely waiting to see the improvements of other toolboxes and frameworks. This fierce competition contributes to the deep learning advances in different aspects. Due to the newness of deep learning, there is still a great deal of work to discover. Pioneer players are attempting to overcome the difficulties to become the leaders of this part of the machine learning field. We realize the importance of deep learning and its frameworks when we look at the list of companies that have made investments. Google, Facebook, Amazon, and Microsoft are only a few examples of the current list of companies. Up until now, many frameworks have been proposed and we describe some popular frameworks in (11.5), mostly mentioned in [421]; furthermore, we discuss our utilized framework as well as provide a short explanation about our reasons for choosing it in (11.5.5).

8.3.3 Building Blocks of Deep CNNs and Relevant Definitions

Our deep CNN model consists of three main layers, input layer, several hidden layers and output layer, for classification of challenging natural plants. There are different types of layers which might be used in hidden part of the deep model and consideration of the nodes shows that each node of the current layer is connected to the nodes of the next layer. A hidden layer of the deep CNN model can be convolutional layer, pooling layer (called also down-sampling layer and sub-sampling layer), fully connected layer and non-linear layer. It should be pointed out that some relevant definitions are provided in (11.4.1).

8.3.4 Convolutional Layer

Let's suppose that we take an input plant image constituting 227×227×3, referring to width and height pixels with 3 channels of RGB image, hence we have actually a 3-dimensional input. If we consider this example, the convolutional layer, called also the convolution layer, is a learnable filter which slides over the input plant image and a dot product is performed between the input plant image and the defined filter. Two learnable parameters that are weight and bias constitute the convolutional layer. It is worth mentioning that weight is usually named kernel filter.

If the size of the filter is 5×5, the depth should be equal to 5×5×3 because of the structure of the input and its 3 channels. Hence, the filter is able to cover all three channels of the image. It should be noted that the result of each taken dot product is scalar.

The convolutional layer is a member of the feature learning in the deep CNN model, and it belongs to feature learning part where each filter is a representative of one specific interest feature. Hence, the CNN model will learn which feature is a component of the first plant species. The power of the output data is independent of the locations of the features. The plant species might be in a different position, but the model is still able to recognize it correctly as the presence of the features is important.

In forward passing, we convolve each filter in width and height directions, and the result is an activation map in 2-dimension for the filter. Intuitively, the network learns the filters that are activated in the input when they view certain features in some places. By stacking the activation maps for all filters along the depth dimension, a complete mass of the output is obtained. Each entry in this volume of the output can be considered as the output of a neuron looking only at the small area of the input, which shares common parameters with the other neurons in the same activation map, because these numbers are the results of applying the same filter. Figure 8.7 shows neurons in a convolutional layer. Each neuron is spatially connected to just one local region in the full depth of three color channels. Meanwhile, mathematical relations of the activation map are explained later.

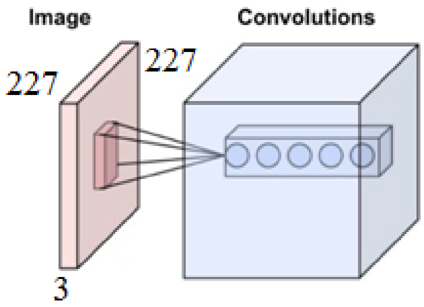


Figure 8.7: Neurons in a convolutional layer

Moreover, it is not practical to create connections between all neurons and all regions of the input volume. The reason is the increase of weights for the training phase. Moreover, the computational complexity increases which costs highly. To solve this problem, we connect each neuron to a small region of the input volume and the area of this small region for connection is a metaparameter called the receptive field [422]. In other words, a local region of the input volume is the receptive field with the same size as the filter. Now, if the receptive field has a size equal to 5×5, it means that each neuron in the convolutional layer has a weight fraction of $5 \times 5 \times 3 = 75$ for a $5 \times 5 \times 3$ input volume.

It should be pointed out that there is another relevant concept called the shared weights. Despite its simplicity, it is so helpful for transformation operations. By using this concept, it is feasible to utilize the same weights for performing the desired operation like convolution.

8.3.5 Activation Layer

The name of this layer interprets its role and responsibility which is to make decision on the final value of the neuron and activate it if it has not reached its ideal value yet. It is also mentioned that this type of layers is actually an element-wise operator. For instance, we have a cell that its ideal value is equal to 1. In practice, it is probably impossible to achieve this target value and the current cell equals 0.75. We use a function to activate this cell by assigning a comparison with a threshold value. For example, we compare the cell value with 0.6 and set it to 1 if the value is greater than 0.6; otherwise, we set it to 0. As a result, the size of both bottom blob and produced top blob remains the same and they will be identical in size. Some types of the activation layer are Rectified Linear Unit (ReLU) [379], hyperbolic tangent (TanH) and sigmoid.

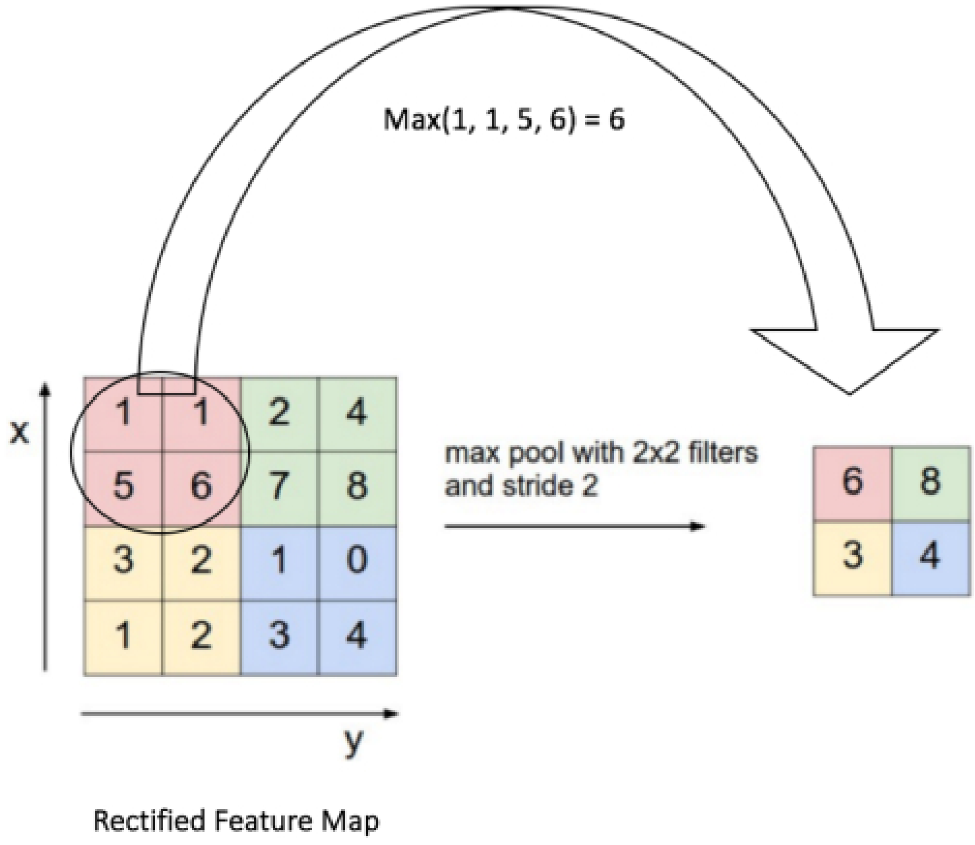
8.3.6 Pooling Layer

Pooling layer is an important layer of the deep CNN models, and operates a nonlinear down-sampling process on the width and height of the image. Therefore, the first outcome is actually reduction of the volume of the image. It should be noted that this layer is a member of the feature learning part of the deep CNN model and its operation on each feature map is independently performed.

One important point is the place of the pooling layer which is usually after the convolutional layer. Reduction of spatial dimensions, width×height, of its input which is actually the output of the convolutional layer helps to gain a reduced representation and less amount of parameters and computations in the model. This reduction can be 75%. We shouldn't forget that the down-sampling is loss of information, but we may benefit from this loss of information. To examine the loss of information, we consider the outcomes of the pooling layer. Reduction of the parameters and computations proves the important advantage of using the pooling layer which is the control of overfitting.

In addition, the pooling layer does not have any effect on the depth dimension of the input volume. There are different types of pooling layers, but the most popular one is called max pooling that gives us the maximum number in every small sub-region of the input volume convolved by the filter. For instance, we have a max pooling with a filter of size 3×3 and stride of 2 in our proposed deep model. Two other types of the pooling layers are average pooling and L2-norm pooling. For instance, if we have a 2×2 matrix and the average pooling is applied, then the result is the average of the four members of the matrix.

Figure 8.8 shows the operation of the max pooling layer where the filter's size is 2×2 with the stride of 2 at every depth slice [423] and the output is $\frac{1}{4}$ size of the input.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 8.8: Example of the max pooling [423]

8.3.7 Fully Connected Layer

Fully connected layer is generally used after convolution and pooling layers. What we have in this layer is the same as a class of the traditional feed-forward neural network, multilayer perceptron (MLP) [424], and outputs of the convolutional neural network will be the input of this layer. In other words, this type of layer is responsible for connecting neurons of one layer to neurons of the other layer. If we use a fully connected layer at the end of the deep model, the result is an N-dimensional vector. N is actually the number of the labels, and it means that we have N neurons finally. It is necessary to point out that this layer belongs to the classification part of the model.

8.3.8 Loss Layer

The driver of the learning is loss which does the comparison process between a predicted label and its true label. The loss layer is basically the final layer and there are different loss functions which can be used for different tasks. The computation of loss is forward-passing, but the gradient computation is backward-passing if we consider the direction of the loss pass. The typical loss function is SoftmaxWithLoss [425] and it can be utilized for doing one-versus-all classification.

Another type of the loss function is called sigmoid cross-entropy [426] and it is useful for prediction of N independent probability values in the range $[0, 1]$. The other type of the loss function is Euclidean, and it is suitable for regression to real-valued labels $(-\infty, +\infty)$. The loss layer is the member of classification part of the deep CNN model.

8.3.9 Local Response Normalization

Lateral inhibition is a concept in neurobiology and refers to an interconnection pattern of neurons in body. In fact, adjacent neurons or receptors inhibit each other. In the same way, our intention is to have peaks and local maxima in deep neural networks. In convolutional layers, a local response normalization (LRN) [427] layer helps us to do normalization across channels and to increase sensory perception. This type of layer is included in Caffe and can be considered as brightness normalization over local input regions. As described in [428], there exists a factor where three different parameters can be selected and each input value can be divided by it (see 8.1).

8.3.10 Blob

Here let's focus on a new definition that is a basic building block of the Caffe framework. There is an important component for the implementation of a deep CNN model in the Caffe framework. This component is called "Blob" and it can be used for storing and communicating purposes. If we consider an individual layer of a deep CNN in Caffe, we find that the layer consists of different blobs. A blob is like a wrapper for easy access to data that encapsulates this information for the CPU and GPU to process. One side of the blob's role is to hide the computational operations of the CPU and GPU and use the memory when there is a demand. It is an N-dimensional structure for the information storage like the batches of the input images, parameters for models and optimization. In addition to the storing property, the framework allows us to communicate among data by means of blobs. Two chunks of the memories, data and gradient values, can be stored by the blobs. They might be saved on the CPU or the GPU. In order to synchronize values between the CPU and the GPU, the blob utilizes a SyncedMem class, conceals the details of a process and minimizes the transfer of the data.

8.3.11 Topology of the Proposed Deep CNN Model

Our goal is to design, develop, and implement a system with a deep CNN core, a set of learnable filters for automatic classification of natural plant species without any user interaction and additional human or non-human actions. As discussed before, a convolutional neural network is composed of different types of layers such as a convolutional layer, pooling layer, fully connected layer, etc. An increase in the number of layers and neurons has made the CNN models closer to the real biological brain, nervous system and related concepts. The name of the model is the proof of the importance of convolutional layers among the common layers of this type of network.

The learnable filters of the model are not spatially large. But, they play an important role for extending the input data, adding depth, and obtaining the full structure of the input data. As a result, a volume of neurons has been obtained. Two main types of layers constitute our proposed CNN model. These are convolutional layers (first five layers) and fully connected layers (last three layers). The role of the fully connected layers is to make connections between current neurons to the whole neurons of the layer which is located before it. Figure 8.9 demonstrates the overview of two types of layers for the proposed CNN model [91].

Now we continue with the detailed explanation of the architecture of the proposed deep learning network and the layers of the implemented deep CNN model. If we stay far away from the deep network, we are able to divide the deep network into three different parts, the input data, the deep CNN and the output data.

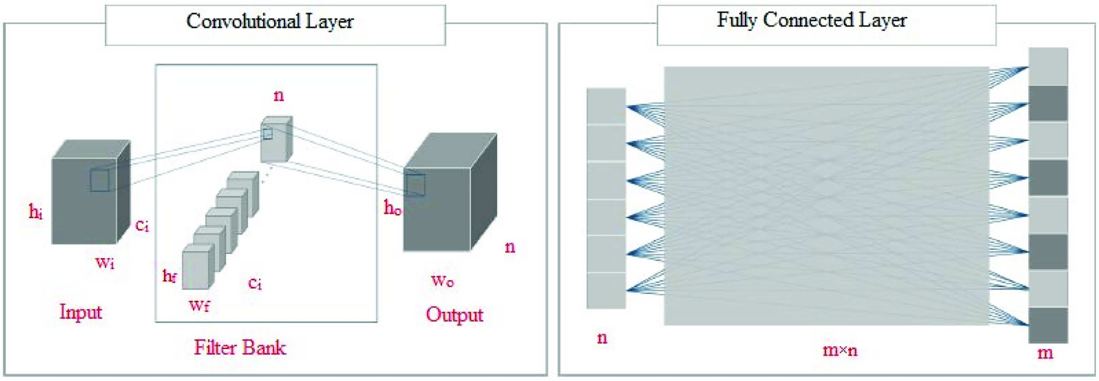


Figure 8.9: The overview of two types of layers for the proposed CNN model [91]

The input dataset consists of RGB natural images of different plant species. In order to provide more details of the first layer, let's look into it deeply. The whole layer is composed of different sublayers such as the convolutional layer and the pooling layer. In this convolutional layer, the weight filter is the Gaussian which means that we initialize the filters with a Gaussian distribution function. In addition, a bias layer is added which starts the biases at zero. The reason for setting the biases at zero is to break the asymmetrical structure which is obtained by some small random numbers in the weights. Then a ReLU layer is provided referring to [379]. If the input of this layer is x, then the output will be equal to x if x is greater than 0; otherwise it equals to $((negative - slope) \times x)$. We do not set any value for the $(negative - slope)$ parameter, therefore, the ReLU function is called standard of getting $\max(x, 0)$. It contributes to maintaining the memory consumption and to having the bottom and the top alike. Then a LRN layer [427] is applied where the local size (n) of it is set to 5 for summing over adjacent channels [427], alpha (α), the scaling parameter, is equal to 0.0001, and beta (β), the exponent of the following equation [427], equals to 0.75. This type of layer performs a kind of lateral inhibition by normalizing over the local input regions [427]. The following equation shows the mathematical relations of the parameters. In this case, each input is divided by the next formula:

$$(1 + (\frac{\alpha}{n})\Sigma_i x_i^2)^\beta \tag{8.1}$$

The last sublayer of the first layer is a max pooling one and it is like a bridge to connect the first layer to the second one.

Before describing the next layer, it should be pointed out that there are two batches, one belongs to the training phase and the other belongs to the testing phase. The size of a batch means the number of inputs in one pass for processing. If we set the batch-size to 250, we may get an error regarding the memory of GPU, so we should decrease the batch-size to a lower value. For example, we can set it to 50. In the testing phase, we should change the batch-size to another lower value. For instance, if the batch-size in the training phase is equal to 50, we can set it to 10. Additionally, we have divided our natural image dataset into two subsets; the number of training images is 800 and the number of test images is 200. The images have been selected for the training and test subsets randomly. If we set the batch-size of the training phase to 50, we can divide the number of training images by this batch-size, so $\frac{800}{50} = 16$. Therefore, the test interval in the solver step can be 16, 32, 48, ..., 16×n (can be a coefficient of 16). We set this parameter to 32.

If we set the batch-size of the testing phase to 10, then we divide the number of test images by this batch-size, so $\frac{200}{10} = 20$. Therefore, the test iteration in the solver step can be 20, 40, ..., 20×n (can be a coefficient of 20). The parameter is set to 20.

$$Epoch = \frac{Maximum \quad iteration}{test \quad interval} \quad \text{(in solver step)} \tag{8.2}$$

$$Test \quad iteration \quad \times \quad batch - size \quad (of \quad test \quad phase) = Number \quad of \quad test \quad images \tag{8.3}$$

Another important point is about possible alternatives for the Gaussian distribution. A feasible alternative is to utilize a constant filter. In this way, the weights will be filled by a constant value. However, it is not a good idea to initialize all the weights to a constant value. The same learning for all neurons is the weakness of this filter as the same outputs will be produced. In fact, it will not be learning different features. Another option is to do the process by using a uniform filter and consequently sampling small values from the uniform distribution. In [429], it has been proposed that completing the training is hard for networks with more than five layers when a uniform initialization is applied.

The next layer, the second layer, is also a convolutional layer. There is a pad parameter which is set to 2. It means that we have defined 2 pixels to add to each side of the input in this convolutional layer. The type of the weight filter in this layer is Gaussian. Moreover, there is a parameter called group and its default value is 1. When we do not talk about this value in a layer, it means that the default value has been utilized. This parameter helps us to limit the connections of each filter to a subset of the input. For instance, if we have 40^2 inputs and set this parameter to 2, then we have two groups of 20^2 connections, and subsequently, the process is accelerated by roughly double speed, although there is a very small loss in the convergence operation. In this layer, the neuron bias has been initialized by using the constant 0.1. The local response normalization and max pooling are the same as the ones used in the first layer. The output of the second layer is the input of the third layer.

The third layer is a convolutional layer without any additional layers of the local response normalization and max pooling types. Furthermore, the pad parameter is set to 1 and the type of weight filter is the same as the previous layer with a standard deviation of 0.01 and a mean of zero. By investigating the standard deviation value, we find out that setting smaller values result in chocking the activations and applying larger values lead to explosion of the activations. The used bias filter has the type of constant, the same as the first layer, and its value is set to 0. Then, one ReLU layer has been applied as it was done previously in the other two layers. It is noteworthy that there are other alternatives for the ReLU. One feasible alternative is actually the TanH which outputs a value in the range (-1, +1) and the center is equal to zero. Our first choice is the ReLU as the TanH involves with expensive computations. Furthermore, the ReLU performs faster than the TanH.

Then, the fourth layer is a convolutional layer, and there is no other pooling or local response normalization. The pad parameter is equal to 1 and the weight filter is a Gaussian distribution with a standard deviation of 0.1 and a mean of the default value which is actually equal to zero. The bias filter remains constant and has the value of 0.1. The last convolutional layer is the fifth layer of the deep CNN model. A max pooling layer has been added to this layer where its kernel size is 3 and the stride value is equal to 2. It is worth mentioning that the Gaussian filter used in the convolutional layer has a standard deviation of 0.01 and a mean of zero. We have adjusted 2 groups and the pad value is equal to 1. Moreover, the neuron biases have been carried out by the constant 0.1 and one ReLU have been utilized. It should be noted that the sensitivity of the value of the stride is important. Using large values leads to increasing the probability of losing information. In such case, the overlap to receptive fields would be reduced and spatial dimensions would consequently be decreased.

The first fully connected layer is the sixth layer of the deep network. The type of the fully connected layer is actually an InnerProduct layer. The input will be supposed as a simple vector, and an output will be produced in the form of a single vector with height and width of 1. In this layer, the Gaussian distribution has a standard deviation of 0.005 and a mean of 0. There is also a constant bias filter with the value of 0.1. To overbear the overfitting problem, the dropout approach has been proposed in [372] and we have applied it to the model. The term can be thought of as dropping out units in a deep neural network and the units might be hidden or visible [373]. The same as a neural network, the dropout layer is a biological inspiration. In [373], it has been mentioned that a theory of the role of the sex in the revolution [430] is a motive in proposing the dropout approach. In addition, the biological neurons have a rate for firing while the inputs have been received, but this rate might be random, which is similar to adding random noise to the current values, and the dropout approach is also based on this fact at each run. It switches off and removes random hidden units (neurons) in the training phase. The proposed approach has been used instead of the expensive methods that use a combination of predictions of different models for the reduction of the test errors [55] [431]. The dropout is also helpful for the convergence since it varies the required iteration value which is approximately a double one [91]. The neurons that have been dropped out are not helpful for passing forward, and they will not be present in the backpropagation. By using this procedure, the deep network samples have a different architecture any time there is an input, but all of the different architectures share weights. In the testing phase, all neurons are utilized and the outputs of them are multiplied by 0.5. Meanwhile, the structure of the seventh is the same as the sixth layer as well.

Let's clarify the dropout by considering a simple example. There are 10 students who are studying mathematics and the lecturer usually asks some questions about previous topics in the beginning of each session. There are only 3 students who are ready to answer the questions quickly. The lecturer is able to ask these students not to answer the questions, hence, the other students can also answer to the questions and learn the topics better. While the answers of these students are not correct, the lecturer supports them to make their answers correct, and this process of correction is similar to updating the weights in a deep CNN. The important outcome of this process is better learning for all the students of the class, all neurons of the layer.

The third fully connected layer is the last layer of the deep CNN model and its output has been fed into a 4-way softmax with loss. The initialization of the weight in the last layer has been performed by a Gaussian distribution with the standard deviation 0.01 and a zero-mean. In general, we compute a probabilistic likelihood per class and then utilize it for the calculation of the error that the network has created [91]. To obtain the accuracy, we have added an accuracy layer, and it shows the score of the network of the present batch. It should be pointed out that the accuracy layer will not be propagated. The softmax function is as below:

$$\sigma(z)_j = \frac{exp(z_j)}{\Sigma_{k=1}^{K} exp(z_k)} \tag{8.4}$$

where z is a vector of the inputs to the output layer. If we have 4 outputs, then there will be 4 elements in z. In addition, j is responsible for indexing the outputs and we have $j = 1, 2, ..., K$.

The optimization process is the third step of deep CNN models. Another part of this process is the generation of parameters' updating. In order to arrange the model optimization, there is a solver and it coordinates the forward inference of the network and also the gradients of backward for doing the update task which helps to improve the loss of the model. A part of the learning process is the solver's responsibility, and the other part of it is the model's responsibility to reduce the loss and obtain the gradients.

In this stage, we would like to investigate the scheme of the solver step and state some important parameters like momentum and weight decay as well as provide details of the learning part. We begin by walking through the stage to become familiar with the parameters therein. Training of the deep CNN model is based on stochastic gradient descent (SGD) [432] with the mentioned batch-size. To start the learning process, a momentum of 0.9 has been proposed as an acceptable and effective value. This parameter makes the model faster and more stable. There is another hyperparameter called the weight decay, and it is utilized for regulating large weights, direct updating of process and decreasing the error of training. Furthermore, it usually gets a real fraction and our chosen value for this hyperparameter is 0.0005. Furthermore, the low value of this parameter is helpful for reducing the error of the model's training and penalizing the large weights. Setting the value of the weight decay depends practically on the network and the goals. By setting this parameter to such low value, we fulfill our desired goal for caring about our predictions and obtaining a high accuracy.

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot < \frac{\partial L}{\partial w}|_{w_i} > D_i \tag{8.5}$$

$$w_{i+1} := w_i + v_{i+1} \tag{8.6}$$

where
i: the index of iteration
v: the momentum variable
ϵ: the rate of learning

The last term which is multiplied by ϵ: the average over the i^{th} batch D_i of the derivative of the objective with respect to w, evaluated at w_i.

There are some other feasible alternatives that can be used instead of the SGD. The first option is adaptive moment estimation (Adam) [433]. Similar to the stochastic gradient descent, it is gradient-based optimization method proposed in [433]. The Adam computes adaptive learning rates for each parameter. It stores and keeps both an exponentially decaying average of past squared gradients and an exponentially decaying average of past gradients like momentum. Considering the momentum as a ball running down a slope, this method is like a heavy ball with friction that prefers flat minima in the error surface. Although the performance of this method is good, it lacks from an important issue, generalization.

The other possible choice is a gradient-based optimization method which is called the AdaGrad [434]. In this method, the attempt is to "find needles in haystacks in the form of very predictive but rarely seen features [434]." This method is based on dividing the features into frequent and infrequent features. As a result, we would be able to adapt the learning rate to the parameters, perform smaller updates for the parameters associated with the frequent features and larger updates for the parameters associated with the infrequent features. As each parameter has its own learning rate and the learning rate is monotonically decreasing, the learning rate may become very small. Then, the system stops learning and it causes a problem.

The next feasible alternative is RMSProp proposed in [435]. The method is an adaptive gradient method. As explained, the AdaGrad suffers from radical diminishing of the learning rates. To overcome the problem of the AdaGrad, this method is helpful as it decays the past accumulated gradient. Hence, only one portion of the past gradients would be considered and the behavior of the

RMSprop is like the moving average. In addition, its contribution is to divide the learning rate by an exponentially decaying average of the squared gradients. Investigation of this method shows that its main problem is generalization as proved in [436]. However, our choice is the SGD due to the importance of generalization.

Due to the structure of the layers, the start of layers' weights is from a Gaussian distribution with zero-mean and standard deviation of 0.01. Neuron biases have been utilized in some layers, layer 2, layer 4, layer 5, and fully connected hidden layers 6 and 7 where the constant value is equal to 0.1 and the early stages of the learning have been sped up by use of ReLUs with positive inputs. In addition, the remaining layers have neuron biases with a constant value of zero. The base learning value has been kept constant and it is equal to 0.00001 which is a real value. By choosing this small value, the model with new data will be changed slowly, but the learning process of the new layer will be fast enough. Therefore, the process will be more reliable.

The used learning policy gets a value with a quoted string and it decides on the changes of the learning rate over time. As the used type of this parameter is by steps, the learning policy drops the learning rate in step sizes of the gamma parameter which is equal to 0.1, and the learning rate is multiplied by the gamma parameter. The next important point is stepsize, which is a positive integer and shows the number of iterations for going onto the next step of training. In addition, we use another parameter to limit the number of iterations, and it is representative of the maximum number of iterations. We are able to decide on the mode that we would like to use in solving the network. There are two modes, GPU and CPU, and our selected option is GPU.

Furthermore, there is a nice feature which helps to indicate how often the Caffe should output a model and solverstate [437]. This value is a positive integer. To complete the parameters of the solver, we introduce two remaining parameters, named test iterations and test interval. The test interval shows how often the test phase of the network will be executed [437], and the test iterations indicate how many test iterations should occur per test interval [437].

Now we would like to discuss an example and make clear the concepts used. We adjust the total number of iterations to 450000 and the optimization process runs for a maximum of 450000 iterations. If we set the stepsize and the learning rate to 100000 and 0.01, respectively, then we have:
1-For the first 100000 iterations, we just utilize the learning rate.
2- For the iterations between 100000 and 200000, the learning rate is multiplied by the gamma which is equal to 0.1, and we do the training at $(0.01) \times (0.1) = 0.001$.
3- For the iterations between 200000 and 300000, the learning rate is $(0.001) \times (0.1)$, which equals to 0.0001.
4- For the iterations between 300000 and 400000, the learning rate is $(0.0001) \times (0.1)$, and it equals to 0.00001.
5- For the iterations between 400000 and 450000, the learning rate is equal to 0.000001.

If the test iteration is set to 10000, the solver can calculate the accuracy of the model by use of the testing set every 1000 iterations.

In this section, we have introduced many new concepts, relevant issues (including deep learning frameworks) and definitions, the proposed approach as well as details of the deep model. Figure 8.10 represents the deep model of the deep natural plant recognition system (DNPRS). Regarding the materials and equipment, we will explain the related important points in detail in the next section to prepare for the experiment section as well.

To summarize the model, Table 8.1 provides an overview of the layers and the parameter values.

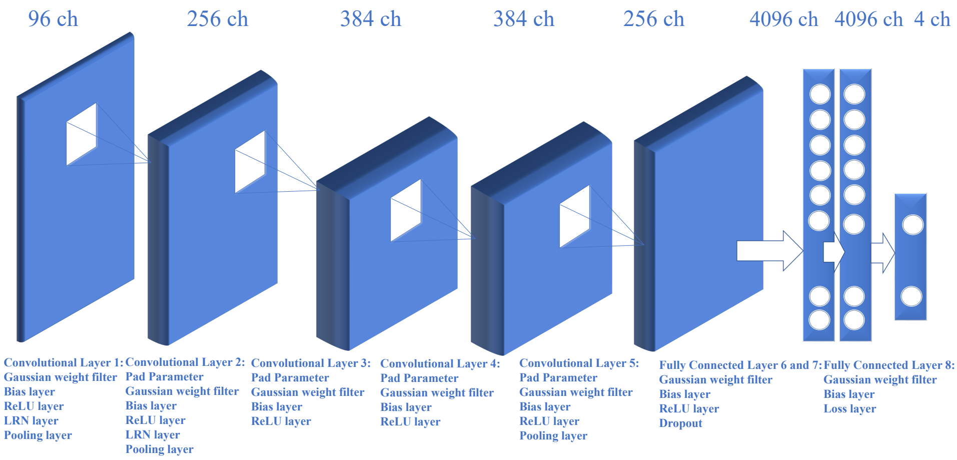


Figure 8.10: Deep model of the DNPRS

Layer 1 LRN layer (n=5, α=0.0001, β=0.75)	Gaussian weight filter (std=0.01) Pooling layer (max type)	Bias layer (constant value of 0)	ReLU layer
Layer 2 ReLU layer	Pad parameter (2) LRN layer (n=5, α = 0.0001, β = 0.75)	Gaussian weight filter (std=0.01) Pooling layer (max type)	Bias layer (constant value of 0.1)
Layer 3 ReLU layer	Pad parameter (1)	Gaussian weight filter (std=0.01)	Bias layer (constant value of 0)
Layer 4 ReLU layer	Pad parameter (1)	Gaussian weight filter (std=0.01)	Bias layer (constant value of 0.1)
Layer 5 ReLU layer	Pad parameter (1) Pooling layer (max type)	Gaussian weight filter (std=0.01)	Bias layer (constant value of 0.1)
Layer 6 Dropout layer	Gaussian weight filter (std=0.005)	Bias layer (constant value of 0.1)	ReLU layer
Layer 7 Dropout layer	Gaussian weight filter (std=0.005)	Bias layer (constant value of 0.1)	ReLU layer
Layer 8	Gaussian weight filter (std=0.01)	Bias layer (constant value of 0)	Loss layer (4-way Softmax type)

Table 8.1: An overview of the layers and the parameter values

8.4 Materials and Equipment

Our research explores the feasibility of high-efficient recognition for plant species from nature. Through the design of the proposed deep CNN model, we would like to become ready for the challenging test and experiment phase. Once the whole system, including the model, materials, and equipment, is designed, built, and trained, it is feasible to start doing the test and to predict the plant species in the testing dataset. Due to an extra defined goal for using the implemented system in real-time situations, there will be an additional challenge.

Furthermore, the used dataset is not clean at all, but what does this mean? It means that we did not consider any specific routine for taking the pictures. We have obtained a variety of images with different qualities, shooting locations, light intensities, point of views, distances, etc. Due to the importance of the generality of the system, there is no specific consideration during the selection process of the testing images. In addition, the images have been selected from the modern dataset randomly, thus the dark sides of the selection have been excluded.

One important point is preparation of the hardware, essential equipment and deciding on the computer mode for performing the training and testing steps. In addition, visualization of the results is also important because of further steps of the work. Furthermore, appropriate equipment helps to investigate the model and its parameters, especially critical ones with more influences on the performance of the model.

8.4.1 Record of Data

Our prepared dataset, called the modern dataset, has been created to support us in moving forward to find effective answers for the unsolved problems on the plant recognition topic. A complete explanation of the dataset and data recording was provided in Chapter 3. It is worth mentioning that the used camera was Canon EOS 600D.

8.4.2 Used PCs

Two different machines have been used to conduct the experiments and train the proposed deep CNN model as we had two options, CPU and GPU, for the training phase due to the selected framework. The first used machine has the following specifications and it has been used for training the deep model by means of the CPU:

Intel® Core$^{\text{TM}}$ i7-4790K, CPU @ 4.00 GHz, Installed memory (RAM) 16.0 GB

The maximum memory size of this type of CPU is 32 GB and its memory type is DDR3-1333/1600, DDR3L-1333/1600 @ 1.5V and the maximum provided memory bandwidth is equal to 25.6 GB/s. Furthermore, we should emphasize that the base frequency of the processor is 4.00 GHz and its price is roughly 350$ [438].

The specifications of the next machine, which has been used for training through the use of the GPU, are as follows:

Intel® Core$^{\text{TM}}$ i7-4820K, CPU @ 3.70 GHz, Installed memory (RAM) 16.0 GB, and graphics GeForce GTX 760/PCIe/SSE2

GeForce GTX 760 is a powerful and mid-range desktop graphics card and its manufacturer is NVIDIA which is providing new and interesting equipment for the deep learning. Its core clock can be in the range of 980-1033 MHz, but it depends on the temperature of the chip and the power consumption.

Although we used both GPU and CPU to do the training phase, we only used the GPU for performing the testing phase in this work. Perhaps someone asks, "How can I choose a GPU for my PC if I am going to implement deep neural networks?" It is really hard to give a narrow and specific answer to this question as the process of selecting the GPU is so complicated. An investigation of some specifications of the GPUs is helpful for making a good decision and choosing an appropriate GPU. The first point is to consider and check the Compute Unified Device Architecture (CUDA) cores. For instance, GeForce GTX 760 has CUDA cores 1152. The memory bandwidth is the other point that someone might probably consider as an important factor. It would be also possible to consider the memory required by the GPU and then decide on the GPUs by the ranking of the GB/s rate. As an example, the memory bandwidth of the GeForce GTX 760 is 192.2 GB/s [439].

The architecture of GeForce GTX 760 is Kapler, not Fermi. As a result, the power consumption is less when the used architecture is Kapler [440]. In addition, thermal specifications can be considered for comparing different the GPUs. The mentioned example, the GeForce GTX 760 has the maximum GPU temperature, 97 centigrade. The price is also another factor which can be used for comparing different GPUs. In many cases, our selection depends on the budget. For instance, the price of the GeForce GTX 760 is 199 Euro [441]. In summary, we selected the GeForce GTX 760 for the PC according to all compared factors, especially our budget. It was the cheapest GK110 GPU with the best performance and acceptable energy consumption to achieve our goal.

8.5 Experiments and Results

Before starting our discussion of performed experiments and obtained results, we would like to look at the application of deep learning in plant species recognition from another perspective. The world population increases daily and the estimated population is set to reach 9.8 billion by 2050 [442]. Moreover, the current amount of food production does not meet the needs of the population of 2050, it will only be sustainable if the production is doubled. This increase in production is very challenging due to, for instance climate change, over-allocated lands for agriculture activities in many countries, water scarcity, etc. Therefore, it is necessary to consider alternative practical approaches in different sectors such as herbicides, pesticides, water, plant growth rate, etc. as well as the development of new technologies to compensate for the lack of food.

New robots can contribute to overcoming the difficulties of population growth and the need to double the food production. Robots can be utilized for detection of weeds, pests and unwanted plant species. Robots will be able to play the human resource roles in gardens and take charge of them in the harvesting of horticultural products and the relevant processes like automatic picking and grading of ripe fruits. Furthermore, the quality of the mentioned processes can be enhanced by the use of robots. In many stages of agricultural processes, the recognition of plant species can be utilized which results in an increase in the number of products and the final outcome of agricultural activities. Regardless of the complexity of the goals, our goal is to apply a sophisticated deep CNN model for identification and recognition of natural plant species and use the deep system in different states of the plant growth. Despite many problems, it is a courageous decision to develop natural plant recognition systems based on deep learning, which is also called the supervised learning, as the deep neural network is trained by using labeled data.

To test our deep model, we would like to examine the DNPRS in the testing phase. We use our modern dataset, which is provided at the Institute of Real-time Learning Systems, University of Siegen [443]. Four different plant species have been collected by using a Canon EOS 600D in different weather conditions, days, time and change of distance between the camera and plant species. Consideration of the mentioned factors results in a natural dataset with high diversity in many parameters such as distance, background clutter, pose, angle, illumination, light intensity, viewpoint, etc. The original modern dataset consists of 1000 natural images, and we have randomly divided it into two sub-datasets, the training dataset (800 images) and testing dataset (200 images). The original modern dataset was also applied in [268] and [151], however, the training and testing datasets were not the same as the datasets used for the DNPRS. In Table 8.2, the number of the images at each defined distance, where the distance is measured from the camera to the plant species, can be observed.

Natural plant dataset	25 cm	50 cm	75 cm	100 cm	150 cm	200 cm
Number of plant images	240	240	240	240	20	20

Table 8.2: Number of images at each defined distance [91]

The DNPRS is based on one component of plants, the leaf, which has a longer lifespan compared to the other components like the fruit and flower. Moreover, there is no additional pre-processing operation such as scale, crop, etc. in the training and testing phases before using the input images and feeding them into the DNPRS.

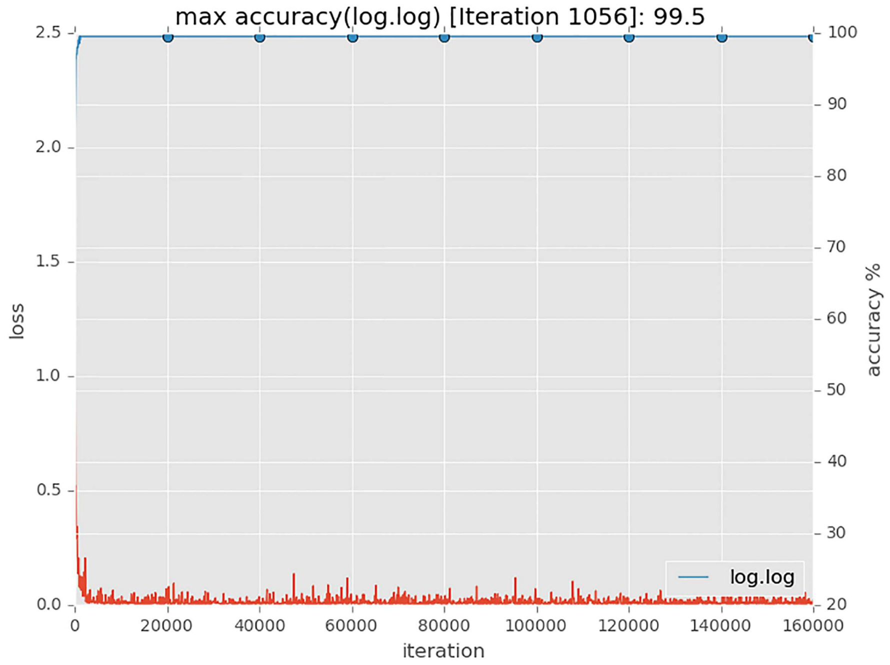


Figure 8.11: The accuracy, iterations and maximum accuracy of the DNPRS [91]

8.5.1 Classification Accuracy

Our results on the modern dataset are summarized in Figure 8.11. The deep system achieves the highest accuracy in the 1056th iteration. It is the best performance during the use of the modern dataset when it is compared to the other implemented systems in [268] and [151]. As it is observable in Figure 8.11 [91], the maximum accuracy occurs in the mentioned iteration. It is constant for the next iterations until all iterations, 160000 iterations, are finished and completed. Figure 8.11 represents the accuracy of the DNPRS in all iterations in the blue color and the maximum accuracy equals 99.5%. The second y-axis on the right side shows the percentage from 0% to 100%. In addition to the accuracy, changes of the loss can be seen in the red color. Its representative part is the first y-axis on the left side. The variation of loss proves that the value of the loss is high at the beginning and it decreases by the increase of the iteration.

By using the DNPRS, the achieved accuracy is larger than all other implemented systems based on the modern combined detection and description approaches using the modern dataset [268] [151].

8.5.2 Runtime by using GPU and CPU

As explained before, spreading a deep model across GPUs speeds up the whole process in comparison to the use of CPUs. We used two different personal computers (PCs) with different specifications. Once we adjusted the deep model to the CPU mode and used one of the PCs for training the model. In addition, we used the GPU mode for training the model too. Our purpose was to compare the runtime of the deep model.

In the GPU mode, the runtime of the deep model was 1248.5088 (sec), 20.8084 (min), when we reached the maximum accuracy in the iteration of 1056.

One question that might arise is, "How long does it take if we use another mode, the CPU mode?" Let’s check the runtime when the CPU mode was applied. Table 8.3 shows the runtime of the model for both GPU and CPU modes. With the GPU acceleration, our deep model training is more than 2422 times faster than using the CPU. In other words, several weeks have been reduced to less than

DNPRS	Runtime (min)
GPU	20.8
CPU	more than 50400

Table 8.3: Runtime of the deep model for the GPU mode and the CPU mode

30 minutes. A reduction of runtime helps us to do the desired changes on our model easily without waiting for a runtime of several weeks. This obtained time shows the importance of using the GPU for deep learning algorithms. Meanwhile, exploitation of the benefits of using the GPU is not limited to runtime. Beside time, energy consumption is also important where this factor decreases by means of GPUs. Therefore, the optimized and formulated solution is to use the GPU and consume much less energy for the power and also the cooling.

Due to the obtained accuracy and runtime with the GPU, there is a good trade-off between the energy consumption, the final deep model and the maximum accuracy. We cannot ignore one important point when choosing the GPU or the CPU and this remaining point is budget. If the budget of a project is not sufficient to utilize the GPU, the option of using the CPU is anyway available. Another option would be to use Google's Cloud tensor processing unit (TPU) [444], but it is currently only available in the USA [445]. Use of one TPU costs 6.50 USD for an hour [445]. In the near feature, it might be possible for the users from other origins to rent Google's tensor processing units (TPUs) and benefit from the fast running time for deep learning algorithms.

8.5.3 Confusion Matrix, Precision and Recall

The first experiment of this section involves building the confusion matrix for the tested images. As it is shown in Table 8.4, there is only one misclassification for one of the plant species. In fact, the plant type is Cornus, but it is identified as Amelanchier Canadensis. We have only one error in the recognition of 200 samples of our four plant species. Therefore, the accuracy (6.8) is 99.5%, as mentioned before.

DNPRS	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	50	0	0	0
Amelanchier Canadensis	0	50	0	0
Acer Pseudoplatanus	0	0	50	0
Cornus	0	1	0	49

Table 8.4: Confusion matrix of the DNPRS

Previously, we proposed different plant recognition systems for identification of plants in the natural environment in [151] and [268]. The idea behind the systems was to use the potential of the combined modern detection and description techniques such as SIFT, SURF, HARRIS-SIFT and FAST-SURF. Through an investigation of the results, we find that the system implemented by the use of the SIFT algorithm has the highest accuracy among all implemented systems, and its accuracy is equal to 94.94%. If we check out the systems implemented, SURF, HARRIS-SURF, and FAST-SURF, based on the SURF algorithm as the description component, we find out that the largest accuracy belongs to the system using the SURF detection and description techniques and the accuracy is 93.96% [268]. The accuracies of the other systems with the FAST-SURF and HARRIS-SURF techniques are 90.94% and 90%, respectively [268]. There is a considerable factor which has effects on the performance in the mentioned systems. This factor is actually the distance from the camera and the plant species. As you remember, we have taken the pictures in various distances. The question is, "How might the distance ruin the implemented systems, though the accuracy is acceptable and good?"

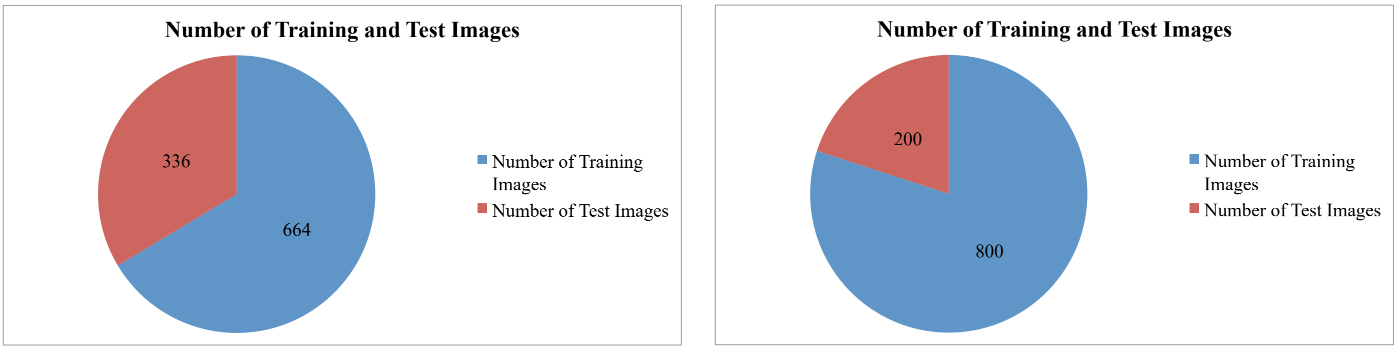


Figure 8.12: Number of training and test images in systems based on the SIFT and SURF description approaches [268] [151] (Left), Number of training and test images for the DNPRS [91] (Right)

Let's consider the system using the HARRIS-SURF as its basis. Once we would like to get the result in a defined distance, we have to construct a vocabulary for this specific distance. In fact, it is necessary to build a new vocabulary when the distance is changed. If the distance is 25 cm and we increase it to 50 cm, then we have to build a new vocabulary for the distance 50 cm and the vocabulary is not useful at all. We have totally designed and implemented six different systems for the natural plant recognition in [151] and [268]. As we have to construct one vocabulary per distance for each system, there are a lot of vocabularies. There is no consideration for the distance if we apply the DNPRS for the natural plant recognition. Thus, the weakness of the previous systems has been solved completely.

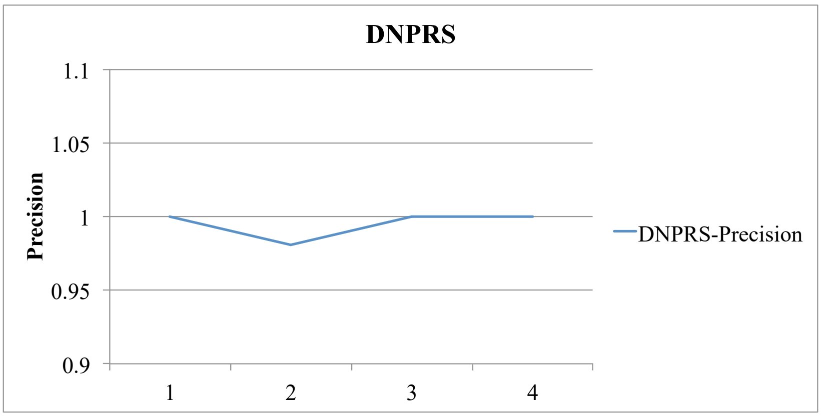


Figure 8.13: Precision measurements for the DNPRS [91]

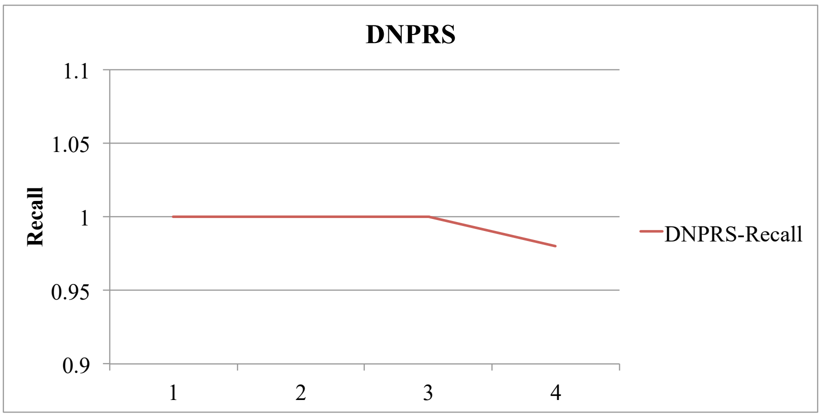


Figure 8.14: Recall measurements for the DNPRS [91]

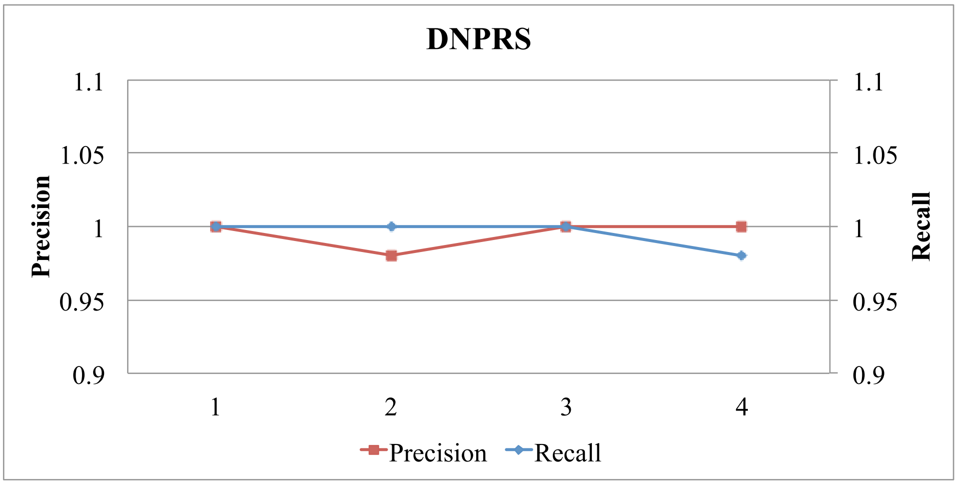


Figure 8.15: Precision and recall measurements for the DNPRS in one figure [91]

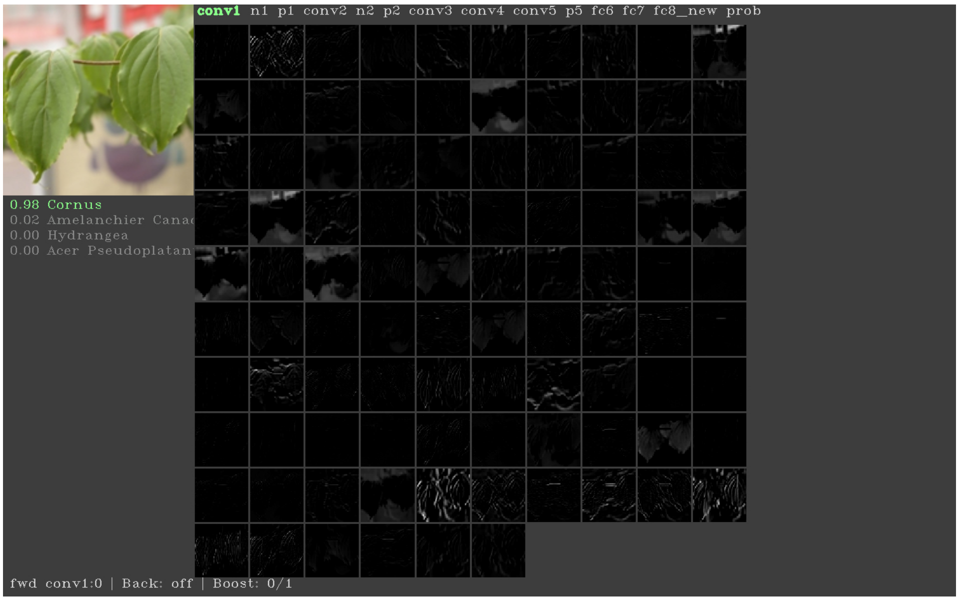
To continue with the implemented systems in [151] and [268] and the DNPRS, we need to consider the difference of the training and testing datasets. As explained before, the number of the testing images and the training images are 664 and 336 images, respectively [151] [268]. However, the training dataset of the DNPRS has 800 images where its testing dataset consists of 200 images. Figure 8.12 shows the split of the original dataset into training and testing datasets for different proposed systems in [91] [151] [268].

To extract new information from the confusion matrix, we compute two important criterions, the precision (6.9) and the recall (6.10), which are beyond the accuracy. Figure 8.13 and Figure 8.14 illustrate the measurements of these two metrics and the four values of the x-axis, 1, 2, 3, and 4, indicate respectively Hydrangea, Amelanchier Canadensis, Acer Pseudoplatanus and Cornus. For instance, the recall value of Cornus is equal to 0.98, whereas, the recall value of Amelanchier Canadensis equals 1 which is the maximum value.

High values of the precision and the recall express a better performance of the DNPRS for the considered plant species. By plotting the precision and recall measurements, we are able to investigate another metric which is the area under the plotted measurements.

The maximum value of the area under plotted precision measurements is 3.00 when the prediction of the testing images is completely correct. Due to the presence of one misclassification for the whole testing images, the related area is less than the highest possible value and its value is 2.9803.

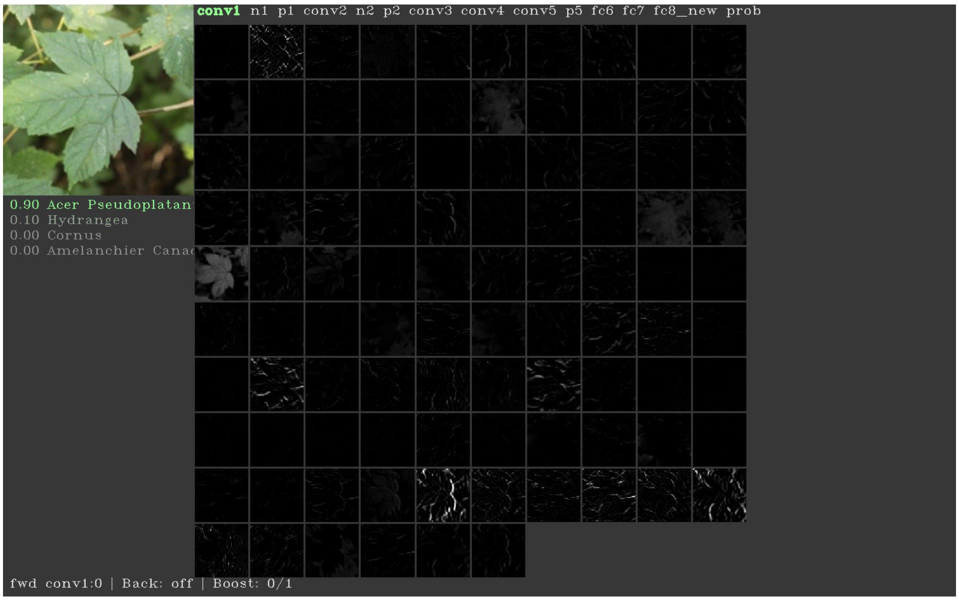
From this point of view, we are able to check the plotted recall measurements. We find out that the maximum value is again 3.00 if there is no misclassification when the images are tested to recognize the types of plants. One misclassification has occurred and the area under the recall measurements has been decreased to 2.9900. In order to compare both recall and precision measurements simultaneously, Figure 8.15 is plotted.


conv1 n1 p1 conv2 n2 p2 conv3 conv4 conv5 p5 fc6 fc7 fc8_new prob

```
0.98 Cornus
0.02 Amelanchier Canac
0.00 Hydrangea
0.00 Acer Pseudoplatan
```

```
fwd conv1:0 | Back: off | Boost: 0/1
```

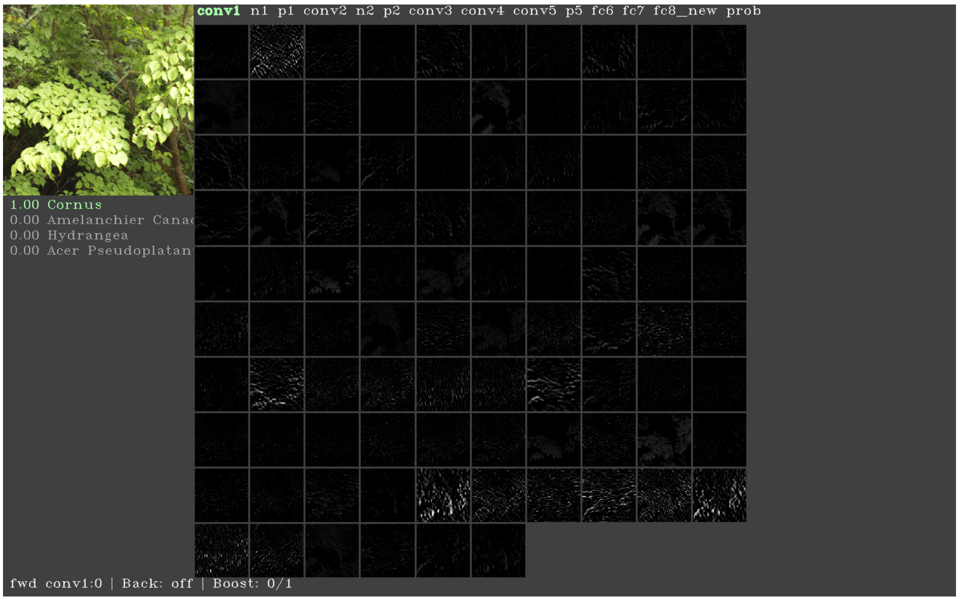
Figure 8.16: Visualization of the test image taken in a short distance and the type of the plant is Cornus in reality [91]


conv1 n1 p1 conv2 n2 p2 conv3 conv4 conv5 p5 fc6 fc7 fc8_new prob

```
0.90 Acer Pseudoplatan
0.10 Hydrangea
0.00 Cornus
0.00 Amelanchier Canac
```

```
fwd conv1:0 | Back: off | Boost: 0/1
```

Figure 8.17: Visualization of the test image while the type of the plant is Acer Pseudoplatanus in reality [91]



```
conv1 n1 p1 conv2 n2 p2 conv3 conv4 conv5 p5 fc6 fc7 fc8_new prob
1.00 Cornus
0.00 Amelanchier Canac
0.00 Hydrangea
0.00 Acer Pseudoplatan
fwd conv1:0 | Back: off | Boost: 0/1
```

Figure 8.18: Visualization of the test image taken in a long distance and the type of the plant is Cornus in reality [91]

8.5.4 Visualization of Proposed Deep Model and Scoring

In 2015, a useful tool for layer visualization of deep models was introduced by [446]. It is helpful for interpretation of constructed deep models in Caffe. This tool is useful for visualization of implemented deep networks. It is also applicable to give scores for testing the process of new images of natural plant species. Furthermore, it helps us to compare the scores of different testing images. As a part of the system, this tool is added to the implemented recognition system. The whole process of testing a new image is automatic and there is no interference. Three samples of the testing images, like an image taken in windy weather, have been processed by the use of this tool and the results are shown in Figure 8.16, Figure 8.17 and Figure 8.18.

Let's consider one of the figures, for instance, Figure 8.17. In this example, the input image that is considered as the testing image is represented on the top left corner and the names of the four plant species have been written under this image:

- Hydrangea, Amelanchier Canadensis, Acer Pseudoplatanus and Cornus

The first name is Acer Pseudoplatanus, and the written score behind it is equal to 0.90. This score means the probability of being Acer Pseudoplatanus is 90%. The second name is Hydrangea, and the score on the left side of it is 0.10. This means that the probability of being Hydrangea for the input test image is 10%. The next written plant species are Amelanchier Canadensis and Cornus and their scores are zero.

In summary, the prediction of the input testing image is visualized in one figure. If two different samples of one plant species are correctly recognized by the system, we are able to compare the testing samples visually. Figure 8.19 shows two different samples which are tested by the system and their recognized species is the same. We would like to use a reverse engineering process and visually compare them with respect the obtained scores of the system.

As we have seen in Figure 8.19, the image on the left side has been captured in the short distance where the image on the right side has been taken at a longer distance. The distance is an important factor for the human brain and visual system to understand the exact shapes of leaves in images and recognize types of plant species. Furthermore, the image on the right side looks blurry and the reason lies in the weather condition that is windy. Strong wind moves leaves and branches of plants and

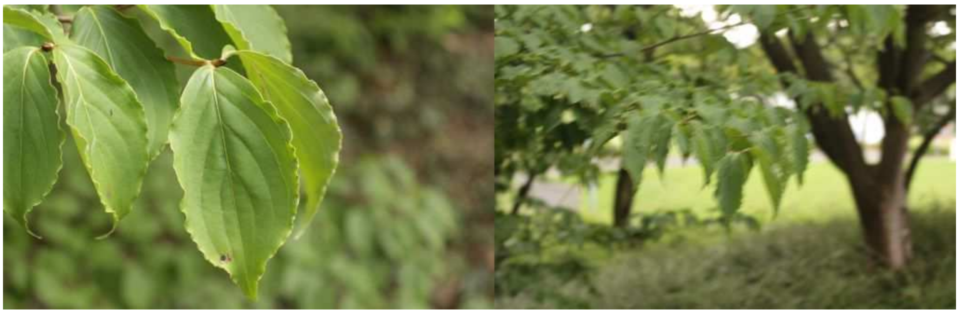


Figure 8.19: Visual comparison of two samples recognized correctely by the system

the human is not able to identify the shapes of leaves for different plant species, especially at longer distances. As humans are often curious about details and pay attention to different colors in a scene, the observer focuses on the trunk in the image on the right side. In addition, there is no single leaf in both images. It makes the problem harder than we expected in controlled photographing conditions, further visual challenges prove the efficiency of the implemented system as well.

8.5.5 Deep CNN, Drawbacks and the Most Recent and Potential Upcoming Breakthrough

As explained, layers play an important role in deep CNN models. The main component is actually the convolutional layer since the name of this type of neural networks is derived from this important component. Close layers to inputs are obviously deeper and they are responsible for detecting and extracting features. Furthermore, simple features will be combined and more complex features will be provided, whereas, the other layers are closer to the last layer. Hence, we find very high level features at the top of deep models for doing the final predictions. Over time, the access to useful hardware for deep learning purposes has increased. Additionally, the deep CNNs are amenable to the chips and the field programmable gate arrays (FPGA). The CNNs are compatible for the hardware, and manufactures are working on producing new hardware due to the needs of the current market. An interesting example is the Intel FPGA which accelerates the artificial intelligence for the deep learning in Microsoft's Bing Intelligent Search [447]. Moreover, some companies, such as NVIDIA, Intel, Samsung, and Mobileye [448], have made a good effort in the development of convolutional network chips and deploying deep learning models in different applications. In addition to high technology companies such as Google, Facebook, Microsoft, Apple, IBM, Yahoo! and Twitter, the number of the startups has grown to begin the research and projects using the CNNs. They have interests primarily in developing new relevant products and services.

The benefit of CNNs is to have layers with three dimensional volume neurons. The turning point is the presence of depth in addition to width and height. Figure 8.20 shows the width, height and depth of such a structure. In the deep CNNs, all neurons of a layer are not fully connected to the next layer, and they are connected to a small region of the layer.

Figure 8.21 shows a 3D input volume with the size $4 \times 4 \times 3$ as follows.

The CNNs are very well-suited for image processing and image classification domains. If we compare CNNs to RNNs in the image domain, we find the superiority of the CNNs in fundamental components. The CNNs have trainable parameters that depend on depth and the used filters at the current layer. On the other hand, many convolution operations are performed on input volumes. The

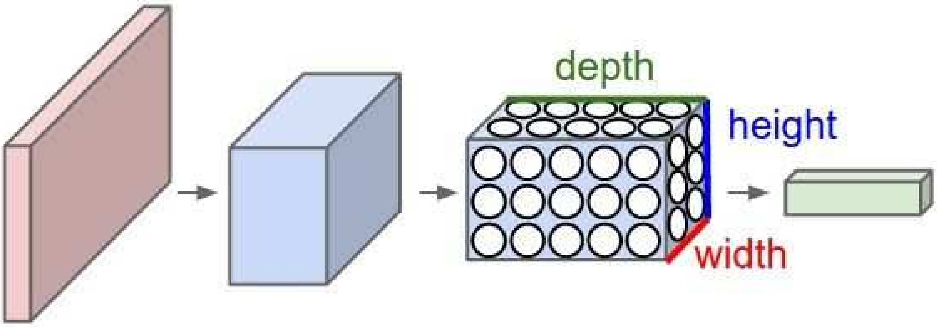


Figure 8.20: Structure of a 3D neuron in CNNs

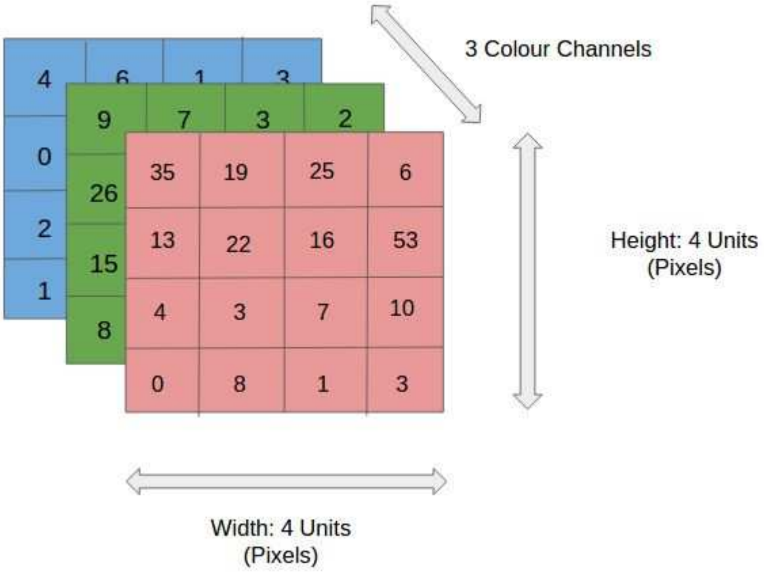


Figure 8.21: Structure of a 3D input volume for the deep CNN [449]

result is to have features and activation maps. Furthermore, huge numbers of filters help to do the learning process and get the spatial features from the volumes. Consequently, we obtain very abstracted representations that lead to the prediction of the outputs through an advanced process. In fact, the deep model learns how to capture important components of an image such as edge, curve, line, etc. and utilizes these components to recognize larger structures which are actually plant species. The RNNs have some filters with the same weights, thus, they are not suitable for classification tasks as they cannot be used for doing the training process with the idea of capturing the information at different levels. The RNNs do not help us to fulfill the generalization goal in our system because there is a single series of the filters which learn to associate the current input at each step, and the RNNs are applicable for the recognition of patterns across the time.

Two important questions are: How does one train such a model with many layers and parameters? How does one overcome the obstructions of the gradient, such as instability and the tendency of vanishing or exploding?

Although we have a big model with many layers, the type of these layers is convolutional one. Using this type of layers contributes to reducing the huge number of the parameters. Consequently the learning process becomes easier. The effect of the dropout is undeniable in our deep model as it avoids overfitting in such a complex neural network. Acceleration of the training process has been done by using the ReLU instead of the other methods like the sigmoid. This process is usually 3 or 5 times faster. Interestingly the ideas seem to be simple, but they are powerful and efficient.

With regard to the drawbacks, the main weakness is the need of the powerful hardware and its expensive costs. Furthermore, it is hard to handle with many hyperparameters and the adjustment of these parameters is not easy at all. Designing a deep model is hard because it is not possible for us to change only a parameter and get a new version of the deep model in a short time. It is time consuming if we would like to make a new change and obtain a new model with the new adjusted parameter. If we concern about changing a layer completely, it will be catastrophe for designers of deep models. Here we would like to consider the last drawback. Important spatial hierarchies between the simple and complex objects are not considered in the internal data representation, although the solution is to use the idea of the capsules and the dynamic routing between capsules [376].

The last explanation of this section is about the most recent and potential upcoming breakthrough in the deep learning which has been expressed by Yann LeCun [450]. He proposed the GANs as the most important advancement in the deep learning, and it is a member of the unsupervised machine learning algorithms. The unique idea is to train two neural networks simultaneously at the same time. The first neural network is called the discriminator, which is typically a convolutional neural network. The other network is called generator, which is typically a deconvolutional neural network [451]. This algorithm has been utilized in different applications such as modeling patterns of the motion in video [452], reconstructing 3D models of the objects from the images [453], the improvement of the astronomical images [454], and the image enhancement [455].

8.5.6 DNPRS, Applications and Future Work

This proposed plant recognition system, DNPRS, is a unique system in different aspects. Its final accuracy is high enough in comparison to the other previous implemented systems in [268] and [151]. In order to recognize the natural plants, the implemented DNPRS can be used in a robot and applied in the real-time application. We are able to mount a camera on a robot or semi-robot system for the classification of the plants when the robot or semi-robot goes through a garden, jungle, or farm, at any time of day, even if it is morning or evening without any consideration about the weather condition and the presence of the sun.

Robots are under the pressure of entering in different fields and our expectation of robots is to solve the remaining problems of each field and obtain accurate results. The future of agriculture is connected to the future of robotics. It is a fundamental demand to be applied in farming and agriculture, although there are many challenges which should be taken into account. Many aspects of modern farming, such as soil development, soil management, pruning, seeding, harvesting, light management, pest control, etc., should be enhanced for the future of world due to its rapidly increasing population. One necessity is to identify all species which have been grown on farms and monitor the products. Automatic recognition of plants is also useful and applicable in the harvesting process. In addition, robots help to accelerate different tasks on farms and save time that is a vital factor in the modern world. To attain peak efficiency, farmers need to use advanced equipment. They tend to have remote access to the farms during the day and afternoon [91]. By using robots, farmers are able to juggle various facts such as weather, level of soil moisture, nutrient content, etc. The DNPRS is an intelligent tool to facilitate identification of plant species and it can be used in natural conditions.

Figure 8.22 shows the schematic of a semi-robot that will be used in the future for testing the DNPRS as a real-time system. The semi-robot is composed of a PC, monitor, camera, unipod and carrier. It is possible to adjust the height of the unipod and rotate the camera if it is needed.

There is also a real agriculture robot at the Institute of Real-time Learning Systems (EZLS) at the University of Siegen, Germany and it is called the Zephyr [17]. Figure 8.23 shows this robot that is equipped with a camera, SJCAM SJ4000 and a wireless modem [17]. If an image is taken by the Zephyr, the image can be transferred over the wireless network to one stationary platform or a cloud

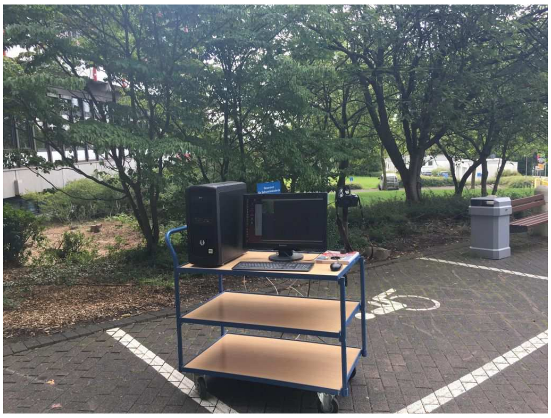


Figure 8.22: The designed semi-robot for the real-time application of the system in the future [91]

server. Then, the DNPRS recognizes the plant species.

Due to the properties of the DNPRS, the real-time system is usable at various distances, different times of day like morning, noon, and evening, and different weather conditions such as sunny, cloudy, etc. It is worth mentioning that the whole process of the plant recognition, from taking a picture of the plant to the final result, the species of the plant, will be completely automatic. The independence of the whole system from the used camera will also be examined by using the other cameras [91]. As you remember, we detected and extracted the features for using them in the traditional machine learning algorithms, SVMs. We are also able to extract the features by using a deep model and then training the features in the algorithms like SVMs. Hence, we can use the rich extracted features obtained by the deep model.

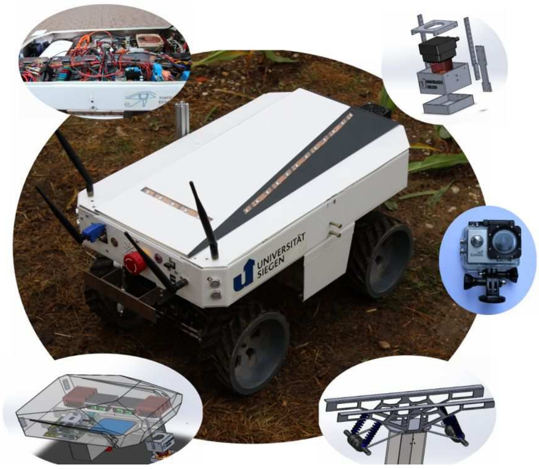


Figure 8.23: The mobile robot, Zephyr, that can be used for the real-time application of the system [17]

8.6 Conclusion

Since the appearance of deep learning, an increasing interest in this unique area of machine learning has led to a diverse set of applications for exploring the complicated and unsolved problems in the machine learning. Another aspect is to supply the new needs of modern life with respect to the capacity of deep learning algorithms. In this chapter, we designed and analyzed a deep convolutional neural network and deployed for classification of a very challenging natural plant species in natural environments. Many different factors have been taken into account according to characteristics and properties of the dataset and the result is promising. This system, DNPRS, fulfills our determined goals such as applicability, generality, etc. The DNPRS classifies efficiently four plant species with an accuracy of 99.5%, which is larger than obtained accuracies by the systems used the modern combined methods like FAST-SURF, FAST-SIFT, HARRIS-SIFT, and SURF, etc. [268] [151], although these systems are also useful in many cases. The key result of the system is not restricted to the final classification accuracy. Providing a scoring tool also helps to compare different input samples of the plant species visually by consideration of the obtained scores. We also provided a discussion on how the implemented system differentiates when the hardware components are changed for the training process. The experiment of training by both CPU and GPU proved the importance of the used hardware as well. Moreover, the application of the DNPRS is not limited to agriculture. It can also be used in different fields such as medicine, drugs, etc. as a fully-automatic plant recognition system. Due to the important position of the system in the industry of the future, we beat the barriers of using mobile robots in natural environments for accurate plant recognition.

Chapter 9

Mobile Plant Recognition Robot (Real-time Application of the DNPRS in Challenging Outdoor Environments)

The horizon of agriculture's future is not clear. Many factors like climate change, population growth and soil infertility might have irrecoverable effects on different aspects of the agriculture. One important aspect of agriculture is plant recognition. It is very challenging in outdoor environments, especially in fields and unstructured places. To walk through a farm and distinguish plants automatically, it is essential to integrate an automatic plant recognition system with mobile robots. Over time, robots are taking up the responsibilities and activities of humans in different places, especially in the natural environment. As previously introduced, our intention is to build stable and reliable robots with the capability of the identification of plants. To develop such an agricultural robot, there are two phases. One is related to the robot and its features. The other is connected to the system capabilities for plant identification including recognition under unfavorable climate and environmental changes. An important component of such a system is undoubtedly the camera which should be mounted on the robot.

However, the robotic market is steadily growing in many fields, its growth is not covering in agriculture as it is expected and it should be broadened. In agriculture, many existing robots are substantially doing elementary operations and simple labor-intensive activities of the growing season such as irrigation, harvesting, seed sowing and pesticide spraying. Some activities like spraying and spreading pesticides are harmful if the chemical materials enter the human body through the inhalation, ingestion or absorption. Thus, it is necessary to replace the workers with the robots to avoid jeopardizing the worker health and safety. Another purpose of using robots in agriculture is to advance precision in this field. As many farming tasks are repetitive, implementing robotic technology contributes to increasing productivity and saving time and energy.

In order to design robot farmers, consideration of the agricultural environment is certainly important. Hence, challenges of the natural environment have to also be considered while designing an automatic plant recognition system. Ours is a four-wheeled robot and enables us to easily do plant recognition tasks in the natural environment. The control of the robot is done by joystick, although it is possible to utilize the robot in farms as an autonomous robot with unique features such as fast navigating through the crop rows, good stability in the agricultural environment, being multipurpose, etc. In addition, a semi-robot is proposed for real-time tests. We investigate both robot and semi-robot in plant recognition tasks.

This work has been published in the 2019 IEEE 15th International Conference on Intelligent

Computer Communication and Processing (ICCP 2019) [18].

9.1 Introduction

A portable plant recognition system has many realistic advantages and usages. While a family is visiting a park, the kids might be curious to know the type and name of plants. They ask the parents many questions and expect to get the right answers. Furthermore, mayoralty of a city may propose some plans for local plants of the city and surroundings. In addition to controlling the invasive species, his aim might be benefiting from energy conversion, reduced labor requirements, local ecosystem supporting, etc. Another approach of such a plan is to expand it to the citizens and invite them to participate. Botanists and scientist of other fields are also interested in recognizing plant species for their own purposes such as gathering data. Furthermore, a wide range of pragmatic farmers are keen to find all types of weeds in their farms. They want to prevent weeds from becoming established as they invade crops. Additionally weeds have negative influences on the productivity of farms, crop quality, quantity and amount of organic matter in soil, etc. To continue our discussion, firstly we would like to have a look at mobile robots.

Let's consider briefly some agricultural robots and their functions. Robots have penetrated the farms and gardens to carry out different agricultural tasks such as harvesting, weed control, sorting and packing, etc. For harvesting, there are several automated systems to perform this task. For instance, Agrobot [456] is like a tractor. The system works within rows, detects ripe strawberries and picks the fresh ones up without any contact. The ripeness of strawberries is determined by the cutting-edge graphic processing units. It has 24 arms and constitutes a team which is working wirelessly. The important part of this robot is its flexible platform; hence, it can be used for other farming purposes and configurations. Interestingly, there is an option to define if it is intended to remove stem or calyx during the harvesting process.

Another example is a project called the Asterix that sprays herbicides on only weeds without impacting crop plants and the ground through the use of machine vision and advanced patented technology that are out of the scope of our work. Weeding is an important task in farms. Several robots have been developed to carry out the task and limit exposure to herbicides. One of the proposed machines is Oz weeding robot [457] that has three different modes: manual mode, track and follow mode and autonomous mode. As a weeder robot, the robot takes care of weeding and saves farmers from the drudgery of this repetitive job. Moreover, this electric robot reduces the workload of farmers and contributes to the improvement of working conditions. The next agricultural robot that we would like to introduce is Vinbot [458]. It is an all-terrain robot which works entirely autonomously with a set of sensors. The sensors are useful for capturing images and data. Cloud computing applications help to analyze vineyard images and 3D data, find the yield of vineyards, predict yield accurately and send the obtained data to owners of the vineyards. Additionally, there are some other applications for robots to tackle the wide range of the tasks in gardens and farms. Robots can be used for sorting and packing purposes in agriculture. An example of such robot is the Pro Packing Robot [459] that was developed to fill cartons with fruits and vegetables [460]. It is equipped with a camera that helps to differentiate the sorted products.

The identity of agricultural robots has not been sufficiently investigated if we ignore the mobile robot for one important task in farms and gardens which is actually plant recognition. Plant recognition contributes to different related areas of agricultural management such as soil management, farm management, dairy management, animal management, etc. A lack of integration of agricultural robots with plant recognition systems is undeniable and there are not many robots for automatic plant recognition. In [461], a mobile robot was developed and applied for the mechanical control of weeds in outdoor environments. The vision system was composed of two sub-systems. The first sub-system recognizes the row structure formed by the crops [461] and guides the robot along the rows. The second sub-system is a color-based vision system for identifying a single crop among weed plants [461]. In [462], they proposed a method for plant species recognition by the use of a 3D light imaging, detection and ranging (LIDAR) sensor and different learning methods by means of the toolbox Weka [463]. They created a set of features, such as a mean of the reflectance values, maximum reflectance value, minimum reflectance value, etc., which are size and rotation invariant. They carried out an experimental evaluation of the best used learning methods. The results have been obtained for identification of plant species in the laboratory. The Asterix [464], an autonomous robot for automatic control of weeds in row-crops owned by the Adigo AS [465], classifies leaves by the use of color information and shape descriptors [466]. It is worth mentioning that the test has been carried out in a carrot field.

Precision farming (PF) is not a limited concept. It includes different farming tasks which target to improve farming operations and obtain ultra-precise information for the crop management. If we would like to use a robot for the purpose of precision farming, the ability of identification of weeds is necessary for the robot. The real-time application is also another side of this work. In [467], a CNN-based semantic segmentation of crop fields has been proposed. The approach helps to separate sugar beet plants, weeds and background information by using input RGB data. Moreover, the proposed system has been utilized on a real agricultural robot, Bonirob [468]. Due to the importance of deep learning and its great influence in different areas, the relationship between deep learning and robotics is getting closer. Ribeiro et al. [469] addressed a real-time deep learning based method for pedestrian detection (PD) [469] to the human-aware robot navigation problem by combining an aggregate channel features (ACF) detector [470] with a deep CNN. In the end, they achieved a fast enough performance. In [471], fusion of a CNN model with a feature-based layered pre-filter was applied to a mobile robot. It resulted in improving the precision and recall results of human detection. The experiment was carried out on two different robots with different GPUs. Neural networks can also be used for autonomous robot navigation in unknown environments as discussed in [472].

Nowadays, the implementation of deep learning models is not a desire. It has become more popular because of the availability of new hardware and the possibility of using embedded systems with limited resources. Furthermore, mobile robots have been developed and used widely in different fields as explained. But the lack of a mobile robot for plant recognition in real world environments is tangible. For instance, an autonomous robot for agriculture (AgriBot) [473] carries out different agricultural activities such as digging holes, putting seeds in holes, covering the hole with soil, applying pre-emergence fertilizers and herbicides along with the marking agent, communicating with another robot by means of Wi-Fi. It is known as a multi-purpose agricultural robot, but there is no operation for plant species recognition. Plant recognition in uncontrolled environments is usually referred as the identification of weeds and non-weeds by the use of robots [474] [475] [476] [461]. The missing part is to use a mobile robot for recognizing different plant species, not just the identification of weeds and non-weeds.

To compete with the difficulties of plant recognition in uncontrolled environments, we decided to utilize our implemented CNN model [91] for real-time tests and undertake the tests with two different mobile systems, a semi-robot and a robot. Deep learning has matured enough to be used in another new field, plant species recognition. It is able to fight with unsolved problems in real-time plant recognition and conquer the peak of related difficulties of recognizing plants in the natural environment. In this work, we address some existing problems of recognizing plants. The first issue is to use an automatic plant recognition system as a real-time system in the natural environment. Two different platforms, a semi-robot and a mobile robot, can benefit from the system and open a new window into agriculture and biology. The other problem refers to the used camera for taking photos as input images. We would like to build two systems which can be utilized independent of the used camera. In fact, we do not want to make any consideration about the camera used. Furthermore, generalization of the plant recognition system will also be provided from another aspect.

In addition, we would like to do the tests at different distances without measuring the distance between the camera and the desired natural plant species. Thus we do not care if the distance is more or less than 1 meter. In this way, we omit one factor which is usually considered in many robots and systems. Meanwhile, the CNN model works in the natural environment without any consideration about the time of day and the weather condition if it is sunny, cloudy, or windy. In agricultural applications, the used robot can be considered as a small robot. This robot is capable of the autonomous navigation in fields. It is also able to keep its distance from crop rows without crashing. Considering the physical aspects of the robot, current conditions in particular areas of plant recognition play a major role. The design of the robot is helpful and usable within farms with different situations and conditions due to some advantages in terms of cost, speed, accuracy and energy consumption in desired operations. It can be considered as the end product if we target the mentioned issues properly. Furthermore, using a semi-robot has economic justifications when there are insufficient facilities and equipment. Therefore, we decided to propose such a system.

The inputs of the CNN model are RGB images that will be recorded in natural environments. The identification of plant species will automatically be done without any pre-processing and pre-segmentation. The output of the plant recognition system can be used for providing a status report of mobile robot or semi-robot navigation through an uncontrolled environment. In addition, using a deep CNN model on agricultural robots is reasonable because of the importance of both of them in the today's world. The industrial market tends to use deep learning algorithms, and the market is thirsty for the agricultural robots. Hence, many startup technology companies are working on different systems based on the deep learning algorithms and developing robots for agricultural purposes. Many companies consider different challenges of the relevant issues such as food supply, lack of workers, the high cost of hiring experienced workers, complexities of farming activities, increase of greenhouses, etc. They attempt to provide for the rising demands of the agriculture field.

One important point is to increase the awareness of the robots and high-tech systems among the farmers and owners of the farms. To convince such people and break the barriers, it is necessary to create precise systems and support implemented systems over time. It should be pointed out that the elasticity of demand is high according to the obtained statistics about shipments of agricultural robots. It will increase in the next years from 32,000 units in 2016 to 594,000 units annually in 2024, and the market is expected to reach 74.1 billion $ in annual revenue [477]. Although we do believe that each region might seek its own demand. Furthermore, the plant recognition is not limited to only one region or one field, it is a demand for many farmers and owners of gardens and also people of other fields such as botanists, scientists in the pharmaceutical industry, etc.

The rest of this chapter is organized as follows: related works in section 9.2, system set-up and schematic in section 9.3, experimental evaluation and results in section 9.4, future work in section 9.5 and conclusion in section 9.6.

9.2 Related Works

In the content of plant classification, one necessity is to automate plant recognition operations. Another necessity is to commercialize a system which can be used in different places and situations. Different approaches and techniques have been proposed for plant recognition. They are mostly based on leaf analysis and identification of leaves. In [478], the focus was on the extraction of feasible characteristics such as shape, morphology, texture and color to obtain a set of features to recognize plant species. Arun et al. [479] proposed texture feature extraction to identify medical plants where the texture features included grey textures, grey tone spatial dependency matrices (GTSDM) [480] and LBP operators. In [481], Zernike moments were utilized to create a plant recognition system, although the Zernike moments are dependent on the scaling and translation of objects where their magnitudes are not dependent on the rotation of the objects [482]. The proposed approach in [483] was to use a contour-based shape descriptor, multi-scale triangular centroid distance (MTCD) [483] and dynamic programming. The best alignment between corresponding points of the shapes was found by the dynamic programming. In [296], using a combination of various features, shape, texture and color features was proposed for doing SVM classification of the plants. Cerutti et al. [298] used high-level geometrical descriptors for building a plant recognition system. The system that will be used in the current work has been proposed in [91] and a deep CNN model has been applied to recognize the natural plant species.

Investigation of the related works in the robotics field shows that most agricultural robots with recognition responsibilities are just doing classification task for two classes, for instance classification of weeds and non-weeds. In [484], normalized excessive green conversion, statistical threshold value estimation, adaptive image segmentation, median filter, morphological feature calculation and ANN were utilized for weed detection and recognition from crop plants. The images were taken by a field robot. Potena et al. [485] presented a perception system for automatic classification of crops and weeds. The input images were RGB+near infra-red and the classification of the pixels was performed by a CNN. A weed detection and classification method including the green segmentation and the feature extraction was proposed in [486]. The goal was to utilize the system for autonomous weed control robots which would be able to classify plants into crops and weeds. In [83], data was acquired from on-board sensors of the gardening rover, also called Autonomous Laboursaving Internet of. Things Veteran Energizer (ALIVE), and sent to a cloud storage platform. They used feature extraction algorithms, SIFT, SURF, ORB and the neural networks to distinguish plant species.

Despite all these important successes and contributions of the mentioned plant recognition approaches and robots, the problem still persists if we consider more realistic parameters and factors in the natural environment. The rationale behind the current work is to build interesting mobile real-time systems which are capable of performing the natural plant recognition task in outdoor environments without any consideration of the distance between plant species and camera, the type of camera, direction of wind, etc. From our perspective, we think that it is time to apply our deep learning approach at a practical level whereby the robot will be able to navigate an environment and recognize plant species from the viewed scene, not only from a single leaf with a specific background like soil. This is the greatest motivation for us in this work. It is worth mentioning that the proposed model in [91] is our rocket engine for the real-time application and its fuel is the natural dataset in used [91].

In sum, we make four key claims, which are the following: Our implemented systems are able to (*i*) accurately perform natural plant recognition for four various plant species with heavily overlapping leaves in different conditions and real world situations (targeting the correct identification of plant species without considering the used camera and the distance); (*ii*) act as a robust classifier that adapts well to different lighting conditions, changes of the original shape of leaves and leaf composition, ages of plants and backgrounds as well as weather conditions not seen in the training dataset; (*iii*) work in real-time on a regular CPU or GPU; (*iv*) be simply extended for the identification of more plant species.

9.3 System Set-Up & Schematic

We would like to explain two different mobile systems which have been used for our experiments where our recognition system is the same for both mobile systems. The mobile systems are actually a semi-mobile robot and a mobile robot, the Zephyr. The semi-robot system is shown in Figure 9.1, and three different cameras have been used for the image acquisition of the testing phase. The main component of the semi-robot system is as below:

- Intel® Core^{TM} i7-4820K, CPU @ 3.70 GHz, Installed memory (RAM) 16.0 GB and graphics GeForce GTX 760/PCIe/SSE2

The mobile robot is shown in Figure 9.2 and we just used one camera, a Canon EOS 600D, for capturing the images during the testing phase of the mobile robot.

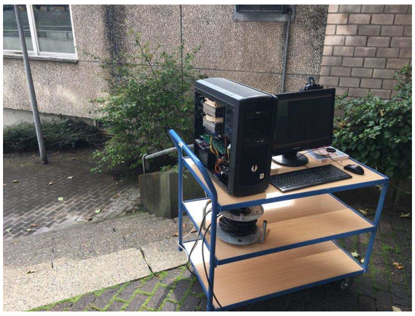


Figure 9.1: Representation of the semi-robot system

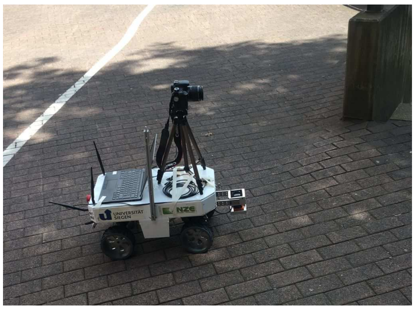


Figure 9.2: The mobile robot, the Zephyr

9.3.1 Image Acquisition and Cameras

In order to capture the images, it is possible to use a camera without the need of a human. This can be considered as the main advantage of using an automatic system for the image acquisition. Using such a system makes the image capturing convenient and natural and guarantees input for the automatic plant recognition system. If we want to take the pictures in the controlled lighting environment, we are able to utilize external lighting equipment and solve the related problems by using approaches in light engineering. In an uncontrolled environment, capturing images happens in a natural outdoor setting. Hence, there is no control of temperature, illumination, light intensity, dust, etc. The efficiency of the plant recognition system depends on the independence of the camera used. Consequently, the system will be generalized. Samsung Galaxy Note 4, iPhone 6s and Canon EOS 600D are the used camera for taking the pictures.

9.3.2 Agricultural Mobile Robot-Zephyr

The Zephyr robot is a small agricultural robot of the Institute of Real-time Learning Systems, University of Siegen, Germany. It has been participated in different competitions. It is possible to

use it as an autonomous robot and control it by joystick. The significant property of the robot is its high speed navigation in fields. To fulfill our purpose, the developed deep CNN model is run on a laptop with the following specifications and the plant recognition system is executed in the CPU mode of the Caffe framework during the test with the Zephyr:
Dell Latitude E7240, Intel Core i5-4300U (2x 1.9 GHz / 3 MB Cache / 64-bit/ 8 GB RAM 128 GB)

9.4 Experimental Evaluation and Results

In this section, the ultimate goal is to demonstrate the applicability and reliability of the deep CNN model in [487] [488] by carrying out different real-time tests in the challenging outdoor environments. Prior to the current real-time investigation, the model was tested after the completion of the critical component of deep learning algorithms, the learning step. The results of the experiments were presented in previous chapter. The interest in deep learning models is continuously increasing. We would like to apply the deep model elaborately in a real-time test and create an intersection between a deep model for the plant recognition and a group of the mobile semi-robots and mobile robots. In addition to using the deep model in the real-time test, there are other important objectives. We want to examine and analyze them through the new experiments.

One main objective is to examine the deep plant recognition system at different times and on different days. The deep system is trained by the modern dataset created in 2015. Our real-time tests have been carried out in 2017 and 2018, a two-year test. Researchers mostly test their classification systems at the same time as developing the systems. But, our intention is to carry out the tests in different years and times. The effect of time is undeniable on experiments and results. We accept this challenge and continue our journey despite the difficulties of the change of time.

Another objective is to investigate the effects of the camera on the mobile plant recognition systems. We examine the model by using three different cameras, the Samsung Galaxy Note 4, the iPhone 6s and the Canon EOS 600D. The deep model is trained by the images of the modern dataset where the distance between the camera and the plant was considered as an important factor during capturing the images of the plants. We measured the distance accurately to have a set of images in each defined distance. For the real-time test, we are not concerned about the distance. In fact, the images are taken at close and far distances without any additional consideration of this factor. Furthermore, the real-time system is tested in different weather conditions such as sunny, cloudy, windy, etc. There is also another objective that we have considered. It is the goal of connecting our scientific research and work to industrial applications in agriculture, medicine, botany, etc.

A human is looking at a leaf of a plant. Surely, he perceives the structure and shape of the leaf if the light intensity and illumination is adequate. If he is looking at a portrait, he is able to count people in it, but what happens if he looks at a plant with a bunch of leaves. We would like to know if he is able to count the leaves of the plant. Figure 9.3 shows a sample image of a plant with plenty of leaves.

As we see in Figure 9.3, a human cannot count the leaves of the plant completely. As many leaves are hidden behind others, it is impossible to count them correctly. For a machine, it is significantly more difficult to count the number of the leaves especially if it is not easy to identify the shape and structure of one leaf. The complexity of the background has an effect for identifying the plant species. One crucial point is that we conduct the real-time experiment on the semi-robots and mobile robots, but we combine the final results for the further evaluation. In addition, the real-time deep system can be run in two different modes, CPU mode and GPU mode. The GPU mode is used for doing the test by our semi-robot and the CPU mode for the test by the use of the mobile robot.

As we have used three different cameras, we show the number of the images taken with each camera, and Table 9.1 represents the related information.

Figure 9.3: A plant with plenty of leaves

	iPhone 6s	Samsung Galaxy Note 4	Canon EOS 600D
Acer Pseudoplatanus	23	0	7
Amelanchier Canadensis	10	0	20
Cornus	16	3	11
Hydrangea	19	4	7

Table 9.1: Number of pictures taken of plant species with each camera

The accuracy of the mobile test is equal to 84.17%. Table 9.2 shows the confusion matrix of the real-time plant recognition results during different times, weather conditions, distances and using different cameras.

Precision and recall are two important parameters which can be extracted from the confusion matrix. Figure 9.4 and Figure 9.5 show the precision and recall measurements, respectively. The sequence of labels is 1, 2, 3, 4 and the members of this sequence are actually Acer Pseudoplatanus, Amelanchier Canadensis, Cornus, Hydrangea.

Real-time Mobile System	Acer Pseudoplatanus	Amelanchier Canadensis	Cornus	Hydrangea
Acer Pseudoplatanus	29	1	0	0
Amelanchier Canadensis	4	20	0	6
Cornus	5	2	22	1
Hydrangea	0	0	0	30

Table 9.2: Confusion matrix for the real-time experiment

The maximum value of the precision measurements is equal to 1 where the minimum value equals to 0.7631. Three out of the four precision measurements are between 0.8 and 1.

The measured recall values indicate that the change of the values is higher than this change in the precision values. The first and the last labels are equal to 1 which is the highest possible value

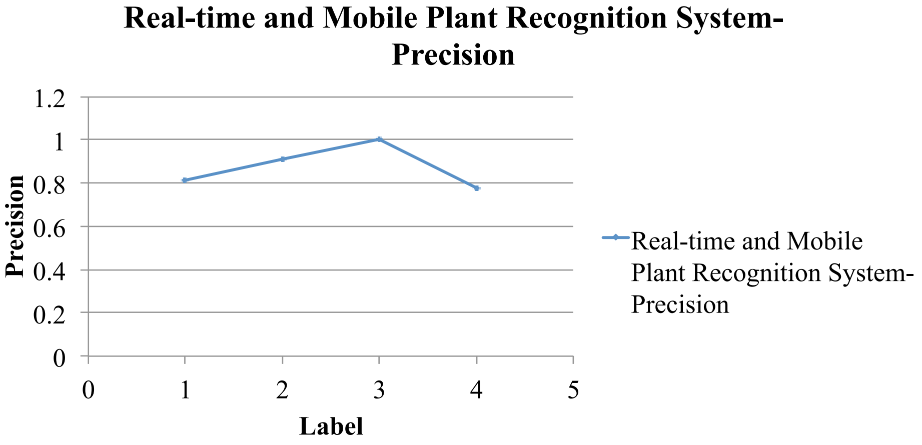


Figure 9.4: The precision measurements for the mobile test

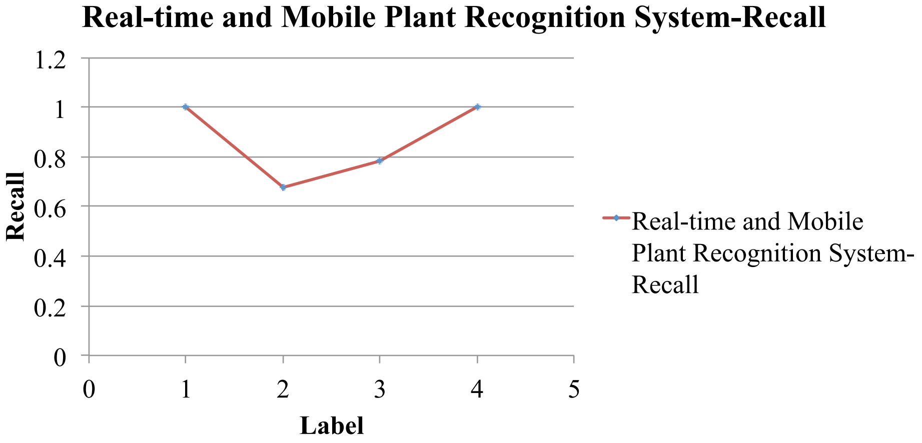


Figure 9.5: The recall measurements for the mobile test

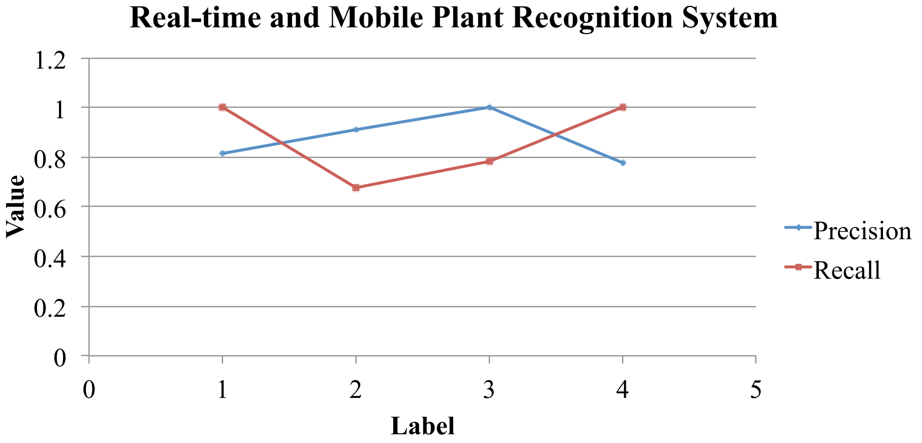
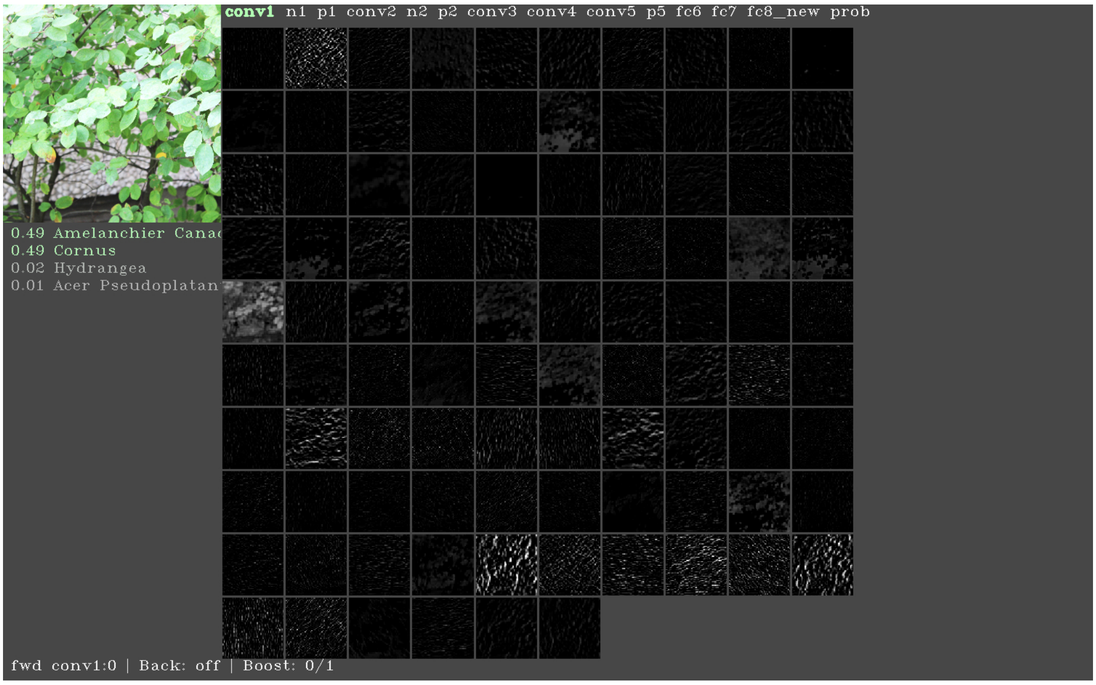


Figure 9.6: The precision and recall measurements for the mobile test

for the recall measurement. All values of the recall measurements are in one range, [0.6666, 1]. In comparison to the graph of the precision measurement, the area under the graph of the recall measurement is smaller. It should be pointed out that the greater area under the graph is the evidence



0.49 Amelanchier Canad
0.49 Cornus
0.02 Hydrangea
0.01 Acer Pseudoplatan

fwd conv1:0 | Back: off | Boost: 0/1

Figure 9.7: Importance of ranking in the recognition process

of good performance. It does not matter if the graph shows the precision or recall measurements. In addition, Figure 9.6 represents the recall and precision measurements in one graph. Therefore, we are able to compare the recall and precision values of the real-time system simultaneously.

During our test, one exceptional case happened and it is worth explaining in this case. If the deep real-time system recognizes a sample as two different plant species with the same percentage, the question is "What is the final result?"

Figure 9.7 shows the sample that is recognized as two different species with the same percentage. But, the first one is Amelanchier Canadensis and the second rank is Corus. According to the obtained ranks, the first rank is Amelanchier Canadensis and the plant species is actually Amelanchier Canadensis. As a result, the importance of the ranking in such cases is undeniable.

A new experiment has been conducted by cropping the input image. In this test, an original image has been cropped and four cropped images have been obtained. The cropped version of the original sample has different dimensions if we compare them with the original one. The dimension of each image has been obtained randomly. There is no special consideration about the dimension and the size. As we see, the cropped images have been predicted correctly, and the plant species are correct. Figure 9.8 shows the original image and the four cropped images.

Figure 9.9, Figure 9.10 and Figure 9.11 represent the predictions of the original image and the four cropped images which have been shown in Figure 9.8.

The system is an end-to-end system and it takes the input image and outputs the plant species. Our recommendation is to select the mode by considering the available equipment. In our tests, autofocus is a feature of the used cameras. As the whole process of plant recognition is automatic, we do not care about this feature if it happens or not. For instance, the needed time for capturing a picture with the Canon increases if the autofocus happens automatically. The implemented real-time plant recognition system works at different times of the day, weather conditions and distances.

One main contribution of this work is a new system to classify the natural plant species using the RGB data taken by one camera without considering the brand, the model and the type. In fact, we remove such a limit. The deep model is based on the CNN, and it is useful for identifying the plants in the challenging natural environments automatically. Due to the structure of the model, it is feasi-

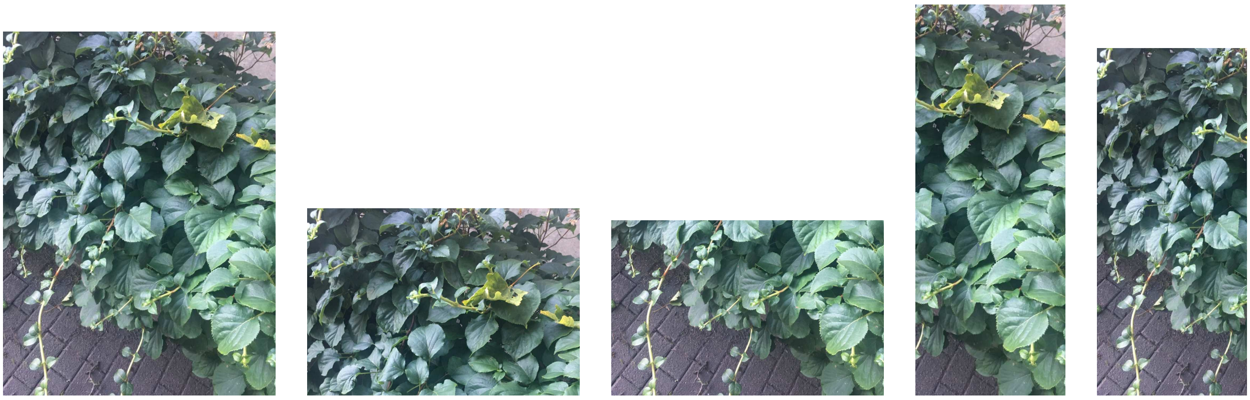


Figure 9.8: The original image and its cropped versions

ble to generalize it to many natural plant species all around the world. We aimed at recognizing the plants with the complex backgrounds in two different years, 2017 and 2018. It is one of the important achievements of the work which shows its applicability to be used for the industrial purposes.

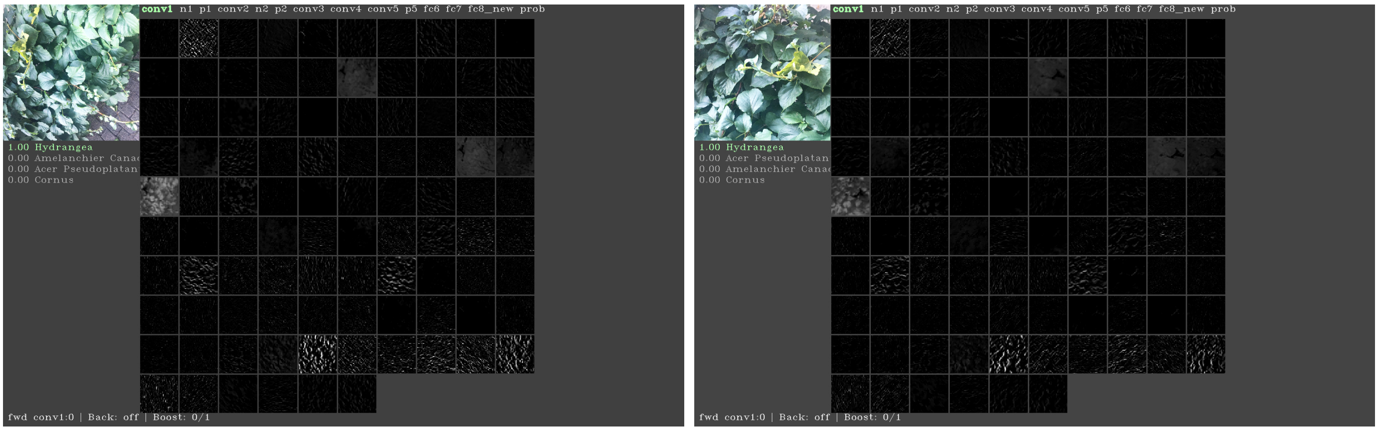
To our best knowledge, there is no similar mobile robot or semi-robot system which is able to carry out the plant recognition in outdoor and natural environments under the mentioned conditions. As explained before, the other mobile systems are mostly capable of identifying two types of plants in the outdoor environment, particularly weed and non-weed plants. Being portable is another additional contribution of this work where the whole software can be installed on a processor and used for all steps, from obtaining the real-time images (capturing the images) to getting the final results.

9.5 Feature Work

The future of this work is not only limited to the plant recognition systems. The mobile component can be considered as a part of the future work. We are able to design and implement a solar power system, a renewable clean energy source, for the mobile robot and consequently reduce the pollution. Such a robot can work autonomously every day, and there is no need for charging or replacing the battery. It is also possible to use it on cloudy days as the solar cells get the power from the sun, converting the light into electricity and storing energy.

From a business perspective, it is so important to reduce the overall cost of the system and increase its efficiency in some agricultural aspects and target applications. Hence, it could be interesting to add other features like a seed spreader and a fertilizer spraying device to the mobile system. There is an additional important point about the system and its real-time application. It should not necessarily be used only by a robot or semi-robot. It can be integrated into the other agricultural machines like tractors, which can be found on most farms. In such case, the ultimate cost of the system is less than the time that the recognition system is implemented as a component of a mobile robot. Due to independence of the system from the camera, it is not essential to buy a unique camera for performing the task of plant recognition. The camera of a cell phone can also be utilized.

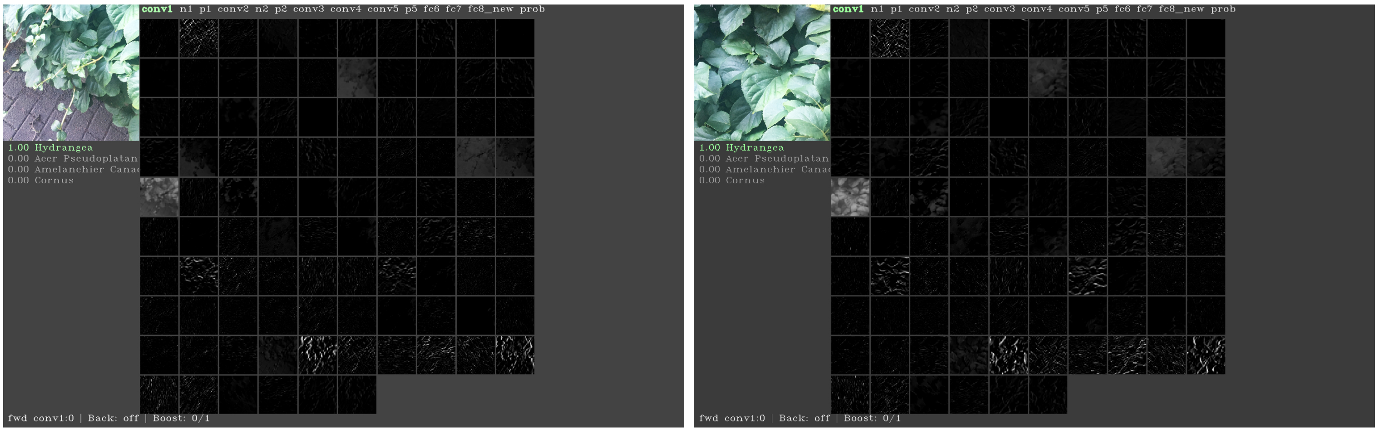
With regard to the natural plant recognition system, it is possible to take a sequence of images from one plant species. Then the system identifies the plant species in all taken images. If 90% of the results show one output, this output is certainly the type of the tested plant. Furthermore, we are able to extend the system for the other plant species if we obtain many images of each plant in the natural environment. Creating a connection between the results of the plant recognition system and the robot based on a semantic camera is also another possible future work. In this case, the output information of the recognition system can be applied in a semantic approach for the mobile robot


conv1 n1 p1 conv2 n2 p2 conv3 conv4 conv5 p5 fc6 fc7 fc8_new prob

```
1.00 Hydrangea
0.00 Amelanchier Canad
0.00 Acer Pseudoplatan
0.00 Cornus
fwd conv1:0 | Back: off | Boost: 0/1
```

```
1.00 Hydrangea
0.00 Acer Pseudoplatan
0.00 Amelanchier Canad
0.00 Cornus
fwd conv1:0 | Back: off | Boost: 0/1
```

Figure 9.9: The classification results for one sample image and its first cropped image


conv1 n1 p1 conv2 n2 p2 conv3 conv4 conv5 p5 fc6 fc7 fc8_new prob

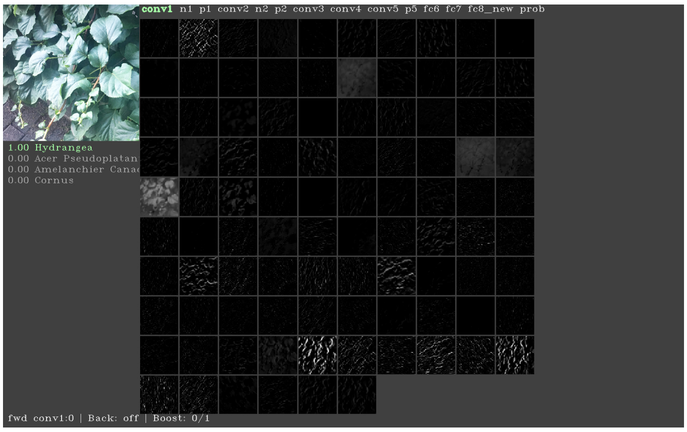
```
1.00 Hydrangea
0.00 Acer Pseudoplatan
0.00 Amelanchier Canad
0.00 Cornus
fwd conv1:0 | Back: off | Boost: 0/1
```

```
1.00 Hydrangea
0.00 Acer Pseudoplatan
0.00 Amelanchier Canad
0.00 Cornus
fwd conv1:0 | Back: off | Boost: 0/1
```

Figure 9.10: The classification results for the second and third cropped images

navigation tours.

As discussed before, the model is based on the deep learning concepts. It is unimaginable to separate the future of the system from the future of the deep learning area where its future depends on developing models closer to the human brain system. Meanwhile, it is stated that the future of the deep learning will be connected to the information bottleneck method which is described in [489] for the first time. Geoffrey Hinton, the godfather of the deep learning, declared that it may be the answer to a really major puzzle in the neural networks [490].


conv1 n1 p1 conv2 n2 p2 conv3 conv4 conv5 p5 fc6 fc7 fc8_new prob

1.00 Hydrangea
0.00 Acer Pseudoplatan
0.00 Amelanchier Canad
0.00 Cornus

fwd conv1:0 | Back: off | Boost: 0/1

Figure 9.11: The classification result for the last cropped image

9.6 Conclusion

Many different countries in various continents such Asia, Africa and South America are dependent on agricultural activities. There are many small farms where a lot of workers are laboring in different environments. One important point is the diversity of farming environments which are mostly unpredictable, especially in India and West Africa. To fulfill the need of plant recognition in such places, it is necessary to consider the potential of the systems based on different variables like cost, efficiency in challenging conditions, simplicity of the system for users, etc. The system developed in this research is an automatic and real-time plant classifier. It has been added to an autonomous agricultural mobile robot, Zephyr, and a semi-robot. In the real-time test, the accuracy of the deep system has been 84.17%, and the images were taken in the natural environment on different days, in different weather conditions, at different distances and by various cameras. The system is independent of the used camera for taking the pictures.

In general, it helps us to overcome the existing challenges in the plant recognition tasks. The system is portable and reliable for the automatic recognition of plants and the monitoring of plant species for managing crops in farms and natural outdoor environments like forests. The achievements of the technical research in this work are the provision of a user-friendly system which leads to robust results as well as short running time for mobile navigation and the building of a commercial real-time robot system for distinguishing the plant species without focusing on only one leaf of the target plant. The automatic real-time system is fully tested to identify four natural plants in two different years, 2017 and 2018, in Siegen, Germany. Evaluations of the results are not only based on the visual tests but also some useful experiments, such as confusion matrix, precision and recall measurements, etc., have also been detailed in this work.

Chapter 10

Conclusions

10.1 Summary

The thesis has addressed the problem of the plant species recognition and understanding the related difficulties. To investigate the existing plant classifiers, we addressed some major weak points of the current systems. The weak points have led them to be unable to deal with complex scenes of the plants in natural environments, the representation of the leaves, the changes of light in outdoor environments and other challenging factors like the time of imaging and weather conditions. Furthermore, some other difficulties may occur due to other factors such as the distance between the used camera and the plant or the viewpoint and angle of photographing. In addition, warmness or coldness of weather results in changes to the shapes of the leaves. Furthermore, the shapes of the leaves vary in day and night for some plant species. Therefore, the appearance of the plants changes and the plant recognition tasks become harder. Hence, it is so important to develop methods and build systems which are capable of handling the challenges in the real world.

To this end, we divided the plant recognition task into plant recognition in a controlled environment and plant recognition in an uncontrolled environment. We proposed six systems based on different approaches for the first environment. Then, we proposed seven systems based on different algorithms, especially deep learning.

Additionally, we built one semi-robot system and a mobile robot system which can be used on the farms and various outdoor environments as well. To evaluate the systems with different approaches, we also conducted different experiments. We compared the results from different points of view. Real robotics experiments were also carried out. To summarize, the proposed systems outperformed many current systems.

With regard to plant classification, the well-known commonly available datasets present controlled environments during photographing. When we look into a dataset consisting of the images of the leaves, which has been taken in a controlled environment, we find small changes among the leaves of one plant, and the images are usually taken with defined and constant background.

In Chapter 3, we pointed out that different datasets provide different plant images with various challenges, and we investigated the problematic challenges which are exactly a part of the natural environments. Moreover, we classified the current datasets based on the level and amount of naturalness. We consequently explained why we chose the used datasets.

By doing the image analysis on different samples of the datasets based on the histogram information, we started checking the effectiveness of the obtained information for matching and the classification of the plant species in Chapter 4. Although the obtained information is useful for comparing different samples, using only the provided information is not sufficient for our goal. The reason is that we expect to develop systems with the capability of identifying the plant species in a general manner. In fact, the generality is one of the most important goals. But as a comparison, the mentioned approaches in Chapter 4 can be used.

To improve the plant recognition, we step into the concepts of the modern keypoint detection and extraction algorithms. We investigate some modern algorithms such as FAST, HARRIS, SIFT, etc. We conducted some experimental tests on a few sample images to check out the performances of the algorithms practically. In addition, to a survey on keypoint techniques, we extended our investigations into another important area of image processing, the matching. In our case, we defined one reference plant image and a target plant image. We applied the modern (keypoint) algorithms as the basis of the matching process to investigate the corresponding features and the reliability of the used algorithms.

The turning point of Chapter 5 is introducing the modern combined detection and description methods such as the HARRIS-SIFT, the HARRIS-SURF, etc. Furthermore, this part fulfilled our ambition for combining different detection and description algorithms.

The proposed approaches in Chapter 5 have been utilized as the base of the implemented systems in Chapter 6. For the aim of the plant identification, we suggested systems based on the combined modern detection and description methods and the BoW technique. Our effort was to model the very rich information of the images in the best way. The difference among the modern combined methods led to having different systems with various characteristics and performances for the recognition of a large number of plant species. In this work, the local features extracted from the images with single leaves and the SVM algorithm was applied for finalizing the classifiers. Consequently, the resulting plant classifiers yielded recognition accuracies over 80%. Especially, the accuracy of the system based on the accuracy of the system based on the SURF detection and description reached more than 92% for more than 30 different plant species.

In addition to the robustness of the systems, the runtime test proved that the systems were fast enough for real-time plant recognition. As the implemented systems can be divided into two groups according to the used description methods, we could investigate the systems of each group separately. For instance, if we look into the results of the systems with the SURF description method, we find an important fact. The system that used the FAST method detected more features and delivered more robustness. Furthermore, this system did not pay too much computational cost for the detection of the features. There is also a good trade-off between the obtained accuracy and computational time in the system based on the SURF detection method. Furthermore, all implemented systems recognize plant species automatically. Hence, there is no need for user action and modification. Nonetheless, the performances of the proposed systems depend on the known scenarios of both detection and description methods. Thus, it is recommended to prioritize our needs first and choose one of the proposed plant recognition systems based on our anticipations and needs.

The next phase was to design and implement recognition systems for natural plant species in outdoor environments. This work was conducted in Chapter 7. This chapter addresses the problems of the state-of-the-art and the lack of a recognition system for plants under natural conditions. The emphasis is placed on building a reliable system for natural plant identification and compensating the gap between the current systems and the real-world needs. We started to examine pre-processing techniques and their applicability concerning whether they can be utilized in the identification systems as the starting step. As the self-generated modern dataset consists of color images, we found different changes in the images of one plant species. It was a big challenge to classify four different plant species in different sunshine and weather conditions, distances, and times of photographing, even when all images were taken in shining or shadow conditions. This task was challenging because other parameters affected the images. Despite the various changes of the scenes in the images, we attempted to find a general solution to decrease the effects of the changes and even to remove the unwanted factors to prepare natural images for the classification task. Due to the variety of the changes, it was not feasible to enhance the images and have them in the same conditions and formats. However, it was possible to divide the images into several groups mainly based on one factor. For instance, it would be possible to divide the images based on the light intensity and illumination. But, some factors did not exist in a number of images and such images could not be set in a group. The existence of many factors proved that it was impossible to have a pre-processing step which could work efficiently under any condition.

We continued our work by developing six different systems based on the BoW model and the local feature detectors and descriptors. Thus, the basis of all systems was the BoW model. The training process was done by SVMs. Different SVMs were used and tested as a part of the implemented systems to judge which one was more efficient and applicable. The system based on the SIFT algorithm obtained the highest accuracy of the classification: 94.94%. They outperformed the other proposed systems in terms of accuracy. By defining our expectations from the system, the other proposed systems were also useful and reliable. For instance, we are able to choose the systems due to the expected runtime. Despite the impact of many factors and parameters like angle, point of view, illumination and light changes, and wind effects, we yielded acceptable and high recognition accuracy in all proposed systems of over 89.99%.

For the next stage, we explored our goals in a new machine learning field. Finally, we explored the applicability of the very modern method of deep learning for natural plant classification. The intention was to check its generality, reliability, stability, etc. In Chapter 8, we developed a new system based on deep neural network concepts. An important advantage of the proposed system, called the DNPRS, was to increase the usability of the system for real-time purposes. The developed system based on CNN put us in the correct road towards a real-time and mobile system for automatic natural plant recognition in outdoor environments. The performance of the DNPRS was interestingly outstanding, and the system achieved an extremely high accuracy of 99.5%. Although the accuracy was not the only important factor in concluding whether the developed system was efficient enough or not. Hence, different experimental evaluations were conducted to compare the previously proposed systems to the DNPRS and the results proved the better performance of the DNPRS in different aspects.

The last chapter, 9, explored the final goal of the plant recognition task which focused on developing a mobile recognition system with the capability of natural plant identification. The developed system was used by a semi-robot and a mobile robot for navigating through outdoor environments and recognizing plant species. The mobile robot could be utilized autonomously and carry out the task of plant recognition. The test of the systems was not done in different years. This meant that we performed the experiments on completely new data in addition to previous challenging factors. In a new year, the growth of leaves might be affected by some environmental parameters and the shapes of leaves could be changed as well. Furthermore, we did the tests using different cameras and proved that the DNPRS could work independently from the used camera. The conducted experiments and results were evidence of the efficiency of the developed system based on deep learning concepts.

In a nutshell, the goal of this work is to develop and implement new automatic systems that enable the plant recognition in both natural and non-natural environments. To explore the goals and findings, various experiments have been conducted for each proposed system and the results have been compared. Since one main challenge of the plant classification is the capability of identifying many plant species, we developed six different systems in [169] and [170] which successfully classified a large number of plant species with high accuracy. In [169] and [170], it was the first time that modern combined methods were introduced for recognizing plant species. The images were captured in the controlled condition without any change in the light intensity, illumination, background, etc. One way to interpret generality is to increase the number of plant species. The developed systems in

[169] and [170] benefit from this characteristic. In order to recognize plant species in uncontrolled environments, six automatic systems based on the modern combined detection and description methods were proposed in [151] and [268]. Although the systems were able to recognize plant species in the presence of environmental factors such as viewpoint, angle, light intensity, brightness, position of light source, background, weather condition, etc., they were still dependent on the pre-information about a non-environmental factor, the distance between the camera and the plant species. The next stage was to design a natural plant recognition system called the DNPRS for the recognition of natural plants in different weather conditions, time of day (morning, noon, afternoon), various backgrounds, short and long distances, etc. [91]. The DNPRS is a deep neural network-based system which can be employed in the presence of environmental and non-environmental factors. Due to the possibility of extending the DNPRS, it was utilized as the basis of a mobile real-time plant recognition system in [18]. This new mobile real-time system provided a degree of freedom for selecting the camera for capturing natural images in uncontrolled outdoor environments. Furthermore, it was applied in two different mobiles systems, semi-mobile robot and mobile robot, with the possibility of using two different modes, for agricultural applications. It is noteworthy that this well-developed system was tested over two years without concerning the distance between the camera and the plant species. In the following, we find a shortlist of the major contributions of the DNPRS:

- Working in different environmental factors such as light intensity, brightness, position of light source, background, weather condition, time of day, etc.
- Being independent of the selected camera for taking pictures of natural plants.
- The ineffectiveness of non-environmental factors (viewpoint, angle, distance between the camera and plants, etc. during photographing) on the system performance.
- Possibility of being used in different field robots as a real-time system for recognition of different plants in natural outdoor environments.
- Usability with lack of hardware (existence of two modes, GPU and CPU).
- Possibility to generalize the DNPRS to identify more plants.
- Being a portable plant recognition system.

10.2 Direction for Future Work

The increase of the world population has an influence on many other parameters and leads to new needs and necessities. For instance, there is a huge increase in the demand for cereal production and rice supply [475]. Consequently, there will be a challenge to secure food for the whole world. Without any doubt, climate change also creates a new battle. Many people in undeveloped countries might be impacted by the shortage of food. Moreover, another undesired effect of the climate change is the variation of the rainfall patterns. Compensation of rainfall is another challenging problem. Thus, we need to equip humans and automatic systems with plant recognition systems to meet the challenges of the future and overcome these difficulties.

This section mainly investigates the future trends in research. The modern classification-based methods for leaf recognition or natural plant species classification (see Chapter [6, 7]) are feature-based. However, the used datasets and images are completely different, and there exist various concerns. The accuracy of such systems are generally high as the modern features are used to train the classifiers. As a result, there is usually a trade-off between the accuracy and runtime.

It is feasible to apply faster components to reduce the processing time. The new engineering process extracts more features and information, so that the robustness increases. However, we need an appropriate equipment, like the GPU, to implement the training process of the deep learning model. Furthermore, the foundation of the model is deep CNN, and we have to pay its computational expenses.

If the users do not have access to adequate and useful equipment for new modeling (for instance, if users would like to increase the number of plant species and have a new system), it causes a big problem for finishing the training process. To tackle the problem, there are three possible solutions which came to mind by combining previously proposed approaches. The first solution is to extract features from the deep model and feed them into the SVMs. It is possible to get different features from the deep CNN model accurately from the input photographs. For instance, it is possible to extract the features from layer 4 or even fully connected layers and then use them for the training processes by the SVMs. The other solution is to apply the local features, extracted by the SIFT or SURF algorithms, as the input layer of the BP neural network [491]. Furthermore, we are able to combine the VLAD approach and the deep model, a trainable end-to-end neural network architecture, and apply it as a new solution. This scenario can be used in two different ways. Our investigation showed that our first idea has been utilized in [492] and a generalized VLAD layers is connected to the CNN architecture. Then it is trained via a backpropagation process. The other possibility is to fuse the extracted features from the deep model and the VLAD vectors and get features with more variety.

There are other possibilities for the future work to focus on improving the performances of the proposed systems. One factor considered in the modern dataset is the weather condition. One possibility is to model the weather conditions and reduce noise before feeding the images into the developed systems. For instance, we see small drops on the leaves of the plants while the weather is rainy. To enhance the images, it is useful to remove the drops and reduce the effects of them in the images. We can target the restoration of an image by approximating and decomposing the weather in the scene which can be inspired by [493].

In [494], the goal was to recognize and then remove the shadows from the monochromatic natural images, and the proposed learning-based approach could be helpful to reproduce the high-quality shadow-free images. Another feasible solution is to add an extra part to the proposed plant recognition system for removing the weather condition influences. In [495], they tried to remove the weather effects from the monochrome images, although it was declared that the methods could be applied to the images captured in RGB format. They focused on the problem of the poor contrast for the images taken in bad weather conditions. They proposed a fast physics-based method without any need for a priori weather-specific and scene information to compute the scene structure. They could restore the contrast of the scene from the two or more images taken in bad weather condition.

Another work has been also proposed in [496] and the goal was to find a mathematical model for the fog by using the deep neural networks and remove the fog for the enhancement in advanced driver assistance systems (ADAS). If we would like to use such an approach to reduce the effects of the bad weather conditions, two extra parts should be added. One part should be added to identify the type of the weather for a sample input image, and an efficient approach is the used to remove the bad effects of the weather condition. Afterwards, it is possible to feed the input into the proposed plant recognition system for recognizing the type of the plant. It should be mentioned that such operations will be computationally expensive. In addition, there is also another idea to classify the images due to the weather conditions and then create new systems based on various datasets where each dataset contains the images of one weather condition. Although adding a pre-processing step for the classification of the input images according to the weather condition is time consuming, it is feasible to achieve higher accuracy in natural plant recognition tasks by dividing the training process and creating a model for each weather condition. Furthermore, the optimization of such a system is a challenge and decreasing the processing time should be considered to try to provide a good trade-off between high accuracy and runtime.

In Chapter 9, we built real-time mobile systems and conducted new experiments in different years. Our image acquisition is stationary, not moving, meaning that the robot stops completely. The used camera captures an image as the input of the recognition system. As the whole process is automatic, one important consideration is about the speed of the image acquisition. Although the deep system works independently from the used camera, it is critical to choose a camera which takes pictures quickly. Clearly, we can utilize a faster camera than the used cameras in capturing images. Our choice can be Optronis CamPerform - High-speed CMOS camera [497] which works properly in the real-time situation. This camera is capable of taking 1051 frames per second at its full resolution. Due to the importance of fast imaging, an investigation into improving the speed of the camera in the image acquisition is really considerable. Our goal is to break the real-time constraint.

Another future direction of this work could be to develop a web-based application on a server and ask several groups of German botanists to take the pictures from the plant species in natural environments over a period of time, e.g. from May to October, and upload the pictures by labeling and marking the exact name of the plant species. In addition, it is necessary to provide a precise protocol about the situations and conditions of photographing as we do not want to get any gap between our goals and the modern dataset which will be provided in the future. In this way, it is feasible to increase the number of natural images and add new plant species to the modern dataset. Another advantage is the possibility of mapping the biodiversity of these plant species in Germany over time. However, it is essential to have an expert researcher or scientist to check the correctness and worthiness of the uploaded images.

The future of the work can be put in the direction of different industrial areas, although other sections, like insurance companies, can use the systems for the accurate estimation of the plants and crops in the farms and the related issues. When such companies are able to recognize the plants in the field and calculate the number of each plant species, they can insure the crops due to the coverage levels and the needs. The plant recognition systems may help these companies to reduce inaccurate payments. Furthermore, an accurate record of the data helps companies to calculate the productivity of each field per year and compare this factor in different years. As a result, they would be able to prepare contracts due to the status of the fields, productivity, etc.

The last idea for some future work is to use the output of the system for a designed system by using other types of information. If we implement a system based on the resource description framework (RDF) [498], we can add the output of our system as the component of information when the mobile robot is navigating through the natural environment by the use of a semantic camera. In this case, if the robot sees a tree or plant, it can add its plant species as one part of the information and the proposed work in [499] can be extended. In fact, our proposed systems contributes to becoming closer and closer to a semantic camera and its respective technology.

Chapter 11

Appendices

11.1 Implementation of Several Pre-processing Algorithms

11.1.1 Canny Algorithm (Edge Detector)

The importance of edge detection is connected to information such as direction, step characteristics, shape, etc. that detected edges provide for us. Edge might exist between two leaves, leaf and background, area and area. In 1986, Canny algorithm [302] was developed by John F. Canny, and has become a popular edge detection technique as a multi-stage algorithm which consists of four different stages. Despite its simplicity, it has been used widely [500] [501]. Thorough deeply researching the Canny algorithm, we investigate the stages of the whole technique, then we carry out our test on several images.

An edge region can be defined as a region where there is a big distance difference between any set of sub-regions. Edge detection is always sensitive to noise and artifacts, hence the first stage of the algorithm is to reduce and clear the image from existing noise. A Gaussian filter (3×3 or 5×5) is applied, and the result is a smoothed image. The second stage is to do an intensity gradient of the smoothed image. In order to compute the first derivative in both horizontal and vertical directions, G_x and G_y, a Sobel kernel [502] is used in both x and y directions. So, we can obtain the gradient and the direction of the edge for each pixel as below:

$$Edge_{Gradient(G)} = \sqrt{G_x^2 + G_y^2} \tag{11.1}$$

$$Angle(\theta) = tan^{-1}(\frac{G_x}{G_y}) \tag{11.2}$$

Concerning the relationship between the gradient direction and edges, the gradient's direction is always perpendicular to edges and the direction of the gradient is rounded to one of four angles showing vertical, horizontal and two diagonal directions. We shouldn't forget that intensities change across the edge, not along the edge.

To form the Canny operator, the third stage is to investigate and check all pixels of the image whether they constitute any edge or not. To remove noisy pixels, each pixel is checked to find if it is a local maximum in its neighborhood in the direction of the gradient. This stage is called non-maximum suppression [503], which is an edge thinning technique. For instance, if the pixel A is on the edge in the vertical direction, the gradient's direction is normal to the edge and the pixel B and the pixel C are in gradient's directions. Therefore, the pixel A is checked with B and C pixels to find if it forms a local maximum. If so, it is considered for the next stage; otherwise, it is suppressed which means, it is put to zero. As a result, we obtain a binary image with thin edges. The last stage for fulfilling the multi-stage algorithm is Hysteresis Thresholding. This stage helps to figure out which edges are real and which are not. In fact, the current stage works like a filter. To obtain the goal, two threshold values, $minTher$ and $maxTher$, should be defined. If the intensity of the gradient of the edge is more than the $maxTher$, then this edge is surely a real edge. In addition, there are two other possible cases that we have to consider them. If the target edge is less than the $minTher$ value, this edge is discarded and defined as the non-edge. The last case occurs when the intensity of the gradient of the edge lies between the $minTher$ and the $maxTher$. In this case, we consider the edge as a real edge if it is connected to pixels of a real edge; otherwise, it is not an edge.

In order to examine the Canny edge detector on natural images, a 3 × 3 kernel Gaussian filter is created. There are two other parameters, $minTher$ and $maxTher$, which should be determined in this edge detection process. We define and adjust the next proportion between the $minTher$ and $maxTher$ values.

$$maxTher = 3 \times minTher \tag{11.3}$$

Here, the maximum value of the $minTher$ is limited to 100; thus, the value of the $minTher$ may lay between 0 and 100. For instance, if we set the $minTher$ value to zero, the result is a binary image. Figure 11.1 depicts the obtained result when we put the $minTher$ to zero in the mentioned process.

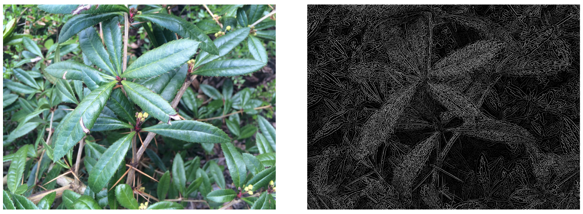


Figure 11.1: The original image and the result when the minimum threshold is set to zero

During the experiment conducted on images by means of the Canny edge detection algorithm, we decided to adjust the $minTher$ value for finding the best output. The final results of input images with the used threshold value, minimum threshold value, are shown in Figure 11.2, 11.3 and 11.4.

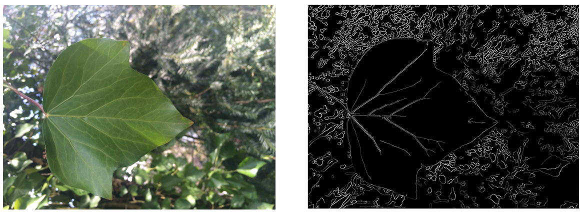


Figure 11.2: The result of the Canny algorithm with the threshold value of 34

The purpose of using an adjustment feature is to find the best result for leaf parts. One disadvantage of this method is the lack of finding a general threshold value which can be used for all samples. We also need to connect the resulting edges to extract the estimate and complete edges that seem so obvious for the human eye system and mind. One of the limitations of the algorithm

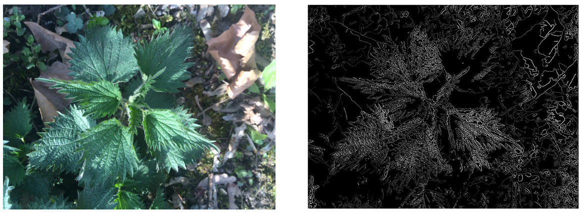


Figure 11.3: The result of the Canny algorithm with the threshold value of 30

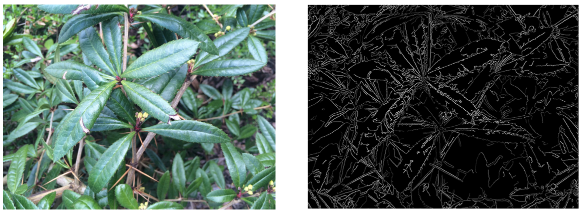


Figure 11.4: The result of the Canny algorithm with the threshold value of 48

is the binary result. Actually, we lose some information that might be valuable for next steps of the plant recognition. We have found another restriction of the method which is the dependency on the size of the Gaussian kernel, and the locations of the edges might be off according to the size of this smoothing filter. Another lack of the method is disconnectedness in some parts, corners and junctions, because the smoothing filter blurs them out. To summarize this part, it should be pointed out that the algorithm is not powerful enough in presence of noise interference, and it performs the detection step weakly. However, it finds the details of the natural image and provides basically thin edges. In our case, the Canny detector isn't able to detect useful information for all sample natural images. Leaves of natural plants can't be segmented efficiently, but the Canny detection algorithm yields surprisingly good results in medical images [504].

11.1.2 K-means Color Clustering

K-means is actually a clustering algorithm which can also be applied to obtain the most dominant colors in a natural image. The main goal of clustering algorithms is to divide the input data into k separate clusters and multiple regions, whereas each cluster might contain n data. The data of each cluster can be assigned to the center of the cluster by using the nearest mean. In fact, the data of the same cluster is assumed as more similar data if we compare the data of the cluster (a set of pixels) to the obtained data of another (another set of pixels). Since one of the main targets in segmentation is to entail the division or the separation of the image into regions of similar attributes, it is useful to apply a K-means color clustering method and extract a set of contours from the entire image. Therefore, we are able to fulfill the most basic part of the image segmentation which is its luminance amplitude for a monochrome image and color components for a color image. Furthermore, all pixels in a specific region are basically similar with respect to some characteristic or computed property, such as color, intensity and texture.

Due to the influence of complicated background and the diversity of natural objects in natural images, it is not possible to do segmentation easily. At this stage, we would like to examine if we are able to do clustering and isolate leaves. For initialization of the color clustering, we would like to make three partitions from the original images. Consequently, the number of clusters is equal to 3. We can also increase the number of clusters. According to the conducted experiments, smaller number of clusters (less than 5) gives better results. It is worth mentioning that the input images are color images constructed by three channels.

To begin the algorithm, we set the input natural image to 3 sets of samples. As goal is to find labels and centers, we apply the K-means. During the clustering step, the termination criteria is the maximum number of iterations and/or the desired accuracy. The defined accuracy is equal to 0.01, although it would be possible to reduce this value if needed. It means, "As soon as each of the cluster centers moves by less than 0.01 on some iteration, the algorithm stops." Five attempts are carried out, and the process yields the best compactness for the samples and labeling of them. The algorithm continues, hence the reshaping of the original image is performed according the obtained groups of pixels and the centers are finally mapped. Similarity between colors of natural leaves and backgrounds of sample images leads to fairly good results for the implemented K-means clustering. Figure 11.5 illustrates the outputs of the implemented algorithm for three natural images.

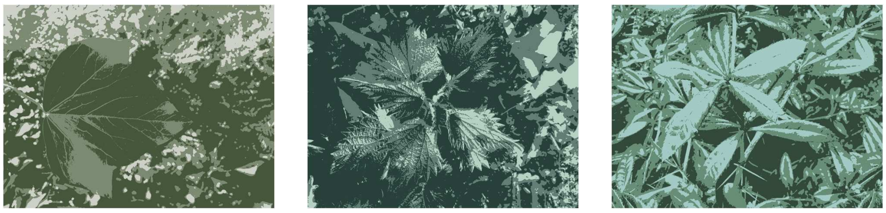


Figure 11.5: The results by using the K-means clustering algorithm

11.1.3 Implementation of Grabcut Algorithm

In this section, we propose implementation of an algorithm, called Grabcut [303]. This algorithm can be considered as an automatic method or graphical initialization method. To have an automatic process, it is essential to set up numerical parameters firstly, and then initialize the algorithm. One possibility is to define the center of the natural image as the target. As a result, the corners of the natural image would be considered as noisy parts without any benefit. This is exactly the dark point of this algorithm in challenging conditions that we are involved in the future. Furthermore, another possibility is to utilize an initial guiding shape in the pipeline of the algorithm, and we call it graphical initialization.

One solution to our problem is to segment foreground/background. Grabcut is the extension of the graph-cut approach [505] which is based on both local and global properties and satisfies the goal of object extraction [269]. The Grabcut is somehow the iterative version of the algorithm that simplifies substantially the user interaction needed for a given quality of result. This algorithm utilizes both edge and region information, hence the algorithm is equipped with a strong and powerful weapon. The obtained information is used to form an energy function which creates the best segmentation when it is minimized. One interesting part of the algorithm is to build a graph for representing pixels of the image as the nodes in the graph. Two important nodes are the Sink and Source nodes where the first one shows the foreground of the image, and the latter, the Sink node, depicts the background of the image. One important point is that each pixel node in the graph is connected to the Source and Sink nodes. In addition, the segmentation of the image depends on the separation of the Source and Sink nodes. The energy function plays the role of weights between the pixel nodes and also weights between the pixel and Source or Sink nodes in the graph. These weights are defined by the edge information in the image. Thus, a weak indication of an edge between two pixels (a small difference in pixel color) results in a very large weight between two pixel nodes. Determination of the weights between the pixels nodes and the Source and Sink nodes is carried out by the region information. These weights are calculated by determining the probability of the pixel node being part of the background or the foreground region. The next step, referred as the clue marking stage, is separating the foreground and background regions. Concerning this issue, some pixels in the image should be labeled before the segmentation as either the foreground or the background. We face a new concept in this step, and it is called the hard labeled. Any labeled pixel of this stage is set as the hard constraint. It means that hard labeled pixels cannot change their labeling during the segmentation process; therefore, they are condemned to have their labels without any change.

Afterwards, a Min-cut/Max-Flow algorithm [505], a graph cut technique, is used to do the graph segmentation. This algorithm is responsible for cutting the graph into two separating Source node and Sink node with a minimum cost function; thus, the minimum cost cut is determined by this algorithm. In order to obtain the cost of cut, the sum of all the weights of the links that are cut will be used. Due to the iterative characteristic of the algorithm, the process continues until the time that the classification converges. However, the iteration number should be defined by user. By separating the Source and Sink nodes, the connected pixels to the Source node are considered as the foreground, and the rest pixels are the background in the end.

One important point is to make the algorithm wholly automatic without any additional user interaction. Firstly, we calculate the size of each image. If we suppose that the image has columns and rows, a rectangle with the top-left vertex at (50, 70), of width (columns-150) and height (rows-180) pixels will be drawn and used for the Grabcut method. Another important point is to define the number of iterations for processing the algorithm, and this parameter is set to 10.

Figure 11.6 shows the outputs of the algorithm applied to the samples. The experiment is carried out by our machine with specifications of Intel® Core^{TM} i7-4790K, CPU @ 4.00 GHz, and installed memory (RAM) 16.0 GB.

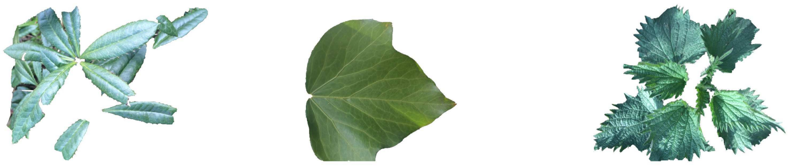


Figure 11.6: The results of the Grabcut algorithm for three samples images

As a summary, the Grabcut algorithm is actually a new algorithm for foreground extraction demonstrated in [303], which obtains foreground alpha mattes of good quality for moderately difficult images with a rather modest degree of user effort [303]. The system is a combination of hard segmentation by iterative graph-cut optimization with the border matting to deal with blur and mixed pixels on object boundaries [303]. Despite good results obtained by the Grabcut algorithm and nearly excellent segmentation, this method is time-consuming. We also centralized the rectangular window, and there might not be any leaf or the leaves might be outside the designed window.

11.1.4 Superpixel-based Segmentation Algorithm

Upon seeing an image containing a plant, a human has no difficulty for understanding the entire structure of the plant, even though the structure is a 3D one and there are a lot of additional objects in the surrounding. However, segmenting all regions of natural images remains extremely challenging for both human and current computer vision systems. Sometimes the human is able to guess the invisible regions and parts, but it is not easy for a computer vision system to estimate or guess invisible parts without prior knowledge. Indeed, in a narrow mathematical sense, it is impossible to separate all parts from the natural plant image taken in hard weather such as windy and rainy, since we cannot predict the effects of undesired particles. In addition, it can also sometimes be difficult to know if it is an outstanding painting of the plant or if it is a picture of a scene in the natural environment. Practically human perceives, and realizes remarkably well given just one image; and we want to give the computers this realization and make the computer-based systems closer to reality for the segmentation of the natural scene. Natural images have usually more details, therefore it is intended to investigate a new algorithm, superpixel-based segmentation algorithm, for natural plant images to know whether it would be possible to utilize this algorithm for obtaining perceptual important regions of the image because these regions are reflecting global aspects of the image.

In comparison to low-level visual processing like edge detection, segmentation approaches cannot commonly run at the same speed. One necessity is to do segmentation tasks in faster way. Using the superpixel image segmentation technique [304] solves the problem of timing in the segmentation. Furthermore, the technique provides relatively good segmentation, and the result obeys neither too coarse (to have too few components) nor too fine [304] [506] [269]. In addition, the technique does not impose an expensive computational cost compared to other segmentation techniques like the Grabcut technique. The implementation of the technique in [269] runs in $O(nlogn)$ time for n graph edges, and it is the same as the original one [304].

Figure 11.7 represents the results of this approach on our captured natural images. In our test, we keep one of the parameter of the segmentation algorithm, called sigma, fixed, and it equals 0.8. Another parameter, k, shows the greedy scale of the algorithm where the higher value of k, the large regions we expect to be segmented. The latter parameter, "Min_size", defines the minimum size for each segmented region. If a region has smaller size than the Min_size, it should be joined to an adjacent region. The specifications of the used machine are Intel® Core$^{\text{TM}}$ i7-4790K, CPU @ 4.00 GHz and installed memory (RAM) 16.0 GB.

Tuned parameters:
k = 50
Min_size=100
Channels = 3

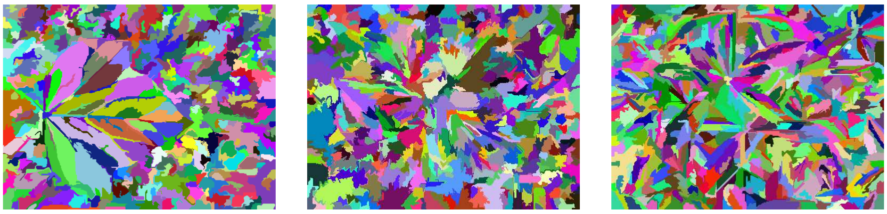


Figure 11.7: The results of the superpixel-based segmentation algorithm on our captured natural images with the number of segmented regions and the processing time, (Left) the number of the segmented regions = 426, (Middle) the number of the segmented regions = 484, (Right) the number of the segmented regions = 553

Looking at an original natural image of one of the plants, we find more complexity in the image, and the result of the implemented algorithm contains more segmented regions which are small regions of interest. Each region seems to be approximately homogeneous in color, and it is completely true in reality. In addition, our investigation of the results proves that a uniform region is usually extracted in a larger size compared with others containing textures and edges. Nonetheless the algorithm performs well and efficiently, it captures nonlocal properties of images. Although nonlocal properties can be as useful as local properties in some cases and the running time of the algorithm is fairly good, matching many small regions is not simple. Seeking useful regions in each image separately is also not an easy task. A carefully reading one must raise the question of how to apply the small regions in which there are interest segments of leaves. In order to answer the question, we have to determine which segmented areas are the regions of interest. Seeking the solution that fulfills the missing point, we first start with applicability of the algorithm. It is applicable if we combine it with the Grabcut algorithm. In this way, we are able to segment the natural leaf image to useful small regions, but this procedure adds more computational cost if we consider it as the pre-processing step.

11.2 Human Nervous System

Human nervous system is the second heart of the human body as it is the leader of human activities. In other words, the controller of the functions of the human body is the brain, and it uses nerve cells, neurons, to carry out different tasks. Neurons are spread through the whole body and formed a network. Interestingly, there is also interaction among the neurons in the whole body, and the other organs of the human body like ears and eyes play the role of receivers for obtaining the information. The whole nervous system can be divided into two main parts: central nervous system (CNS) and peripheral nervous system (PNS). The main components of the CNS are brain and spinal cord. Main components of the PNS, nerve cells, are responsible for transferring the information from/to the CNS. Due to the importance of the functions of the nervous system, we firstly have a look into some main functions, and then we investigate the human visual system.

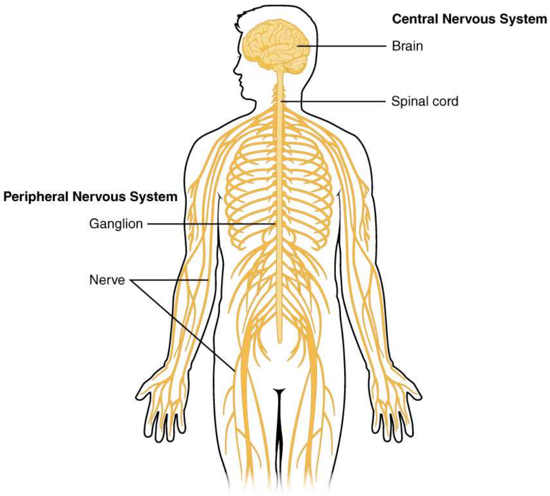


Figure 11.8: Overview of the nervous system

In reality, we are not able to divide the system on the basis of the functions. However, we have anatomical and functional divisions. The main problem is how we can fit functional differences into anatomical divisions, because we have sometimes the same structure and it can be a part of various functions. Hence, it is very hard to have certainty in general. Let's explain this fact by an example. In human body, the optic nerve is responsible for carrying signals from the retina. The signal may be used for the conscious perception of the visual stimuli. In addition, they may be used for the reflexive responses of the smooth muscle tissue. Hence, it is not easy to specify the border of differences. To solve this problem, we need to change our standpoint; therefore, we try to divide it according to another basis. Our decision is to do division on the basis of basic functions: sensation, response and integration. Another option is to divide the system on the basis of the functional difference in responses and the control of the body, but we skip this basis and continue with the first proposed solution.

The same as many systems, the nervous system receives environmental information, and responds to the received information whereas the first part is called the sensation and the second one is named the motor responses. In addition, a third type of function exists, and it is called integration which is responsible for associating and integrating sensory information with other sensations which are higher cognitive functions like memories, emotion and learning. Hence, it is feasible to divide the nervous system into three major functions: sensory functions, motor functions and integration functions.

Human visual system consists of eyes, the biological camera, and a part of his brain and pathways for making connections. The brain part is responsible for all related image processing tasks which are really complex. The retina processes the neural signals, and the signals go via axons of the ganglion cells through the optic nerves. The signals continue their journey, and information are divided and crossed over the optic chiasm. Then, the signals go through the optic tracts to the lateral geniculate nucleus (LGN), and continue from the LGN to the place that the visual processing occurs. This place is the primary visual cortex.

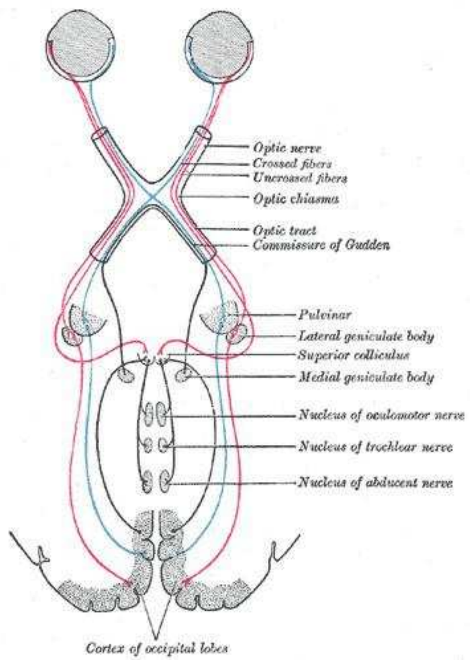


Figure 11.9: Visual cortex

The sensory functions confirm the presence of any change from homeostasis or a specific event occurred in the environment, known as a stimulus. The main senses are usually smell, touch, sight, hear and taste, and human organs are mostly the resources of gaining the senses. Due to characteristics of the taste and the smell, the stimuli of them are chemical substances. There is another type of stimuli, physical or mechanical one. For instance, touch is a physical interaction with the human skin, sight is the light stimuli, and hear is the perception of the sound that is a physical stimulus. Furthermore, there are additional sensory stimuli which can be considered as internal one like stretch of ligaments.

The basis of the nervous tissue is neuron cell, and the neurons are responsible for the electrical signals communicating information about sensations and responding to stimuli. Three dimensional shape of neurons provides the possibility of having a vast numbers of connections within the nervous system. In addition, the main part of the neuron is the cell body called soma. As we consider, neurons are mostly described as having one axon which is a fiber emerging from the cell body and projecting to target cells. An axon is able to branch for communicating with the targets and propagating the impulse of the nerve. Dendrites are also other highly branched processes of the neuron, and they are responsible for obtaining the information from other neurons at specific areas of the contact which are called synapses. From dendrites, the flow of the information is in one direction, and it goes through neuron across the cell body and also down the axon.

There is an action potential which propagates down the axon and it is the basis of the electrical signal in a neuron. In fact, a neuron is able to produce an action potential if it receives the input from another neuron or a sensory stimulus. There are also synapses which give permission to neurons to pass signals, that are either chemical or electrical in nature, among themselves or to target another cell which might be more effective. However, we have to mention that chemical synapses are far more common.

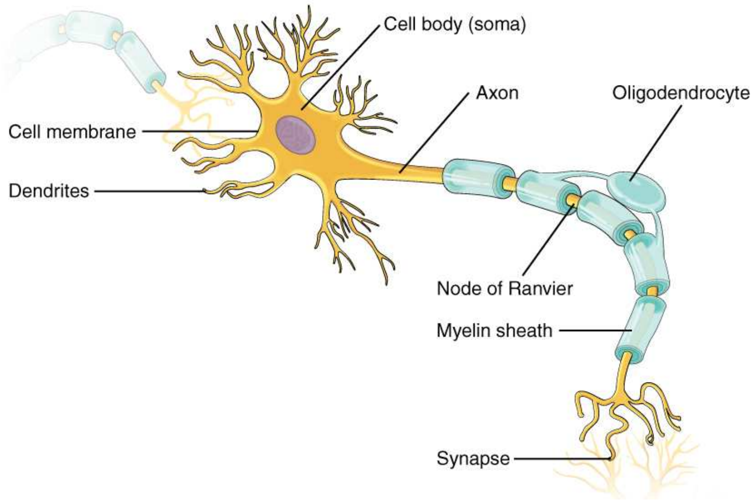


Figure 11.10: Human neuron

If we compare the artificial neural network to the biological one, there are some similarities as the artificial one is inspired by the biological network. In general, artificial neural network learns how to do new functions and it takes place by adjustment of the topology and weights. For the biological network, the learning process takes also time for human. Furthermore, the time that takes for a student to learn a new mathematical theorem depends on different parameters like genetics, previous background, etc. In biological neural networks, the learning is obtained from the interconnections between myriad neurons in the brain and the nervous system, and repeating makes the task easier for the student as the neurological connections become stronger. In the artificial neural network, there exist interactions between the nodes. By the end of learning process, the nodes and weights will be finalized.

11.3 Human Learning vs Machine Learning

In general, machine learning is a subset of artificial intelligence. The algorithms of this subset are rapidly growing and the speed of advances is greatly increasing. One cannot deny the huge progress of the machine learning algorithms. The term learning usually refers to the humans and the process of the human learning, but it has been used for machines in the artificial intelligence as well. Firstly, we would like to have a look at the learning concepts related to the human brain and psychology.

Then our goal is to compare the human and machine learning.

The human learning is a unique process in nature and it can be divided into different levels. A child may learn something new by memorizing it. For instance, a child is able to sing, if he/she memorizes the lyrics of a song. In addition to the memorization, the ability of remembering is also another aspect of being capable of singing.

Learning and understanding are close concepts and they might be wrongly considered as the same process. Understanding is only one of the learning levels. The other side of the understanding that might happen is misunderstanding. When a child enters primary school, one important part of the learning is to understand the facts even if the fact seems to be simple. Understanding helps the child to compare the main concepts and ideas. The next level is applying what has been learned. If a child learns how to sum two numbers, he/she will be able to apply this knowledge to new questions for summing two other new numbers. Analyzing this mathematical task is also another part of the learning. Therefore, humans are able to analyze new tasks and examine the components of the problems when learned.

Apart from the analysis, another level of the learning is evaluating the problem. The evaluation process consists of making judgment and comparing the facts based on some other concepts and standards. The last level of the learning is the ability of combining different concepts and ideas, and creating a new problem. Furthermore, an important part of human's creativity is related to human's learning. In this level, a child is able to solve a mathematical problem through the use of the alternative methods and by proposing new solutions. The proposed examples can be expanded in other scenarios too. Additionally, one point should not be forgotten "Practice is an important factor in the human learning."

As discussed before, the first idea of learning is to imitate the human learning and provide different levels of learning. The neural networks have helped us to get closer to the desire of learning and obtaining the human behavior and activities in machines. Basically, the components of the neural networks are not adaptive or self-organizing. Thus, the algorithms start to teach the networks how to perform the desired task correctly and responsibly. Although it is very hard to achieve all levels of the human learning, a main goal is to build knowledge-based machines. The final goal is to use the machines and the related knowledge in appropriate ways. If a machine learns the human activities, it is possible to save the human time and effort by replacing them with machines. In addition, it is also possible to have online access to the obtained knowledge and gather more advantages from relying on the machines in the real life.

In order to use and teach the machines, they need the human learning and knowledge, and a combination of the human and the machine learning makes sense. For instance, we would like to build a machine which can be used in remote surgery. To do the process learning of such a machine, it is essential to combine the human knowledge and the human learning. Therefore, the machine learning is not completely independent of the human learning.

The human brain does not work the same as computers, which are actually digital. Computers' language is basically composed of only two digits, 0 and 1. However, we are able to compare them through the consideration of the status of the neuron. For instance, we can suppose a triggered neuron as 1 and a non-triggered neuron as 0. Furthermore, the human memory is unlike the machine memory which has a limited capacity. For instance, RAM is similar to the short-term memory of the human. However, the human memory does not have a fixed capacity while some parameters, such as expertise, experience and familiarity, may affect the differential capacity from one person to another. Furthermore, a machine is usually not able to repair its components if one part is not usable, but the human body and its organs help to cure a damaged component in many cases. Although we find similarities between humans and machines, they are still very dissimilar in many aspects. Researchers and scientists are still doing their best to go up and increase the likeness of the machines and human in different aspects and areas.

11.4 Feedforward Neural Network

Traditional neural network reckons on shallow networks [507] and its simplest and the first type of it consists of 3 different layers, one input layer, one output layer and mostly one hidden layer between the first layer and the last layer, input layer and output layer, respectively. If the main architecture of the neural network has more than these three layers including input and output layers, the network is qualified as deep learning. A neural network with three layers is named feedforward neural network if the data moves only in one forward direction [508] which means the movement of the input data through the hidden layer and ending in nodes of the last layer. In this type of neural network, we find neither cycles nor loops. Figure 11.11 shows architecture of a feedforward neural network, and it is observable that there is only one direction of the data flow.

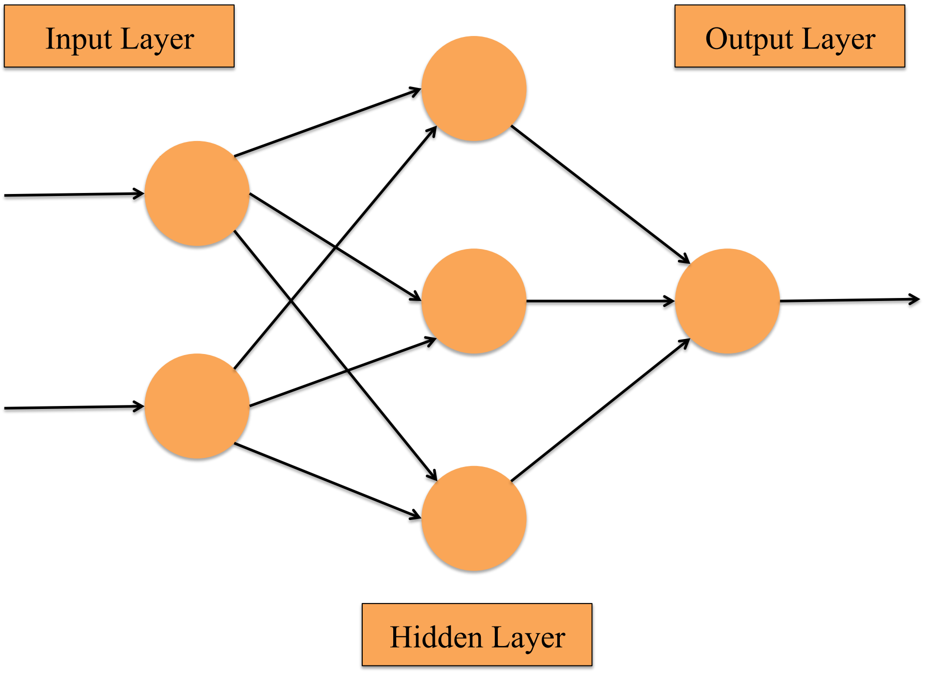


Figure 11.11: A part of a simple feedforward neural network

It should be noted that shallow neural networks usually refers to neural networks with only one hidden layer, and deep neural networks are usually neural networks with several hidden layers; therefore the difference lies in the number of intermediate layers.

11.4.1 Definitions of Concepts

In this section, we would like to introduce some concepts and terms which might be used in deep CNN models.

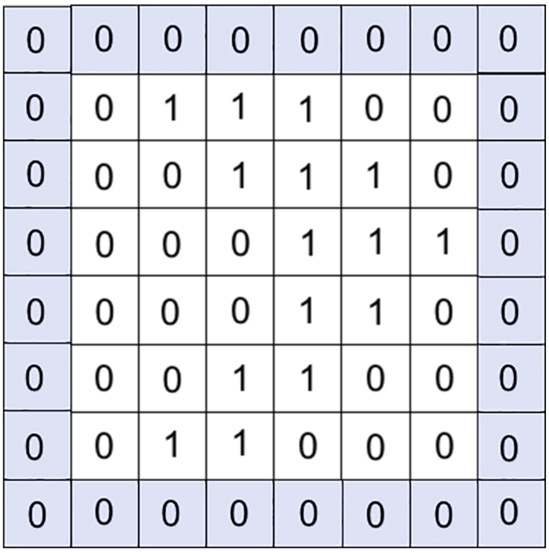
The first term which has been explained before is activation map. It is also called convolved feature and feature map, and it is the output volume which is obtained by sliding a filter over the input plant image and computation of the dot product.

The next term that might be utilized is "Depth" which is the number of the applied filters.

Fibre, called also depth column, is a set of neurons pointing to the one receptive field.

Another important term is stride, and it is responsible for defining the intervals that we should apply the filters to the input. It produces spatially smaller output volumes and its common value is 2. In order to set the stride, it is necessary to obtain an integer value for the output volume, not a fraction. For instance, we set the stride to 2, the filter will then shift by 2 pixels when it convolves with the input volume.

Zero padding is a process of adding zero to the outside of the input volume, making it ready for convolution operations and obtaining finally the same number of outputs as inputs. Without the zero padding operation, we lose some information of the outer part of the volume after convolutional layers, and decrease of the size of volume destroys the performance of the deep model. Figure 11.12 shows the zero padding operation for a volume of information [509].



0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

Figure 11.12: Example of the zero padding operation

Due to the mentioned concepts, we would like to propose an example to show how we compute the output volume mathematically. The input volume is $[Width \times Height \times Depth]$ and the size of the receptive field is considered as F. Other parameters, stride, zero padding and depth, are S, P and K, respectively. The output volume, $[Width_{out} \times Height_{out} \times Depth_{out}]$, is calculated as follows:

$$Width_{out} = \frac{(Width - F + 2P)}{S + 1} \tag{11.4}$$

$$Height_{out} = \frac{(Height - F + 2P)}{S + 1} \tag{11.5}$$

$$Depth_{out} = K \tag{11.6}$$

11.5 Common Deep Learning Frameworks

11.5.1 Theano

The first deep learning framework that we would like to introduce and study is Theano [510] which is actually the first widely adopted framework for deep learning algorithms and a numerical computation library for Python. The creator of this framework is Prof. Yoshua Bengio [511] and it has been maintained by Montreal Institute for Learning Algorithms (MILA) [512]. In September 2017, they officially stated that they wouldn't work on Theano any more after the last release of it in 2018 [513]. The framework is user friendly as the user is able to do computations by means of NumPy-like syntax, build the models and do the training on either CPU or GPU. Some positive and negative points of this framework are listed as below:

Positive Points:

1- Possibility of using Python and NumPy
2- Presence of computational graph as a useful abstraction
3- Availability of high level wrappers, Keras and Lasagne [514].
4- RNNs.

Negative Points:

1- Long compiling time if model is large
2- Single GPU
3- Useless error messages
4- Finding bugs on amazon web services (AWS) [515]

11.5.2 Torch

Torch [516] is a deep learning framework and it is available for public use as an open source for scientific computing framework. Providing various deep learning algorithms has become this framework so popular, and it is already used by Google, Twitter and Facebook [516]. In addition, the core of the PyTorch [517] is Torch, and it has the role of heart in the human body. The Lua programming language [518] is the basis of the Torch for fast scripting, and implementation of the Torch is in C/CUDA with a wrapper in the Lua [513].

Popularity of the Torch is also connected to its simplicity to use, flexibility for implementing complicated neural network models, possibility of creating arbitrary graphs of deep networks and availability of parallelizing the models over CPUs and GPUs efficiently.

Some main features of the Torch are listed as below:

1- Strong N-dimensional array
2- Availability of many procedures for doing some processes like indexing, slicing and transposing
3- Providing C interface through the LuaJIT [519]
4- Possibility of using linear algebra and numeric optimization methods
5- Neural networks and energy-based models
6- GPU supporting
7- Ability to being embedded with different mobile operating systems, the iOS, the Android and FPGA

11.5.3 TensorFlow

TensorFlow [520], an open source software library developed by the Google Brain Team [521], is introduced for performing numerical computation using flow graphs and conducting deep learning algorithms. Without any doubt, it is one of the most common deep learning frameworks used widely by many companies and scientists. The first version of it suffered from slow running, but the released version of the framework in January 2018 is much faster, and new features are added to enhance the framework and increase its flexibility [522]. It should be pointed out that two programming languages, C++ and Python, have been used for the framework.

Due to the presence of a giant company behind the framework, many huge companies such as eBay, Coca Cola, Twitter, Deep Mind, airbnb, Uber, etc. are users of the framework. Moreover, it is possible to run its model on CPUs, GPUs and TPUs.

In addition to the mentioned points, a comparative list of other points is provided and some advantages and disadvantages are announced.

Advantages:
1- Python and NumPy
2- Less compiling time than the Theano
3- Possibility of parallelizing the data and models
4- Presence of a visualization tool, TensorBoard
5- Similarity to the Theano concerning the possibility of the computational graph abstraction

Disadvantages:
1- Lack of commercial support
2- Working slower than similar frameworks
3- Slow computational graph as it uses only Python
4- Absence of many pretrained models
5- Being fatter than the Torch

11.5.4 Keras

In this part, we would like to have a glance at Keras [523] which is a high-level deep learning API written in Python, and the creator of it is François Chollet [524] who is a researcher in Google [513]. One important advantage of the Keras is the possibility of running it on the top of the TensorFlow or the Theano, and it benefits from supporting convolutional networks, recurrent networks and combinations of them, thus it has covered multiple frameworks. As backend, the Keras framework has both TensorFlow and Theano. In addition, Google has selected the Keras for the high-level API service of the TensorFlow; hence the user enjoys more flexibility. Another advantage of the Keras is its simplicity, and the user is not involved with complex mathematical concepts, although he is able to utilize them correctly and do fast prototyping for advanced and complicated deep learning models. Furthermore, it is possible to run the Keras on both CPUs and GPUs.

In the following, there is a list of properties of the Keras.
1- Working with the TensorFlow and the Theano
2- Growing fast due to its simplicity
3- Supporting only Python and R [525]
4- Lack of many pretrained models

11.5.5 Caffe

Here, we would like to introduce Caffe (stands for Convolutional Architecture for Fast Feature Embedding) [526] which was developed in 2013 by Berkeley Artificial Intelligence Research (BAIR) [527] and community contributors. This framework is considered the brainchild of Yangqing Jia who is working as a researcher at Facebook [528]. Having many contributors and a large repository of pre-trained deep neural network models made Caffe very popular for image classification tasks. The NVIDIA Deep Learning GPU Training System (DIGITS) [529] uses Caffe as one of its powerful frameworks for building deep neural networks to do image classification, segmentation, and object detection tasks. Furthermore, two other used frameworks are Torch and TensorFlow. One important point is the release of Caffe under the BSD 2-Clause license [530].

In order to design and implement our deep neural network model, our choice is Caffe. Many users have grievances concerning the difficulties of installing this framework, and it is, therefore, important

to be aware of the pros and cons of use. Below some important properties of Caffe are listed:
1- Uses plaintext for modeling and optimizations and decreases programming.
2- Acceptable speed to be used in academic and industrial applications. For instance, it has been mentioned in [531] that Caffe is able to process more than 60M images per day if we use a single NVIDIA K40 GPU.
3- Possibility of extending the model and creating new settings easily as it is a modular and flexible framework.
4- Open source framework.
5- A large community of users from different academic, industrial and startup sectors.
6- Opportunity of compiling a Caffe model on different devices and porting to Windows and Linux.
7- Although it is actually based on C++, it supports other programming interfaces, Matlab and Python.
8- Possibility of building complex layers and deep components in a low-level language.
9- CUDA library.
10- Possibility of switching between CPU and GPU.

There are still other reasons that motivated us to select Caffe for our natural plant recognition system. The point is its flexibility for CNN implementation and classification tasks which is our final desire. Additionally, the option of fine tuning is also available in this framework. We are able to extend the model by adding layers and linking it to other toolboxes if it is needed. In addition, the concentration of the framework is on convolutional neural networks. Another point is the activeness of the Caffe framework. Its development has not stopped, therefore, we would be able to use new features if needed. Furthermore, there is a unique facility for Caffe users. They are able to enjoy access to a repository of models that developers have designed, implemented and shared on Caffe Model Zoo [532] [533]. It should be pointed out that Caffe is overall a good choice for us because our intention is to design a feedforward deep network. The last reason is the possibility of changing from CPU to GPU and vice versa, and it has been useful in our real-time application.

11.6 Constructed Confusion Matrix for each Proposed System

Proposed system used the SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	18	0	0	2
Amelanchier Canadensis	1	17	0	2
Acer Pseudoplatanus	1	0	19	0
Cornus	0	0	0	20

Table 11.1: Confusion matrix (distance 25 cm) [151]

Proposed system used the SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	19	1	0	0
Amelanchier Canadensis	0	19	0	1
Acer Pseudoplatanus	1	0	19	0
Cornus	0	0	0	20

Table 11.2: Confusion matrix (distance 50 cm) [151]

Proposed system used the SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	18	0	0	2
Amelanchier Canadensis	0	19	0	1
Acer Pseudoplatanus	0	0	20	0
Cornus	2	0	0	18

Table 11.3: Confusion matrix (distance 75 cm) [151]

Proposed system used the SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	22	1	0	1
Amelanchier Canadensis	0	24	0	0
Acer Pseudoplatanus	0	0	23	1
Cornus	0	0	0	24

Table 11.4: Confusion matrix (distances 100 cm, 150 cm 200 cm) [151]

Proposed system used the FAST-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	16	0	0	4
Amelanchier Canadensis	1	16	0	3
Acer Pseudoplatanus	0	0	20	0
Cornus	0	3	0	17

Table 11.5: Confusion matrix (distance 25 cm) [151]

Proposed system used the FAST-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	18	1	0	1
Amelanchier Canadensis	0	17	0	3
Acer Pseudoplatanus	0	1	19	0
Cornus	0	0	0	20

Table 11.6: Confusion matrix (distance 50 cm) [151]

Proposed system used the FAST-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	19	0	0	1
Amelanchier Canadensis	0	19	0	1
Acer Pseudoplatanus	1	0	19	0
Cornus	2	1	0	17

Table 11.7: Confusion matrix (distance 75 cm) [151]

Proposed system used the FAST-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	21	2	0	1
Amelanchier Canadensis	0	24	0	0
Acer Pseudoplatanus	0	0	24	0
Cornus	1	0	0	23

Table 11.8: Confusion matrix (distances 100 cm, 150 cm and 200 cm) [151]

Proposed system used the HARRIS-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	16	0	0	4
Amelanchier Canadensis	1	15	0	4
Acer Pseudoplatanus	0	0	20	0
Cornus	0	0	0	20

Table 11.9: Confusion matrix (distance 25 cm) [151]

Proposed system used the HARRIS-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	20	0	0	0
Amelanchier Canadensis	1	18	0	1
Acer Pseudoplatanus	1	0	19	0
Cornus	0	0	0	20

Table 11.10: Confusion matrix (distance 50 cm) [151]

Proposed system used the HARRIS-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	18	0	0	2
Amelanchier Canadensis	0	19	0	1
Acer Pseudoplatanus	1	0	19	0
Cornus	2	0	0	18

Table 11.11: Confusion matrix (distance 75 cm) [151]

Proposed system used the HARRIS-SIFT detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	20	3	0	1
Amelanchier Canadensis	0	24	0	0
Acer Pseudoplatanus	0	0	24	0
Cornus	0	0	0	24

Table 11.12: Confusion matrix (distances 100 cm, 150 cm and 200 cm) [151]

Proposed system used the SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	20	0	0	0
Amelanchier Canadensis	2	17	0	1
Acer Pseudoplatanus	1	0	19	0
Cornus	0	0	0	20

Table 11.13: Confusion matrix (distance 25 cm)

Proposed system used the SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	20	0	0	0
Amelanchier Canadensis	3	17	0	0
Acer Pseudoplatanus	0	0	20	0
Cornus	1	0	0	19

Table 11.14: Confusion matrix (distance 50 cm)

Proposed system used the SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	16	0	4	0
Amelanchier Canadensis	0	19	0	1
Acer Pseudoplatanus	0	0	20	0
Cornus	1	2	0	17

Table 11.15: Confusion matrix (distance 75 cm)

Proposed system used the SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	22	0	1	1
Amelanchier Canadensis	0	23	1	0
Acer Pseudoplatanus	0	0	23	1
Cornus	0	0	0	24

Table 11.16: Confusion matrix (distances 100 cm, 150 cm and 200 cm)

Proposed System used the FAST-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	19	0	0	1
Amelanchier Canadensis	2	17	1	0
Acer Pseudoplatanus	1	0	19	0
Cornus	0	0	0	20

Table 11.17: Confusion matrix (distance 25 cm)

Proposed system used the FAST-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	20	0	0	0
Amelanchier Canadensis	2	15	0	3
Acer Pseudoplatanus	0	0	20	0
Cornus	1	1	0	18

Table 11.18: Confusion matrix (distance 50 cm)

Proposed system used the FAST-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	14	0	6	0
Amelanchier Canadensis	0	19	0	1
Acer Pseudoplatanus	0	1	19	0
Cornus	1	3	0	16

Table 11.19: Confusion matrix (distance 75 cm)

Proposed system used the FAST-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	21	0	2	1
Amelanchier Canadensis	0	22	1	1
Acer Pseudoplatanus	0	0	23	1
Cornus	0	0	0	24

Table 11.20: Confusion matrix (distances 100 cm, 150 cm and 200 cm)

Proposed system used the HARRIS-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	19	0	0	1
Amelanchier Canadensis	1	18	1	0
Acer Pseudoplatanus	1	0	19	0
Cornus	0	0	0	20

Table 11.21: Confusion matrix (distance 25 cm)

Proposed system used the HARRIS-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	18	0	0	2
Amelanchier Canadensis	4	14	0	2
Acer Pseudoplatanus	0	0	20	0
Cornus	1	1	1	17

Table 11.22: Confusion matrix (distance 50 cm)

Proposed system used the HARRIS-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	13	0	6	1
Amelanchier Canadensis	0	19	0	1
Acer Pseudoplatanus	0	1	19	0
Cornus	2	0	0	18

Table 11.23: Confusion matrix (distance 75 cm)

Proposed system used the HARRIS-SURF detection and description techniques	Hydrangea	Amelanchier Canadensis	Acer Pseudoplatanus	Cornus
Hydrangea	22	0	1	1
Amelanchier Canadensis	0	21	0	3
Acer Pseudoplatanus	0	0	23	1
Cornus	0	0	0	24

Table 11.24: Confusion matrix (distances 100 cm, 150 cm and 200 cm)

References

[1] Wilson Nichols Stewart, Wilson Stewart, and Gar Rothwell. *Paleobotany and the evolution of plants*. Cambridge University Press, 1993.

[2] Daniel Ladinsky. *The Gift: poems by the Great sufi master*. Penguin, 1999.

[3] Ji-Xiang Du, Xiao-Feng Wang, and Guo-Jun Zhang. Leaf shape based plant species recognition. *Applied mathematics and computation*, 185(2):883–893, 2007.

[4] Yanhua Ye, Chun Chen, Chun-Tak Li, Hong Fu, and Zheru Chi. A computerized plant species recognition system. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004*, pages 723–726. IEEE, 2004.

[5] Miao Zhenjiang, Marie-Helene Gandelin, and Yuan Baozong. An oopr-based rose variety recognition system. *Eng. Appl. Artif. Intell.*, 19(1):79–101, February 2006.

[6] Rodrigo de Oliveira Plotze, Maurício Falvo, Juliano Gomes Pádua, Luís Carlos Bernacci, Maria Lúcia Carneiro Vieira, Giancarlo Conde Xavier Oliveira, and Odemir Martinez Bruno. Leaf shape analysis using the multiscale minkowski fractal dimension, a new morphometric method: a study with passiflora (passifloraceae). *Canadian Journal of Botany*, 83(3):287–301, 2005.

[7] Allen White. *The history of the Washington state university, college of pharmacy, 1891-1991*. College of Pharmacy, Washington State University, 1996.

[8] International Organization For Standardization. *9241-11. Ergonomic requirements for office work with visual display terminals (VDTs)*. ISO, 1998.

[9] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, pages 430–443. Springer, 2006.

[10] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[11] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[12] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: speeded up robust features. In *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[13] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, June 2010.

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[15] Massimo Bernaschi, Annarita Di Lallo, Riccardo Fulcoli, Emanuele Gallo, and Luca Timmoneri. Combined use of graphics processing unit (gpu) and central processing unit (cpu) for passive radar signal & data elaboration. In *2011 12th International Radar Symposium (IRS)*, pages 315–320, 2011.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[17] Jan Kunze, Simon Hardt, and Klaus-Dieter Kuhnert. *Team Zephyr university of Siegen*. Haßfurt, Germany, 2016.

[18] Masoud Fathi Kazerouni, Nazeer T. Mohammed Saeed, and Klaus-Dieter Kuhnert. Exploration of autonomous mobile robots through challenging outdoor environments for natural plant recognition using deep neural network. *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP 2019)*, 2019.

[19] Li Fei-Fei, Rob Fergus, and Antonio Torralba. Recognizing and learning object categories, cvpr 2007 short course. *Cambridge, MA*, 2007.

[20] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[21] Vladimir Vapnik. *The natural of statistical theory*. New York: Springer-Verlag, 1995.

[22] Gardenerdy Staff. Different Types of Plants. `https://gardenerdy.com/different-types-of-plants`. [Online; accessed 2018-08-07].

[23] Otto Schmeil, Jost Fitschen, Karlheinz Senghas, and Siegmund Seybold. *Flora von Deutschland und angrenzender Länder*. Quelle und Meyer Wiesbaden, 1996.

[24] Larry Morse. Computer-assisted storage and retrieval of the data of taxonomy and systematics. *Taxon*, pages 29–43, 1974.

[25] James Cullen and Crynan Alexander. The european garden flora. *Curtis's Botanical Magazine*, 1(3):119–122, 1984.

[26] Ken Thompson. Name that plant: a book is still the best place to look. url=https://www.telegraph.co.uk/gardening/11245168/Name-that-plant-a-book-is-still-the-best-place-to-look.html, 2014. [Online; accessed 2018-08-07].

[27] Zhiyong Wang, Zheru Chi, David Dagan Feng Feng, and Qing Wang. Leaf image retrieval with shape features. In *Advances in Visual Information Systems*, volume 1929, pages 477–487. Springer, 2000.

[28] Herbert Freeman and John Saghri. Generalized chain codes for planar curves. In *Proceedings of the 4th International Joint Conference on Pattern Recognition*, pages 701–703, 1978.

[29] Babu Mehtre, Mohan Kankanhalli, and Wing Foon Lee. Shape measures for content based image retrieval: a comparison. *Information Processing & Management*, 33(3):319–337, 1997.

[30] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[31] Charles Zahn and Ralph Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21(3):269–281, 1972.

[32] Roy Davies. Machine vision: Theory, algorithms and practicalities. 1990, 1997.

[33] Xianfeng Ding, Weixing Kong, Changbo Hu, and Songde Ma. Image retrieval using schwarz representation of one-dimensional feature. In *International Conference on Advances in Visual Information Systems*, pages 443–450. Springer, 1999.

[34] George Nagy and Jie Zou. Interactive visual pattern recognition. In *Object recognition supported by user interaction for service robots*, volume 2, pages 478–481. IEEE, 2002.

[35] Farzin Mokhtarian and Sadegh Abbasi. Matching shapes with self-intersections: application to leaf classification. *IEEE Transactions on Image Processing*, 13(5):653–661, 2004.

[36] Takeshi Saitoh, Kimiya Aoki, and Toyohisa Kaneko. Automatic recognition of blooming flowers. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 1, pages 27–30. IEEE, 2004.

[37] Xiao-Feng Wang, Ji-Xiang Du, and Guo-Jun Zhang. Recognition of leaf images based on shape features using a hypersphere classifier. In *International Conference on Intelligent Computing*, pages 87–96. Springer, 2005.

[38] Qingfeng Wu, Changle Zhou, and Chaonan Wang. Feature extraction and automatic recognition of plant leaf using artificial neural network. *Advances in Artificial Intelligence*, 3:5–12, 2006.

[39] Abdolvahab Ehsanirad and Sharath Kumar. Leaf recognition for plant classification using glcm and pca methods. *Oriental Journal of Computer Science and Technology*, 3(1):31–36, 2010.

[40] David Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of Remote Sensing*, 28(1):45–62, 2002.

[41] Leen-Kiat Soh and Costas Tsatsoulis. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2):780–795, 1999.

[42] Robert Martin Haralick, Kumarasamy Shanmugam, and Its'Hak Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973.

[43] Ian Jolliffe. *Principal component analysis.* Springer Berlin Heidelberg, 2011.

[44] Basavaraj Anami, Suvarna Nandyal, and Aliseri Govardhan. A combined color, texture and edge features based approach for identification and classification of indian medicinal plants. *International Journal of Computer Applications*, 6(12):45–51, 2010.

[45] Javed Hossain and Ashraful Amin. Leaf shape identification based plant biometrics. In *13th International Conference on Computer and Information Technology (ICCIT)*, pages 458–463. IEEE, 2010.

[46] Zhiyong Wang, Bin Lu, Zheru Chi, and Dagan Feng. Leaf image classification with shape context and sift descriptors. In *Proceedings - 2011 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2011*, pages 650–654. IEEE, 2011.

[47] Naomi Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[48] N. Valliammal and Dr. S. N. Geethalakshmi. Automatic recognition system using preferential image segmentation for leaf and flower images. *Computer Science & Engineering: An International Journal (CSEIJ)*, 1(4):13–25, 2011.

[49] Jyotismita Chaki and Ranjan Parekh. Plant leaf recognition using shape based features and neural network classifiers. *International Journal of Advanced Computer Science and Applications*, 2(10), 2011.

[50] Kian-Lee Tan, Beng Chin Ooi, and Lay Foo Thiang. Retrieving similar shapes effectively and efficiently. *Multimedia Tools and Applications*, 19(2):111–134, 2003.

[51] Arun Priya, Thiruvambalam Balasaravanan, and Antony Selvadoss Thanamani. An efficient leaf recognition algorithm for plant classification using support vector machine. In *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, pages 428–432. IEEE, 2012.

[52] Pavan Kumar Mishra, Sanjay Kumar Maurya, Ravindra Kumar Singh, and Arun Kumar Misra. A semi automatic plant identification based on digital leaf and flower images. In *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*, pages 68–73. IEEE, 2012.

[53] Sandeep Kumar. Leaf color, area and edge features based approach for identification of indian medicinal plants. *Indian Journal of Computer Science and Engineering (IJCSE)*, 3(3):436–442, 2012.

[54] Akhil Arora, Ankit Gupta, Nitesh Bagmar, Shashwat Mishra, and Arnab Bhattacharya. A plant identification system using shape and morphological features on segmented leaflets: Team iitk, clef 2012. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[55] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[56] Chih-Ying Gwo and Chia-Hung Wei. Plant identification through images: Using feature extraction of key points on leaf contours1. *Applications in Plant Sciences*, 1(11):1200005, 2013.

[57] Mohamad Faizal Ab Jabal, Suhardi Hamid, Salehuddin Shuib, and Illiasaak Ahmad. Leaf features extraction and recognition approaches to classify plant. *Journal of Computer Science*, 9(10):1295–1304, 2013.

[58] Sandeep Kumar and Viswanath Talasila. Leaf features based approach for automated identification of medicinal plants. In *2014 International Conference on Communication and Signal Processing*, pages 210–214. IEEE, 2014.

[59] Gábor Szűcs, Dávid Papp, and Dániel Lovas. Viewpoints combined classification, method in image-based plant identification task. In *CLEF Conference*, volume 1180, pages 763–770, 2014.

[60] Carlo Tomasi. Estimating gaussian mixture densities with em–a tutorial. *Duke University*, pages 1–8, 2004.

[61] Nursuriati Jamil, Nuril Aslina Che Hussin, Sharifalillah Nordin, and Khalil Awang. Automatic plant identification: Is shape the key feature? *Procedia Computer Science*, 76:436–442, 2015.

[62] Alceu Ferraz Costa, Gabriel Humpire-Mamani, and Agma Juci Machado Traina. An efficient algorithm for fractal analysis of textures. In *25th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 39–46. IEEE, 2012.

[63] Nisar Ahmed, Usman Ghani Khan, and Shahzad Asif. An automatic leaf based plant identification system. *Science International-Lahore*, 28(1):427–430, 2016.

[64] Jing Zhang, Choong-Woong Yoo, and Seok-Wun Ha. Roi based natural image retrieval using color and texture feature. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, volume 4, pages 740–744. IEEE, 2007.

[65] Olfa Mzoughi, Itheri Yahiaoui, Nozha Boujemaa, and Ezzeddine Zagrouba. Advanced tree species identification using multiple leaf parts image queries. In *2013 IEEE International Conference on Image Processing*, pages 3967–3971. IEEE, 2013.

[66] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang, and Qiao-Liang Xiang. A leaf recognition algorithm for plant classification using probabilistic neural network. In *2007 IEEE International Symposium on Signal Processing and Information Technology*, pages 11–16. IEEE, 2007.

[67] Hamid Laga, Sebastian Kurtek, Anuj Srivastava, Mahmood Golzarian, and Stanley J Miklavcic. A riemannian elastic metric for shape-based plant leaf classification. In *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–7. IEEE, 2012.

[68] Bolla Vijaya Lakshmi and Vasudev Mohan. Plant leaf image detection method using a midpoint circle algorithm for shape-based feature extraction. *Journal of Modern Applied Statistical Methods*, 16(1):461–480, 2017.

[69] Juan Carlos Flores-Bastida, Asdrúbal López-Chau, Rafael Rojas-Hernández, and Valentin Trujillo-Mora. Automatic classification of lobed simple and unlobed simple leaves for plant identification. *Research in Computing Science*, 139:9–18, 2017.

[70] Sue Han Lee, Yang Loong Chang, and Chee Seng Chan. Lifeclef 2017 plant identification challenge: Classifying plants using generic-organ correlation features. *CLEF Conference*, 2017.

[71] Sue Han Lee, Yang Loong Chang, Chee Seng Chan, and Paolo Remagnino. Hgo-cnn: Hybrid generic-organ convolutional neural network for multi-organ plant classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 4462–4466. IEEE, 2017.

[72] Sukhvir Kaur, Shreelekha Pandey, and Shivani Goel. Plants disease identification and classification through leaf images: a survey. *Archives of Computational Methods in Engineering*, pages 507–530, 2018.

[73] Donald Specht. Probabilistic neural networks. *Neural Networks*, 3(1):109–118, 1990.

[74] Jiazhi Pan and Yong He. Recognition of plants by leaves digital image and neural network. In *2008 International Conference on Computer Science and Software Engineering*, volume 4, pages 906–910. IEEE, 2008.

[75] Windhya Rankothge, Samantha Dissanayake, Kanchana Gunathilaka, Chamarams Gunarathna, Chamitha Mudalige, and Rohana Thilakumara. Plant recognition system based on neural networks. In *2013 International Conference on Advances in Technology and Engineering (ICA-TE)*, pages 1–4. IEEE, 2013.

[76] Vijay Satti, Anshul Satya, and Shanu Sharma. An automatic leaf recognition system for plant identification using machine vision technology. *International Journal of Engineering Science and Technology (IJEST)*, 5(4):874–879, 2013.

[77] Redmond Ramin Shamshiri, Cornelia Weltzien, Ibrahim Hameed, Ian Yule, Tony Grift, Siva Balasundram, Lenka Pitonakova, Desa Ahmad, and Girish Chowdhary. Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering*, 11(4):1–14, 2018.

[78] Kongskilde. VIBRO CROP Intelli. `http://www.kongskilde.com/de/de-DE/Agriculture/Soil/Weed%20Control/Weed%20Control/VIBRO%20CROP%20Intelli`. [Online; accessed 2019-03-03].

[79] Sai Kirthi Pilli, Bharathiraja Nallathambi, Smith Jessy George, and Vivek Diwanji. eAGROBOT–a robot for early crop disease detection using image processing. In *2014 International Conference on Electronics and Communication Systems (ICECS)*, pages 1–6. IEEE, 2014.

[80] Heba Al-Hiary, Sulieman Bani-Ahmad, Mohammad Reyalat, Malik Braik, and Zainab ALRahamneh. Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*, 17, 2011.

[81] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011.

[82] Marius Muja and David Lowe. Flann – fast library for approximate nearest neighbors. `https://www.cs.ubc.ca/research/flann/`, 2018. [Online; accessed 2018-04-14].

[83] Sathiesh Kumar, Ilango Gogul, Mohan Deepan Raj, S.K. Pragadesh, and Sarathkumar Sebastin. Smart autonomous gardening rover with plant recognition using neural networks. *Procedia Computer Science*, 93:975–981, 2016.

[84] Charles Mallah, James Cope, and James Orwell. Plant leaf classification using probabilistic integration of shape, texture and margin features. *Signal Processing, Pattern Recognition and Applications (SPPRA 2013)*, 3842, 2013.

[85] James Cope, Thibaut Beghin, Paolo Remagnino, et al. One-hundred plant species leaves data set, 2012.

[86] Waghmare. Leaf shapes database. `http://www.imageprocessingplace.com/`, 2007. [Online; accessed 2014-04-14].

[87] Oskar Söderkvist. Computer vision classification of leaves from swedish trees, 2001.

[88] De-Shuang Huang, Jianhua Ma, Kang-Hyun Jo, and Michael Gromiha. *Intelligent Computing Theories and Applications: 8th International Conference, ICIC 2012, Huangshan, China, July 25-29, 2012, Proceedings*, volume 7390. Springer, 2012.

[89] Gaurav Agarwal, Peter Belhumeur, Steven Feiner, David Jacobs, John Kress, Ravi Ramamoorthi, Norman Bourg, Nandan Dixit, Haibin Ling, Dhruv Mahajan, Rusty Russell, Sameer Shirdhonkar, Kalyan Sunkavalli, and Sean White. First steps toward an electronic field guide for plants. *Taxon*, 55(3):597–610, 2006.

[90] ImageCLEF. Plant identification task 2011, https://www.imageclef.org/2011/plants, Online; accessed 2017-04-03.

[91] Masoud Fathi Kazerouni, Nazeer T. Mohammed Saeed, and Klaus-Dieter Kuhnert. Fully-automatic natural plant recognition system using deep neural network for dynamic outdoor environments. *SN Applied Sciences, Springer*, 1(7):756, 2019.

[92] Coding Lab TechOnTechnology. Image comparison in matlab [matrix laboratory] using histograms. `http://codinlab.blogspot.com/2013/10/image-comparison-in-matlab-matrix.html`, 2007. [Online; accessed 2018-07-14].

[93] Jack Meyer and Michael Ormiston. The comparative statics of cumulative distribution function changes for the class of risk averse agents. *Journal of Economic Theory*, 31(1):153–169, 1983.

[94] Dominik Sankowski and Jacek Nowakowski. *Computer vision in robotics and industrial applications*. World Scientific Publishing Co., Inc., 2014.

[95] Eric Grimson and Joseph Mundy. Computer vision applications. *Communications of the ACM*, 37(3):45–51, 1994.

[96] Shingo Kagami. High-speed vision systems and projectors for real-time perception of the world. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops*, pages 100–107. IEEE, 2010.

[97] Zhijun Pei, Ping Zhang, Shoumei Sun, and Jinqing Gu. Fisher information analysis for matching feature extraction. In *2009 International Conference on Information Technology and Computer Science*, volume 1, pages 425–428. IEEE, 2009.

[98] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.

[99] Concetta Morrone and David Burr. Feature detection in human vision: A phase-dependent energy model. *Proceedings of the Royal Society of London. Series B, Containing papers of a Biological character. Royal Society (Great Britain)*, 235(1280):221–245, 1988.

[100] Naoki Sakai, Satoshi Yonekawa, Akio Matsuzaki, and Hiroko Morishima. Two-dimensional image analysis of the shape of rice and its application to separating varieties. *Journal of Food Engineering*, 27(4):397–407, 1996.

[101] Alberto Perez-Jimenez, Fernando Lopez, José Vicente Benlloch-Dualde, and Svend Christensen. Colour and shape analysis techniques for weed detection in cereal fields. *Computers and Electronics in Agriculture*, 25(3):197–212, 2000.

[102] Thibaut Beghin, James Cope, Paolo Remagnino, and Sarah Barman. Shape and texture based plant leaf classification. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 345–353. Springer, 2010.

[103] Peter Belhumeur, Daozheng Chen, Steven Feiner, David Jacobs, John Kress, Haibin Ling, Ida Lopez, Ravi Ramamoorthi, Sameer Sheorey, Sean White, and Ling Zhang. Searching the world's herbaria: A system for visual identification of plant species. In *European Conference on Computer Vision*, pages 116–129. Springer, 2008.

[104] Carlos Caballero and Carmen Aranda. Plant species identification using leaf image retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 327–334. ACM, 2010.

[105] Sean White, Steven Feiner, and Jason Kopylec. Virtual vouchers: Prototyping a mobile augmented reality user interface for botanical species identification. In *3D User Interfaces (3DUI'06)*, pages 119–126. IEEE, 2006.

[106] Vishakha Metre and Jayshree Ghorpade. An overview of the research on texture based plant leaf classification. *International Journal of Computer Science and Network*, 2, 2013.

[107] Beril Sirmacek and Cem Unsalan. Urban-area and building detection using sift keypoints and graph theory. *IEEE Transactions on Geoscience and Remote Sensing*, 47(4):1156–1167, 2009.

[108] Jin Zhao, Sichao Zhu, and Xinming Huang. Real-time traffic sign detection using surf features on fpga. In *2013 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2013.

[109] Suhas Salve and Kalpana Jondhale. Shape matching and object recognition using shape contexts. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 9, pages 471–474. IEEE, 2010.

[110] Ji-Xiang Du, De-Shuang Huang, Xiao-Feng Wang, and Xiao Gu. Computer-aided plant species identification (capsi) based on leaf shape matching technique. *Transactions of the Institute of Measurement and Control*, 28(3):275–285, 2006.

[111] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[112] Yong Seok Heo, Kyoung Mu Lee, and Sang Uk Lee. Illumination and camera invariant stereo matching. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[113] Supheakmungkol Sarin, Michael Fahrmair, Matthias Wagner, and Wataru Kameyama. Holistic feature extraction for automatic image annotation. In *2011 Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering*, pages 59–66. IEEE, 2011.

[114] Nishchal Kumar Verma, Ankit Goyal, Harsha Vardhan, Rahul Kumar Sevakula, and Al Salour. Object matching using speeded up robust features. In *Intelligent and Evolutionary Systems*, pages 415–427. Springer, 2016.

[115] Oskar Andersson and Steffany Reyna Marquez. A comparison of object detection algorithms using unmanipulated testing images: Comparing sift, kaze, akaze and orb (dissertation). `http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-186503`, 2016. [Online; accessed 2018-08-23].

[116] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[117] Bhaskara Rao, Vara Prasad, and Pavan Kumar. Feature extraction using zernike moments. *International Journal of Latest Trends in Engineering and Technology*, 2(2):228–234, 2013.

[118] Marcin Novotni and Reinhard Klein. 3d zernike descriptors for content based shape retrieval. In *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, pages 216–225. ACM, 2003.

[119] Tomoki Watanabe, Satoshi Ito, and Kentaro Yokoi. Co-occurrence histograms of oriented gradients for pedestrian detection. In *Pacific-Rim Symposium on Image and Video Technology*, pages 37–47. Springer, 2009.

[120] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[121] Jerod Weinman, Allen Hanson, and Andrew McCallum. Sign detection in natural images with conditional random fields. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing*, pages 549–558, 2004.

[122] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[123] Caroline Silva, Thierry Bouwmans, and Carl Frélicot. An extended center-symmetric local binary pattern for background modeling and subtraction in videos. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISAPP 2015*, 2015.

[124] Di Huang, Caifeng Shan, Mohsen Ardabilian, Yunhong Wang, and Liming Chen. Local binary patterns and its application to facial image analysis: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):765–781, 2011.

[125] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555. IEEE, 2011.

[126] David Nistér and Henrik Stewénius. Linear time maximally stable extremal regions. In *European Conference on Computer Vision*, pages 183–196. Springer, 2008.

[127] Per-Erik Forssén. Maximally stable colour regions for recognition and matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[128] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517. IEEE, 2012.

[129] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142. Springer, 2002.

[130] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *European Conference on Computer Vision*, pages 228–241. Springer, 2004.

[131] Ling Shao, Timor Kadir, and Michael Brady. Geometric and photometric invariant distinctive regions detection. *Information Sciences*, 177(4):1088–1122, 2007.

[132] Tinne Tuytelaars and Luc Van Gool. Content-based image retrieval based on local affinely invariant regions. In *International Conference on Advances in Visual Information Systems*, pages 493–500. Springer, 1999.

[133] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85, 2004.

[134] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *BMVC*, pages 412–425, 2000.

[135] Hans Peter Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, 1980.

[136] Hans Peter Moravec. Techniques towards automatic visual obstacle avoidance. 1977.

[137] Badrul hisham Hisham, Shahrul Nizam Yaakob, Rafikha Aliana Raof, Amir Nazren, and Mohd Wafi. Template matching using sum of squared difference and normalized cross correlation. In *2015 IEEE Student Conference on Research and Development (SCOReD)*, pages 100–104. IEEE, 2015.

[138] Harris corner detection. `https://docs.opencv.org/3.0-beta`. [Online; accessed 2018-11-12].

[139] Rick Szeliski. Notes on the harris detector.

[140] Jianbo Shi Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994.

[141] Ren Yan. A survey of corner detection algorithms. *Mechanical Engineering & Automation*, 1, 2009.

[142] Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305. Interlaken, 1987.

[143] Stephen Smith and Michael Brady. Susan–a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.

[144] Leonardo Trujillo and Gustavo Olague. Automated design of image operators that detect interest points. *Evolutionary Computation*, 16(4):483–507, 2008.

[145] Asif Masood and Muhammad Sarfraz. Corner detection by sliding rectangles along planar curves. *Computers & Graphics*, 31(3):440–448, 2007.

[146] Xiaoming Peng, Chengping Zhou, and Mingyue Ding. Corner detection method based on wavelet transform. In *Image Extraction, Segmentation, and Recognition*, volume 4550, pages 319–324. International Society for Optics and Photonics, 2001.

[147] Jie Chen, Li-hui Zou, Juan Zhang, and Li-hua Dou. The comparison and application of corner detection algorithms. *Journal of Multimedia*, 4(6), 2009.

[148] Farzin Mokhtarian, Nasser Khalili, and Peter Yuen. Multi-scale free-form 3d object recognition using 3d models. *Image and Vision Computing*, 19(5):271–281, 2001.

[149] Farzin Mokhtarian and Miroslaw Bober. *Curvature scale space representation: theory, applications, and MPEG-7 standardization*, volume 25. 2003.

[150] Alan Bovik. *Handbook of image and video processing*. Academic press, 2010.

[151] Masoud Fathi Kazerouni, Jens Schlemper, and Klaus-Dieter Kuhnert. Modern detection and description methods for natural plants recognition. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(8):1497–1512, 2017.

[152] Tony Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[153] David Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[154] Tony Lindeberg. Scale-space. 2009.

[155] Francisco Estrada, Allan Jepson, and David Fleet. Local features tutorial 2. 2004.

[156] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1. IEEE, 2001.

[157] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[158] Matthew Brown and David Lowe. Invariant features from interest point groups. In *Proceedings of the British Machine Vision Conference 2002*, volume 13, 2002.

[159] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*, pages 778–792. Springer, 2010.

[160] ORB (Oriented FAST and Rotated BRIEF). `http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html`. [Online; accessed 2018-08-07].

[161] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, volume 99, pages 518–529. Morgan Kaufmann Publishers Inc., 1999.

[162] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, 1936.

[163] Brute-force Approach. `http://intelligence.worldofcomputing.net/ai-search/brute-force-search.html#.W-rW3DhKhdg`. [Online; accessed 2018-08-07].

[164] Feature Matching. `https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_matcher/py_matcher.html`. [Online; accessed 2018-08-07].

[165] Mohammad Norouzi, David Fleet, and Ruslan Salakhutdinov. Hamming distance metric learning. In *Advances in Neural Information Processing Systems*, pages 1061–1069. Curran Associates, Inc., 2012.

[166] Common Interfaces of Descriptor Matchers. `https://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_descriptor_matchers.html`. [Online; accessed 2018-08-23].

[167] Cordelia Schmid, Roger Mohr, and Christian Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[168] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[169] Masoud Fathi Kazerouni, Jens Schlemper, and Klaus-Dieter Kuhnert. Comparison of modern description methods for the recognition of 32 plant species. *Signal & Image Processing*, 6(2):1, 2015.

[170] Masoud Fathi Kazerouni, Jens Schlemper, and Klaus-Dieter Kuhnert. Efficient modern description methods by using surf algorithm for recognition of plant species. *Advances in Image and Video Processing*, 3(2):10, 2015.

[171] First land plants and fungi changed earth's climate, paving the way for explosive evolution of land animals, new gene study suggests. `http://science.psu.edu/news-and-events/2001-news/Hedges8-2001.htm`, 2001. [Online; accessed 2018-10-15].

[172] Joel Nolan. Mitigating heat island effect in the urban environment.

[173] Montague Yudelman, Annu Ratta, and David Nygaard. Pest management and food production: looking to the future. 25, 1998.

[174] Anthony Bloom, Julia Lee-Taylor, Sasha Madronich, David Messenger, Paul Palmer, David Reay, and Andy McLeod. Global methane emission estimates from ultraviolet irradiation of terrestrial plant foliage. *New Phytologist*, 187(2):417–425, 2010.

[175] Study gives green light to plants' role in global warming. `https://www.sciencedaily.com/releases/2010/04/100429111021.htm`, 2010. [Online; accessed 2018-10-17].

[176] Takeshi Saitoh and Toyohisa Kaneko. Automatic recognition of wild flowers. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 2, pages 507–510. IEEE, 2000.

[177] Wang Xiaofeng, Huang Deshuang, Du Jixiang, et al. Feature extraction and recognition for leaf images. *Computer Engineering and Applications*, 42(3):190–193, 2006.

[178] Qing-Kui Man, Chun-Hou Zheng, Xiao-Feng Wang, and Feng-Yan Lin. Recognition of plant leaves using support vector machine. In *International Conference on Intelligent Computing*, pages 192–199. Springer, 2008.

[179] Shanwen Zhang and Xianfeng Wang. Method of plant leaf recognition based on weighted locally linear embedding. *Transactions of the Chinese Society of Agricultural Engineering*, 27(12):141–145, 2011.

[180] Maliheh Shabanzade, Morteza Zahedi, and Seyyed Amin Aghvami. Combination of local descriptors and global features for leaf recognition. *Signal & Image Processing: An International Journal*, 2(3):23, 2011.

[181] Shanwen Zhang and Ying-Ke Lei. Modified locally linear discriminant embedding for plant leaf recognition. *Neurocomputing*, 74(14-15):2284–2290, 2011.

[182] Bo Li, Chun-Hou Zheng, and De-Shuang Huang. Locally linear discriminant embedding: An efficient method for face recognition. *Pattern Recognition*, 41(12):3813–3821, 2008.

[183] Rongxiang Hu, Wei Jia, Haibin Ling, and Deshuang Huang. Multiscale distance matrix for fast plant leaf recognition. *IEEE Transactions on Image Processing*, 21(11):4667–4672, 2012.

[184] Jun Luo, Yong Ma, Erina Takikawa, Shihong Lao, Masato Kawade, and Bao-Liang Lu. Person-specific sift features for face recognition. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, volume 2, pages II–593–II–596. IEEE, 2007.

[185] Donghoon Kim and Rozenn Dahyot. Face components detection using surf descriptors and svms. In *2008 International Machine Vision and Image Processing Conference*, pages 51–56. IEEE, 2008.

[186] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[187] Xin Chen, Xiaohua Hu, and Xiajiong Shen. Spatial weighting for bag-of-visual-words and its application in content-based image retrieval. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 867–874. Springer, 2009.

[188] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 270–279. ACM, 2010.

[189] Michael Villamizar, Jorge Scandaliaris, Alberto Sanfeliu, and Juan Andrade-Cetto. Combining color-based invariant gradient detector with hog descriptors for robust image detection in scenes under cast shadows. In *2009 IEEE International Conference on Robotics and Automation*, pages 1997–2002. IEEE, 2009.

[190] Jun Yang, Yu-Gang Jiang, Alexander Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*, pages 197–206. ACM, 2007.

[191] David Picard, Nicolas Thome, and Matthieu Cord. An efficient system for combining complementary kernels in complex visual categorization tasks. In *2010 IEEE International Conference on Image Processing*, pages 3877–3880. IEEE, 2010.

[192] Chih-Fong Tsai. Bag-of-words representation in image annotation: A review. *ISRN Artificial Intelligence*, 2012, 2012.

[193] Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele. Local features for object class recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1792–1799. IEEE, 2005.

[194] Tinne Tuytelaars, Krystian Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.

[195] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009.

[196] Anna Bosch, Xavier Muñoz, and Robert Martí. Which is the best way to organize/classify images by content? *Image and Vision Computing*, 25(6):778–791, 2007.

[197] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[198] Trupti Patel and Sandip Panchal. Corner detection techniques: an introductory survey. 2014.

[199] Jeffrey Beis and David Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pages 1000–. IEEE, 1997.

[200] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[201] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1508–1515. IEEE, 2005.

[202] Christopher Manning and Hinrich Schütze. *Foundations of statistical natural language processing.* MIT press, 1999.

[203] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval.* New York: ACM Press; Harlow, England: Addison-Wesley, 2010.

[204] Maryam Bugaje and Gobinda Chowdhury. Data retrieval= text retrieval? In *International Conference on Information*, pages 253–262. Springer, 2018.

[205] Zellig Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[206] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1470–1477. IEEE, 2003.

[207] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.

[208] Iván González Díaz, Vincent Buso, Jenny Benois-Pineau, Guillaume Bourmaud, and Rémi Megret. Modeling instrumental activities of daily living in egocentric vision as sequences of active objects and context for alzheimer disease research. In *Proceedings of the 1st ACM International Workshop on Multimedia Indexing and Information Retrieval for Healthcare*, pages 11–14. ACM, 2013.

[209] Gaurav Sharma, Frédéric Jurie, and Cordelia Schmid. Discriminative spatial saliency for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3506–3513. IEEE, 2012.

[210] Eleonora Vig, Michael Dorr, and David Cox. Space-variant descriptor sampling for action recognition based on saliency and eye movements. In *European Conference on Computer Vision*, pages 84–97. Springer, 2012.

[211] Fei-Fei Li. Machine learning in computer vision. *Lecture Notes.*

[212] Vincent Delaitre, Ivan Laptev, and Josef Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *Proceedings of the British Machine Vision Conference*, pages 97.1–97.11. BMVA Press, 2010.

[213] Weilong Yang, Yang Wang, and Greg Mori. Recognizing human actions from still images with latent poses. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2030–2037. IEEE, 2010.

[214] Yang Wang, Duan Tran, Zicheng Liao, and David Forsyth. Discriminative hierarchical part-based models for human parsing and action recognition. In *Gesture Recognition*, pages 273–301. Springer, 2017.

[215] Martin Fischler and Robert Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, (1):67–92, 1973.

[216] Robert Fergus, Pietro Perona, and Andrew Zisserman. A sparse object category model for efficient learning and exhaustive recognition. In *CVPR (1)*, pages 380–387, 2005.

[217] Juan Carlos Niebles and Li Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[218] Thomas Deselaers, Lexi Pimenidis, and Hermann Ney. Bag-of-visual-words models for adult image classification and filtering. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[219] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[220] Jiang Hao and Xu Jie. Improved bags-of-words algorithm for scene recognition. In *2010 2nd International Conference on Signal Processing Systems*, volume 2, pages V2–279–V2–282. IEEE, 2010.

[221] Ross Walker, Paul Jackway, and Ian Dennis Longstaff. Recent developments in the use of the co-occurrence matrix for texture recognition. In *Proceedings of 13th International Conference on Digital Signal Processing*, volume 1, pages 63–65. IEEE, 1997.

[222] Tapas Kanungo, David Mount, Nathan Netanyahu, Christine Piatko, Ruth Silverman, and Angela Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (7):881–892, 2002.

[223] Sheng Xu, Tao Fang, Deren Li, and Shiwei Wang. Object classification of aerial images with bag-of-visual words. *IEEE Geoscience and Remote Sensing Letters*, 7(2):366–370, 2010.

[224] Zisheng Li, Jun-ichi Imai, and Masahide Kaneko. Face and expression recognition based on bag of words method considering holistic and local image features. In *2010 10th International Symposium on Communications and Information Technologies*, pages 1–6. IEEE, 2010.

[225] Kiyo Tomiyasu. Tutorial review of synthetic-aperture radar (sar) with applications to imaging of the ocean surface. *Proceedings of the IEEE*, 66(5):563–583, 1978.

[226] Jie Feng, LC Jiao, Xiangrong Zhang, and Dongdong Yang. Bag-of-visual-words based on clonal selection algorithm for sar image classification. *IEEE Geoscience and Remote Sensing Letters*, 8(4):691–695, 2011.

[227] Sandra Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo de Albuquerque Araújo. Bossa: Extended bow formalism for image classification. In *2011 18th IEEE International Conference on Image Processing*, pages 2909–2912. IEEE, 2011.

[228] Bharath Ramesh, Cheng Xiang, and Tong Heng Lee. Shape classification using invariant features and contextual information in the bag-of-words model. *Pattern Recognition*, 48(3):894–906, 2015.

[229] Yi Yuan and Xiangyun Hu. Bag-of-words and object-based classification for cloud extraction from satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(8):4197–4205, 2015.

[230] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150:109–125, 2016.

[231] Qiqi Zhu, Yanfei Zhong, Bei Zhao, Gui-Song Xia, and Liangpei Zhang. Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. *IEEE Geoscience and Remote Sensing Letters*, 13(6):747–751, 2016.

[232] Yanshan Li, Weiming Liu, Qinghua Huang, and Xuelong Li. Fuzzy bag of words for social image description. *Multimedia Tools and Applications*, 75(3):1371–1390, 2016.

[233] Ross Quinlan. Program for machine learning. *C4. 5*, 1993.

[234] Kevin Murphy. Naive bayes classifiers. *University of British Columbia*, 18, 2006.

[235] Yoav Freund and Robert Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[236] Robert Schapire. *Explaining adaboost.* Springer, 2013.

[237] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society, Series B*, 39(1):1–38, 1977.

[238] Seyed Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[239] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE, 1995.

[240] Iñigo Barandiaran. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[241] Andy Liaw. Package 'randomforest'. 2018.

[242] RANDOM FORESTS Trademark of Health Care Productivity, Inc. - Registration Number 3185828 - Serial Number 78642027 :: Justia Trademarks. `https://trademarks.justia.com/786/42/random-78642027.html`. [Online; accessed 2018-11-07].

[243] Brijain Patel and Kushik Rana. A survey on decision tree algorithm for classification. 2014.

[244] Harry Zhang. Exploring conditions for the optimality of naive bayes. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(02):183–198, 2005.

[245] Kalid Azad. An intuitive (and short) explanation of bayes' theorem, 2007.

[246] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[247] scikit-learn developers. 2.7. Novelty and Outlier Detection. `http://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection`. [Online; accessed 2017-11-07].

[248] Mark Aizerman. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

[249] Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[250] Xiao-dan Wang and Ji-qin Wang. Research and application of support vector machine. *Journal of Air Force Engineering University (Natural Science Edition)*, 3:013, 2004.

[251] Zhijie Liu, Xueqiang Lv, Kun Liu, and Shuicai Shi. Study on svm compared with the other text classification methods. In *2010 Second International Workshop on Education Technology and Computer Science*, volume 1, pages 219–222. IEEE, 2010.

[252] David Crisp and Christopher Burges. A geometric interpretation of v-svm classifiers. In *Advances in Neural Information Processing Systems 12*, pages 244–250. MIT Press, 2000.

[253] Christopher Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[254] Edgar Osuna and Federico Girosi. Reducing the run-time complexity of support vector machines. In *International Conference on Pattern Recognition (submitted)*, 1998.

[255] Chih-Chung Chang. "libsvm: a library for support vector machines,äcm transactions on intelligent systems and technology, 2: 27: 1–27: 27, 2011. *http://www. csie. ntu. edu. tw/˜ cjlin/libsvm*, 2, 2011.

[256] OpenCV library. `https://opencv.org/`. [Online; accessed 2014-04-01].

[257] Hao Guan, William Smith, and Peng Ren. Corner detection in spherical images via the accelerated segment test on a geodesic grid. In *International Symposium on Visual Computing*, pages 407–415. Springer, 2013.

[258] Krishna Singh, Indra Gupta, and Sangeeta Gupta. Svm-bdt pnn and fourier moment technique for classification of leaf shape. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 3(4):67–78, 2010.

[259] N. Valliammal and S.N. Geethalakshmi. An amalgam approach for feature extraction and classification of leaves using support vector machine. In *Advances in Computer Science, Engineering & Applications*, pages 847–855. Springer, 2012.

[260] Lamis Hamrouni, Ramla Bensaci, Mohammed Lamine Kherfi, Belal Khaldi, and Oussama Aiadi. Automatic recognition of plant leaves using parallel combination of classifiers. In *Computational Intelligence and its Applications*, pages 597–606. Springer, 2018.

[261] William Hawthorne and Anna Lawrence. *Plant identification: creating user-friendly field guides for biodiversity management.* Routledge, 2013.

[262] James Castner. *Photographic atlas of botany and guide to plant identification.* Ingram, 2004.

[263] James Cullen, Sabina Knees, Suzanne Cubey, and J. M. H. Shaw. *The European garden flora flowering plants: a manual for the identification of plants cultivated in Europe, both out-of-doors and under glass*, volume 1. Cambridge University Press, 2011.

[264] Reverse image search. `http://research.omicsgroup.org/index.php/Reverse_image_search#cite_note-1`. [Online; accessed 2018-11-01].

[265] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* Ö'Reilly Media, Inc.", 2008.

[266] David Picard and Philippe-Henri Gosselin. Improving image similarity with vectors of locally aggregated tensors. In *2011 18th IEEE International Conference on Image Processing*, pages 669–672. IEEE, 2011.

[267] Xin Zhao, Yinan Yu, Yongzhen Huang, Kaiqi Huang, and Tieniu Tan. Feature coding via vector difference for image classification. In *2012 19th IEEE International Conference on Image Processing*, pages 3121–3124. IEEE, 2012.

[268] Masoud Fathi Kazerouni, Jens Schlemper, and Klaus-Dieter Kuhnert. Automatic plant recognition system for challenging natural plant species. 2017.

[269] Duong-Van Nguyen. Vegetation detection and terrain classification for autonomous navigation. 2013.

[270] Leaf Morphology: Shape. `http://forestry.sfasu.edu/faculty/stovall/dendrology/index.php/fact-sheets-sp-916/morphology-photos/465-leaf-morphology-shape`. [Online; accessed 2018-08-07].

[271] Abdurrasyid Hasim, Yeni Herdiyeni, and Stephane Douady. Leaf shape recognition using centroid contour distance. In *IOP Conference Series: Earth and Environmental Science*. IOP Publishing, 2016.

[272] Trishen Munisami, Mahess Ramsurn, Somveer Kishnah, and Sameerchand Pudaruth. Plant leaf recognition using shape features and colour histogram with k-nearest neighbour classifiers. *Procedia Computer Science*, 58:740–747, 2015.

[273] Xiao-Yuan Jing and David Zhang. A face and palmprint recognition approach based on discriminant dct feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(6):2405–2415, 2004.

[274] Andrzej Ruta, Yongmin Li, and Xiaohui Liu. Real-time traffic sign recognition from video by class-specific discriminative features. *Pattern Recognition*, 43(1):416–430, 2010.

[275] Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Context-aware local binary feature learning for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1139–1153, 2018.

[276] Alberto Del Bimbo, Pietro Pala, and Simone Santini. Image retrieval by elastic matching of shapes and image patterns. In *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, pages 215–218. IEEE, 1996.

[277] Rajiv Mehrotra and James Gary. Feature-based retrieval of similar shapes. In *Proceedings of IEEE 9th International Conference on Data Engineering*, pages 108–115. IEEE, 1993.

[278] Wayne Niblack and John Yin. A pseudo-distance measure for 2d shapes based on turning angle. In *Proceedings., International Conference on Image Processing*, volume 3, pages 352–355. IEEE, 1995.

[279] Eli Saber and Murat Tekalp. Image query-by-example using region-based shape matching. In *Image and Video Processing IV*, volume 2666, pages 200–212. International Society for Optics and Photonics, 1996.

[280] Stan Sclaroff. Deformable prototypes for encoding shape categories in image databases. *Pattern Recognition*, 30(4):627–641, 1997.

[281] Stan Sclaroff and Alex Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, 1995.

[282] Sadegh Abbasi, Farzin Mokhtarian, and Josef Kittler. Reliable classification of chrysanthemum leaves through curvature scale space. In *International Conference on Scale-Space Theories in Computer Vision*, pages 284–295. Springer, 1997.

[283] Chia-Ling Lee and Shu-Yuan Chen. Classification of leaf images. *International Journal of Imaging Systems and Technology*, 16(1):15–23, 2006.

[284] Fernando Gouveia, Vitor Filipe, Manuel Reis, Carlos Couto, and Jose Bulas-Cruz. Biometry: the characterisation of chestnut-tree leaves using computer vision. In *ISIE '97 Proceeding of the IEEE International Symposium on Industrial Electronics*, pages 757–760. IEEE, 1997.

[285] Hossam Zawbaa, Mona Abbass, Sameh Basha, Maryam Hazman, and Abul Ella Hassenian. An automatic flower classification approach using machine learning algorithms. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 895–901. IEEE, 2014.

[286] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using sift features and mean shift. *Computer Vision and Image Understanding*, 113(3):345–352, 2009.

[287] Xuelong Hu, Yingcheng Tang, and Zhenghua Zhang. Video object matching based on sift algorithm. In *2008 International Conference on Neural Networks and Signal Processing*, pages 412–415. IEEE, 2008.

[288] Jing Liu, Fang Meng, Fangcheng Mu, and Yichun Zhang. An improved image retrieval method based on sift algorithm and saliency map. In *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 766–770. IEEE, 2014.

[289] Neeraj Kumar, Peter Belhumeur, Arijit Biswas, David Jacobs, John Kress, Ida Lopez, and João Soares. Leafsnap: A computer vision system for automatic plant species identification. In *European Conference on Computer Vision*, pages 502–516. Springer, 2012.

[290] Andrew Rabinovich, Andrea Vedaldi, and Serge Belongie. *Does image segmentation improve object categorization?* Department of Computer Science and Engineering, University of California, San Diego, 2007.

[291] Mary Nielsen and Michael Stokes. The creation of the srgb icc profile. In *Color and Imaging Conference*, volume 1998, pages 253–257. Society for Imaging Science and Technology, 1998.

[292] Yoshi Ohno. Cie fundamentals for color measurements. *International Conference on Digital Printing Technologies*, 2000(2):540–545, 2000.

[293] N. Valliammal and S. N. Geethalakshmi. Plant leaf segmentation using non linear k means clustering. *International Journal of Computer Science Issues (IJCSI)*, 9(3), 2012.

[294] Chin-Hung Teng, Yi-Ting Kuo, and Yung-Sheng Chen. Leaf segmentation, classification, and three-dimensional recovery from a few images with close viewpoints. *Optical Engineering*, 50(3), 2011.

[295] Dalcimar Casanova, Joao Batista Florindo, Wesley Nunes Gonçalves, and Odemir Martinez Bruno. Ifsc/usp at imageclef 2012: Plant identification task. In *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[296] Berrin Yanikoglu, Erchan Aptoula, and Caglar Tirkaz. Automatic plant identification from photographs. *Machine Vision and Applications*, 25(6):1369–1383, 2014.

[297] João Camargo Neto, George Meyer, and David Jones. Individual leaf extractions from young canopy images using gustafson–kessel clustering and a genetic algorithm. *Computers and Electronics in Agriculture*, 51(1-2):66–85, 2006.

[298] Guillaume Cerutti, Laure Tougne, Julien Mille, Antoine Vacavant, and Didier Coquin. Understanding leaves in natural images–a model-based approach for tree species identification. *Computer Vision and Image Understanding*, 117(10):1482–1501, 2013.

[299] Andreas Kårsnäs, Robin Strand, and Punam Saha. The vectorial minimum barrier distance. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 792–795. IEEE, 2012.

[300] Robin Strand, Krzysztof Chris Ciesielski, Filip Malmberg, and Punam Saha. The minimum barrier distance. *Computer Vision and Image Understanding*, 117(4):429–437, 2013.

[301] Filip Malmberg, Robin Strand, Joel Kullberg, Richard Nordenskjöld, and Ewert Bengtsson. Smart paint a new interactive segmentation method applied to mr prostate segmentation. *MICCAI Grand Challenge: Prostate MR Image Segmentation*, 2012, 2012.

[302] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[303] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[304] Pedro Felzenszwalb and Daniel Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[305] Gonzalo Ruz, Pablo Estevez, and Claudio Perez. A neurofuzzy color image segmentation method for wood surface defect detection. *Forest Products Journal*, 55(4):52–58, 2005.

[306] Ali Moghaddamzadeh and Nikolaos Bourbakis. A fuzzy region growing approach for segmentation of color images. *Pattern Recognition*, 30(6):867–881, 1997.

[307] James Bezdek. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1):1–8, 1980.

[308] Diane Larlus and Frédéric Jurie. Latent mixture vocabularies for object categorization. In *Proceedings of the British Machine Vision Conference 2006*, volume 3, pages 959–968. The British Machine Vision Association, 2006.

[309] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[310] Demir Gokalp and Selim Aksoy. Scene classification using bag-of-regions representations. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[311] CIE 1931 color space. `https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/cie-1931-color-space`. [Online; accessed 2018-04-03].

[312] Simone Bianco, Davide Mazzini, Danilo Pietro Pau, and Raimondo Schettini. Local detectors and compact descriptors for visual search: a quantitative comparison. *Digital Signal Processing*, 44:1–13, 2015.

[313] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[314] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 143–151. Morgan Kaufmann Publishers Inc., 1997.

[315] David Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *European Conference on Machine Learning*, pages 4–15. Springer, 1998.

[316] Azizi Abdullah, Remco Veltkamp, and Marco Wiering. Ensembles of novel visual keywords descriptors for image categorization. In *2010 11th International Conference on Control Automation Robotics Vision*, pages 1206–1211. IEEE, 2010.

[317] Jason Farquhar, Sandor Szedmak, Hongying Meng, and John Shawe-Taylor. Improving "bag-of-keypointsïmage categorisation: Generative models and pdf-kernels. *Univ. Southampton*, 68, 2005.

[318] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178. IEEE, 2006.

[319] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[320] Juan Caicedo, Angel Cruz, and Fabio Gonzalez. Histopathology image classification using bag of features and kernel functions. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 126–135. Springer, 2009.

[321] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, 2010.

[322] Mahmudur Rahman, Sameer Antani, and George Thoma. Biomedical cbir using "bag of keypoints" in a modified inverted index. In *2011 24th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 1–6. IEEE, 2011.

[323] Jingyan Wang, Yongping Li, Ying Zhang, Chao Wang, Honglan Xie, Guoling Chen, and Xin Gao. Notice of violation of ieee publication principles bag-of-features based medical image retrieval via multiple assignment and visual words weighting. *IEEE Transactions on Medical Imaging*, 30(11):1996–2011, 2011.

[324] Jingyan Wang, Yongping Li, Ying Zhang, Honglan Xie, and Chao Wang. Boosted learning of visual word weighting factors for bag-of-features based medical image retrieval. In *2011 Sixth International Conference on Image and Graphics*, pages 1035–1040. IEEE, 2011.

[325] Abe Shigeo. Support vector machines for pattern classification. *Advances in Computer Vision and Pattern Recognition*, 2005.

[326] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Introduction to data mining: Pearson addison wesley. *Boston*, 2005.

[327] Jin-Hyuk Hong and Sung-Bae Cho. A probabilistic multi-class strategy of one-vs.-rest support vector machines for cancer classification. *Neurocomputing*, 71(16-18):3275–3281, 2008.

[328] Kernel Functions-Introduction to SVM Kernel & Examples. `https://data-flair.training/blogs/svm-kernel-functions/`. [Online; accessed 2018-04-03].

[329] César Souza. Kernel functions for machine learning applications. *Creative Commons Attribution-Noncommercial-Share Alike*, 3:29, 2010.

[330] Daphne Koller, Nir Friedman, and Francis Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[331] Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen Walker. *Bayesian nonparametrics*, volume 28. Cambridge University Press, 2010.

[332] Shitala Prasad, Krishna Mohan Kudiri, and Ramesh Chandra Tripathi. Relative sub-image based features for leaf recognition using support vector machine. In *Proceedings of the 2011 International Conference on Communication, Computing & Security*, pages 343–346. ACM, 2011.

[333] Yuan Tian, Chunjiang Zhao, Shenglian Lu, and Xinyu Guo. Svm-based multiple classifier system for recognition of wheat leaf diseases. In *World Automation Congress 2012*, pages 189–193. IEEE, 2012.

[334] Performance Metrics for Classification problems in Machine Learning. `https://medium.com/thalus-ai`. [Online; accessed 2018-08-26].

[335] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Machine Learning Technology*, 2, 2011.

[336] Yanwei Pang, Wei Li, Yuan Yuan, and Jing Pan. Fully affine invariant surf for image matching. *Neurocomputing*, 85:6–10, 2012.

[337] Itamar Arel, Derek Rose, and Thomas Karnowski. Deep machine learning-a new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine*, 5(4):13–18, 2010.

[338] Scott Neslin. Customer relationship management (crm). In *The History of Marketing Science*, pages 289–317. World Scientific, 2014.

[339] Yann LeCun. Une procedure d'apprentissage ponr reseau a seuil asymetrique. *Proceedings of Cognitiva 85*, pages 599–604, 1985.

[340] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[341] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009.

[342] Convolutional Neural Networks (LeNet). `http://deeplearning.net/tutorial/lenet.html`. [Online; accessed 2017-04-14].

[343] Geoffrey Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[344] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, volume 2, pages 1045–1048, 2010.

[345] Geoffrey Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems*, pages 3–10. Morgan-Kaufmann, 1994.

[346] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10):255–258, 1998.

[347] Steve Lawrence, Lee Giles, Ah Chung Tsoi, and Andrew Back. Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[348] A weird introduction to Deep Learning. `https://towardsdatascience.com/a-weird-introduction-to-deep-learning-7828803693b0`. [Online; accessed 2018-06-14].

[349] Warren McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[350] Ling Zhang and Bo Zhang. A geometrical representation of mcculloch-pitts neural model and its applications. *IEEE Transactions on Neural Networks*, 10(4):925–929, 1999.

[351] Computing Machinery. Computing machinery and intelligence-am turing. *Mind*, 59(236):433, 1950.

[352] Frank Rosenbaltt. The perceptron–a perceiving and recognizing automation. Technical report, Report 85-460-1 Cornell Aeronautical Laboratory, Ithaca, 1957.

[353] Henry Kelley. Gradient theory of optimal flight paths. *ARS Journal*, 30(10):947–954, 1960.

[354] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*, pages 267–285. Springer, 1982.

[355] John Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[356] David Rumelhart, Geoffrey Hinton, and Ronald Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

[357] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[358] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[359] A History of Deep Learning. `https://www.import.io/post/history-of-deep-learning/`. [Online; accessed 2018-10-01].

[360] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160. MIT Press, 2007.

[361] Alex Krizhevsky. `https://www.cs.toronto.edu/~kriz/`. [Online; accessed 2018-10-01].

[362] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[363] Yann Le Cun, Lionel Jackel, Brian Boser, John Denker, Henry Graf, Isabelle Guyon, Don Henderson, Richard Howard, and William Hubbard. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46, 1989.

[364] Yann LeCun. Generalization and network design strategies. *Connectionism in Perspective*, pages 143–155, 1989.

[365] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404, 1990.

[366] Bernhard Boser, Eduard Sackinger, Jane Bromley, Yann Le Cun, and Lawrence Jackel. An analog neural network processor with programmable topology. *IEEE Journal of Solid-State Circuits*, 26(12):2017–2025, 1991.

[367] Ofer Matan, Christopher Burges, Yann LeCun, and John Denker. Multi-digit recognition using a space displacement neural network. In *Advances in Neural Information Processing Systems*, pages 488–495, 1992.

[368] Régis Vaillant, Christophe Monrocq, and Yann Le Cun. Original approach for the localisation of objects in images. *IEE Proceedings - Vision, Image and Signal Processing*, 141(4):245–250, 1994.

[369] Yann LeCun, Lawrence Jackel, Léon Bottou, Corinna Cortes, John Denker, Harris Drucker, Isabelle Guyon, Urs Muller, Eduard Sackinger, Patrice Simard, and Vladimir Vapnik. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural Networks: the Statistical Mechanics Perspective*, 261:276, 1995.

[370] Yann Le Cun, Leon Bottou, and Yoshua Bengio. Reading checks with multilayer graph transformer networks. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 151–154. IEEE, 1997.

[371] Léon Bottou, Yoshua Bengio, and Yann Le Cun. Global training of document processing systems using graph transformer networks. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 489–494. IEEE, 1997.

[372] Geoffrey Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[373] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[374] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680. Curran Associates, Inc., 2014.

[375] KDnuggets. `https://www.kdnuggets.com/2016/08/yann-lecun-quora-session.html`. [Online; accessed 2018-10-01].

[376] Sara Sabour, Nicholas Frosst, and Geoffrey Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866. Curran Associates, Inc., 2017.

[377] Christoph Käding, Erik Rodner, Alexander Freytag, and Joachim Denzler. Fine-tuning deep neural networks in continuous learning scenarios. In *Asian Conference on Computer Vision*, pages 588–605. Springer, 2016.

[378] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 693–700. PMLR, 2010.

[379] Vinod Nair and Geoffrey Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807–814. Omnipress, 2010.

[380] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising autoencoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907. Curran Associates, Inc., 2013.

[381] Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In *Predicting structured data*. MIT Press, 2006.

[382] Li Deng, Dong Yu, and John Platt. Scalable stacking and learning for building deep architectures. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2133–2136. IEEE, 2012.

[383] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J. Lang. Phoneme recognition using time-delay neural networks. In *Readings in Speech Recognition*, pages 393–404. Elsevier, 1990.

[384] Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.

[385] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[386] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Hybrid deep learning for face verification. In *2013 IEEE International Conference on Computer Vision*, pages 1489–1496, 2013.

[387] Guoqiang Zhong, Li-Na Wang, Xiao Ling, and Junyu Dong. An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*, 2(4):265–278, 2016.

[388] Wei Zhang. Shift-invariant pattern recognition neural network and its optical architecture. In *Proceedings of annual conference of the Japan Society of Applied Physics*, 1988.

[389] Wei Zhang, Kazuyoshi Itoh, Jun Tanida, and Yoshiki Ichioka. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied Optics*, 29(32):4790–4797, 1990.

[390] David Hubel and Torsten Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[391] Alex Waibel, Hidefumi Sawai, and Kiyohiro Shikano. Modularity and scaling in large phonemic neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):1888–1898, 1989.

[392] Léon Bottou, Fogelman Soulié, Pascal Blanchet, and Jean-Sylvain Lienard. Experiments with time delay networks and dynamic time warping for speaker independent isolated digits recognition. In *First European Conference on Speech Communication and Technology*, 1989.

[393] Patrice Simard, Dave Steinkraus, and John Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963. IEEE, 2003.

[394] Steven Nowlan and John Platt. A convolutional neural network hand tracker. *Advances in Neural Information Processing Systems*, pages 901–908, 1995.

[395] Dave Steinkraus, Ian Buck, and Patrice Simard. Using gpus for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, volume 2, pages 1115–1120. IEEE, 2005.

[396] Kumar Chellapilla, Sidd Puri, and Patrice Simard. High performance convolutional neural networks for document processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006.

[397] Christopher Poultney, Sumit Chopra, and Yann Cun. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems*, pages 1137–1144, 2007.

[398] Jia Deng, Alex Berg, Sanjeev Satheesh, Hao Su, Aditya Khosla, and Li Fei-Fei. Imagenet large scale visual recognition competition 2012 (ilsvrc2012). *See net.org/challenges/LSVRC*, 2012.

[399] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. 2013.

[400] Matthew Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[401] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[402] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. IEEE, 2015.

[403] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[404] BBC News Services. Google AI defeats human Go champion. `https://www.bbc.com/news/technology-40042581`. [Online; accessed 2019-06-22].

[405] CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more .... `https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5`. [Online; accessed 2018-10-01].

[406] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[407] Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[408] Yuehao Pan, Weimin Huang, Zhiping Lin, Wanzheng Zhu, Jiayin Zhou, Jocelyn Wong, and Zhongxiang Ding. Brain tumor grading based on neural networks and convolutional neural networks. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 699–702. IEEE, 2015.

[409] Wei Shen, Mu Zhou, Feng Yang, Caiyun Yang, and Jie Tian. Multi-scale convolutional neural networks for lung nodule classification. In *International Conference on Information Processing in Medical Imaging*, pages 588–599. Springer, 2015.

[410] Gustavo Carneiro, Jacinto Nascimento, and Andrew Bradley. Unregistered multiview mammogram analysis with pre-trained deep learning models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 652–660. Springer, 2015.

[411] Jelmer Wolterink, Tim Leiner, Bob de Vos, Robbert van Hamersvelt, Max Viergever, and Ivana Išgum. Automatic coronary artery calcium scoring in cardiac ct angiography using paired convolutional neural networks. *Medical Image Analysis*, 34:123–136, 2016.

[412] Yu-Yun Dai and Robert Braytont. Circuit recognition with deep learning. In *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 162–162. IEEE, 2017.

[413] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[414] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European Conference on Computer Vision*, pages 512–528. Springer, 2014.

[415] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1274. IEEE, 2015.

[416] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 647–655. PMLR, 2014.

[417] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732. IEEE, 2014.

[418] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

[419] Kumar Chellapilla and David Fogel. Evolving neural networks to play checkers without relying on expert knowledge. *IEEE Transactions on Neural Networks*, 10(6):1382–1391, 1999.

[420] Yoon Kim. Convolutional neural networks for sentence classification. pages 1746–1751, 2014.

[421] Welcome to Deep Learning. `http://www.deeplearning.net/`. [Online; accessed 2018-05-24].

[422] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 4898–4906. Curran Associates, Inc., 2016.

[423] Convolutional Neural Networks (CNNs / ConvNets). `http://cs231n.github.io/convolutional-networks/`. [Online; accessed 2018-05-23].

[424] Hind Taud and Jean-Francois Mas. Multilayer perceptron (mlp). In *Geomatic Approaches for Modeling Land Change Scenarios*, pages 451–455. Springer, 2018.

[425] Caffe. `http://caffe.berkeleyvision.org/doxygen/classcaffe_1_1SoftmaxWithLossLayer.html`. [Online; accessed 2018-05-23].

[426] Caffe-BAIR. Sigmoid cross-entropy loss layer. `http://caffe.berkeleyvision.org/doxygen/classcaffe_1_1SigmoidCrossEntropyLossLayer.html`. [Online; accessed 2019-06-22].

[427] Local Response Normalization (LRN). `http://Caffe.berkeleyvision.org/tutorial/layers/lrn.html`. [Online; accessed 2018-05-28].

[428] Park Chunduck-Hallym University. Caffe framework tutorial2. `https://de.slideshare.net/ssuserf45ab2/caffe-framework-tutorial2`. [Online; accessed 2019-06-22].

[429] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[430] Adi Livnat, Christos Papadimitriou, Nicholas Pippenger, and Marcus Feldman. Sex, mixability, and modularity. *Proceedings of the National Academy of Sciences*, 107(4):1452–1457, 2010.

[431] Robert Bell and Yehuda Koren. Lessons from the netflix prize challenge. *SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[432] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the Trade*, pages 421–436. Springer, 2012.

[433] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[434] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[435] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[436] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017.

[437] Solver Prototxt. https://github.com/BVLC/Caffe/wiki/Solver-Prototxt. [Online; accessed 2018-06-01].

[438] Product Specifications. https://ark.intel.com/products/80807/Intel-Core-i7-4790K-Processor-8M-Cache-up-to-4_40-GHz. [Online; accessed 2018-06-07].

[439] Specifications Geforce GTX 760. https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-760/specifications. [Online; accessed 2018-06-07].

[440] NVIDIA CUDA: Kepler Vs. Fermi Architecture. http://blog.cuvilib.com/2012/03/28/nvidia-cuda-kepler-vs-fermi-architecture/. [Online; accessed 2018-06-07].

[441] User Benchmark. http://gpu.userbenchmark.com/Compare/Nvidia-GTX-760-vs-Nvidia-GTX-580/2159vs3150. [Online; accessed 2018-06-07].

[442] World population projected to reach 9.8 billion in 2050, and 11.2 billion in 2100 | UN DESA Department of Economic and Social Affairs. https://www.un.org/development/desa/en/news/population/world-population-prospects-2017.html. [Online; accessed 2018-06-07].

[443] Universität Siegen. https://www.uni-siegen.de/start/. [Online; accessed 2018-11-23].

[444] An in-depth look at Google's first Tensor Processing Unit (TPU) | Google Cloud Blog. https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu. [Online; accessed 2018-06-06].

[445] Pricing | Cloud TPU | Google Cloud. https://cloud.google.com/tpu/docs/pricing. [Online; accessed 2018-06-08].

[446] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[447] Intel FPGAs Accelerate Artificial Intelligence for Deep Learning in Microsoft's Bing Intelligent Search. https://newsroom.intel.com/editorials. [Online; accessed 2018-06-09].

[448] Autonomous Driving and ADAS (Advanced Driver Assistance Systems). https://www.mobileye.com. [Online; accessed 2018-06-09].

[449] Convolutional Neural Networks (CNNs): An Illustrated Explanation. https://xrds.acm.org/blog/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/. [Online; accessed 2018-06-09].

[450] Yann LeCun's Home Page. http://yann.lecun.com/. [Online; accessed 2018-06-10].

[451] Matthew Zeiler, Dilip Krishnan, Graham Taylor, and Rob Fergus. Deconvolutional networks. 2010.

[452] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621. Curran Associates, Inc., 2016.

[453] Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. http://3dgan.csail.mit.edu/. [Online; accessed 2018-06-10].

[454] Kevin Schawinski, Ce Zhang, Hantian Zhang, Lucas Fowler, and Gokula Krishnan Santhanam. Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Monthly Notices of the Royal Astronomical Society: Letters*, 467(1):L110–L114, 2017.

[455] Mehdi Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4501–4510. IEEE, 2017.

[456] Strawberries Harvester. http://agrobot.com/. [Online; accessed 2018-06-14].

[457] Autonomous Oz weeding robot. https://www.naio-technologies.com/en/agricultural-equipment/weeding-robot-oz/. [Online; accessed 2018-06-15].

[458] VINBOT. http://vinbot.eu/. [Online; accessed 2018-06-15].

[459] Robots in Agriculture. https://www.intorobotics.com/35-robots-in-agriculture/. [Online; accessed 2018-06-16].

[460] SORTER- polskie maszyny sortownicze. http://www.sorter.pl/en. [Online; accessed 2018-06-16].

[461] Björn Åstrand and Albert-Jan Baerveldt. An agricultural mobile robot with vision-based perception for mechanical weed control. *Autonomous robots*, 13(1):21–35, 2002.

[462] Ulrich Weiss, Peter Biber, Stefan Laible, Karsten Bohlmann, and Andreas Zell. Plant species classification using a 3d lidar sensor and machine learning. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 339–345. IEEE, 2010.

[463] Weka 3: Data Mining Software in Java. https://www.cs.waikato.ac.nz/ml/weka/. [Online; accessed 2018-06-16].

[464] Asterix. https://www.adigo.no/portfolio/asterix/?lang=en. [Online; accessed 2018-06-16].

[465] Adigo AS. https://www.adigo.no/. [Online; accessed 2018-06-17].

[466] Øystein Grændsen. Automatic visual weed recognition-detection and classification of weed in row cultures combining machine vision and artificial intelligence. Master's thesis, NTNU, 2014.

[467] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2229–2235. IEEE, 2018.

[468] BoniRob - Multipurpose farm robot / weeding by Bosch Deepfield Robotics | AgriExpo. http://www.agriexpo.online/prod/bosch-deepfield-robotics/product-168586-1199.html. [Online; accessed 2018-06-17].

[469] David Ribeiro, André Mateus, Pedro Miraldo, and Jacinto Nascimento. A real-time deep learning pedestrian detector for robot navigation. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 165–171. IEEE, 2017.

[470] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, 2014.

[471] Eric Martinson and Veera Ganesh Yalla. Real-time human detection for robots using cnn with a feature-based layered pre-filter. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1120–1125. IEEE, 2016.

[472] Xiyang Song, Huangwei Fang, Xiong Jiao, and Ying Wang. Autonomous mobile robot navigation using machine learning. In *2012 IEEE 6th International Conference on Information and Automation for Sustainability*, pages 135–140. IEEE, 2012.

[473] Gulam Amer, Syed Mujtaba Mahdi Mudassir, and Abdul Malik. Design and operation of wi-fi agribot integrated system. In *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pages 207–212. IEEE, 2015.

[474] Ji Li and Lie Tang. Crop recognition under weedy conditions based on 3d imaging for robotic weed control. *Journal of Field Robotics*, 35(4):596–611, 2018.

[475] Ji Li. 3d machine vision system for robotic weeding and plant phenotyping. 2014.

[476] Megha Arakeri, Vijaya Kumar, Shubham Barsaiya, and H. V. Sairam. Computer vision based robotic weed control system for precision agriculture. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1201–1205. IEEE, 2017.

[477] Agricultural Robots. `https://www.tractica.com/research/agricultural-robots/`. [Online; accessed 2018-06-18].

[478] Mohammad Ali Jan Ghasab, Shamsul Khamis, Faruq Mohammad, and Hessam Jahani Fariman. Feature decision-making ant colony optimization system for an automated recognition of plant species. *Expert Systems with Applications*, 42(5):2361–2370, 2015.

[479] Arun Kumar, Sam Emmanuel, and Christopher Durairaj. Texture feature extraction for identification of medicinal plants and comparison of different classifiers. *International Journal of Computer Applications*, 62(12):1–9, 2013.

[480] The Grey Level Co-occurrence Matrix, GLCM (also called the Grey Tone Spatial Dependency Matrix). `http://www.ucalgary.ca/mhallbey/glcm1`. [Online; accessed 2018-06-19].

[481] Abdul Kadir, Lukito Edi Nugroho, Adhi Susanto, and Paulus Insap Santosa. Experiments of zernike moments for leaf identification. *Journal of Theoretical and Applied Information Technology (JATIT)*, 41(1):82–93, 2012.

[482] Amir Tahmasbi, Fatemeh Saki, and Shahriar Shokouhi. An effective breast mass diagnosis system using zernike moments. In *2010 17th Iranian Conference of Biomedical Engineering (ICBME)*, pages 1–4. IEEE, 2010.

[483] Chengzhuan Yang, Hui Wei, and Qian Yu. Multiscale triangular centroid distance for shape-based plant leaf recognition. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, pages 269–276. IOS Press, 2016.

[484] Hong Jeon, Lei Tian, and Heping Zhu. Robust crop and weed segmentation under uncontrolled outdoor illumination. *Sensors*, 11(6):6270–6283, 2011.

[485] Ciro Potena, Daniele Nardi, and Alberto Pretto. Fast and accurate crop and weed identification with summarized train sets for precision agriculture. In *International Conference on Intelligent Autonomous Systems*, pages 105–121. Springer, 2016.

[486] Amir Kargar and Ali Shirzadifar. Automatic weed detection system and smart herbicide sprayer robot for corn fields. In *2013 First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*, pages 468–473. IEEE, 2013.

[487] Klaus Müller, Charam Ram Akupati, Felix Graf, Yuwei Guo, Sven Höhn, Jan-Marco Hütwohl, Thomas Köther, Samir Nezir Osman, Goetz Poenaru, Saeid Sedighi, Jan-Friedrich Schlemper, Whangyi Zhu, Jan Kunze, and Klaus-Dieter Kuhnert. Zephyr - university of siegen, germany. In Jurij Rakun, editor, *Proceedings of the 13th Field Robot Event 2015*, pages 122–132. University of Maribor : Faculty of Agriculture and Life Sciences, Department of Biosystems Engineering, 2015, 2015.

[488] Jan Kunze, Simon Hardt, Sebastian Zeller, Sven Höhn, Daniel Patoka, Oliver Tiebe, Matthias Kölsch, and Klaus-Dieter Kuhnert. Zephyr. In *14th Field Robot Event 2016*, pages 154–164, 2016.

[489] Naftali Tishby, Fernando Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, pages 368–377, 2000.

[490] New Theory Cracks Open the Black Box of Deep Learning. `https://www.quantamagazine.org/new-theory-cracks-open-the-black-box-of-deep-learning-20170921/`. [Online; accessed 2018-07-02].

[491] Shu-e YANG and Li Huang. Financial crisis warning model based on bp neural network. *Systems Engineering-theory and Practice*, 1:12–18, 2005.

[492] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307, 2016.

[493] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[494] Mingliang Xu, Jiejie Zhu, Pei Lv, Bing Zhou, Marshall Tappen, and Rongrong Ji. Learning-based shadow recognition and removal from monochromatic natural images. *IEEE Transactions on Image Processing*, 26(12):5811–5824, 2017.

[495] Srinivasa Narasimhan and Shree Nayar. Removing weather effects from monochrome images. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001.

[496] Farhan Hussain and Jechang Jeong. Visibility enhancement of scene images degraded by foggy weather conditions with deep neural networks. *Journal of Sensors*, 2016, 2016.

[497] Optronis CamPerform. `https://www.stemmer-imaging.com/en-gb/products/series/optronis-camperform/`. [Online; accessed 2018-07-23].

[498] Suleyman Ozarslan, Erhan Eren, Xinzhi Wang, Vijayan Sugumaran, Hui Zhang, and Zheng Xu. Resource description framework (rdf) is a commonly used format for semantic web processing. it basically contains strings representing items and their relationships which can be queried or inferred. in this paper, we propose a framework for processing large rdf data sets. it is based on brute-force string matching on gpus (bfg). graphics processing units (gpus) are used as a parallel platform that... *Information Systems Frontiers*, 20(4):863–882, 2018.

[499] Nazeer T. Mohammed Saeed, Masoud Fathi Kazerouni, Madjid Fathi, and Klaus-Dieter Kuhnert. Robot semantic protocol (robosemproc) for semantic environment description and human-robot communication. *International Journal of Social Robotics, Springer*, pages 1–14, 2019.

[500] Peng Xu and Dezhong Yao. A study on medical image registration by mutual information with pyramid data structure. *Computers in Biology and Medicine*, 37(3):320–327, 2007.

[501] Xun Wang and Jian-Qiu Jin. An edge detection algorithm based on improved canny operator. In *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, pages 623–628. IEEE, 2007.

[502] Sobel Edge Detector. http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm. [Online; accessed 2018-08-07].

[503] Xu Li, Zheng-yong Wang, Xiao-hong Wu, and Qi-zhi Teng. An canny edge detection algorithm base on improved non-maximum suppression. *Journal of Chengdu University of Information Technology*, 5, 2011.

[504] Harpreet Kaur and Lakhwinder Kaur. Performance comparison of different feature detection methods with gabor filter. *International Journal of Science and Research (IJSR)*, 3:1880–1886, 2014.

[505] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Graph cut based image segmentation with connectivity priors. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[506] Ashutosh Saxena, Min Sun, and Andrew Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009.

[507] Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems*, pages 666–674, 2011.

[508] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[509] Convolutional Neural Network(CNN). http://www.renom.jp/notebooks/tutorial/basic_algorithm/convolutional_neural_network/notebook.html. [Online; accessed 2018-05-23].

[510] Welcome. http://www.deeplearning.net/software/theano/. [Online; accessed 2018-05-24].

[511] Yoshua Bengio. https://www.iro.umontreal.ca/~bengioy/yoshua_en/index.html. [Online; accessed 2018-05-24].

[512] MILA. https://mila.quebec/en/. [Online; accessed 2018-05-24].

[513] Battle of the Deep Learning frameworks - Part I: 2017, even more frameworks and interfaces. https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750. [Online; accessed 2018-05-24].

[514] Welcome to Lasagne. https://lasagne.readthedocs.io/en/latest/. [Online; accessed 2018-05-24].

[515] Amazon Web Services (AWS) - Cloud Computing Services. https://aws.amazon.com/. [Online; accessed 2018-05-24].

[516] What is Torch? http://torch.ch/. [Online; accessed 2018-05-24].

[517] PyTorch. https://pytorch.org/. [Online; accessed 2018-05-24].

[518] The Programming Language Lua. https://www.lua.org/. [Online; accessed 2018-05-24].

[519] The LuaJIT Project. http://luajit.org/. [Online; accessed 2018-05-24].

[520] TensorFlow. https://www.tensorflow.org/. [Online; accessed 2018-05-24].

[521] Brain Team – Google AI. https://ai.google/research/teams/brain. [Online; accessed 2018-05-24].

[522] A Comparison of Deep Learning Frameworks. https://www.exastax.com/deep-learning/a-comparison-of-deep-learning-frameworks/. [Online; accessed 2018-05-24].

[523] Keras: The Python Deep Learning library. https://keras.io/. [Online; accessed 2018-05-24].

[524] François Chollet. https://scholar.google.com/citations?user=VfYhf2wAAAAJ&hl=en. [Online; accessed 2018-05-24].

[525] The R Project for Statistical Computing. https://www.r-project.org/. [Online; accessed 2018-05-25].

[526] Caffe. http://Caffe.berkeleyvision.org/. [Online; accessed 2018-05-24].

[527] Berkeley Artificial Intelligence Research Lab. http://bair.berkeley.edu/. [Online; accessed 2018-05-24].

[528] Yangqing Jia. http://daggerfs.com/. [Online; accessed 2018-05-24].

[529] NVIDIA DIGITS. https://developer.nvidia.com/digits. [Online; accessed 2018-05-24].

[530] BVLC/caffe. https://github.com/BVLC/Caffe/blob/master/LICENSE. [Online; accessed 2018-05-24].

[531] Which deep learning network is best for you? https://www.cio.com/article/3193689/artificial-intelligence/which-deep-learning-network-is-best-for-you.html. [Online; accessed 2018-05-24].

[532] Model Zoo. https://github.com/BVLC/Caffe/wiki/Model-Zoo. [Online; accessed 2018-05-24].

[533] Caffe Model Zoo. http://Caffe.berkeleyvision.org/model_zoo.html. [Online; accessed 2018-05-24].

Acronyms

FAST features from accelerated segment test

SIFT scale-invariant feature transform

SURF speeded up robust features

VLAD vector of locally aggregated descriptors

GPUs graphics processing units

GPU graphics processing unit

CPU central processing unit

CPUs central processing units

CNNs convolutional neural networks

CNN convolutional neural network

BoW bag of words

SVM support vector machine

SVMs Support vector machines

MCH moving center hypersphere

MMCH moving median centers hypersphere

GLCM gray-level co-occurrence matrix

PCA principal component analysis

RBENN radial basis exact fit neural network

KNN k-nearest neighbor

M-I moments-invariant

C-R centroid-radii

DMFs digital morphological features

GMM Gaussian mixture model

SFTA segmentation-based fractal texture analysis

ROI region of interest

HGO-CNN hybrid generic-organ convolutional neural network

PNN probabilistic neural network

APIS advanced plant identification system

2D-FFT 2D-fast Fourier transform

ANN artificial neural network

ANNs artificial neural networks

ORB oriented FAST and rotated BRIEF

FLANN fast library for approximate nearest neighbours

IoT internet of things

RGB red-green-blue

sRGB Standard Red Green Blue

ImageCLEF Image Combined Lab Evaluation Forum

MNPD modern natural plant dataset

CDF cumulative distribution function

MDP modified dynamic programming

HOG histogram of oriented gradients

CoHOG co-occurrence histograms of oriented gradients

LBP local binary patterns

BRISK binary robust invariant scalable keypoints

MSER maximally stable extremal regions

FREAK fast retina keypoint

EBR edge-based regions

IBR intensity-extrema-based regions

SSD sum of squared differences

DoG difference of Gaussian

LoG Laplacian of Gaussian

SUSAN smallest univalue segment assimilating nucleus

BRIEF binary robust independent elementary features

LSH locality sensitive hashing

LLE locally linear embedding

WLLE weighted LLE

MLLDE modified locally linear discriminant embedding

MMMC modified maximizing margin criterion

A/D analog-to-digital

BBF best-bin-first

VSLAM vision simultaneous localization and mapping

NLP natural language processing

IR information retrieval

SAR synthetic aperture radar

HSR high spatial resolution

LGFBOVW local-global feature bag-of-visual-words

AdaBoost adaptive boosting

EM expectation-maximization

RBF radial basis function

ms milliseconds

AST accelerated segment test

VLAT vector of locally aggregated tensors

CSS curvature scale space

HP Hewlett-Packard

GAC guided active contour

CMOS complementary metal-oxide semiconductor

CRM customer relationship management

RNNs recurrent neural networks

RNN recurrent neural network

DNN deep neural network

LSTMs long short-term memory networks

DBN deep belief network

GANs generative adversarial nets

GAN generative adversarial net

DBM deep Boltzmann machine

RBM restricted Boltzmann machine

DSN deep stacking network

TDNN time delay neural network

SPN sum product network

RAM random-access memory

SIANN space invariant artificial neural network

ImageNet LSVRC ImageNet Large Scale Visual Recognition Competition

ReLUs Rectified Linear Units

ReLU Rectified Linear Unit

BAIR Berkeley Artificial Intelligence Research

DIGITS NVIDIA Deep Learning GPU Training System

LRN local response normalization

SGD stochastic gradient descent

Adam adaptive moment estimation

CUDA Compute Unified Device Architecture

DNPRS deep natural plant recognition system

FPGA field programmable gate arrays

EZLS Real-time Learning Systems

GPS Global Positioning System

LIDAR light imaging, detection and ranging

PF Precision farming

PD pedestrian detection

ACF aggregate channel features

AgriBot autonomous robot for agriculture

GTSDM grey tone spatial dependency matrices

MTCD multi-scale triangular centroid distance

ALIVE Autonomous Laboursaving Internet of. Things Veteran Energizer

SLR single-lens reflex

BSI Backside Illumination

QHD Quad High Definition

HDR High Dynamic Range

ISO International Organization of Standardization

DIGIC Digital Imaging Integrated Circuit

CNC computer numerical control

UCB UDOO Connector Board

ROS Robot Operating System

ADAS advanced driver assistance systems

RDF resource description framework

CNS central nervous system

PNS peripheral nervous system

LGN lateral geniculate nucleus

TanH hyperbolic tangent

MLP multilayer perceptron

MILA Montreal Institute for Learning Algorithms

AWS amazon web services

PC personal computer

PCs personal computers

TPUs tensor processing units

TPU tensor processing unit