Investigation and development of machine learning algorithms for analysis of large volumes of data

by

Athanasios Anagnostis

A thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

University of Thessaly
Department of Computer Science and
Telecommunications
2022

Date: 14/11/2022

UNIVERSITY OF THESSALY
DEPARTMENT OF COMPUTER SCIENCE AND TELECOMMUNICATIONS

Title

INVESTIGATION AND DEVELOPMENT OF MACHINE LEARNING ALGORITHMS FOR ANALYSIS OF LARGE VOLUMES OF DATA

by Athanasios Anagnostis

Chairperson of the Supervisory Committee:

Dr. Dionysis Bochtis
Centre for Research and Technology Hellas
Institute for Bio-Economy and Agri-Technology

Member of the Supervisory Committee:

Dr. Elpiniki Papageorgiou
University of Thessaly
Department of Energy Systems

Member of the Supervisory Committee:

Dr. Konstantinos Kolomvatsos
University of Thessaly
Department of Informatics and Telecommunications

DECLARATION

I hereby declare that the work presented in this thesis has not been submitted for any other degree or professional qualification, and that it is the result of my own independent work.

COPYRIGHTS

ABSTRACT

The current era is defined by the abundancy of data. Globally, the accumulation of data is exponentially increasing each year, leading to huge amounts of data, containing useful information. Machine learning, a set of data-driven approaches for developing mathematical models, based on self-learning algorithms, has found fertile ground to grow. The adaptation of machine learning has been also vast not only due to the recent data availability, but also because of the easy access to high performance hardware. In most domains, the implementation of machine learning approaches brought previously unattainable solutions to complex problems and overall high performances to their operations, leading to a significant paradigm shift. However, a gap has been identified in the domain of Agriculture, the only domain that is steadily behind in technological advancements, even though it is highly significant and belongs in the primary production. The need for fast technological adaptation to agriculture is grave, because of the rapid increase in global population, leading to overpopulation and the adverse effect of climate change.

The problem at hand is the development of a methodology for proper identification of diseases on trees, that are located within orchards of high valued crops. Agricultural operational environments are highly complex and the disease detection is problem based on vision, therefore large amounts of imaging data and sophisticated machine learning algorithms are employed to tackle all specific issues. Specifically, a significant number of data has been collected from various walnut orchards and has been manually labelled by expert agronomists. Additionally, multiple convolutional neural network architectures are utilized, since they are highly capable of extracting features and useful information from complex images. The proposed methodology comprises of three consecutive tasks, aiming to offer a holistic solution towards the identification of infected trees, in order to mitigate the disease's spread and preserve the crop production and yield. The first task tackles the issue of tree localization within orchards with the use of semantic segmentation, from images taken by unmanned aerial vehicles, under a large variety of conditions and throughout all seasons. Once the proper localization is complete, unmanned ground vehicles receive the information to autonomously navigate through the orchard and inspect each tree individually, leading to the next phase. The second step concerns the detection of disease infected leaves, specifically anthracnose, in images of walnut trees, a high value crop. Final step is to properly classify the images of disease-infected leaves and approximate the level of infection in each tree. Outcome of this methodology is a variability map of the orchard and the extent of the disease spread within it, which can then be used by automatic precision spraying systems or manual operations to treat the trees with the minimum possible resources.

Each task of the proposed methodology has been thoroughly investigated, tested and evaluated for application in real-life agricultural environments. Aim of the present thesis is to develop a methodology that is not just applicable in real-life operational environments, but also achieves high performance, robustness, and generalization, a demanding task due to the complex and variable nature of the agricultural environments. All targets have been met for each one of the tasks, leading to the publication of three scientific papers in esteemed peer-reviewed journals, proving the scientific soundness of all approaches. Future goals of the proposed methodology include, but are not limited to, further development based on the potential of the methodology, and direct implementation in real-life farming systems.

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my primary advisor Prof. Dionysis Bochtis for the continuous support of my PhD study, both with his clear insight on the principal idea that we tackled, as well as the acquisition of funding that allowed me to accomplish my goal. His guidance was crucial with the research and writing of this PhD thesis and I am sincerely glad I have had him as a supervisor through some of the most challenging years of my life.

Besides my primary advisor, I would like to extend my gratitude to the rest of my PhD advisors: Prof. Elpiniki Papageorgiou and Prof. Konstantinos Kolomvatsos, for their support and corroboration, but also for the encouragement which was much needed thought the duration of the PhD.

My thanks also go to Prof. Aspasia Daskalopoulou, Prof. Georgios Stamoulis, Prof. Christos Athanasiou, and Prof. Nikolaos Tziritas, who, as a part of the thesis committee, reviewed my PhD thesis, provided with insightful comments, and asked the hard questions during the defence.

I sincerely thank my past collaborators, Apostolis Chondronasios and Serafeim Moustakidis who have introduced me to this exciting domain called Artificial Intelligence, and current teammates, especially Gavriela Asiminari, for the efficient collaboration and countless brainstorming sessions, that resulted in numerous published papers. I wish them the best to their endeavours as well.

Lastly, I would like to thank my parents for supporting me all my life the best way they could, and my brother who set the academic bar very high years ago, for me to admire.

Special thanks go to my wife who has shown an immense level of support since we met, mostly emotionally but also in any other way imaginable, including through my PhD journey, which happened to coincide with hard times. Ultimately, I want to thank my son, who has come to our lives recently and has been yelling at me enough to motivate me to finish writing this thesis faster.

To my angels…

TABLE OF CONTENTS

ABBREVIATIONS

Acronym	*Description*
1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
CNN	Convolutional Neural Network
ML	Machine Learning

RGB	Red Green Blue
R-CNN	Region-based Convolutional Neural Network
IDC	International Data Coorporation
ZB	Zettabytes
GB	Gigabytes
CPU	Central Processing Unit
RAM	Random Access Memory
AI	Artificial Intelligence
ANN	Artificial Neural Network
AP	Average Precision
AR	Average Recall
CLAHE	Contrast Limited Adaptive Histogram Equalization
COCO	Common Objects in Context
CV	Computer Vision
DNN	Deep Neural Network
DRNN	Deep Residual Neural Network
DSS	Decision Support System
DT	Decision Trees
EQ	Histogram Equalization
FCN	Fully Convolutional Network
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
FPS	Frames Per Second
GB	Gradient Boosting
GP	Gaussian Process
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HSV	Hue Saturation Value
IoU	Intersection over Union
IR	Infra-red
IT	Information Technology
LASSO	Least Absolute Shrinkage and Selection Operator
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
mAP	mean Average Precision
MAPE	Mean Absolute Percentage Error

MLP	Multi Layered Perceptron
MSE	Mean Squared Error
NIR	Near Infra-red
OBIA	Object Based Image Analysis
ORB	Oriented FAST and Rotated BRIEF
R2	Coefficient of Determination
RAM	Random Access Memory
ROM	Read Only Memory
ReLU	Rectified Linear Unit
RF	Random Forests
RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Network
RTK	Real Time Kinematic
SSD	Single Shot Detector
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
TPU	Tensor Processing Unit
UAS	Unmanned Aerial Systems
UGV	Unmanned Ground Vehicle
VOC	Visual Object Classes
YOLO	You Only Look Once

1 INTRODUCTION

1.1 BACKGROUND

1.1.1 Age of data

Decades have passed since the computer started to become an integral part in almost everyone's lives, invoking radical changes throughout business and personal aspects of life, leading us towards the Age of Information. The Information Age, also known as the Computer Age or the Digital Age is a historical period that began in the middle of the 20th century, and whose main hallmark was the rapid epochal change from the traditional industry, constituted by the Industrial Revolution, to an economy mainly based on information technology [1][2][3][4]. The beginning of the Information Age has coincided with the development of transistor technology, not by accident, since these two are directly associated [5].

The exponential increase in all crucial tokens of the digital revolution such as processing capabilities, storage capacity, data management, and transmission speeds, together with the associated reduction in costs per unit, marked the beginning of a new era: the "Age of Data" [6]. This period, as we are currently experiencing it, has been defined by the collection, storing and analysis of data, that cascaded in a rapid transformation of the economy, society, and in general, all aspects of life. These changes have brough monumental effects to the way humanity works, but also how we entertain ourselves and socially interact with each other. The exponential increase in the use of computers, brought an even bigger increase in the creation and utilization of data. A popular term for the massive amount of data that is created, collected, and processed by all electronic devices, is called "Big Data" [7].

This term has been used, overused and abused, mainly for the wrong reasons, however, the question remains: "what is Big Data?". Arguably a marketing term and a meme, but ultimately the keyword that signifies advancing trends in technology, namely on data-driven approaches, that are used for understanding data, their underlying information and their usefulness for making important decisions. Data are continuously created at an increasing rate across the globe at an accelerative way, currently doubling every two years, as estimated by IDC (International Data Corporation) [8]. What is noteworthy is that the main reason for the increasing number of data is not solely the result of increasing streams of existing data sources, but the creation of entirely new streams, mainly from electronic devices packed with sensors or human interaction, such as smartphones. As a matter of fact, countless of digital sensors that can measure all types of variables, are installed, and used worldwide in automobiles, industrial equipment, energy meters and more. A noteworthy example is the United Stated of America, where, in the past decade, the digital economy has been a major growth factor [9]. An average annual growth of 5.6% annually was achieved by the digital sector between 2005 and 2016, even if the country's economy as a whole grew only 1.5% [10]. Cloud computing fuelled that growth, by boosting efficiency and enabling new business models [11].

1.1.2 Data accumulation

"Data is the new oil" is probably one of the most prevalent mottos of the 21st century, accentuating the increasing significance and value of data [12]. This metaphor shows accurately how, in this new era, data are "fuelling" the digital transformation the mankind is experiencing. This is apparent, simply by observing the changes brought by the collective online footprint concerning the global economy and the digital lifestyle [13].

The global data accumulation that is taking place at the moment is astounding, and yet, it will keep on increasing exponentially in the years to come [14]. Some interesting facts reveal the magnitude of all data accumulation and transactions, and hint to the levels they will reach in the future. The creation of data will grow to such a degree, that will surpass 180 ZB (zettabytes) by 2025, which will be approximately 118.8 ZB more than that which was available in 2020 [15]. Between 2010 and 2020, the creation, capturing, copying, and consumption of data went up by 5000%. Considering absolute values, usage grew from 1.2 trillion GB (gigabytes), to an astounding 60 trillion GB [16]. Between 2018 and 2020, 90% of the world's total amount of data was created. In 2020, the total data generation per day was 146,880 GB, and if we consider that the world population is 8 billion people, it is easily derived that every person was generating on average 1.7 MB (megabytes) per second [17]. At a daily basis, over $306 \cdot 10^9$ emails are sent, $5 \cdot 10^6$ tweets are posted on Twitter, and $95 \cdot 10^6$ photos and videos are shared on Instagram. However, the discussion on photos and videos requires further investigation, since, vision is the dominant sense in the majority of human beings for interacting with their environment, perform tasks, convey information and acquire knowledge.

1.1.3 Digital images

A significantly large part of all created data, both online and offline, is digital images. With the term digital images, all different types of data that can contain visual information are included, such as static and dynamic images, as well as videos, which can be defined as a sequence of rapidly recorded static images. In general, the majority of traditional media have migrated to the digital domain, therefore it is only natural that there would be a radical increase in digital images globally. Images can convey messages in a denser fashion than text, as it is also mentioned in the famous adage "a picture is worth a thousand words", and therefore are more prominent to be used for faster transfer and richer storage of information. In technical terms, there are three pillars that are of great importance concerning digital images.

1.1.3.1 Remote Sensing

Remote sensing was generally considered as the non-contact information acquisition of an instance or an event. The absence of physical contact comes in contrast to the in-situ observation, which in general is restricting, especially in real-life applications. The term remote sensing has been associated mainly to the information acquisition from Earth (or other planets when possible), and is applied in large number of scientific areas, such as agriculture, geography, meteorology, geology, oceanography, hydrology, land surveying and generally, all ecology-related sciences. Some other application concern intelligence, military operations, planning, and other human-centric applications.

The contemporary use of the term refers mainly to image acquisition by sensors installed in airborne and spaceborne vehicles, such as aircrafts and satellites, and the classification or detection of items on the surface. There are two categories concerning remote sensing, the "active" which relies on emitted and reflected signals from said airborne and spaceborne vehicles, and the "passive" which relies solely on the natural reflection of the sunlight. Nevertheless, both of them utilize the signature of the propagated signals from a wide spectrum of electromagnetic radiation, to map land, oceans and atmosphere from the Earth's surface.

1.1.3.2 Digital image processing

Digital image processing is the term that describes the utilization of a digital computer and an algorithm, in order to perform any type of process on digital images. It is a subdivision of digital signal processing, considering that images are just two (or more) dimensional signals, and has contributed vastly in tasks such as noise reduction. Various reasons have driven the evolution of digital image processing as a scientific and technological area. One reason is the advancements in mathematical concepts that apply to the particular domain, such as discrete mathematical theory, while on the same time, another reason is the technological progress in computer hardware such as CPUs (Central Processing Unit), GPUs (Graphics Processing Unit), and the memory capacity increase, both in storage ROM (Read Only Memory) but also as RAM (Random Access Memory). However, undoubtably one of the most important drives for the evolution of this domain is the practical demand for real-life application in a variety of areas such as agriculture, medical science and industry amongst other. Digital image processing has had a significant impact in the IT (Information Technology) scientific area as well as in applied solutions by providing with practical and effective methods such as multi-scale signal analysis, feature extraction and pattern recognition.

1.1.3.3 Computer vision

Computer vision (CV) takes things one step further and focuses on the extraction of meaningfully information from digital images (including videos). This interdisciplinary scientific field aims to enable computers to gain a high-level understanding of a problem, in the same fashion as the human visual system can do [18][19][20]. This classifies CV as part of AI (Artificial Intelligence), since it goes beyond the functions of simple processing, and allows systems to learn to derive helpful knowledge that can assist with decision making and recommendations. This broader understanding of digital images relies on some well-known CV methods such as acquisition, process, analysis, and knowledge extraction from high-dimensional data, with final aim to produce symbolic or numerical information in a definitive structure i.e. decisions or recommendations [21][22][23][24].

Key element of CV is the understanding of context, which can be translated as the transformation of visual information to a cognitive process, the same way an eye's retina captures the light's photons, and the brain transforms this input into knowledge. Image understanding can be regarded as the symbolic information disentanglement of image data, with the use of mathematical and computational models based on principles of linear algebra, geometry, physics, statistics and learning theory [25].

Therefore, CV can be considered as the scientific discipline that enables artificial intelligence to extract information from images. As mentioned above, digital image data can exist in various types, such as static or dynamic images, videos, as well as multi-angle input recordings (from multiple cameras), multi-dimensional images from 3D (three dimensional) scanners, and multi-spectral images with large variety of bandwidth capturing such as IR (Infra-red) or NIR (Near Infra-red). The technological discipline of CV embraces all image capturing systems and aims to develop theories, methods and models that apply to all input variants. Some well-known CV sub-domains are object recognition, object detection, semantic segmentation, video tracking, 3D pose estimation, motion recognition, image restoration and image enhancement [23].

1.2 Problem definition

1.2.1 Computer vision issues due to complexity

Vision is an inherent attribute to most living creatures, and the dominant sense in most of them, including humans. For those who are lucky enough to have it, it enables them to process their surroundings in their visual cortex by the information they receive when photons enter their eyes. Different living creatures see different bandwidths of light, however they all perceive their environment with a high level of intuitive understanding.

This task is inherently hard for a computer to perform. Understanding visual information is complicated because it involves intelligence, hence, an “unintelligent” machine is ill-equipped to perform such a task. Scientists have managed to develop techniques and algorithms to allow machines classify images or recognize objects, however these techniques fell short when the visual information increased in complexity.

Since all these systems mainly aim to assist humans, either directly or indirectly, it is important to research and develop methodologies that will enable them to “understand” all environments in which people are present and operate within. Controlled, or at least anticipated settings, can be achieved for indoor locations such as warehouses or shopfloors, however humans operate a fair deal on open air surroundings. One example is the operational environment of self-driving cars, where they need to be outside and drive around various locations. A more specific example of open-air scenes that are important for humans are agricultural environments, where operations take place in rural locations which are highly complex from a visual point of view. Agricultural environments can contain enormous amounts of information within a range of aspects such as crops, trees, leaves, seeds, weeds, etc., as well as cover large areas for notable variations. Such open-air environments, besides their intrinsic complexity, suffer from various variabilities, either global, such as the daily day/night switch, the yearly seasonal fluctuations, or specific, such as the cropping periods depending on each crop or other seasonal operations. This time-dependent environment can radically change their characteristics with the change of time, maintaining however, similar needs concerning operations.

1.2.2 Lack of technological advancements in Agriculture

Agriculture, an important and critical domain for human prospect and existence, has itself seen little improvement since the Industrial Revolution. Even though the Information Age has impacted almost all other large industries and pushed towards their 4th age, i.e., Industry 4.0, Agriculture is still lacking the deep and integral application of information technologies. On the same time, the Earth's population is continuously increasing, and a large majority will be facing a realistic threat concerning famine, unless the primary production of Agriculture as a whole, increases as well [26]. The rapid swift towards the Age of Data, together with the accessible computational power, is steadily bringing Agriculture closer to new technologies, but with more to be done. This evolution of traditional Agriculture goes by the name Precision Agriculture, and it's based on state-of-the-art technologies for various domains.

1.2.3 Precision Agriculture

Precision agriculture (PA) is the practise which aims towards the increase of agricultural yield and mitigation of environmental risks, via monitoring and measuring the variability in a plethora of farming management parameters. Research effort focusing on precision agriculture research aim on developing a DSS (Decision Support Systems) for farm management, with the ultimate goal of optimizing processes, procedures and returns, while preserving natural and operational resources [27][28].

Precision agriculture has also reaped the benefits from the wide-ranging availability of unmanned vehicles, both ground- and aerial, since they systematically become less expensive, and can be operated by relatively inexperienced pilots. These ground robots and agricultural drones have the ability to be equipped with RGB (Red Green Blue), IR (infrared), hyperspectral or multispectral cameras, which are capable of capturing different types of images. Special techniques of field images can be stitched together with the use of photogrammetric methods, in order to create orthophotos. Such multispectral images contain additional values per pixel, such as near infrared and red-edge spectrum values on top of the standard red, green blue values, which are used to process and analyze vegetative indexes such as NDVI (Normalized Difference Vegetation Index) maps [29].

Such unmanned vehicles are capable of capturing imagery in a remote and even unsupervised fashion, and can provide detailed visual information about a farming installation, from a few centimetres, up to kilometres. Additionally, geographical references such as elevation, can be used to build precise topography maps, which are used for correlating topography with crop health. This correlation can offer information which is valuable for the optimization of variable-rate application of crop inputs such as water, fertilizer, and chemicals (herbicides and growth regulators) [30].

1.2.4 Field inspection by human experts

One of the most critical agricultural tasks, the identification of diseases that affect food/nut production trees within orchards, has been conducted with the same way for centuries. Diseases, viral or fungal, infect the trees and reduce their potential for large or at minimum the normal, expected yield. The adverse effects of trees infection, besides the obvious loss of crops, can be easily translated into economical figures once the crops are of high value.

In large orchards, early signs of infection can be completely missed since human presence can be sparse. Omitted infections can swiftly lead to the contagion of larger areas, since the infection spread obeys the same exponential formula as all diseases do, except of course the mobility parameter that appears in human and animals. In essence, given a non-constrained system that usually applies in orchards, the larger the number of infected trees is, the faster the infection spreads to the rest, and consequently, the more drastic measures need to be taken to deal with the spread. Aim of every producer is to achieve the maximum reduction in resources and yield losses, therefore, early detection systems could be immensely valuable for agricultural operations. Deploying experts to conduct field inspections periodically would be costly and impractical, therefore there is a gap for autonomous systems to be developed and employed for such tasks. Additionally, the human factor is not negligible should be taken into consideration when performing tasks, since human errors can easily occur, especially when there is fatigue, or other obstructive factors present. This issue might not have a threatening impact in an orchard (when compared to a hospital's Emergency Room), however, supports the importance of achieving a human expert-level accuracy for a developed system. Nevertheless, yield loss or increased expenses can incur significant financial impact to producers, which can lead to debts, closures and bankruptcies, which can lead to physical or mental health issues.

1.2.5 Disease detection

A variety of studies have attempted to tackle the issue of disease detection on leaves, however, most of them had promising results only when the images were properly curated before the classification [31][32][33]. Placing leaves on single-coloured backgrounds, or manually removing background information from real-life images increased the accuracy of trained classifiers, however such approaches lack applicability in real conditions. Ideal or controlled environments are useful for proof of concept as well as for some specific applications, however, the vast portion of agriculture is conducted on open air environments, and therefore it is mandatory that such methodologies are applicable in them, regardless of the external conditions.

1.2.6 Tree localization

On top of the existing tasks performed by on-field experts, new set of tasks have emerged with the introduction of new technologies as they gradually become available to the public. Tasks like tree location for autonomous orchard operations, rely on remote sensing images taken by GPS (Global Positioning System) satellites, however there is inherent complexity in them mostly due to the existence of same-coloured canopies and weeds, or tree trunks and ground [34]. The next generation of farming, i.e. Agriculture 4.0, relies heavily on robotics, therefore efficient automated and unsupervised operations, require efficient tools and systems that overcome issues with the use of intelligence.

1.2.7 High complexity of agricultural environments

A severe obstacle computer vision faces towards its effectiveness in precision agriculture applications, is the high complexity environments as they are represented in digital images. The human visual perception is by far more capable is resolving challenging inputs, than any type of RGB sensors, especially the commercially available ones. Such challenging inputs could be captured sceneries where both illuminated and shadowed portions exist in the same frame, and the camera can only clearly capture one of those, but also in images of high-density information, where only a small portion contains useful information and needs to be investigated. Both examples apply in precision agriculture, especially when tackling images with trees and canopies that have a high number of leaves, both with light and shadow, and the focus needs to be specifically on the leaves of the tree, but the background contains out-of-focus trees, sky, ground, weeds and much more. What is noteworthy is that these obstacles are hardly ever mentioned by human experts, firstly because the human visual system is by far a better vision system than any of the digital ones in existence, and secondly because they have obtained "trained eyes" from knowledge and experience, that can pinpoint an issue with impeccable ease and speed.

1.2.8 The proposed solution: combination of large volumes of data and machine learning algorithms

Farming activities and operations within agricultural environments cover a large range, most of them conducted, lead or supervised by trained domain experts or expert workers with multiple years of involvement in the field. Agriculture 4.0 aims on assisting human in the entirety of this range of tasks, with any piece of technology available. Tasks that are dependent on such level of expertise are difficult to automate and model with traditional programming, especially the ones that are reliant on visual information, such as field inspection. Taking a step further, there is dire need for the development of methodologies, and consecutively systems, that will be able to assist human experts with high level of accuracy in complex and difficult tasks, such as disease identification in agricultural environments. These factors have led to the selection of Precision Agriculture, and specifically disease detection in real environments and in-field application, as a use case for the present thesis. Alongside with the increasing capacity for data collection, closely related to the constantly lowering cost of digital sensors, and in tandem with the rise of AI algorithms and the cost reduction of powerful chipsets, the problem at hand can be addressed and useful outcomes can be extracted.

1.3 Objective

Main objective of the present thesis is to develop methodologies that tackle computer vision issues that were up to now, ineffective when applied in real-world applications. These methodologies focus on automatic operations based on visual stimuli in precision agriculture tasks, and specifically in open-air complex environments. All methodologies are developed based on obtained large amounts of data, in the form of digital images, and utilize deep learning algorithms, namely deep neural networks, which have been proven that can learn to successfully perform computer vision tasks of high complexity with increased accuracy [35].

Similar works have been tackling issues of computer vision in precision agriculture by applying their methodologies in ideal or controlled environments. Leaves' features are extracted only when placed in high-contrast backgrounds, weeds are recognized when present in clear ground, seeds and fruits are detected when their colour is distinct compared to the leaves, and trees are located from aerial images when the ground is free of weeds. These ideal conditions, even though they would be also optimal for the farmers, are scarcely found. Additionally, open air environments suffer from temporal fluctuations (day-night and yearly cycles) and weather conditions (sun, rain or snow), resulting in major variations in the macroscopic picture of the said environments.

This work, however, attempts to solve the issues created by the on-field applicability and complexity. Tasks that traditionally are conducted with the on-field presence of expert agronomists and farmers, such as disease identification on leaves, are attempted to be automated by modular methodologies, in order to initially assist and later on replace the human presence on the fields.

Goal of this work is to establish methodologies, based on various machine and deep learning methodologies that tackle the issues of computer vision tasks in complex environments with large amounts of data, and examine their in-field applicability and robust performance. The range of vision tasks should attempt to cover a complete operation performed in agricultural environments, where all comprising tasks are performed in open-air complex environments and all data are collected by cameras placed on UGVs (Unmanned Ground Vehicles) or UAVs. Ultimate goal is to achieve close-to-human levels of accuracy by unsupervised robotic agents, in tasks that are conducted exclusively by visual inspection of experts, in order to integrate this knowledge extraction mechanism into subsequent tasks.

The approach followed in the present work is conducted by establishing a workflow for multi-level computer vision tasks, that can be directly applied on real-life agricultural environments. And since these novel methodologies will be applied on high-value crops productions such as orchard farming, there is significantly higher interest both on their economic and environmental impact.

In short, the present work covers the following objectives:

- Main objective is to develop methodologies that tackle these issues considering computer vision in precision agriculture.

- The developed methodologies are based on machine learning algorithms, which learn to successfully perform computer vision tasks of high complexity with increased accuracy.
- Similar works have been tackling issues of computer vision in precision agriculture by applying it in ideal or controlled environments, however this work attempts to solve the issues created by the on-field applicability and complexity.
- Goal of this work is to establish the applicability and performance of various machine and deep learning methodologies that tackle the issues computer vision tasks in complex environments, with large amounts of data.
- This is conducted by establishing a workflow for multi-level computer vision tasks, that can be directly applied on real-life agricultural environments.

2 STATE OF THE ART

2.1 MACHINE LEARNING

Up until the 1950s, computers were considered to be "dumb" since they were able to only do what they were programmed to, in a pre-determined manner. Along came AI which brought to computers the ability to learn from their errors by trying to correct them [36][37], in the same way it is done in living beings [38][39]. The brain's cognitive and learning skills have been extensively studied and consequently implemented in computer science [40][41]. Probably the most influential area of computer science of the modern world][42][43][44], AI's applications cover almost all domains including agriculture [45], medicine [46], energy [47] and manufacturing [48]. AI's penetration to our everyday lives, even though mostly unrecognizable, it is very deep, as it is applied in our smartphones, smartwatches, search engines, shopping carts, autonomous cars, video platforms and music players [49]. Machines can now understand more than they used to and we are gradually becoming more used to this fact [50].

Mathematical formulas are the core of AI, used in such combinations and ways so that a machine becomes able to learn from data [51]. A wide range of algorithmic approaches and mathematical concepts found fertile ground in the domain of data science, because they could enable the machines to learn from data. The set of all of these mathematical concepts, methods and algorithms is known as machine learning [52]. Machine learning (ML) is the foundation on which the machines are programmed iteratively learn in order to achieve intelligence. By looking at the large picture, ML is "only" a part of the general concept of AI [53], visually shown in Figure 1, nevertheless it is the most significant one and definitely the one that has drawn a huge amount of research and academic interest.

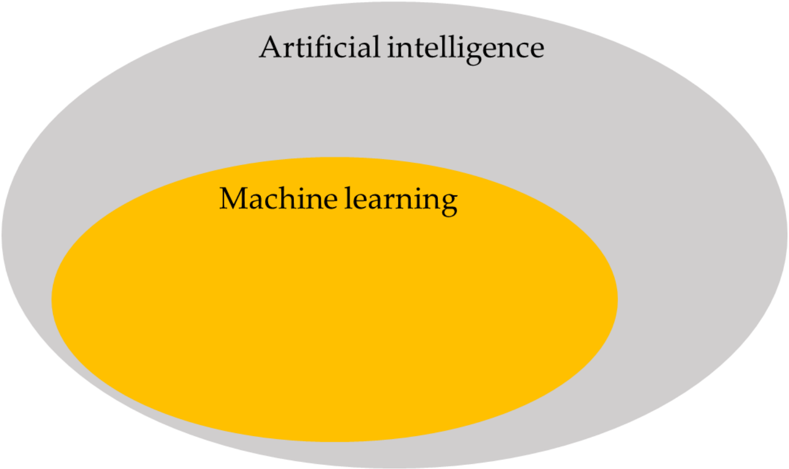


Figure 1: Machine learning as part of artificial intelligence.

2.1.1 Machine learning versus conventional programming

Conventional programming and ML's difference lies in one major aspect. Even though both are parts of computer science [54]. When algorithms are developed with specific instructions and rules in order to "explain" the computer in each step exactly what to do, it is called conventional programming. On the other hand, the premise is completely the opposite in ML since what is provided to the computer are the input-output sets, and the tools to learn to derive the latter from the former. The two types of programming can be illustrated in Figure 2.

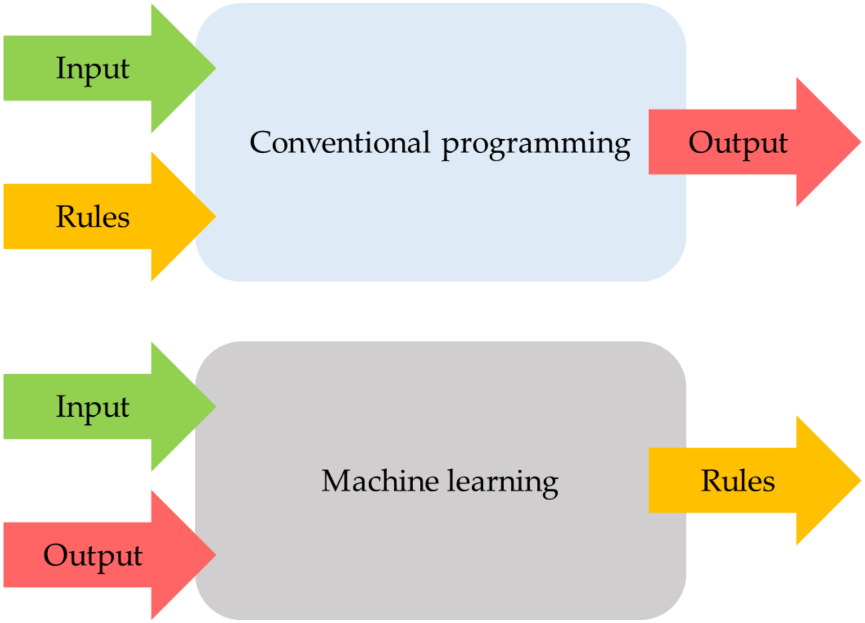


Figure 2: Conventional programming versus machine learning.

The outcome model that derives from an ML approach, contains rules which can either be clear or obscure, based on which of the many ML algorithms is implemented. Unawareness of the ML model's rules can cause uncertainties and insecurity, however ML applications have achieved extremely high efficiency most applications that have been tested.

2.1.2 Fundamental features of machine learning

Machine learning strongly depends on data. In most cases, the more data it uses, the better it works. This make sense, given that machine learning is built on "experience". As far as living creatures are concerned, they learn from examples; the first human that touched fire probably never did it a second time. The more experiences someone has, the better understanding is accomplished with a subject or task. The same goes with physical abilities. Someone must throw a rock many times to learn how to throw it far, or to build strength for it. The same applies also to the machines. In an attempt for a machine to build a set of rules that describe the function that turns input onto output, it needs a lot of input and output data to be used for tryouts. Again, tryouts stand for the testing of as many possibilities that will lead to the desired outcome in the same manner as babies try to find the right shape that goes to the right hole. The mathematics behind this thought process is a bit more complicated than this example, however the idea is simple; more data means more experience, leading to more tryouts, causing deeper learning, and consequently, better understanding.

At the time being, machine learning allows for achieving something called narrow artificial intelligence [55]. This is the intelligence that is confined within some specific limits. This means that we can build models, where models are the rules that a machine learning algorithm produces given explicit input/output) that can be applied only to specific tasks, e.g., identification of faces in images. The greater picture is general artificial intelligence [56]. This is the hypothetical, for the moment, intelligence that gives a machine the capacity to perform intellectual tasks like an adult human.

A simplified version of machine learning working pipeline reads as follows:

- Input and output data are obtained;
- The desired algorithm is selected and fed the data;
- The algorithm makes an attempt to solve the problem;
- After the attempt, results are evaluated;

- The algorithm proceeds to correct its parameters and attempts again;
- Repeat until a performance condition is met.

The final condition is highly relative to the type of problem, the desired outcome and the algorithm that has been chosen, but simplified again, is to build a model that covers most cases successfully or generalizes well. This means that the model should not only be able to predict the provided examples, but also examples that are completely unknown to it. This is achieved by using the data that are in possession in a smart way. Thus, in most of the cases of a finite amount of data, data are split into three categories. Before starting anything, an amount of data is concealed from the rest of the process. This set is called "testing data" and will be used in the final step again. The rest of the data, that is usually the larger amount compared to the testing set, will be used for the learning process. Out of the training data however, a small portion will be used for the validation of the learning process itself. This validation data helps the model to learn from its mistakes and improve its predictions after each iteration, until it reaches a condition that is being set. After the model has completed training, it is evaluated against the testing data. Given on how well it performs on the testing data, its accuracy and general efficiency is evaluated. The splitting of the data is visualized in Figure 3.

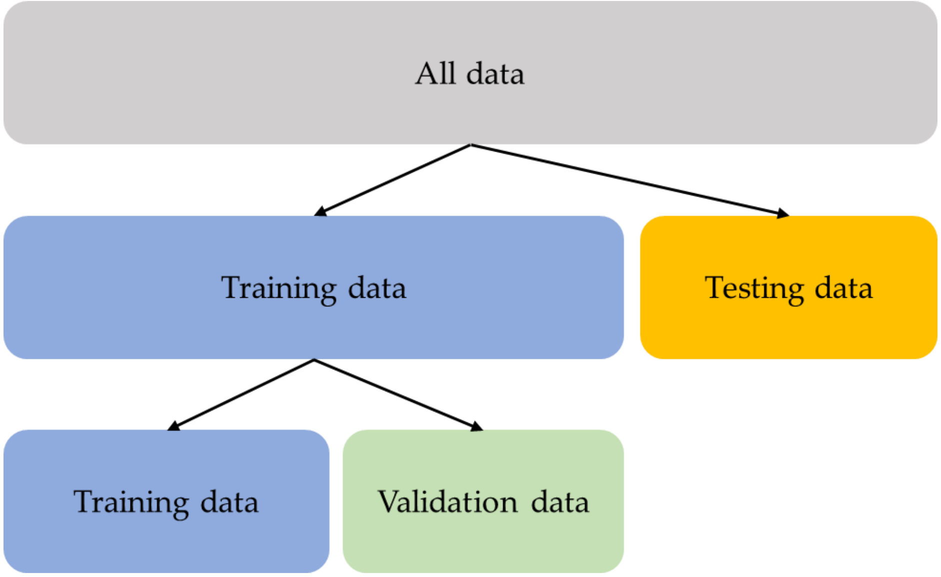


Figure 3: Splitting the data for machine learning.

The above analysis is a description on how supervised machine learning works [57]. However, supervised learning is not the only type of machine learning. There are more which will be elaborated next.

2.2 TYPES OF MACHINE LEARNING METHODS

There are four types of ML methods, depending on the problem at hand and the available data. When there is specific knowledge on the outcome, as it has been described in detail above, the ML method is called supervised learning. Nonetheless, sometimes the outcome is not specifically known, albeit some general intuition can point towards a direction, and only after the predictions are produced there can be a clear view on their usefulness. This method is called unsupervised learning [58]. Semi-supervised learning, recently gaining a lot of attention, is a combination of supervised and unsupervised learning, however it is not considered a category on its own [59]. A highly popular ML method, especially amongst the gaming community, is based on algorithms that "reward" desired actions and "punish" undesired ones as they happen. Reinforcement learning, as it is called, has been behind some famous examples like AlphaGo [60], Super Mario [61] and in any simulated human models trying to walk/run/jump [62]. Last but not least, active learning is another particular category of ML, used for building recommender systems. Recommender systems, as hinted by their name, aim at providing the most fitting recommendations, based on patterns that humans exhibit. Platforms like Netflix [63] and Spotify [64] use recommenders for movie and songs recommendations that fit the taste of the user in order to engage them into using the platform more. Another example is Amazon that recommends complementary items to the ones someone is buying, so that they have more chances that the user will spend more money [65]. The largest example is Google that utilizes ML to almost every business aspect, such as in its search engine, mail service, maps and most importantly, advertising [66].

2.2.1 Supervised learning

Supervised learning is the most common category of ML [57] and the main focus of the present thesis since in used in each step of the methodology, the output is known. There are two model categories that fall under supervised learning, namely regression and classification.

2.2.1.1 *Regression*

Regression refers to a continuous variable output, meaning that in can take any value. Common examples of regression are word prediction in texts, real-estate values prediction, or energy consumption forecasting [67]. In all these examples, a common denominator is that the prediction can be almost any value. A visual example of regression be seen in Figure 4, where the blue dots are actual data, and the green line is the prediction curve.

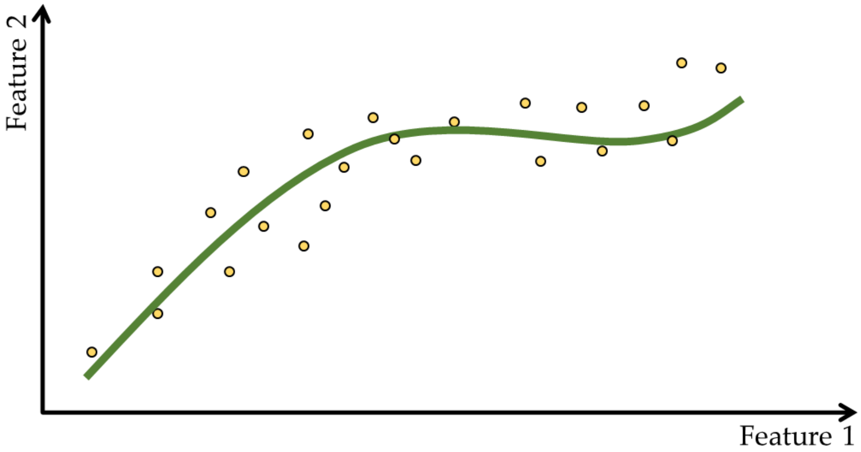


Figure 4: Qualitative representation of regression.

2.2.1.2 *Classification*

Classification is the method where the input is related to specific outcomes, or predefined classes. Classification is widely in imaging applications, such as agricultural [68] or medical[69], where the goal is to classify an image based on its content and appoint it to a class. Therefore, in medical applications for example, machine learning algorithms are used to identify cancers and metastases in whole body scans or other imaging methods [70][71][72]. Classification is in essence regression except the selected activation function that assigns new values to the outcome and serves as a final step. A visual representation of classification is shown in Figure 5 where the blue line is the classification border that splits the data into two categories.

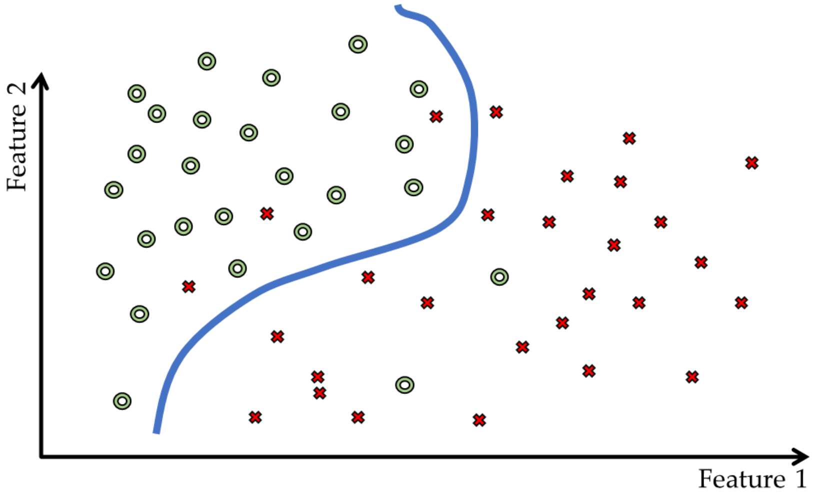


Figure 5: Qualitative representation of Classification.

2.2.2 Unsupervised learning

Unsupervised learning [58] is governed by an "intuition" for the solution of a problem, and reaches to an output without prior knowledge set by examples. Three categories are the main parts of unsupervised learning, clustering, dimensionality reduction, and association.

2.2.2.1 *Clustering*

Clustering is the task of grouping (or dividing) data based on common characteristics and patterns without having specific classes as a target. It differs from classification because the classes are not predefined, but instead, assumed. A visual representation is given in Figure 6 with three main clusters of similar characteristics and some random outlier points.

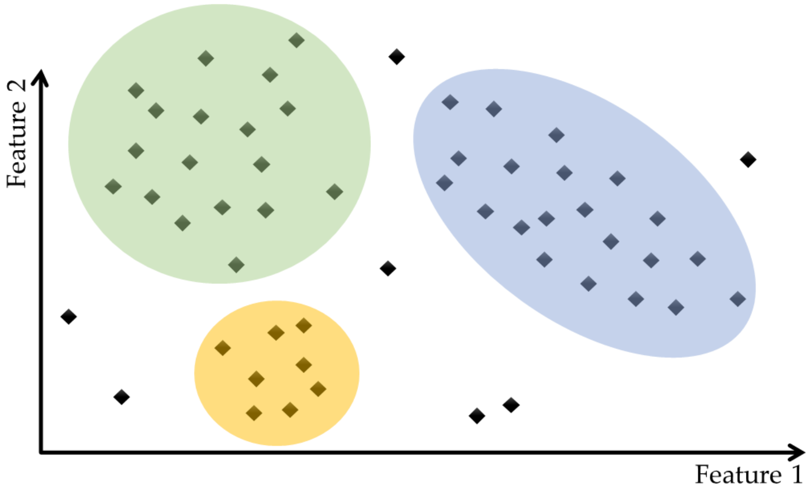


Figure 6: Qualitative representation of Clustering.

2.2.2.2 *Dimensionality reduction*

Dimensionality reduction aims to reduce the amount of data that is available, with the minimum possible information loss. Large amounts of data increase the computational complexity and cost during processing, and do not necessarily imply valuable information, therefore, dimensionality reduction techniques aim in finding relations between the data, in order to remove features that do not offer any value. A visual example of dimensionality reduction from 2-dimensions (2D) space to 1-dimension (1D) space is depicted in Figure 7.

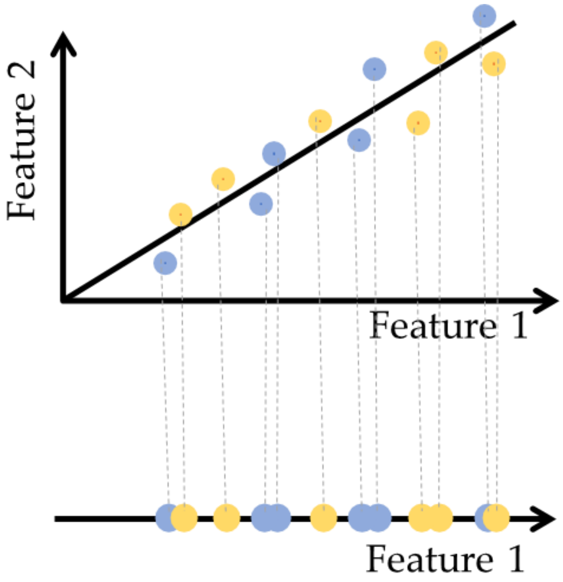


Figure 7: Dimensionality reduction schematic illustration.

.

2.2.2.3 *Association*

Association aims to find association rules between large amounts of data in the same way as the human cognitive function equivalent. Via ML, computers can perform this task, sometimes better than humans, especially in cases where huge amounts of data with thousands of attributes each are available. A simplified visualization is shown in Figure 8.

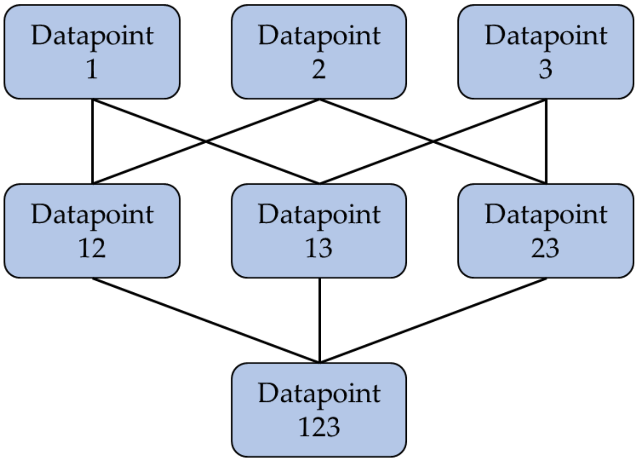


Figure 8: Association rules in data sets.

.

2.2.3 Reinforcement learning

Reinforcement learning might be the closest method to how most living creatures learn, since it rewards "good" actions and punishes "bad" actions. A simplified visual depiction on how reinforcement learning works, can be shown in Figure 9, where the relationship is shown between an agent that acts and its actions are rewarded on the basis of the outcome they produce.

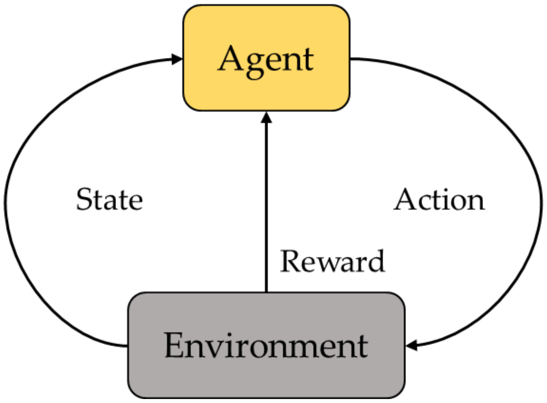


Figure 9: Reinforcement learning.

Reinforcement learning is mainly used for two practices, classification, and control.

2.2.3.1 Classification

Classification under reinforcement learning has one main difference than that of the supervised learning; the temporal component where the assigned class can change over time as the outcome of an action. Other than that, the aim still is to appoint a class to an input.

2.2.3.2 Control

The most famous application of reinforcement learning because its applications, such as controlling driverless cars and robotic arms without explicit programming, are widely marketable. Based on the hardware and the problem, a set of actions is provided as input and based on the outcome, the ML model chooses the appropriate action in order to reach the desired goal. When mistakes are made, they are “punished”, and correct behaviors that lead to the target are “rewarded”.

2.2.4 Recommender systems (active learning)

Active learning is different from the other types of learning because the learning algorithm can actively request feedback from the information source in order to label previously unlabelled data. Another definition that describes this type of learning is the “iterative supervised learning”. A visual representation of an example if shown in Figure 10.

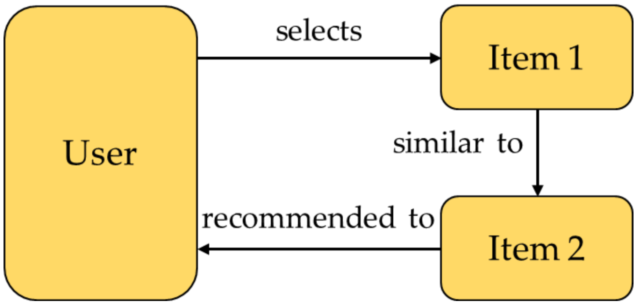


Figure 10: Recommender system.

Two main categories exist for recommender systems, namely content based and collaborative filtering.

2.2.4.1 Content based

In content-based systems, the algorithm aims to continuously add information about a user with the intention of improving its predictions. Invoking the user into providing more data, this is an iterative process provides adaptability for any changes in usual behavioural pattern from the user’s side.

2.2.4.2 Collaborative filtering

Collaborative filtering systems, as opposed to content-based systems, are developed explicitly on past interactions of a user towards their targets. Main premise of the design is that that historical information is usually enough to make predictions about the user.

2.2.5 The main pillars of machine learning types

Summing up, ML methods are designed to tackle a large range of problems with diverse ways. Data science and ML are constantly expanding, with more algorithms being developed continuously. The main pillars however, as described above, remain the same and are presented in Figure 11.

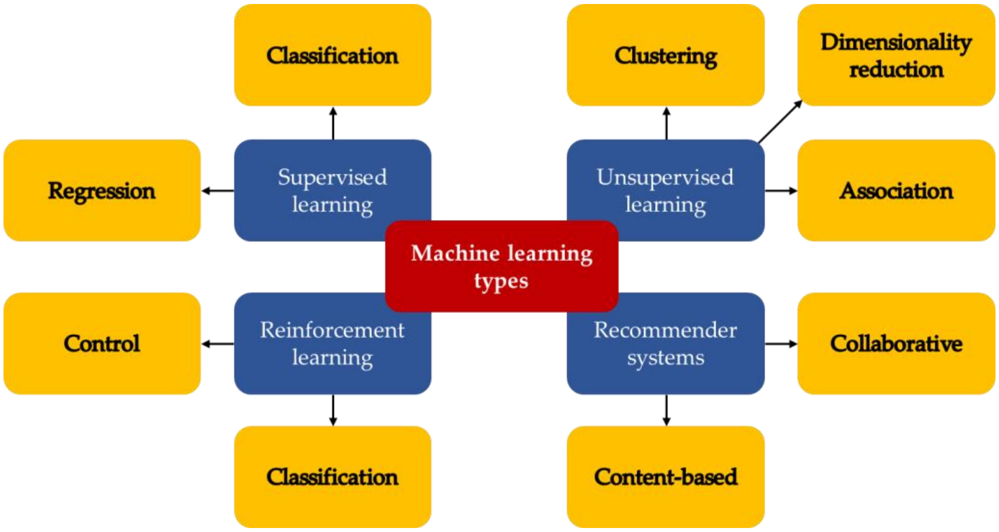


Figure 11: Machine learning types and their respective categories.

2.3 FAMILIES OF MACHINE LEARNING ALGORITHMS

Further categorization can be applied to the ML algorithms, depending on the way they apply the learning method. Naturally, each algorithm is designed to learn in certain ways, different than the others, however, distinctive similarities exist in a number of approaches based on their basic principles and functions, enabling this categorization.

2.3.1 Regression

Regression is the most fundamental ML family, used for finding simple correlations between variables. At its core lies linear regression, and as the name implies, it fits a line to represent 2D data [73]. Some other notable examples are multiple [74], logistic [75], stepwise [76], ordinary least squares [77], multivariate adaptive splines [78], and locally estimated scatterplot smoothing [79].

2.3.2 Regularization

Regularization is effectively regression, however, their main difference is that during their learning process, the algorithms apply terms of regularization in order to penalize complex models and favor simple ones, for achieving generalization. Notable examples are least angle [80] and ridge [81] regression, elastic-net [82] and LASSO [83] which stands for Least Absolute Shrinkage and Selection Operator.

2.3.3 Bayesian

Bayes' theorem is the base for this family of learning algorithms that base their learning methods on the probability that something will happen, based on what have already happened in the past [84]. It is one of the oldest documented mathematical concepts of the modern world and widely used in statistics [85]. Naïve [86], Gaussian naïve [87], Bayesian network [88], and belief networks [89] are some of the most famous Bayesian ML algorithms.

2.3.4 Instance-based

Instance-based algorithms take new instances (testing example or unknown data), compare them with all the other previous instances, and create predictions based on similarity metrics. Space representation of data is the key component, especially in algorithms such as support vector machines (SVM) [90], the most famous in this family and in general. Another name for this family of algorithms is memory-based methods with some noteworthy examples being k-nearest neighbors [91], self-organizing maps [92], learning vector quantization [93] and locally weighted learning [94].

2.3.5 Decision Tree

Built tree-like structures of decisions based on if-else conditions, decision tree (DT) methods are applied on the features of all examples, and once an output is reached, the optimal branch is selected. Characterized by speed, DT methods are accurate, however, they are prone to suffer from not being able to generalize (overfit). Famous examples are classification and regression tree [95], conditional trees [96], iterative dichotomizer [97], and C4.5- C5.0 [98].

2.3.6 Ensemble

Ensemble methods combine weaker models in order to create a strong model that generalizes better that its counterparts. Random forest (RF) [99] being a prime example, builds numerous decision trees and aggregates their outcome, thus reducing the overfitting issue. Boosting algorithms are a major part of this algorithmic family with most prominent examples being adaptive [100] and gradient boosting [101], bootstrapped aggregation [102], and stacked generalization [103].

2.3.7 Clustering

In the clustering family, algorithms build models by being provided with reference points, also known as centroids, and then try to assign the data based on their proximity or relationship to these centroids. Famous algorithms are the k-means [104] and k-medians [105], as well as variations that operate on hierarchical pre-existing structures.

2.3.8 Dimensionality reduction

The algorithms in this family are designed to reduce the number of variables, to achieve minimization of data with as little impact as possible on the retained information. Prominent examples are principal component [106], quadratic [107], mixture [108] and flexible discriminant analysis [109], partial least squares [110] and principal component regression [111], multidimensional scaling [112], and projection pursuit [113].

2.3.9 Association Rule

Association rule algorithms aim to find associations between data variables. The apriori [114] algorithm is well known for large datasets with a big number of variables albeit computationally complex and relatively slow. On the other hand, the eclat [115] algorithm is the faster and most suitable solution, however, for small and medium datasets.

2.3.10 Artificial Neural Networks

Artificial neural networks (ANN) are by far the most trending algorithmic family in the past decades, heavily associated with ML. Designed with the human brain as equivalent, neurons in ANN are represented by nodes with units and synapses as mathematical operations [116]. Layers are formed by stacking nodes, positioned as such so that each layer processes data consecutively. Perceptron [117], the original design of ANNs, an input layer, a hidden layer and an output layer comprise the architecture. Multilayer perceptron is based on the same premise but with more hidden layers [118]. ANNs became increasingly complicated and thus, it became harder to optimize their parameters. Stochastic gradient descent, an optimization method, has evolved as a standalone ANN [119]. Backpropagation, a groundbreaking method, was designed to compute the gradients of variables and reiterate differential operations backwards in order to optimize the model's weight towards a better prediction [120]. Mentionable algorithms in this family are also the radial basis function [121] and the Hopfield [122] networks.

2.3.11 Deep Neural Networks

Deep neural networks (DNN) is a direct byproduct of ANNs, introduced for solving more complex problems [123]. The most recent addition to ML and data science, they've sparked academic and corporate research due to the availability of vast amounts of data, and abundancy of inexpensive computational power like central, graphical and tensor processing units (CPUs, GPUs, and TPUs). By being deeper and more complex architectures than ANNs, allows them to extract from large amounts of data, deeper and more complex features, without prior feature engineering [124]. A recently coined and highly used term, deep learning, includes the type of learning methods based on DNNs [125]. Two notable variations of DNNs, designed with specific characteristics are presented below.

Recurrent neural networks (RNN), developed for forecasting purposes such as timeseries predictions [126], take previous time steps of a variable and store it in memory cells in order to make future predictions. Originally designed for word prediction when typing texts, RNNs have applications to weather, energy, stock market value and any type of forecasts. Most famous representations are the long short-term memory (LSTM) [127] and the gated recurrent unit (GRU) [128] algorithms.

Convolutional neural networks (CNN) have immensely contributed to the DNNs' popularity due to their wide applicability, especially in vision applications [129]. The application of filters, or convolutions in mathematical terms, reveals patterns and characteristics, such as edges or corners, that assist with the model optimization. Their fame originated by outperforming ANNs by a large margin in the identification of handwritten digits, a problem of image classification [130]. On top of image classification, CNNs are used for object detection applications, where objects are located and classified within images [131], and instance aware semantic segmentation where pixel-wise class appointment takes place in images [132]. Surveillance systems and autonomous cars run based on CNNs.

The CNNs' superiority on image-related tasks performance, led to the development of even more complicated architectures and concepts. Variational autoencoders [133] architectures where the input images are encoded based on their features, and consequently other images are decoded, originating from the encoded latent space, resulting in similar but definitely not identical images. Generative adversarial networks [134] are developed as two antagonizing networks, where one (the generator) is generating images and the other (the discriminator) is comparing them against to real images. Such algorithms have found extended application in generation tasks as well as counterfeiting problems. Deep belief networks [135] and deep Boltzmann machines [136] are also notable mentions, whereas the former are generative models, and the latter are recurrent models with applied stochastic principles.

2.4 MACHINE LEARNING IN PRECISION AGRICULTURE

Agriculture is vital for a country's economy, however, its true necessity resides in the global food security, especially for the upcoming years [137]. Serving as the main source of food and raw materials, agriculture is the means for employment and income for a large percentage of the global population. Population increase however, presents a significant demand challenge to agriculture [138][139] especially since arable land decreases smaller and becomes poorer through the years[140]. It is imperative that the global food system provides with healthy and nutritious food, and on the same time minimizes the environmental impact during its production. Agricultural systems need to become more sophisticated and comprehensible via means of data collection and analysis for multiple physical aspects and phenomena, in order to overcome the aforementioned challenges [141][142].

Machine learning has only recently been introduced in agriculture, even though it is proven to be highly efficient in processing large amounts of data, and coping with complex, non-linear tasks [143], [144]. Several agricultural applications have applied ML modelling, most commonly for crop management, a valuable part of agriculture since it provides information to producers about on-field operations which eventually contributes to decision-making [145], [146]. Other applications of ML methodologies in agriculture involve soil, water and livestock management, however, they are fewer since many data availability is smaller.

A preliminary scholarly literature survey was conducted, aiming to capture the latest progress in crop management studies, focusing on the application of ML methodologies. Variations of keywords such as "machine learning", or "precision agriculture" were used in the Google Scholar and Scopus search engines. The reviewed studies are all published between 2018 and 2020, and include scientific journals publications, conference articles and Masters/PhD theses.

A keyword information clustering was created with the use of approximately 130 keywords were utilized from a total of 26 publications, as listed in the corresponding part of the manuscripts. The 10 most frequent keywords are presented as a word-cloud in Figure 12.

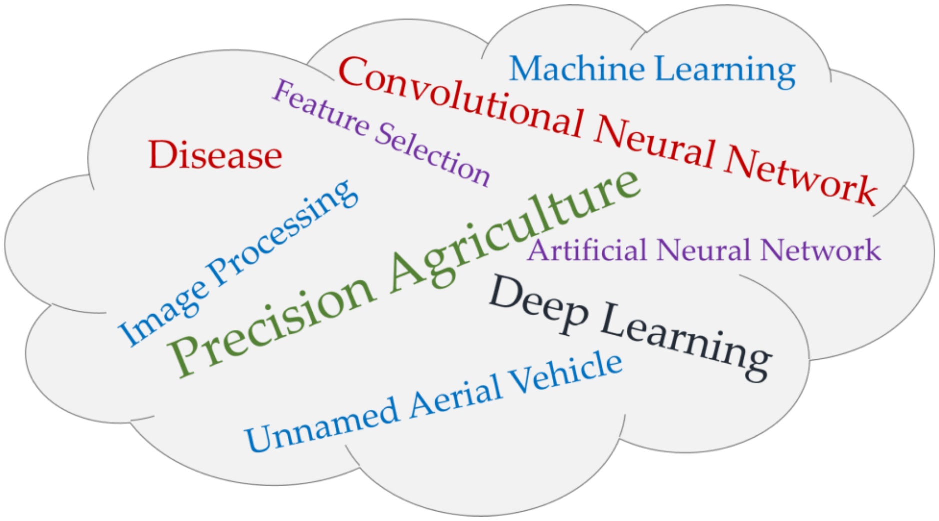


Figure 12: Keyword information clustering of the 26 reviewed articles.

The appearance frequency is represented by the font size, while same-coloured keywords indicate similar frequency. "Precision agriculture", a dominant research topic, was the commonest keyword, followed by "Deep Learning", arguably the hottest scientific topic of the present era. The "Convolutional Neural Network" keyword denotes the importance of the algorithm, and the "Disease" keyword, the importance of the disease detection problem. "Image Processing", "Machine Learning" and "Unnamed Aerial Vehicle" (or UAV) followed, with "UAV" being present in works that require high resolution images. "Image Processing" appeared together with "Machine Learning" since they are complimentary methods. Lastly, "Artificial Neural Network" and "Feature Selection" appeared the same frequency, indicating the usefulness of ANNs in precision agriculture, alongside with proper methodologies for feature extraction.

The applications of the reviewed studies are classified into four categories based on their purpose, namely crop disease detection, yield prediction, weed detection and quality assessment, however, since the aim of the present thesis is to propose a methodology for disease detection on trees, an analytical overview of reviewed studies of only the crop disease detection category is presented.

2.4.1 Crop disease detection

A significant problem throughout all agriculture, crop diseases are a threat to production, able to implicate catastrophic economic impact to producers. Multiple studies attempted to tackle issues of automatic detection and disease classification with ML approaches, and consequently resulted in the several publications throughout the years. Since it is a vision-based problem, the majority of works use images containing leaves or seeds, which are the parts that exhibit signs of infection on a tree.

A leaf disease detection model was developed by the authors in [147], namely a one-class SVM model for each condition the plant displays (healthy, downy mildew, powdery mildew and black rot), based on images of vine leaves, accomplishing a high generalization behaviour when applied in other crops as well. A total accuracy of 95% was achieved, signifying that 44 of the 46 tested plant-condition combinations were classified successfully correctly. The authors of [148] combined a CNN with colour information for disease detection in vineyards, from images taken by UAVs. Colour spaces and vegetation indices were combined, aiming to improve the model's performance resulting to an overall accuracy of 95.8%. The improvement of accuracy in maize leaf diseases attempted in [149], where GoogLeNet and Cifar10 were trained and tested on images from nine kinds of maize leaf and eight kinds of maize leaf diseases, achieving accuracy of 98.9% and 98.8% was achieved by each model respectively. A hybrid method was introduced [150], for the detection and classification of diseases in citrus plants. The first step aimed at the detection of lesion spots on leaves and fruits, with the second step performing classification of citrus diseases including anthracnose, black spot, canker, scab, greening, and melanose, based on a multi-class SVM . This technique reached 97% classification accuracy on image gallery dataset showing citrus disease images, 90.4% on a local dataset and 89% on combined dataset.

A clustering algorithm, namely k-Means was applied in [151] in conjunction with an SVM, for the classification of papaya diseases from images taken from mobile devices. K-Means was responsible for the segmentation of the diseased region while SVM for the feature extraction and classification, resulting to an approximately 90% classification accuracy. Detection of canker in leaves was tackled in [152], with the use of the kNearest Neighbor (kNN) method, achieving detection accuracy 96% for late disease stage, however, in indoor conditions. A pre-trained CNN model, namely VGG16 [153], was implemented in [154] for mildew disease identification in pearl millet, achieving accuracy of 95%. Lastly, a powerful algorithm from the family of ANNs, Deep Residual Neural Network

(DRNN), was implemented in [155] for detecting at the early stage multiple plant diseases in wheat. The approach aim at model deployment on a smartphone, and managed to reach 87% accuracy under exhaustive testing and 96% accuracy on a pilot test conducted in Germany.

2.4.2 Summary of the reviewed studies

A total number of 26 articles were surveyed, including sub-categories of crop management additional to crop disease detection. Table 1 provides a summary of these articles alongside relative information such as publication year, crop type, the purpose of the study, the applied algorithm and the results for each study.

Table 1: Summary of the reviewed publications.

Ref	Year	Cat.	Crop	Purpose	Algor.	Results
[156]	2019	YP	Corn	Predictions of yields for new hybrids planted	DNN	RMSE=12% av. yield RMSE=50% STD for validation dataset
[157]	2019	YP	Wheat Malting barley	Crop YP from NDVI and RGB data	CNN	Early period: MAE=484.3kg/ha MAPE=8.8% Later: MAE=624.3kg/ha MAPE=12.6%
[158]	2019	YP	Rice grain	Acquire important features associated with rice grain yield	CNN	RGB and multispectral images: R^2=0.464~0.499 MAPE=26.61%
[159]	2019	YP	Strawberry	Strawberry flower detection system for YP	R-CNN	Av. accuracy=84.1% Av. occlusion=13.5%
[160]	2019	YP	Wheat	Predict wheat yield across Australia	SVM RF NN	YP at the statistical division level: R^2 ~0.75
[161]	2020	YP	Wheat	Winter wheat YP based on multi-source data	SVM GPR RF	R^2 >0.75 Yield error <10%

[162]	2019	YP	Maize	Meta models for the prediction of crop model outputs	XG Boost RF	R^2 >0.96
[68]	2020	CDD	Walnut	Classify leaves to healthy and infected	CNN	Accuracies ranging from 92.4% to 98.7%
[147]	2019	CDD	Vine	Identify crop disease on leaf sample images	One Class SVM model	Total success rate of 95%
[148]	2018	CDD	Vine	Identify infected areas of grapevines	CNN	Accuracy more than 95.8%
[149]	2018	CDD	Maize leaf	Improve identification accuracy of maize leaf diseases	CNN	Accuracy: 98.9% for GoogleLeNet 98.8% for Cifar10
[150]	2018	CDD	Citrus	Detect and classify diseases in citrus plants	M-SVM	Accuracy: 97% on image gallery dataset 90.4% on local dataset
[151]	2018	CDD	Papaya	System that determines the papaya diseases	Clusteri ng SVM	More than 90% classification accuracy
[152]	2019	CDD	Citrus	Remote sensing technique to detect citrus canker	KNN	Accuracy: 94% healthy and asymptomatic trees 96% healthy and canker-infected trees
[154]	2019	CDD	Pearl millet	identify mildew disease in pearl millet	CNN	95% accuracy 90.50% precision 94.5% recall 91.75% f1-score

[155]	2019	CDD	Wheat	Detect many plant diseases in real conditions	RNN	Accuracy: 0.87 under exhaustive testing 0.96 on a pilot test
[163]	2019	WD	Bermuda grass	Detect weed in bermudagrass	CNN	F1 score value>0.99
[164]	2018	WD	Sugar beet	Create pattern based on shape features for different weeds	ANN SVM	Correctly classified weeds: ANN:92.50% SVM:93.33% plants: ANN:93.33% SVM:96.67%
[165]	2018	WD	Bean spinach	Learning method with unsupervised training data collection	CNN	Differences in accuracy: 1.5% in spinach, 6% in beam compared to supervised training data labeling
[166]	2018	WD	Maize	Weed detection in early season maize field	RF	Accuracy=0.945 Kappa value=0.912
[167]	2018	WD	Cotton sunflower	Design prescription maps	RF	Accuracy: 84% of weeds in cotton field, 81.1% in sunflower field
[168]	2018	WD	Sugar beet	Crop-weed classification system	FCN	Precision: 98.3%, 99.1% and 85.5% for crop, weed and intra weed
[169]	2018	WD	Rice	Generate a weed cover map	FCN	Overall accuracy: 0.935
[170]	2018	QA	Pepper	Classify seeds to high and low quality	MLP	99.4% stability rate
[171]	2019	QA	Soybean	Classify 10 soybean varieties	GA-BP and T-S fuzzy NN	Av. accuracy: 96% for training set, 84% for test set
[172]	2018	QA	Beer	Evaluate intensity levels of sensory descriptors in beer	ANN	High correlation (R=0.91) to predict the intensity levels of 10 sensory descriptors

Algor.: Algorithms; ANN: Artificial neural network; Av: Average; Cat.: Categories; CDD: Crop disease detection; CNN: Convolutional neural networks; DNN: Deep neural network; FCN: Fully convolutional network GPR: Gaussian process regression; KNN: k-nearest neighbors; MAE: Mean absolute error; MLP: Multilayer perceptron; NN: Neural network; R-CNN: Region based convolutional network; RF: Random forest; RNN: Recurrent Neural Network; RMSE: Root mean square error; STD: Standard deviation; SVM: Support vector machine; WD: Weed detection; QA: Quality assessment; XG-Boost: Extreme gradient boosting; YP: Yield prediction;

Nine (9) out of the total twenty-six (26) reviewed articles, constituting the majority, were related to crop disease detection (34.62%). Seven (7) articles were in reference to yield prediction (26.92%) as well as weed detection (26.92%). The minority of articles was related to quality assessment with three (3) articles (11.54%). A visual representation of the crop management's four groups distribution is given in Figure 13.

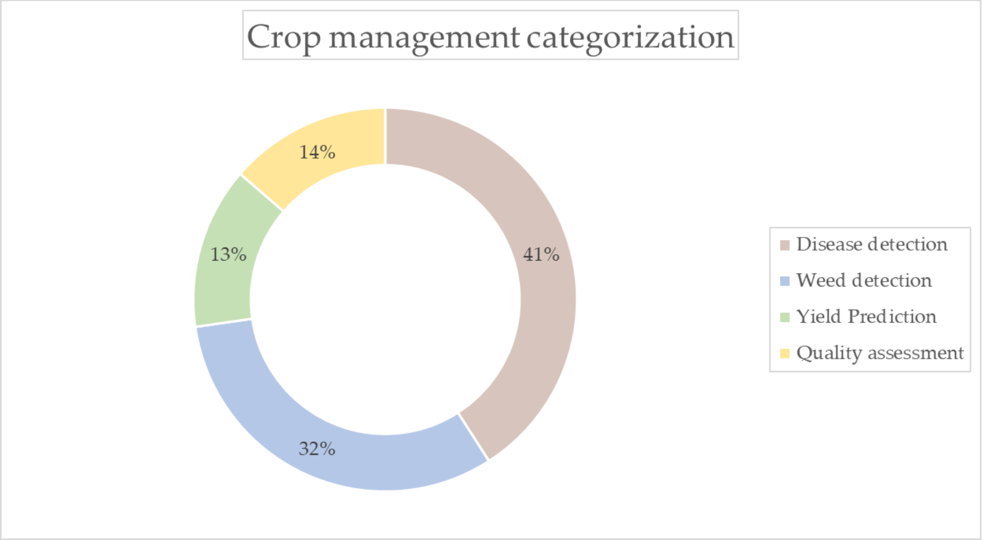


Figure 13: Distribution of the four categories of crop management.

Within these reviewed articles, a total of twelve (12) different ML algorithms were applied. The most implemented algorithm was CNN, found in nine (9) studies (28.13%), which is logical since most studies utilized images with high complexity for feature extraction. Second was SVM, applied in six (6) studies (15.63%), with RF following, appearing in five (5) works (15.63%). ANN and FCN (Fully Convolutional Network) appeared in two (2) articles (6.25%) and finally, DNN, Gaussian Process (GP) regression, kNN, MLP, RNN and XGBoost were found in one study (3.13%). An analytical plot can be seen in Figure 14

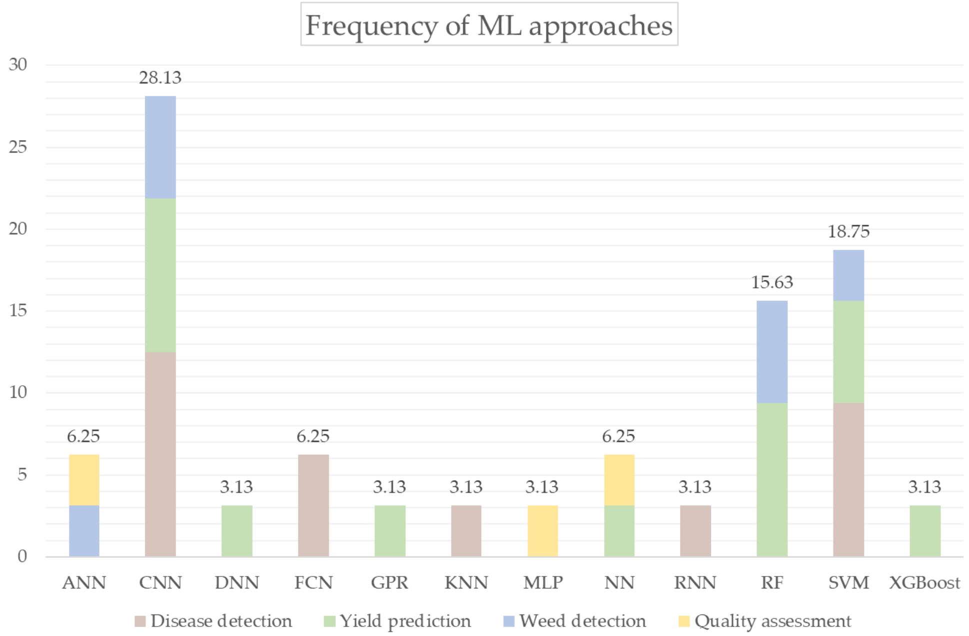


Figure 14: Frequency (%) of machine learning algorithms appearing in the reviewed studies.

Crop disease detection is the largest and most common problem associated with precision agriculture. Diseases can have significant impact on production and its financial consequences, especially in high value crops. The economic impact crop diseases can incur to farmers and producers can be significant, therefore, there is imperative need for the development of detection tools. Simultaneously, the technology is ripe for the development and deployment of such systems, since the necessary major components are: accessibility ease, hardware affordability, availability of vast amounts of data, collected by sensors and/or cameras, and sophisticated ML algorithms to train models for performing highly complex and demanding tasks, to assist the works experts conduct on the field.

3 METHODOLOGY

Aim of this thesis is to develop a multi-level methodology with ultimate goal to identify disease-infected trees in operational, agricultural environments. Such environments are extremely rich in information, because they contain a variety of diverse, different sized objects, such as trees, fruits, leaves, and weeds, each one with their own different features. This information can be captured with a variety of optical and imaging sensors, however, aim of the methodology is to rely on the cheaper and more available RGB camera sensors, similarly to the perception of a human expert. A crucial factor for the development of the proposed methodology is the large amount of data that can be gathered, thus leading the way for the implementation of machine and deep learning algorithms, which can achieve robustness and high accuracy performance.

The developed methodology is divided into three levels. Initially the precise location of the trees in an orchard is defined, based on the shape of their canopies, with images collected by an unmanned aerial vehicle (UAV) that flies above the orchard. Then, once the tree's location is accurately located, it is possible for an unmanned ground vehicle (UGV) to navigate through the orchard and inspect its canopy from a distance that is convenient for the UGV's manoeuvrability. The inspection aims to detect disease-infected leaves within the canopy, similarly to how a human expert would do. Finally, in order to be able to detect disease-infected leaves inside a plethora of leaves and background information, a classifier is designed and trained with sole purpose the successful distinction between healthy and disease-infected leaves, but in real conditions.

The multi-level methodology a whole, as well as each level individually, can be considered as critical steppingstones for a plethora of agricultural operations. Path planning, a crucial agricultural aspect with huge impact on operations, relies heavily on the proper mapping of an orchard. Accurate tree localization allows accurate orchard mapping that consequently enables optimized path planning. Variability maps are a useful tool, used in tandem with remote sensing applications, and can display areas with common characteristics within a range of interest. They can be used to demonstrate disease spread within an orchard and enable precision spraying for efficient treatment and reduction in costs and energy. Therefore, correct disease identification in tree level allows to create accurate variability maps, and consequently help the producers reduce fungicide costs, overall energy costs, and increase yield and therefore profits.

Each of the three levels of the methodology is described and presented in detail in the following paragraphs, starting from the efficacy of classifying disease-infected leaves in real-life conditions, to detecting a number of infected leaves within leaf-rich canopies and finally to the accurate localization of trees in the orchard by canopy segmentation. For the training, validation and testing of the developed deep learning models, images were collected from a walnut orchard, that partially suffered from the anthracnose fungi, well known for its catastrophically impact to high value crops.

3.1 Leaf-based image classification for disease detection with convolutional neural networks

The first level of the methodology focuses on the development of a model that can successfully classify on-field images of healthy and anthracnose-infected leaves. Initial scope is to validate that this can be achieved at a high accuracy, so that it can be used for on-site inspections. An approach was investigated for the proper classification of anthracnose-infected leaf images, based on recognizing brown-yellowish marks that are present on the leaf as circular spots or along its perimeter, as seen in Figure 15.

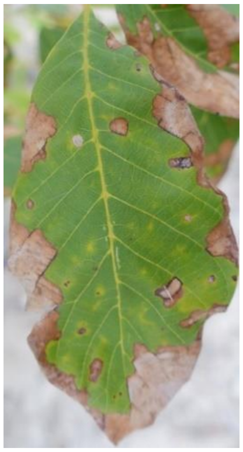


Figure 15: Anthracnose presence on a walnut leave.

Deep learning algorithms, like CNNs, have been chosen for the proper training of a classifier that will be able to distinguish images with infected leaves from images with healthy ones. The fundamental principles behind CNNs are the convolutions and pooling operations, which when applied on an image, they transform it in such a way that enhances its desired features. This process creates several images, each of whose impact is constantly calculated and optimized. The images with the most useful features are ultimately the ones that affect the distribution of weights in the final model. Convolutions are the mathematical operations that give CNNs the advantage, compared to other algorithms, in the task of image classification. As a consequence, CNNs have been consistently outperforming ANNs and SVMs, which were the state-of-the-art in computer vision and image analysis, up until the time CNNs were introduced [24]. On top of that, a Fast Fourier Transform (FFT)-based preprocessing technique was implemented for feature extraction on the images. An optional background removal method was ultimately investigated, for evaluating the background's effect on the performance of the classifier.

3.1.1 Process Description

The process followed in this level of the methodology is shown in Figure 16. There are three stages:

- Preprocessing, where the images are gathered, prepared, and finally split,
- Learning process, i.e. ML training, where the images are used to train a classifier until it reaches its maximum potential by self-improving methods, and
- Inference/predictions, where characterized test images are provided to the classifier to evaluate its actual performance.

The stages are presented descriptively at Figure 16.

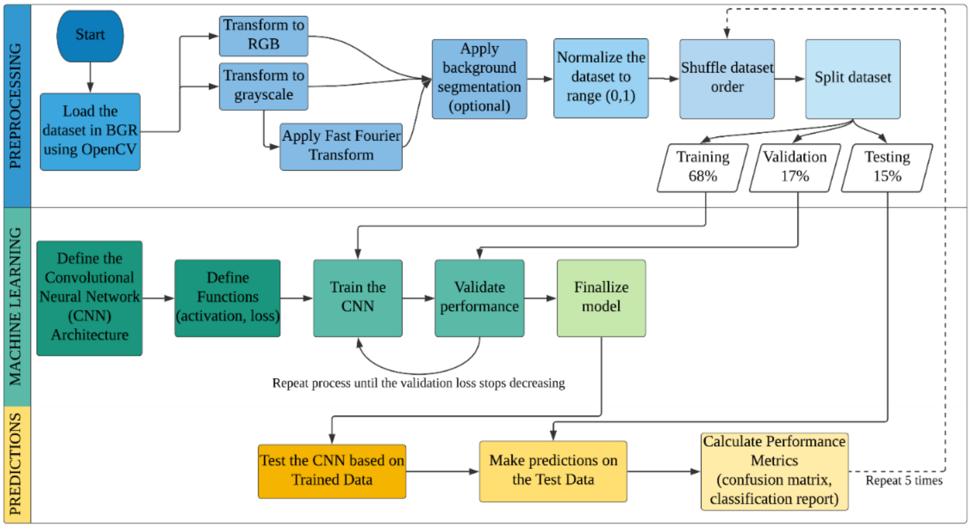

PREPROCESSING
Start
Load the dataset in BGR using OpenCV
Transform to RGB
Transform to grayscale
Apply Fast Fourier Transform
Apply background segmentation (optional)
Normalize the dataset to range (0,1)
Shuffle dataset order
Split dataset
Training 68%
Validation 17%
Testing 15%
MACHINE LEARNING
Define the Convolutional Neural Network (CNN) Architecture
Define Functions (activation, loss)
Train the CNN
Validate performance
Finallize model
Repeat process until the validation loss stops decreasing
PREDICTIONS
Test the CNN based on Trained Data
Make predictions on the Test Data
Calculate Performance Metrics (confusion matrix, classification report)
Repeat 5 times

Figure 16: Process flowchart of the approach followed for the image classification.

3.1.1.1 Preprocessing

The preprocessing stage of a data-driven methodology is a crucial part for the overall performance. It includes methods that load, clean, transform, resize and engineer data is such ways, that make them suitable for usage with the selected algorithm and application. An important function of the preprocessing stage is to shuffle and split the data into training, validation and testing portions, for ensuring an unbiased training procedure and reliable testing results. The steps followed in this methodology are presented here.

Load the dataset

The images are recursively loaded based on their pathnames. A label value of '0' or '1' is appended to each loaded image, depending on the prefix of their filename.

Colour conversion

Two colour conversion functions were implemented: a BGR to RGB colour-space conversion, where the image's colour channels' order switches to the most common (RGB) format, as well as a BGR to the grayscale colour-space conversion, where all the colour information is removed from the images, and only the brightness and saturation for each pixel is maintained.

Feature extraction (grayscale only)

A feature extraction method, namely the Fast Fourier Transform (FFT), is additionally applied for the grayscale images. FFT is particularly valuable in bringing out edge features of objects in images (i.e., leaves), thus potentially assisting the model's performance improvement, compared to a straightforward approach of grayscale images.

Background segmentation (optional)

This optional method removes the background from images with the use of commonly used classical computer vision techniques such as thresholding, dilation and erosion.

Data normalization

The dataset values are normalized within the [0,1] range in order to ensure that the loss function, which is usually not convex, finds the global minimum as easy as possible. Reducing the range of input values also assists the convergence of the backpropagation algorithm.

Dataset shuffling

The dataset is being shuffled by implementing the “random.shuffle” method, which is an order-shuffling algorithm based on a random number generator. After the application, the order of the images, which originally was alphabetical, is now mixed throughout the dataset.

Dataset splitting

The dataset is being split in the following fashion: First, 15% of the total dataset is held out and set as the testing set, then the remaining dataset is split again in the 80/20 fashion, where the small portion is used as the validation set. The rest is used for training the algorithm. The training and validation sets are used for the training of the prediction algorithm, while the testing dataset is used only after the classifier is trained.

3.1.1.2 Learning process

Learning process refers to the training of the model based on training data and the monitoring via validation for ensuring improvement in order to achieve the best performance possible.

CNN architecture definition

The architecture of the CNN is defined based on experimentation and trials. The number, type and order of layers is the important aspect that can lead to effective network architecture.

Functions definition

Various functions that are used throughout the network are being defined through bibliographic research and trials. The activation function is carefully chosen for defining the output of the layer, and the loss function used for the optimization of the network’s weights.

Training

This is the process where the algorithm tries to create a function that describes the desired relation, based on the training data. It then makes predictions based on this function and moves to the next step.

Validation

Following the previous step, the function that the algorithm created is being evaluated against the validation dataset. The error in the predictions is being calculated, and the algorithm tries to find a way to minimize this error. This part defines the "learning" of the process.

Finalizing the model

The training-validation process repeats itself until the algorithm cannot improve itself anymore. When the training loss and the validation loss have become almost the same, lowest-possible value, and before the validation loss starts to increase, the procedure stops, and model is finalized to its best state.

3.1.1.3 Inference

Inference is the process where the trained model makes predictions. The predictions are done on data that have been withheld from the training process, therefore the performance achieved during this process is more reliable than that of the training and validation. The steps of this stage are described.

Test the CNN based on trained data

The performance of the finalized model is measured by its validation accuracy, i.e., the level of accuracy that could be achieve based on the data that it has been given to train and evaluate. This is only indicative of the model performance since the actual performance is only evident when predictions on unknown data take place.

Make predictions on the test data

Testing data, which is completely unknown to the model, are used to make predictions on each image's class. For each image, the classifier provides a predicted class, which is stored alongside its true class.

Calculate performance metrics

This is the final step where the predicted classes are being cross-checked over the true classes of the testing images. Then, the testing accuracy, precision, recall and f1-score of the model are calculated. A confusion matrix is also used to visually evaluate the performance of the model, and also check if it is biased over any class.

After the performance metrics are calculated, they are stored, and the algorithm returns to the shuffling of the original dataset and continues the process. For the particular approach, the algorithm runs a total of five repetitions, and finally the mean value of the performance metrics is being calculated. The reason why this repetitive loop is performed is to ensure the stability of the model and eliminate outliers that are not representative.

3.1.2 Feature Extraction Preprocessing with Fast Fourier Transform

Fast Fourier Transform (FFT) is used in image preprocessing for the easy detection of abrupt changes in images, such as the presence of anthracnose in leaves. The image is considered as a signal described in a two-dimensional spatial domain. Abrupt changes in images are mainly considered as high-frequency signals. As a result, image representation in the frequency domain is a powerful tool for the detection of such changes, so that they can be enhanced or removed, depending on the required task. Moreover, applying filters to images in the frequency domain is computationally faster than to do the same in the spatial domain. Similar applications of the FFT combined with ML algorithms have been tested in image classification applications of different domains with promising results [26].

When applied on an anthracnose-infected leaf image (Figure 17.a), FFT decomposes the signal into its periodic components, so that it produces their frequencies. After implementing the FFT, the image is converted from the spatial domain to the frequency domain. At this new representation, each point denoted a specific frequency that is included in the original image. After the FFT application, the image is shifted in such a way that the fixed-value DC-component $F(0,0)$, which corresponds to the average brightness, is displayed in the center of the image.

The next step is to calculate the magnitude of the Fourier Transform, as it contains most of the information of the image structure. More specifically, if the magnitude changes abruptly, the signal is considered to be high frequency. Figure 17.b shows the magnitude spectrum after the implementation of FFT with the pixels in the center of the image representing its low frequencies. In general, anthracnose is considered as an abrupt change, and is consisted of high frequencies. As a result, a high pass filter which allows only high frequencies should be implemented. Since the low frequencies are near the center of the Fourier image, a radius around the center is determined, and all the frequency components within that radius are constricted. For that reason, each image gets multiplied by the Gaussian high pass filter where a smooth cut off process is used. The cut off frequency is considered to be equal to the standard deviation σ in the frequency domain and is equal to 0.3.

The magnitude spectrum after the use of the filter where the low frequencies were successfully removed, is shown in Figure 17.c. Finally, the inverse Fourier transform was implemented in order to obtain the original image without low frequencies, as it is shown in Figure 17.d.

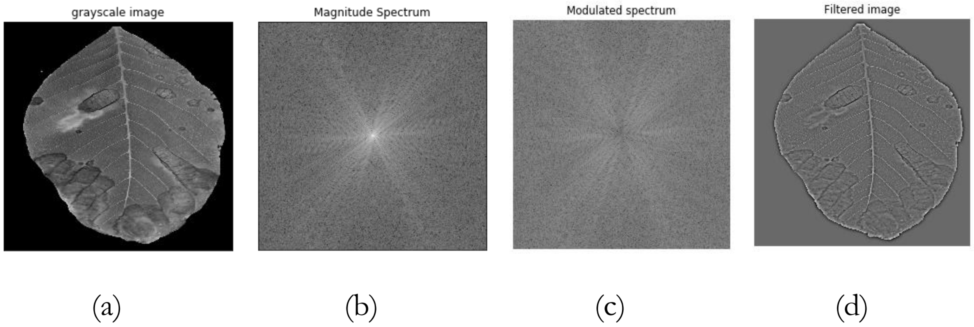


Figure 17. Fast Fourier transform (FFT) steps where: (a) the original image; (b) is analysed into a magnitude spectrum; (c) into a modulated spectrum; (d) and finally into the feature-rich image.

The choice of simple FFT over wavelet transforms or more advanced techniques is because the time domain is irrelevant to the specific problem. The photographed leaves have no temporal information, therefore there is need to only focus on the frequency precision, and FFT is the most appropriate method for this application [27].

3.2 Object detection with single-shot detector algorithm for tree-level disease classification in orchards

The second level of the proposed methodology focuses on the implementation of the aforementioned classifier to an object detector. The object detector was trained so that to able to accurately detect the presence of anthracnose-infected leaves in images containing healthy leaves. The task at hand, poses the highest level of complexity regarding disease detection since it takes distant images of canopies containing many leaves, and tries to find signs of infection in individual leaves. Such leaves could be hiding in shadows, behind others, or simply be “invisible” due to the extremely information-rich image, however, the developed methodology is considering all associated problems for tackling each one.

3.2.1 Process Pipeline

The methodology pipeline developed in this level, can be broken down into three sequential tasks. The first task regards the proper training of a model that is able to identify infected leaves in contrast to healthy ones, in leaf-rich images. The second task focuses on the development of a model that can locate anthracnose-infected leaves at tree-level, from images that are taken in real field conditions, from a relatively far distance (approximately 2-3m). The third task aims to utilize the produced knowledge that derives from the detection and localization of anthracnose-infected leaves, with ultimate goal to classify the infection level of each tree within the orchard, potentially creating a variability map. These tasks are described in the following paragraphs.

3.2.2 Model Training

The open source TensorFlow Object Detection API was exploited to enable the deployment of object detection models. Instead of training the SSD (Single Shot Detector) model from scratch, the final layers of pre-trained models that had already been trained for image classification on benchmark datasets were re-trained on the acquired data. Three well-established architectures were trained, tested and evaluated with the SSD; Resnet50 [173], Inception v2 [174] and Mobilenet v2 [175]. These architectures, used as CNN feature extractors for the object detection algorithms, were originally pretrained on a large-scale benchmark object detection dataset (COCO) [176]. These particular pretrained models were selected because they provide a balance between good performance accuracy and high execution speed [177].

3.2.3 Anthracnose-infected leaves detection on tree level

Capturing a whole tree in one single image by any high resolution RGB camera, produces an image size that is relatively large compared to the relative size of the single leaves, the target objects to be detected. If the model was trained considering the original image, this size difference would make the object detection task extremely challenging due to the limited number of pixels representing the targets (leaves). Extensive investigation and testing took place verifying that at raw image size level the trained models performed poorly. An alternative approach was developed to tackle this issue.

According to proposed approach, images acquired from each tree side are segmented into sections. Different image sizes were investigated, leading to the selection of sub-image sizes of 1,280×1,280 and 640×640 pixels, where the best results were achieved. That signifies that the 5,472×3,648 pixels raw images can be segmented into 12 or 48 smaller images (Figure 18a). A script was developed for the automatic segmentation of the original images. The automatic segmentation does not produce exact pixel sizes for all the sub-images. hence, the ones located at the edges of the original image were clipped. Since the surrounding pixels mainly represented indifferent content, such as soil or sky, thus rendering them unusable, they can be removed without information loss.

During the experimentation it has been observed that poorly trained models occurred when the images that were larger than 1,280×1,280 pixels, due to the ratio of leaf size over total image size. The smaller size of leaves is related to less pixels for describing their characteristic features, and consequently, noisier information. Distinguishing infected from healthy from infected leaves at this level is already a complex task, the model training was conducted with the most obvious examples of infected leaves (true positives). Ground truth bounding boxes were created as annotations in each sub-image, that ranged in size from 50×100 to 320×320 pixels, containing the most evidently infected leaves. The threshold was set to a maximum 3 infected leaves were for each sub-image as seen in Figure 18b. Extensive investigation on the dataset resulted to the definition of this particular threshold. In detail, a maximum of 3 leaves were distinctly and undeniably classified as infected for every sub-image, since the majority of leaves were overlapped, shaded or at angles that made extremely challenging to distinguish them from the neighboring leaves even to the human eye. An attempt to annotate more leaves per image, lead to the overlapping of the boundaries of the leaves, resulting to a significant performance drop. Thus, for each sub-images, no more than 3 infected leaves were annotated.

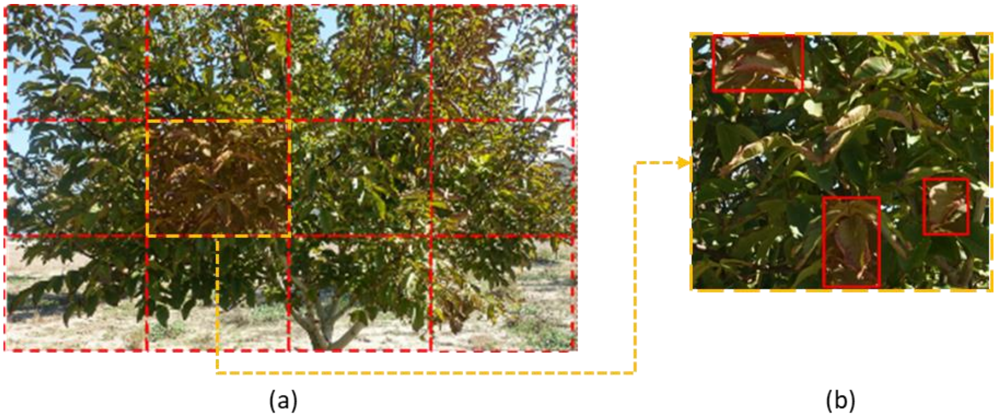


Figure 18. Proposed image segmentation of walnut tree side view image into sub-images sized 1,280×1,280 pixels (a) and ground truth bounding boxes containing anthracnose-infected leaves (b).

3.2.4 Classification of trees

This methodology's main goal is to develop a tool to automatically appoint a class on the trees within the orchards based on their health status (infected or healthy). Therefore, at a second level, the previously trained model is applied to all the walnut trees of the orchard. Ultimately, the system's performance was evaluated in real field conditions. At this stage, the trained model should be able to recognize the target (i.e., the tree), from a distance that ideally covers the total surface of the canopy. Therefore, all sides of the trees that were tested in this approach, were meticulously captured with the RGB camera. This requirement was addressed by capturing images from all four sides, 90° apart from each other, and at approximately a constant distance from the tree,. This setup was used to maximize the coverage of the foliage of the trees with coverage angle higher that 360° (including overlapping) (Figure 19).

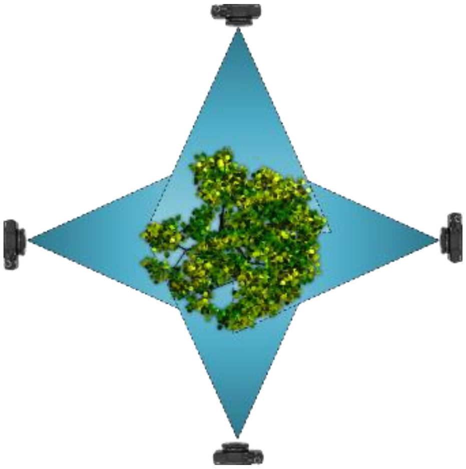


Figure 19. The image coverage of the walnut tree taken from four different angles 90° apart from each other.

To sum up, the developed methodology that is being followed in the current system level, is shown in Figure 20. The methodology is divided in three phases: a) the training phase which includes data acquisition, image preprocessing and model training, b) the testing phase including acquisition and preprocessing images from 4 sides of the remaining trees, and tree classification, and c) the evaluation phase where the predicted classification is cross-checked with the real classification of the trees according to experts' knowledge.

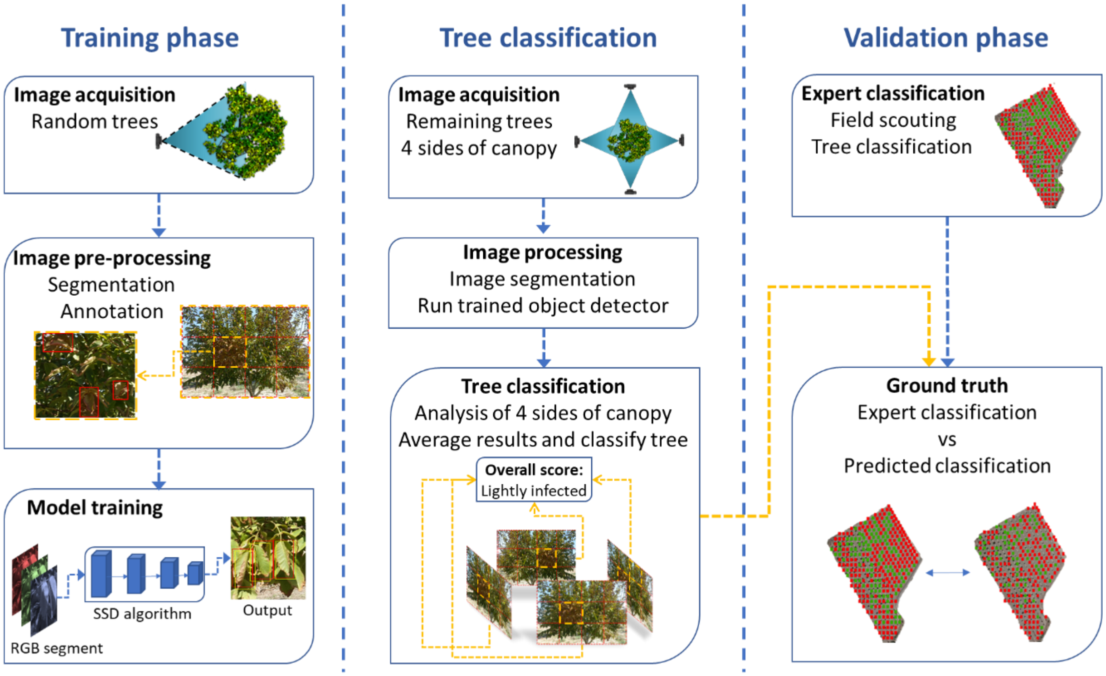


Figure 20. Process flow of the proposed methodology.

3.3 Orchard mapping in complex environments with deep learning semantic segmentation

The third and final level of the developed methodology focuses on the accurate localization of trees within orchards, from aerial images. This is a task that is more contemporary, since digital imagery from aerial means became available only recently comparing with the eons of agricultural activity of humans. However, the inherent difficulty of locating full-canopy trees in weed-ridden orchards or canopy-less trees in barren grounds, is easily understandable. This level aims to complete the methodology as a whole, by allowing unsupervised localization of trees by canopy segmentation and other traditional computer vision techniques, namely centroid calculation via mask moments.

This way, the entire operation can be performed in an unsupervised manner, starting from tree trunk localization from geo-referenced aerial imagery collected by UAVs, and then, on-field inspection by images collected by UGVs navigating through the orchard as part of operational planning.

The third level of the methodology is also structured around data-driven algorithms and computer vision techniques. A large dataset was generated from a large number of UAV captured images, and the annotation was conducted by masking the canopies of the trees in order to create a large dataset for supervised learning. A deep-learning algorithm was selected to train a model on this annotated dataset, with aim to properly identify tree canopies and segment them from the background and the rest of the objects. A mask image is produced as the output, containing the shapes of all predicted tree canopies. Right after segmenting the canopies' shape, the moment of each mask, also known as weighted average, is used for the calculation of its centroid. This is used as the most reasonable approximation of the location of the tree's trunk. The tree trunk locations can finally be computed with high accuracy, provided that the geodetic coordinates of the photographed location are retained in the orthomosaic images.

In order to tackle with the orchards' complex environments, in terms of the weed presence in the image background, and the high variability in the phenomenology of canopies due to seasonality, a deep learning algorithm, namely U-net, was considered for base for deployment, and was further tweaked to fit the problem's requirements.

3.3.1 Semantic Segmentation Architecture

U-net is an advanced type of convolutional neural network which consists of two modules, an encoder and a decoder. These networks encode the input into a latent space with the aim of creating the desired output based on the aforementioned input. The characteristic feature that distinguishes U-net from the simple encoder-decoder networks, is that it contains direct "skip" connections. These are present between the shallow encoder and decoder layers alongside the sequential structure of the architecture [178]. This way, useful features from the encoding/input layers can be directly fed to the decoding/output layers. Two modifications were implemented to the standard U-net architecture for the developed approach; the input layer was tweaked to handle both 3- and 6-channel images and the insertion of a dropout layer took place between each of the convolutional layers per block, to avoid the overfitting tendency towards that occurs in small datasets with similar visual representations. A schematic of the U-net used in this work is shown in Figure 21.

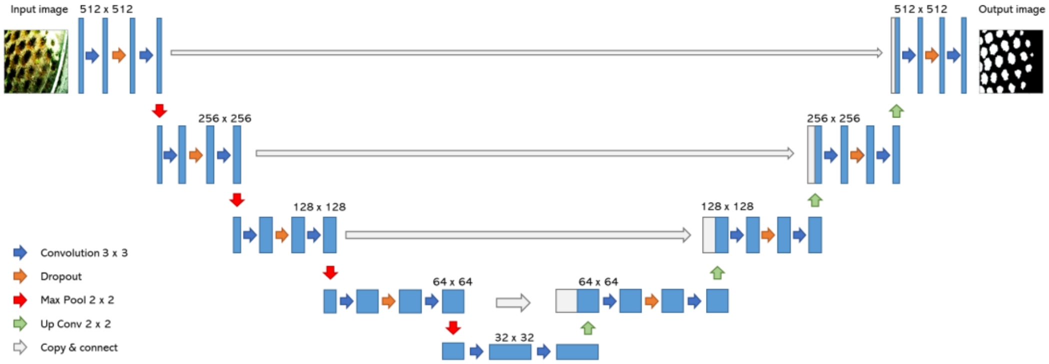


Figure 21. Architecture of the modified U-net network implemented in the approach.

3.3.2 Process Flow

The methodology developed for the canopy segmentations follows a sequential order which consists of several preprocessing, training, and evaluation steps throughout the whole process. The complete process flow can be summarized as follows:

- Data are imported and split into train and test sets. To achieve generalization and robustness on the implementation of the approach, the test set is required to contain at least one image from each use case.
- Image reshaping into a predefined aspect ratio and, colour enhancements such as contrast equalization are applied.
- The training dataset is fed to the U-net and the trained model learns to create proper segmentations for each image. An evaluation metric is used across a randomly selected validation set comprising 10% of the training set, so that the trained model can iteratively learn to create better segmentation masks.
- The trained model predicts segmentation masks for the test images and the evaluation metric is applied in order to evaluate the model's performance.
- The segmentation masks are compared with the real masks annotated by experts, and the presence of false positive or false negative segmentations is manually investigated.
- The model's overall performance is defined by the accuracy it achieved on the test dataset, as well as the ratio of false positives and false negatives over the total amount of trees in the image.

A visual representation is shown in Figure 22.

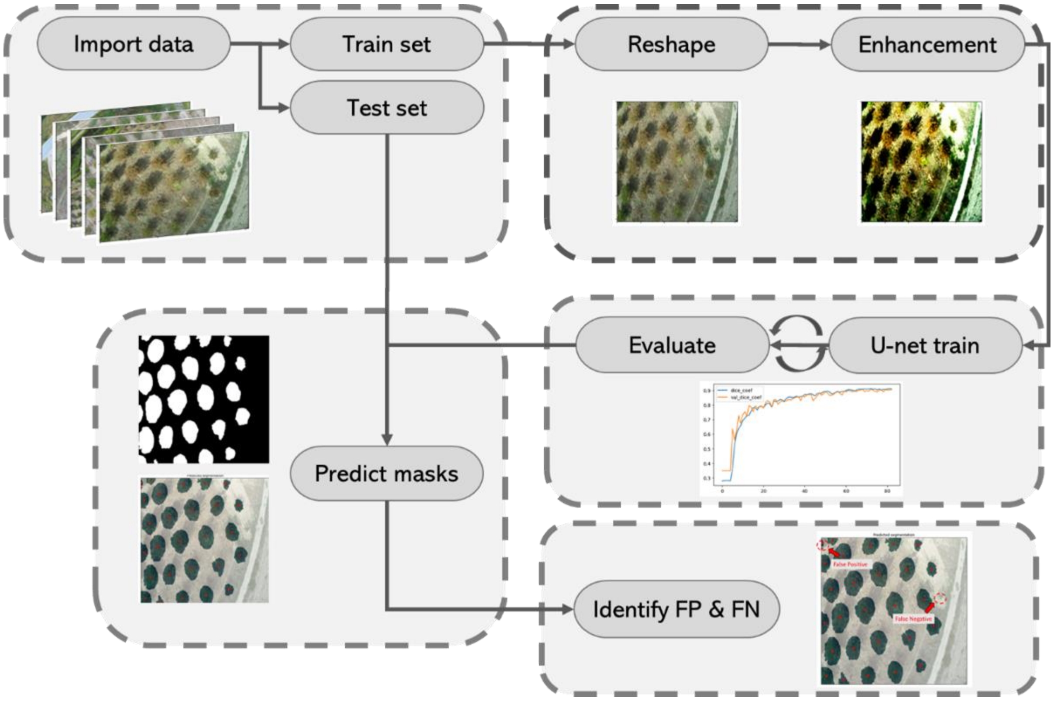


Figure 22. Process flow of the proposed methodology for creating segmentation predictions. FN: false negative; FP: false positive.

3.4 Process automation from locating the tree to detecting the disease on tree and leaf level

The system as a whole is designed to follow the logic of a linear sequence of operations. The methodology aims to integrate the aforementioned, highly complex tasks, in order to enable unsupervised robotic systems to perform the said tasks with as close as possible to human expert level accuracy. The methodology's core task is disease identification, however, the order in which these tasks should be deployed in a real-life scenario are different.

Initially, an aerial inspection takes place to identify the locations of all trees within the orchard. This will provide with exact coordinates that will be used for the navigation of the ground vehicles and their placement to each tree. Then, camera sensors will capture images of the trees' canopies in order for the detection and classification of infected leaves. Finally, this information will be used to create a variability map of the orchard, and thus provide detailed information of the disease spread within the orchard. Based on that information, additional automated systems can perform precision spraying or any other operations that are deemed necessary. A schematic of the overall process can be seen in Figure 23.

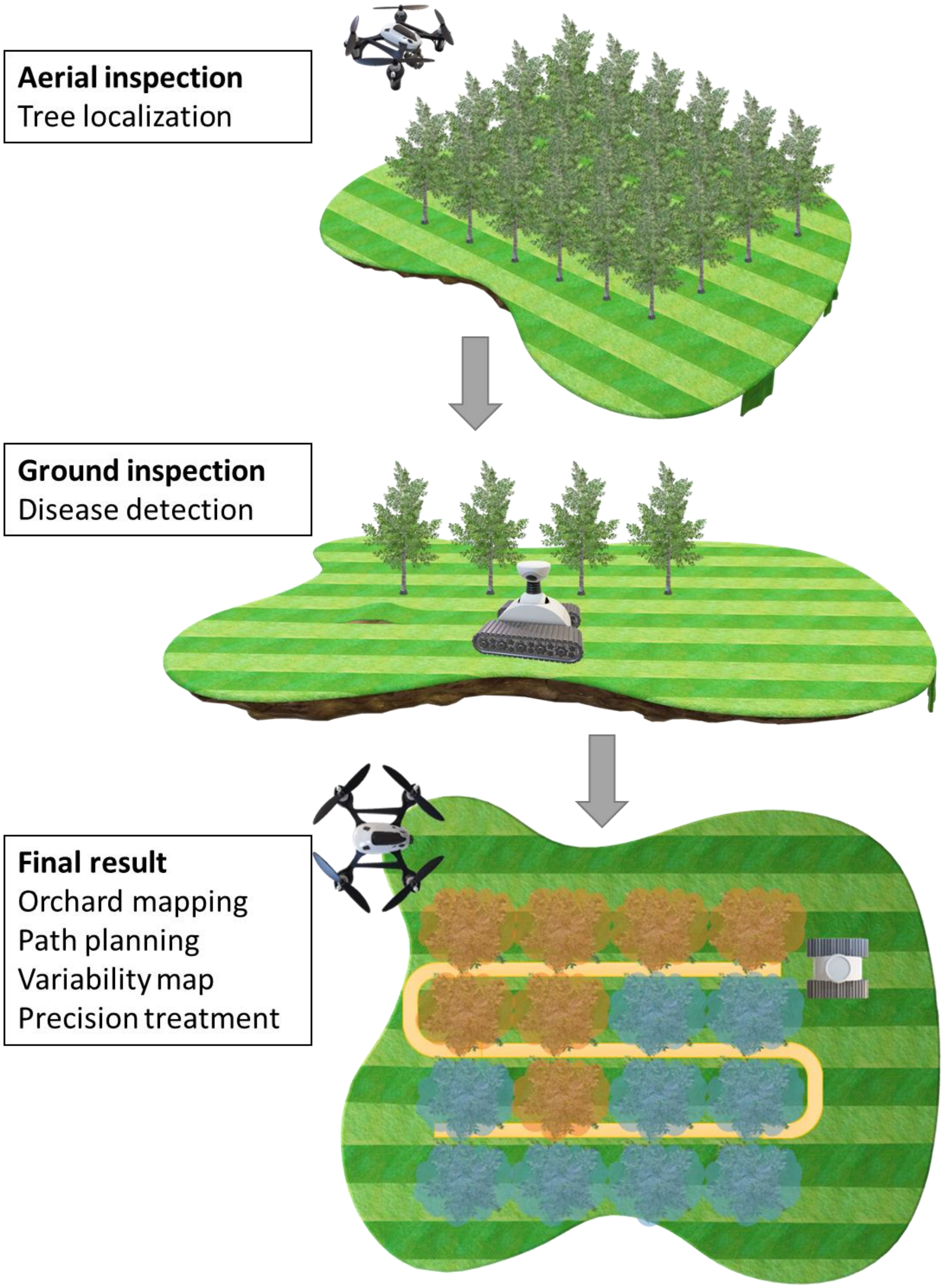


Figure 23. Schematic of the overall process automation methodology.

The automation of the entire process can be achieved only on the premise that each task is successfully completed. For instance, if the tree localization fails, important ground operations such as path planning, will not be able to be achieved since the trees will be located in different positions that the ones expected by the operating vehicles. Furthermore, if due to erroneous detection, the trees' disease classification is wrong, the variability map that will be misrepresenting the reality, and the corrective actions will be wrong and useless.

As previously described, agricultural operational areas are information-rich environments, where a plethora of visual stimuli can offer useful insights on the condition of individual trees as well as orchards as a whole. Even from relatively small orchards, the amount of data that can be extracted is large due to numerous different areas of interest, thus covering the main volume requirement for applying data-driven deep learning approaches. However, volume alone is of low value unless there is variety in data. Agricultural environments cover the requirement of variety as well, since there are numerous types of trees, plants, weeds and other objects present in orchards. Most of the aforementioned flora, especially in high-value crops, is being susceptible to the yearly change of seasons, leading to being able to collect large amounts of high-variability data within one year, when a full seasonal cycle is complete. These requirements, volume and variety, pose as ideal characteristics datasets should have, for applying machine and deep learning algorithms on them. The implementation of deep learning algorithms on the available data, for each step of the proposed methodology, is presented in the following section.

4 IMPLEMENTATION

4.1 METHOD IMPLEMENTATION – HARDWARE AND SOFTWARE SPECIFICATIONS

In this paragraph, the hardware specifications and software information, used for all tasks and experiments, is provided in detail.

4.1.1 Camera, UAV

A Sony RX100 II digital camera with a F1.8 (T) lens was used to take photos of raw format 5,472×3,648 pixels resolution at a 3:2 aspect ratio. This camera has been used for ground photo acquisition, as well as for aerial photography, since it can be mounted on UAVs. In total, two different types of UAV were used, a quadcopter (Phantom 4, DJI Technology Co., Ltd., Shenzhen, China) and a fixed-wing UAV (eBee, senseFly, Cheseaux-sur-Lausanne, Switzerland), both equipped with high-accuracy GNSS (real-time kinematic (RTK) positioning), which was also used for accurate geotagging.

4.1.2 Computer hardware and software

The personal computer that was used for all algorithmic and model development was equipped with Intel® Core™ i7-6950X CPU (@3.00GHz) and 64GB RAM PC, running Ubuntu 18.04 LTS OS. All models were trained on a Nvidia Titan GeForce GTX 1080 Ti. For the image classification task, all images were annotated manually, however, for object detection task, each image was annotated by using "LabelImg", a free (MIT license) graphical image annotation tool [179]. With this tool, boxes containing leaves infected with anthracnose were created on all training images. The produced annotations containing the coordinates of the annotated objects within the image, were saved as XML files in PASCAL VOC [180] format. The XML files were initially converted into .csv and then into TFRecords files, containing all training and testing data, in a form suitable for the TensorFlow Object Detection API [177]. For the semantic segmentation task, all images were segmented and annotated with the use of Supervise.ly, an online platform for computer vision tasks [181].

The Jupyter Notebook [30] interactive computing product under the Python programming language was used for encoding the whole process, with all developed neural network architectures, being programmed with Keras (with Tensorflow backend [31]). Additionally, data normalization, data splitting, confusion matrices and performance metric reports were programmed with Sci-Kit Learn, and OpenCV was used for loading and manipulating images. SciPy was used for additional operations such as the FFT application, Glob was used for reading filenames from a folder, Matplotlib was used for plot visualizations, and finally Numpy was used for all mathematical and array operations.

4.2 LEAF-BASED IMAGE CLASSIFICATION FOR DISEASE DETECTION WITH CONVOLUTIONAL NEURAL NETWORKS

4.2.1 Data Acquisition

Classification of images is a complex process that needs large volumes of feature-rich data, and an “intelligent” self-trainable algorithm that can learn from the data to classify correctly. All walnut trees located in the orchard, where photographed under various lighting conditions. Specifically, photos were taken during morning, noon and afternoon times, having the sun across or behind the camera. This detail is important because the position of the sun affects the diffusion/refraction of light through/on the leaf, and creates a difference of how anthracnose is captured on the image, as dark brown spots on a light green leaf or light brown spots on dark green leaf. The leaves were manually cropped and split into categories. Acquired from a walnut orchard in Volos, Greece, a total of 4,491 images contained close-ups of leaves, both with, and without anthracnose. The number of the anthracnose-infected leaf images amounted to 2,356, slightly higher than the healthy leaf images which were 2,135. In Figure 24 (a) and (c) an indicative image from a leaf infected with anthracnose is seen, with and without background, and in Figure 24 (b) and (d), a healthy leaf again with and without background.

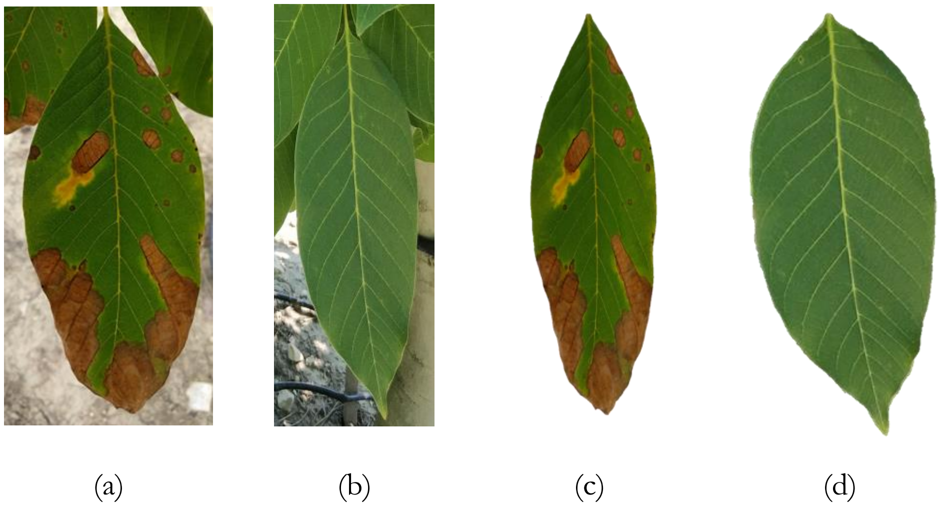


Figure 24. Images of anthracnose-infected ((a) with and (c) without background) and healthy ((b) with and (d) without background) walnut leaves.

4.2.2 Data Preparation

Since all images had different proportions, their size was modified to a 256×256 pixels proportion. This resolution was deemed satisfactory for both maintaining the images features, as well as keeping computational time to a minimum.

All images were originally saved in an RGB 3-channel, and small script was set up in order to assign a numerical value as a label to each image, according to its prefix. Therefore, images with the "anthracnose_" prefix were assigned with the value '0', and images with the "healthy_" prefix were assigned with the value '1'.

The images are loaded into the memory alphabetically due to their order in the containing folder, as well as the script used to load them. If the dataset is split as is, in training/validation/testing, it is certain that at least the training and testing datasets will not contain sufficient or any images from one or the other class. For example, the testing set will only contain healthy images if it derives only from the last images of the loaded dataset, thus there will be no proper measure for the classifier performance. Therefore, since the order of the images was defined by the name of each file, a shuffling method based on a random-number generator is used to randomly reorder the images so that the sampling is as unbiased as possible, thus avoiding improper training.

4.2.3 Data Split

Splitting the dataset occurs in a three way fashion: A training portion destined for training the classifier, a validation portion for model improvement during the training process, and a testing (held-out) portion which is a part of the dataset that is completely hidden from the training process and will be used for validating the classifier on unknown data. Each aforementioned portion is created as a subset that contains a 50/50 ratio of healthy and anthracnose-infected leaves photos. This happens to avoid any unnecessary class imbalance that would incur if a dataset would contain images from only one category.

Python's random.shuffle generator was used to shuffle the order of images and avoid having datasets with similar external conditions. This way, images of leaves with different shapes, angles, levels of infection, main leaf colour, brightness, ambient lighting, etc., are all included in all categories in order to achieve the highest possible variability. Variability in the datasets ensures that the model will be trained in the most generalized fashion possible and will be evaluated and tested under all conditions [25].

The first data split concerns the removal of 15% of the total dataset for later use as testing dataset. The remaining 85% of the dataset is then split again into an 80/20 ratio, resulting in the following portions, as seen in Table 2.

Table 2. Data splitting percentages and purpose.

Dataset	Training	Validation	Testing
% of total dataset	68%	17%	15%
# of images	3053	764	674

The validation set, which is used during the training process, allows the model to update its weights in such a way, so that its performance improves, as well as to avoid overfitting. After the model has finished training, it is tested on the testing data, and verified if it has classified the test images correctly, thus creating the need to keep the testing set hidden.

4.2.4 Performance Metrics

In this paragraph, the performance metrics that were used to evaluate the performance of this approach, are described. In general, the performance metrics are used in order to provide a common measure of the performance of the trained classifier, against new images from the testing set. The outcome of this prediction, in comparison to the actual class label that was assigned to the image, can take one of the four values, true positive (TP) or true negative (TN) if it is classified correctly, and false positive (FP) or false negative (FN) if it is misclassified.

These values are then used to calculate the performance metrics that are most commonly used in classification problems. In Table 3, the performance metrics used in this approach for evaluation of the performance of the trained classifier, together with their descriptions, as well as their mathematical formula, are described.

Table 3. Performance metrics used for the image classification.

Name	**Description**	**Formula**
Accuracy	ratio of correctly predicted observation to the total observations (preferred in balanced datasets)	(TP+TN/TP+FP+FN +TN)
Precision	ratio of correctly predicted positive observations to the total predicted positive observations	(TP/TP+FP)
Recall	ratio of correctly predicted positive observations to all observations in actual class	(TP/TP+FN)
F1 score	is the weighted average of Precision and Recall (preferred in unbalanced datasets)	[2·Recall·Precision] / [Recall+Precision]

Additional to the performance metrics, a helpful way to visualize the prediction results is the confusion matrix. The confusion matrix is a table that displays the aforementioned values in such a way that one can easily view the number of properly classified examples, as well as false positives and false negatives. In this level of the proposed methodology, which is a binary classification, the confusion matrix is of size 2×2. The template used for the confusion matrix is shown in Table 4.

Table 4. The confusion matrix template.

Confusion Matrix		Predicted	
		Anthracnose	Healthy
True	Anthracnose		
	Healthy		

All the confusion matrices with the results for each tested scenario are located in Appendix A.

Finally, a loss function (or objective function, or cost function) is used to evaluate how well the specific algorithm performs on the given data. The mean squared error (MSE) was used as a performance metric, where the average squared difference between the estimated values and the real values is calculated during the training. It is preferred because large errors create larger consequences than equivalent smaller errors, and it always has non-negative values.

The value of the loss function is important for the evaluation of the model's performance, because it can show us if the model can improve its performance and how well the predictions correspond to the real values. The closest that this value is to 0, the better will be the performance of the model.

4.2.5 Convolutional Neural Network Architecture

For this approach, a deep neural network was developed, that utilizes convolution and pooling operations, which have proven to be very effective on problems regarding image classification [182]. Classic feature extraction techniques, used in computer vision, required the manual feature selection in order to find the appropriate feature to utilize. CNNs, being a type of ANNs, can perform feature extraction automatically by applying numerous filters on the input images, and consequently learn to pick the ones that are useful for the images' proper classification.

A typical CNN structure starts with a convolutional layer, as the name states, and is generally followed by a pooling layer. This combination is repeated as many times as necessary for the defined architecture, followed by fully connected layers, before the final output layer. Moreover, when there are cases of overfitting during training, a dropout layer can be added after either a convolution, a pooling, or a fully connected layer.

In the proposed implementation, a deep-layer network is designed and developed, with five convolutional–pooling layers, a dense layer followed by a dropout layer, and finally, the output layer with one node since it is a binary classification. A layout of the network's architecture along with the layers' shapes and the number of trainable parameters, is given in Table 5. A more detailed figure of the proposed CNN is presented in Appendix B.

Table 5. The selected CNN architecture.

Layer (type)	**Output Shape**	**Param #**
conv2d_1 (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d_1 (MaxPooling2)	(None, 127, 127, 32)	0
conv2d_2 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_2 (MaxPooling2)	(None, 62, 62, 64)	0
conv2d_3 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_3 (MaxPooling2)	(None, 30, 30, 128)	0
conv2d_4 (Conv2D)	(None, 28, 28, 256)	295168
max_pooling2d_4 (MaxPooling2)	(None, 14, 14, 256)	0
conv2d_5 (Conv2D)	(None, 12, 12, 512)	1180160
max_pooling2d_5 (MaxPooling2)	(None, 6, 6, 512)	0
flatten_1 (Flatten)	(None, 18432)	0
dropout_1 (Dropout)	(None, 18432)	0
dense_1 (Dense)	(None, 512)	9437696
dense_2 (Dense)	(None, 1)	513
Total params: 11,006,785		
Trainable params: 11,006,785		
Non-trainable params: 0		

The images enter the network at a 256×256 pixels dimension. The first (input) convolutional layer consists of 32 filters (kernels) of size 3×3, always followed by a max-pooling layer of size 2×2, which chooses the maximum value of each consecutive four (2×2) pixels. The number of filters is doubled at each next convolutional layer ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512$), and at the same fashion, the max-pooling layers that follow. A flattening operation is placed in the sequence, right after the convolution and pooling operations, transforming the 2-dimensional matrices, to 1-dimensional arrays. This transformation sets up the hidden, fully connected (dense) layer with 512 nodes. Following, a dropout layer drops randomly 20% of the learned weights in order to avoid overfitting. Last is the output layer with one node, since the problem at hand is binary classification.

The activation function that is used in all convolutional and fully connected (dense) layers is the Rectified Linear Unit (ReLU) and the final activation function is the sigmoid function. The algorithm is set to train for 100 epochs, however an early stopping function prevents the network to over-fit by stopping its training when the validation loss starts to increase and diverge from the training loss. For the loss, the binary cross-entropy function is calculated with an Adam optimizer [28], and accuracy is used as the measurable metric. The ImageDataGenerator class [29] was used for the model training, in order to perform seamless augmentations on images such as rotations, shifting, zoom and flips.

4.2.6 Visualization of Convolutions

Even though CNNs are considered "black boxes", there is a way to visualize some of the computations that take place, as well as their effect on the input image at each step. Here, some of the filters that are being applied at each layer are presented, as well as the activation maps that are produced after the convolutions.

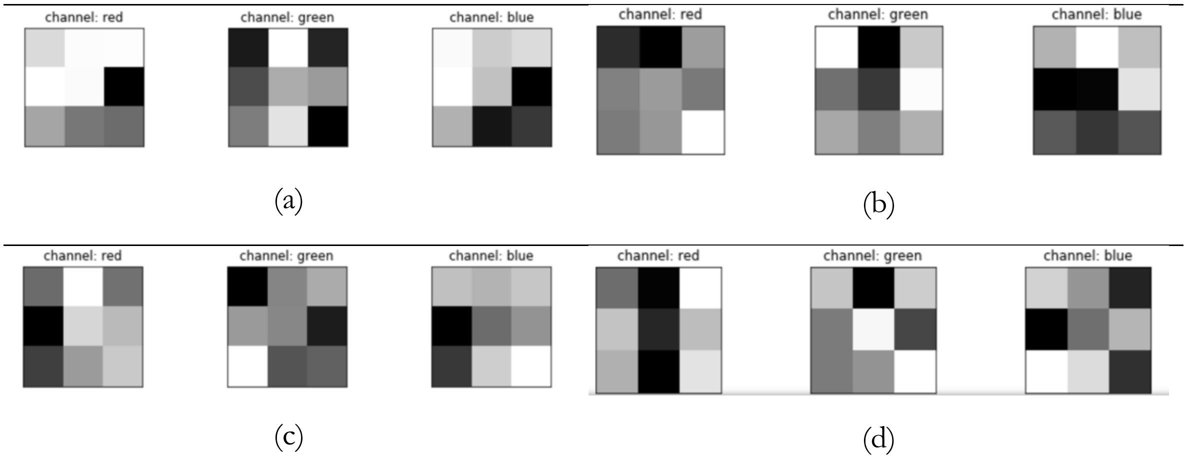
4.2.6.1 Filters

The filters are mathematical kernels that are being applied on the matrix that represents the image. In computer vision applications, they are constructed in such ways so that when they are multiplied with the image values, they bring out specific features, such as edges. For this methodology, a CNN with five convolutional layers is used (Table 6).

Table 6. Filter dimensions for each convolutional layer.

Layer	Filter Dimensions	Number of filters
Conv2d_1	3×3	32
Conv2d_2	3×3	64
Conv2d_3	3×3	128
Conv2d_4	3×3	256
Conv2d_5	3×3	512

While for the grayscale images the filter application is straightforward, for RGB images, each channel has its own filters being applied. A visual representation of the filters for an RGB image is shown in Figure 25.



(a)

(b)

(c)

(d)

Figure 25. The three RGB colour channels' filters: (a) for the first convolutional layer; (b) the second convolutional layer; (c) the third convolutional layer; (d) and the fourth convolutional layer.

These filters are important for the machine learning processing, since they bring out features of the images that will be used for the proper classification.

4.2.6.2 *Activation Maps for RGB*

The activation maps are the result of the filter application on the input image or on other activation maps when they enter the convolutional stage. These maps highlight features of the image that can potentially be useful for the classification. Two activation maps cases of a leaf infected with anthracnose are presented, one with background information, and one with the background removed. The two images are shown in Figure 26.

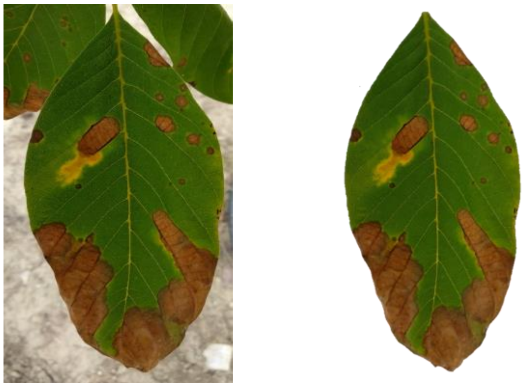
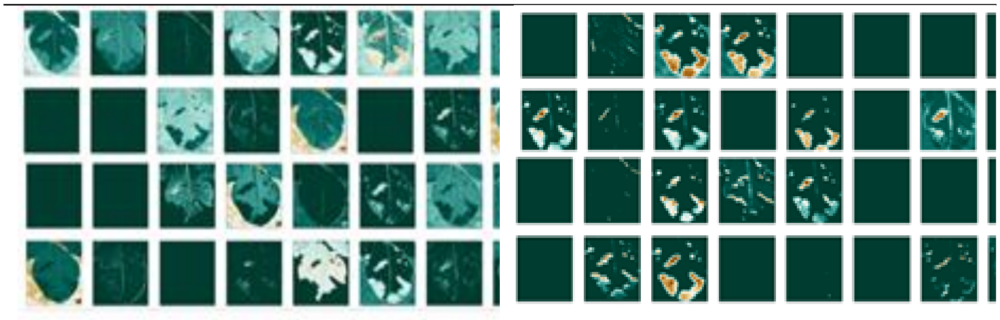
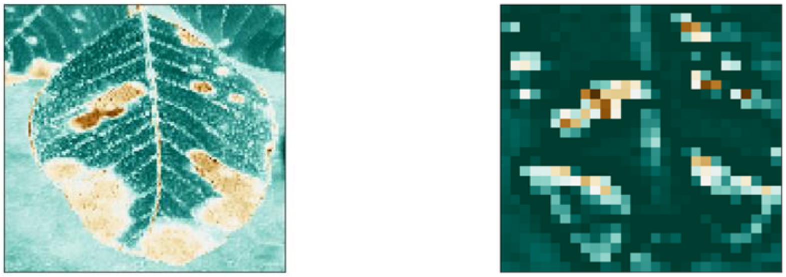

(a) (b)

Figure 26. (a) Walnut tree leaf before; (b) and after the background removal.

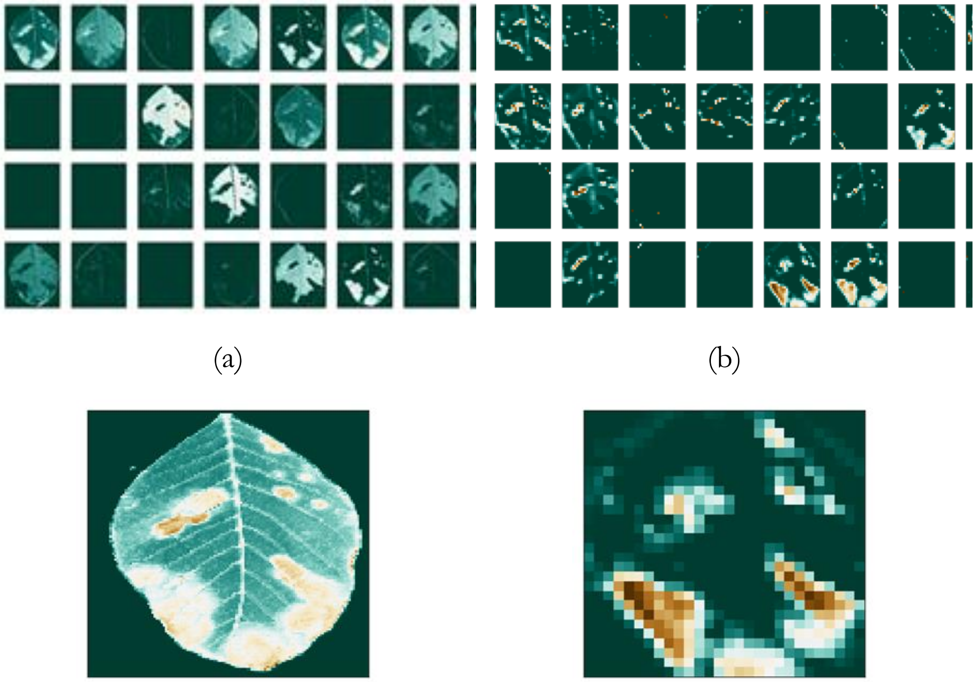
Since the images go through various transformations via mathematical operations, it is useful to see how the activation maps look like after the first and after the last convolutional layer. Activation maps for the leaf image that contains background information are shown in Figure 27, while activation maps of the leaf image that had its background removed are shown in Figure 28.


(a) (b)



(c) (d)

Figure 27. Activation maps for the leaf image that contains background information: (a) for the first convolutional layer; (b) and last convolutional layer; and individual maps for: (c) the first; (d) and last convolutional layer.



(a) (b)

(c) (d)

Figure 28. Activation maps for the leaf image without background information (a) for the first convolutional layer; (b) and last convolutional layer; and individual maps: (a) for the first; (d) and last convolutional layer.

The filters effect on each channel of the RGB images independently is also investigated. First, the image is split into each channel, and then three separate images are created, as shown in Figure 29.

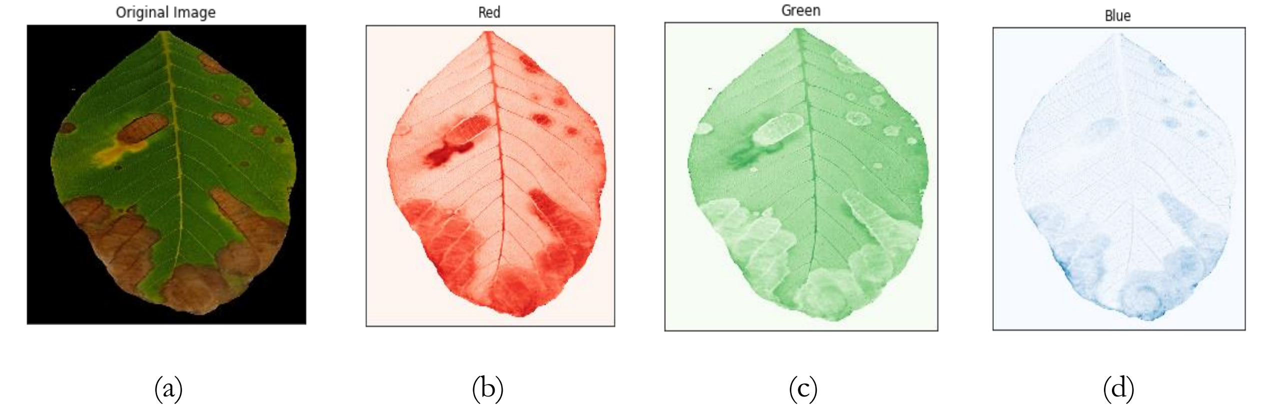
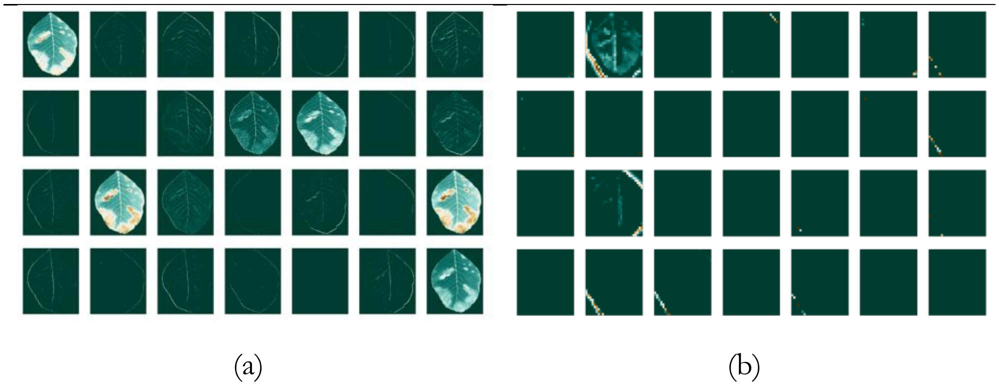


Figure 29. Anthracnose-infected leaf image after background removal (a), the image's red (b), green (c) and blue channel.

The filters are then applied and the activation maps visualized for each image on the first and last convolutional layers of the proposed network. The separate feature maps of each RGB colour are presented in Figure 30 for the red channel, in Figure 31 for the green channel and Figure 31 for the blue channel. It is clear that the filters have different effects on the different channels, obvious both in the first layer, as well as the last.



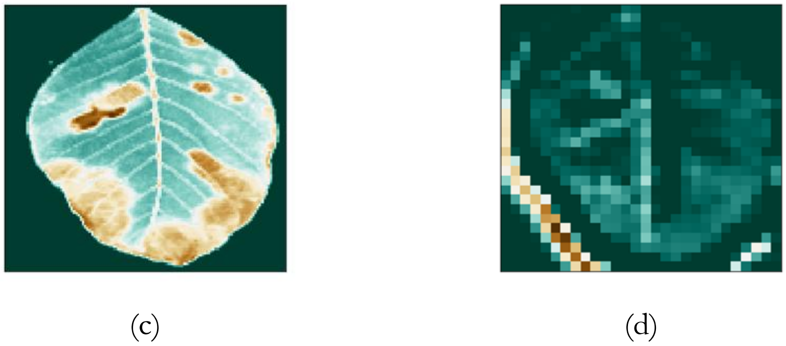


Figure 30. Feature maps of the red channel: (a) for the first; (b) and last convolutional layer; and individual maps: (c) for the first; (d) and last convolutional layer.

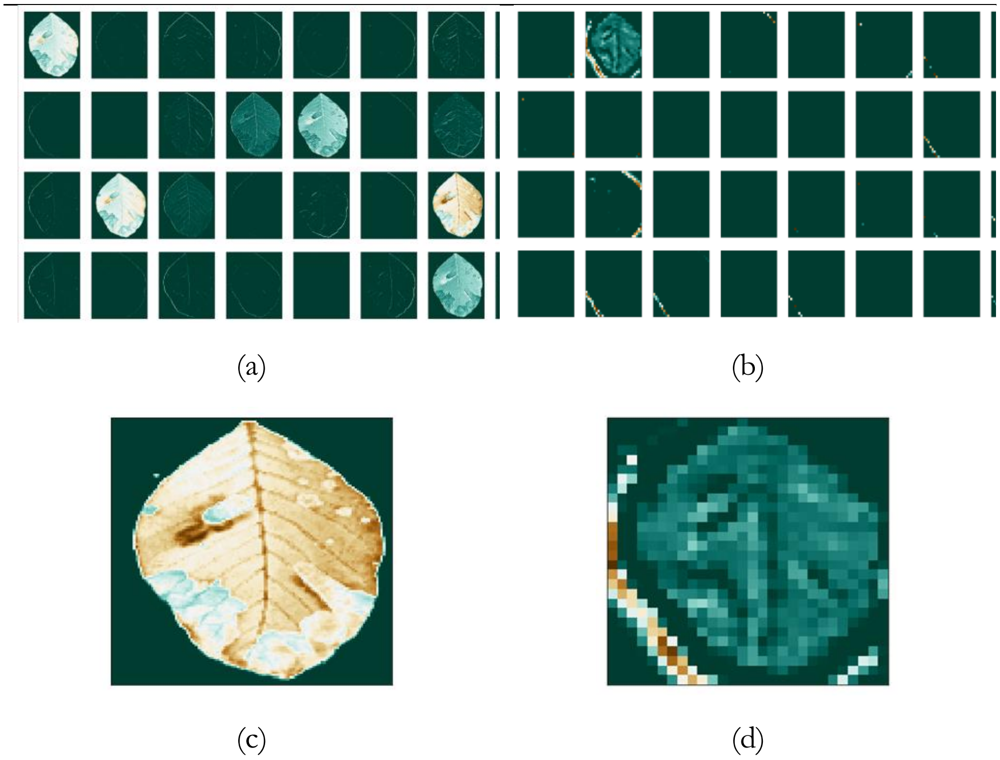


Figure 31. Feature maps of the green channel: (a) for the first; (b) and last convolutional layer; and individual maps: (c) for the first; (d) and last convolutional layer.

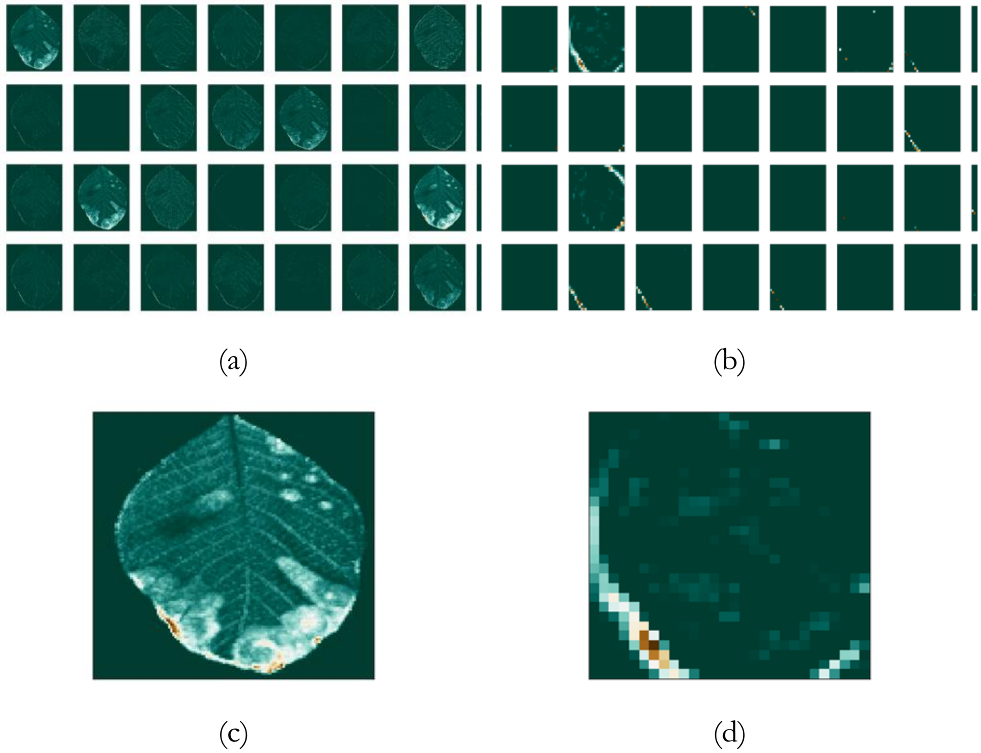
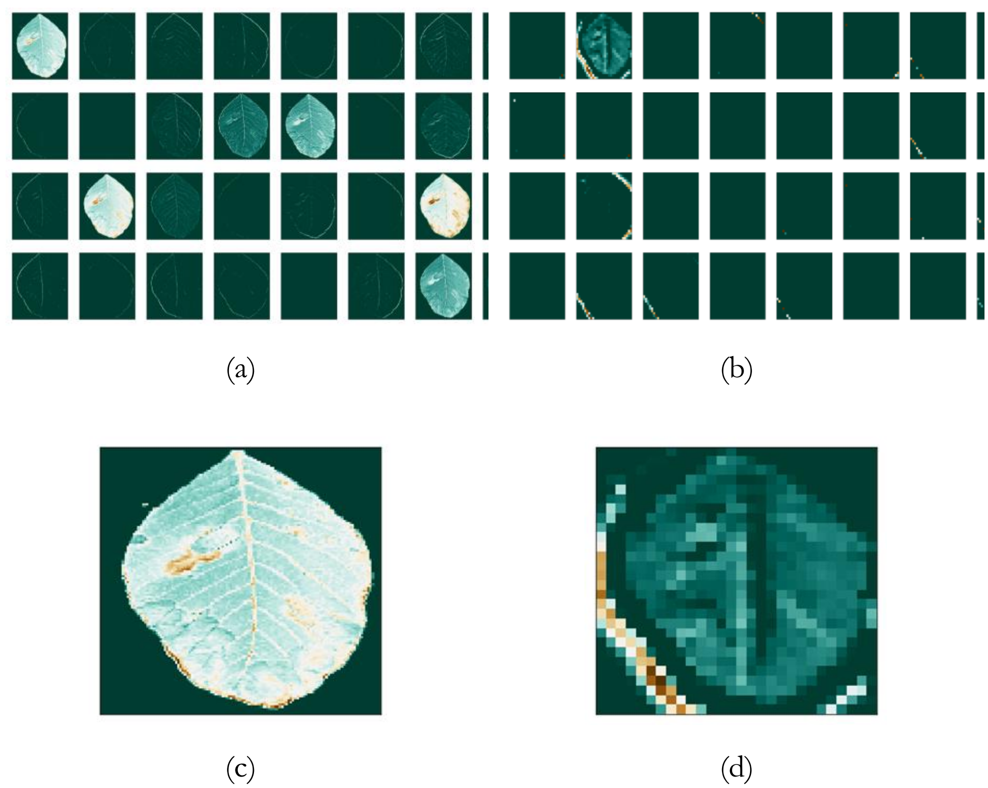


Figure 32. (a) Feature maps of the blue channel for the first; (b) and last convolutional layer (c) and individual maps for the first (d) and last convolutional layer.

4.2.6.3 Activation Maps for Grayscale

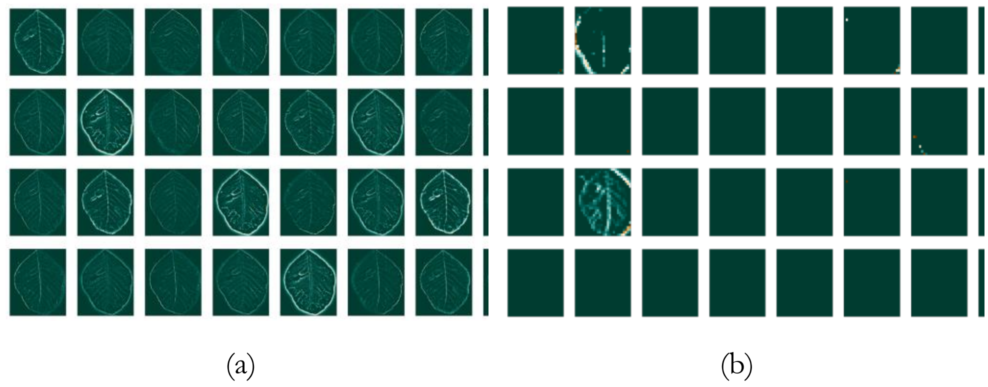
The same method is applied on three categories of the grayscale images in Figure 33 and the images that have been transformed with the FFT method in Figure 34.



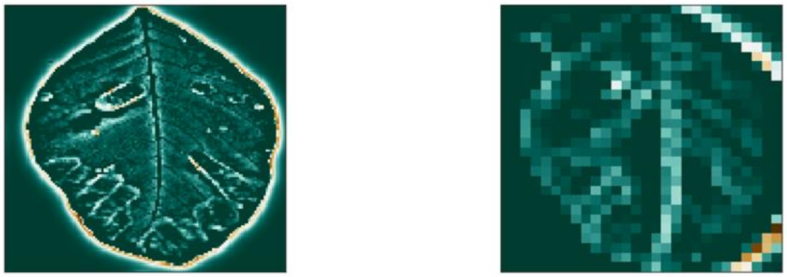
(a) (b)

(c) (d)

Figure 33. Feature maps for the image that was transformed to grayscale, without background, (a) for the first; (b) and the last convolutional layer; (c) and individual maps for the first; (d) and last convolutional layer.



(a) (b)

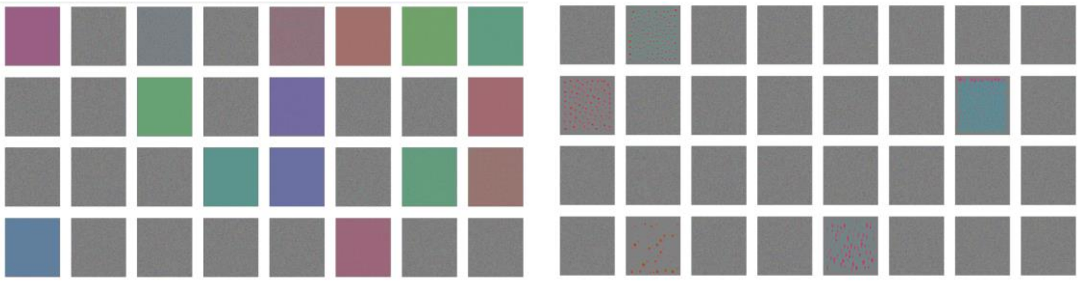


(c) (d)

Figure 34. Feature maps for the image that was transformed to grayscale, without background, and after the application of the fast Fourier transform: (a) for the first (b) and the last convolutional layer and individual maps (c) for the first (d) and last convolutional layer.

4.2.6.4 *Filter Effect on Images*

The effect of the filters can be examined by visualizing the optical pattern of the filter itself, both on a white image, and on the image of the leaf. This is done by applying gradient ascent to the input image, in order to maximize the response of the particular filter. The filters of the first and last convolutional layers are applied on a blank image, just to visualize the filter itself, as shown in Figure 35.



(a) (b)

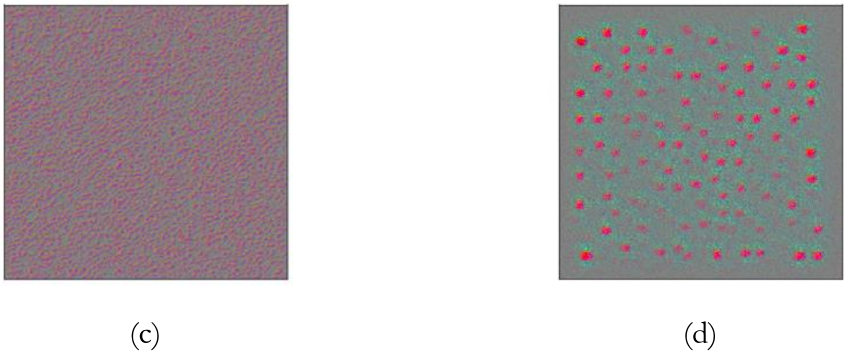


Figure 35. Filters' effect on a blank image: (a) for the first; (b) the last convolutional layer; and individual filters: (c) for the first; (d) and last convolutional layer.

Following this, the image with the anthracnose-infected leaf is used as input and the effect of the filters on the leaf image itself can be seen in Figure 36.

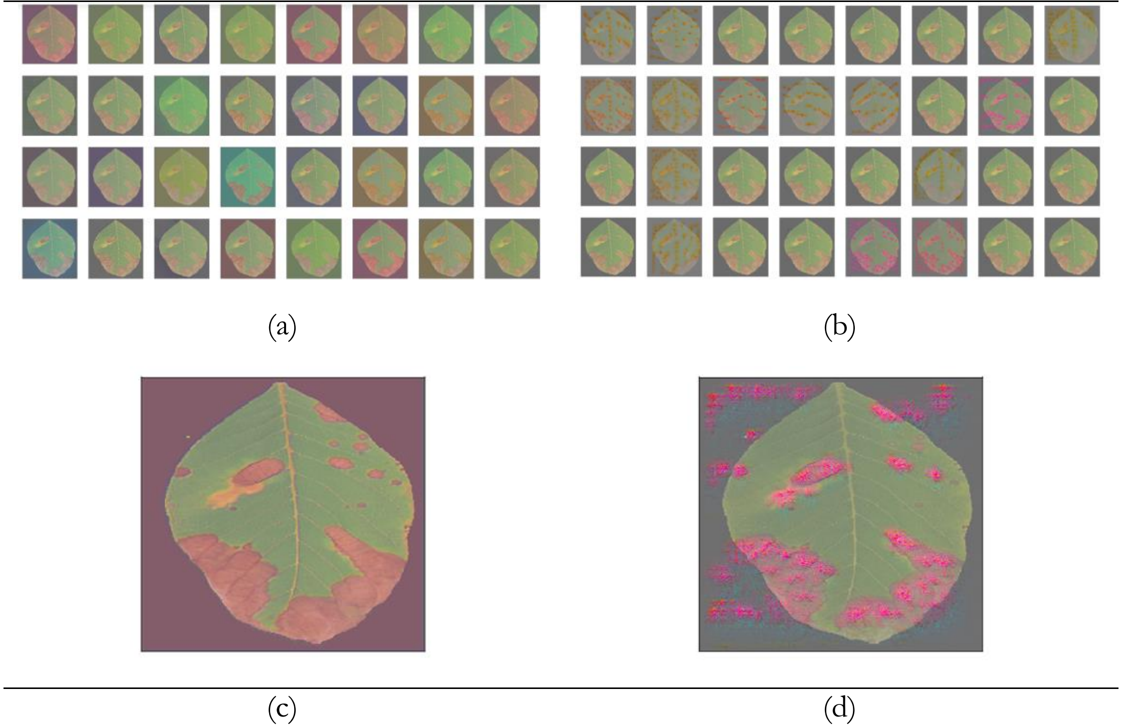


Figure 36. Filters' effect on the anthracnose-infected leaf image, without background: (a) for the first; (b) the last convolutional layer; and individual filter effect: (c) for the first; (d) last convolutional layer.

This way, insight of the depths of the deep neural network are obtained, and specifically how the convolutions affect the features of an image that contains leaves.

4.3 Object detection with single-shot detector algorithm for tree-level disease classification in orchards

4.3.1 Case study

In order to appropriately train a model capable of the proper detection of leaves infected with anthracnose, a significantly large volume of images with distinct features of the objects of interest, is required by object detection algorithms. In the proposed approach, 100 images were acquired from all trees of a commercial walnut orchard, located in Rizomylos in Central Greece. These images were used as a training dataset for the object detector. A 20% of the aforementioned images were randomly chosen in each run, in order to be used as validation dataset for the training process of the object detector. The orchard size used for this case study was 4 ha and contained a total of 379 walnut trees. The images from the training dataset contained trees that were selected based on their location within the orchard, their infection percentage and the size of the canopy. This approach aims at training an object detector with the highest possible accuracy and evaluate its performance with unknown images. For that reason, the remaining 279 trees that were excluded from the object detection training phase, were used to examine the effectiveness and robustness of the detector in real orchard environments.

The images were acquired in raw format with resolution of 5,472×3,648 pixels, the same as it was used in the previous approach. The camera was positioned at 1.5m from the ground centered towards the side canopy center to capture as much of the tree's canopy as possible, demonstrated as a schematic in Figure 37.

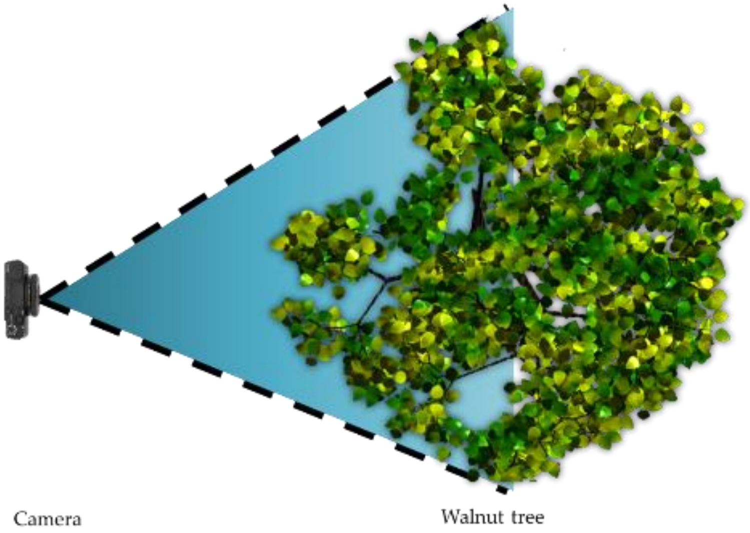


Figure 37. Camera setup for aiming the walnut tree's canopy from one side.

4.3.2 Object detection algorithms

The SSD object detection algorithm was the basis of the proposed methodology. The selection of the SSD for this methodology was based on two factors. Number one, the detection of anthracnose-infected leaves on walnut trees is the main aim, therefore, lower prediction accuracy is not an issue, since there is a large number of leaves on the tree's canopy, that will be detected nevertheless. If enough infected leaves are detected, the tree will be classified as infected in total, even without having all infected leaves located by the model. Number two, future goal of this application is the on-field deployment and its operation in real time, either by the acquisition and processing of images, or video feed. Consequently, adequate accuracy and fast performance is preferred over the highest accuracy but slow performance.

In literature, a comparison of the accuracy, by measuring the mean Average Precision (mAP), and speed by measuring the Frames Per Second (FPS), between the the most famous object detectors including Faster R-CNN, SSD and YOLO (You Only Look Once), on the VOC2007 [180] dataset, has been documented [183]. The results are presented in Table 7. R-CNN and Fast R-CNN are omitted from the table due to their very low processing speed (<1fps).

Table 7. Comparison of object detectors on the VOC2007 dataset.

Method	**mAP**	**FPS**	**Batch size**	**# boxes**	**Input resolution**
Faster R-CNN (VGG16)	73.2	7	1	~ 6.000	~ 1000 × 600
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8.732	300 × 300
SSD512	76.8	19	1	24.564	512 × 512
SSD300	74.3	59	8	8.732	300 × 300
SSD512	76.8	22	8	24.564	512 × 512

The mAP described in detail later within the document, FPS denotes the frames-per-second and measures inference speed, batch size is the number of training examples used in a single iteration, # boxes is the number of predicted boxes, and the input resolution is the pixel size of the images used as input.

According to Table 7, the fastest and most accurate detector is the SSD300. SSD [184] is a popular architecture for detecting objects in images, by using a single deep neural network. This technique discretizes the created bounding boxes into a set of default boxes, through different scales and aspect ratios as shown in Figure 38. During inference, the trained network provides predictions concerning the objects' class in each default box, and on the same time, it adjusts every box appropriately, to fit the shape of desired objects. Feature maps with different resolutions are used as a basis for the produced predictions, resulting in a more natural objects' treatment of diverse sizes. Furthermore, a single network conducts all computations, making SSD very easy to train and be integrated into full systems that entail detection components [185].

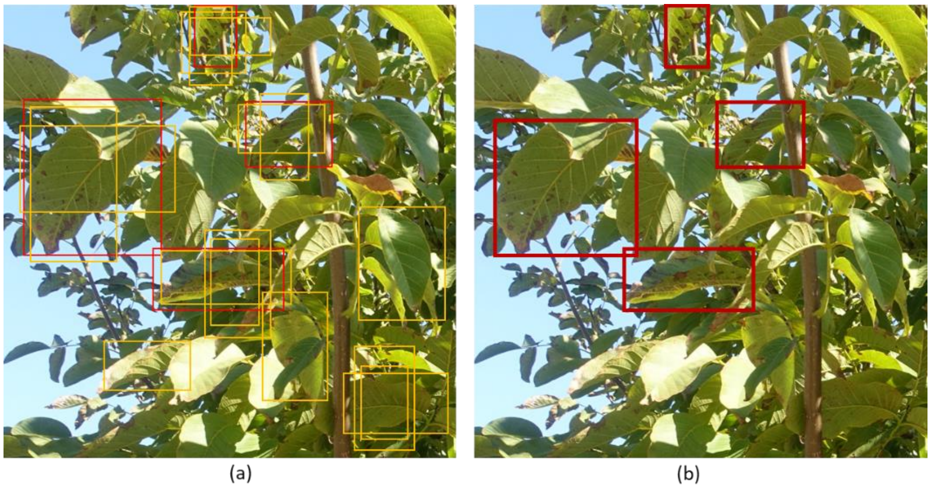


Figure 38. Predicted bounding boxes on an image acquired from the experimental orchard (a) and the final selection over the actual position (b).

4.3.3 Validation and evaluation of methodology

The proposed methodology's validation was based on the application of the trained object detection model, to all the orchard's trees that were not used for training the models. All four of the trees' canopies perimetrical images were segmented, according to the methodology mentioned previously. Namely, for the images sized 1,280×1,280 pixels there would be 12 sub-images × 4 sides = 48 sub-images, or 48 sub-images × 4 sides = 192 sub-images of 640×640 pixel resolution, to be examined with the trained object detector for anthracnose-infected leaves.

Expert knowledge was recorded and used as ground truth for the tree's classification. Expert agronomists, specialized in orchards production, inspected thoroughly throughout the entire orchard and classified each tree into two classes. The first class was constructed based on trees that were severely infected, and the other based on healthy or lightly infected trees. The second class was mixed (both healthy and lightly infected trees) because the non-infected trees were extremely limited as a result of the spreading of the inoculum throughout the orchard.

After extensive investigation, the number of 10 detected leaves per side was set as threshold for the classification between the two classes. The aforementioned threshold was defined based on the knowledge of the experts. In principle, images from the training dataset were given to the expert agronomists, for defining the threshold, i.e. the number of the infected leaves within each image, and re-classify the trees if it is deemed necessary. The threshold was the average number of leaves, above which, the trees were classified as severely infected. In cases where the classifications for the different sides vary, the majority of outcomes is considered for the final classification of the trees. In the rare occasions when equal sides are categorized in both classes, a secondary classification mechanism is based on the total number of classified leaves. The classification of the tree is then passed along with its ID, the coordinates, and additional metadata, to a data file which can then be used to create a point map of the infected trees, ultimately being able to visualize the way and speed the infection is spreading throughout the orchard.

Merging healthy and lightly infected trees in one class was an attempt to balance the classes, however, the severely infected trees were still significantly more. This class imbalance was unfortunate; however, model generalization was more important. The imbalance was tackled by selecting a proper metric for measuring the performance of the developed system.

4.3.4 Performance Metrics for Object Detection

4.3.4.1 Intersection over Union (IoU)

IoU focuses on the localization part of the prediction and is the measure of how much overlapping occurs between the areas of the ground truth bounding box and the predicted bounding box. When a prediction box matches 100% the ground truth box, the IoU is 1. The mathematical formula is given in Equation (1).

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} \qquad (1)$$

4.3.4.2 Average Precision (AP)

For object detection problems, average precision (AP) [186] is usually used as a primary metric because it can evaluate both the classification and the detection. The AP shows the accuracy with which the detector can define the area where the object is really, at a specific overlap percentage (i.e. 50%, 75%, 95%). It is defined as the mean precision at a set of eleven equally spaced recall levels [0, 0.1 ,…, 0.9 ,1], represented as r, and for the PASCAL VOC standard that was adopted in this methodology level, a prediction considered positive only if IoU

≥ 0.5. Any AP between (0.5:0.95) is considered a positive match and corresponds to the average over multiple IoU within the image. The mathematical equation for the AP is given in Equation (2).

$$AP = \frac{1}{11} \sum_{r=(0,0.1,\dots,1)} p_{interp}(r) \tag{2}$$

4.3.4.3 *Average Recall (AR)*

Average Recall (AR) is an significant metric for object detection problems, due to the fact that it summarizes proposal performance, i.e. recall, across IoU thresholds. The AR measures how many of the objects that were supposed to be detected, are indeed detected. In essence, AR correlates with detection performance [187]. The mathematical formula used for its computation is given in Equation (3).

$$AR = 2\int_{0.5}^{1} recall(IoU)dIoU \tag{3}$$

Generally, for object detector applications, mean average precision (mAP) and mean average recall (mAR) are preferred, due to the fact that the most popular benchmark datasets deal with multiple classes of objects and the mean value of precision and recall is necessary. This can also be seen in Table 7. However, since only one class is present (anthracnose infected leaves), mAP and mAR are considered as AP and AR respectively.

4.4 ORCHARD MAPPING IN COMPLEX ENVIRONMENTS WITH DEEP LEARNING SEMANTIC SEGMENTATION

Three sites of commercial walnut orchards, located in Thessaly, Central Greece, were used for testing the proposed methodology. The orchards covered a range of tree ages and soil surface features. On the same premise of data variability, the images were collected and selected for representing different seasons. Aim was the capture of different tree conditions and stages, that occur throughout the growing season, such as defoliated, canopy developing, canopy fully developed, and brown canopy before defoliation. Moreover, the orchards covered a range of background soil surface conditions, such as free from weeds, partly covered by weeds, and untreated soil with complete weeds coverage. A sum of 106 images from the aforementioned three orchards had led to defining seven different use cases, used for training and testing the proposed methodology. A detailed list of the characteristics' use cases is presented in Table 8.

Table 8. Characteristics and categorization of the orchards into separate use cases.

Use Case No.	Yearly Season	Weeds Coverage	Canopy Size	Foliage Colour	Ground Colour
1	Autumn	Low	-	Brown	Brown
2	Autumn	Low	-	Mixed	Brown
3	Summer	Low	Small	Green	Brown
4	Summer	Low	Medium	Green	Brown
5	Summer	Low	Medium	Green	Mixed
6	Summer	Low	Large	Green	Brown
7	Summer	High	Large	Green	Green

All use cases were adequately represented by several images in the training set and, more importantly, the test set was constructed so that it contains always, one image of each use case at minimum. This way, the trained models would be tested for all different combinations of characteristics, ensuring the maximum generalization. Sample images for each use case are presented in Appendix C.

4.4.1 Data Acquisition

Several test flights were conducted from 2018 to 2020 in order to acquire a large number of images, from multiple orchards, under different conditions. To maximize the exploration ability, the automated flights were maintained with the necessary criteria to produce high-accuracy orthomosaics. Each automated flight's parameters, such as UAV flight height, speed, number of captured images, side overlap and forward overlap ratio, were fine-tuned with aim to produce high-resolution orthomosaics (below-centimeter pixel size), and are presented in detail in Table 9.

Table 9. Details on the UAV flights that were conducted for each use case, for acquiring images and creating the orthomosaics used in the approach.

Use Case No.	Acquisition Date	Number of Trees	Number of Images	Overlap	GSD	Air Speed (m/s)	Cloud Coverage (%)

1	1/11/2018	1399	283	75%	1.3	<3	49
2	30/8/2020	569	522	75%	1.3	3	32
3	19/6/2020	358	330	75%	1.3	<3	5
4	3/06/2020	506	244	75%	1.5	<3	35
5	12/8/2020	2118	510	75%	1.5	<3	40
6	07/05/2019	296	193	75%	1.3	<3	12
7	15/05/2020	632	465	75%	1.3	<3	5

4.4.2 Data Preprocessing

Image preprocessing is a fundamental aspect of computer vision tasks, especially when employing self-learning algorithms. The reason for this is the need to transform the images into proper sizes/shapes, in order for the numerical computations to take place. Each of the raw images captured from the orchards occupied over 30 MB of storage each and had a 5472 × 3648 pixel rectangular shape. Size reduction and reshaping was applied to all images in order to transform them to dimensions of 512 × 512 pixels.

This approach also investigated the effect of image preprocessing in terms of colour and colourspaces. Histogram equalization (EQ) [188] and contrast-limited adaptive histogram equalization (CLAHE) [189] are two methods usually used for contrast enhancement in RGB images, both of which expand the contrast by adapting the range of the image's pixel values either globally or locally. Besides the RGB spectrum, the HSV colourspace—which represents colour with hue, saturation, and value, all assigned to cylindrical coordinates—was also investigated since it amplifies different features of an image, which could lead to increased performance.

A novel approach for contrast increase and feature extraction was attempted in this methodology. The approach was based on the combination of an RGB contrast-enhanced instance of an image, and its HSV colourspace instance, fused into a single 6-channel image. These fused images contain "double" information when compared to a regular 3-channel image; however, the increase in added value, due to more colour channels, is not directly implied [70]. A visual representation of how the 3- and 6-channel images are constructed is shown in Figure 39.

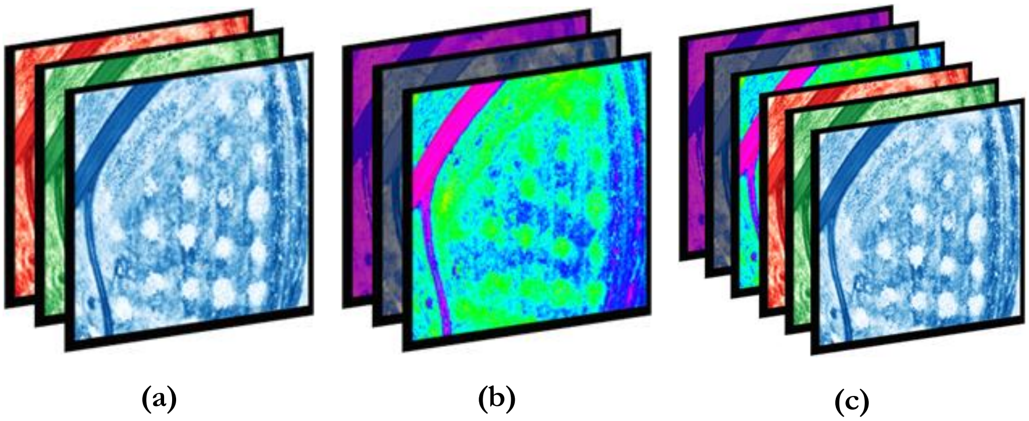


Figure 39. Channel deconstruction of (a) RGB, (b) HSV, and (c) fused images.

Two variants of the fused images were tested, namely the RGB image without any contrast enhancement and the CLAHE method for adaptive contrast enhancement, alongside the HSV colourspace image. The visual differences between all methods are presented in Figure 40.

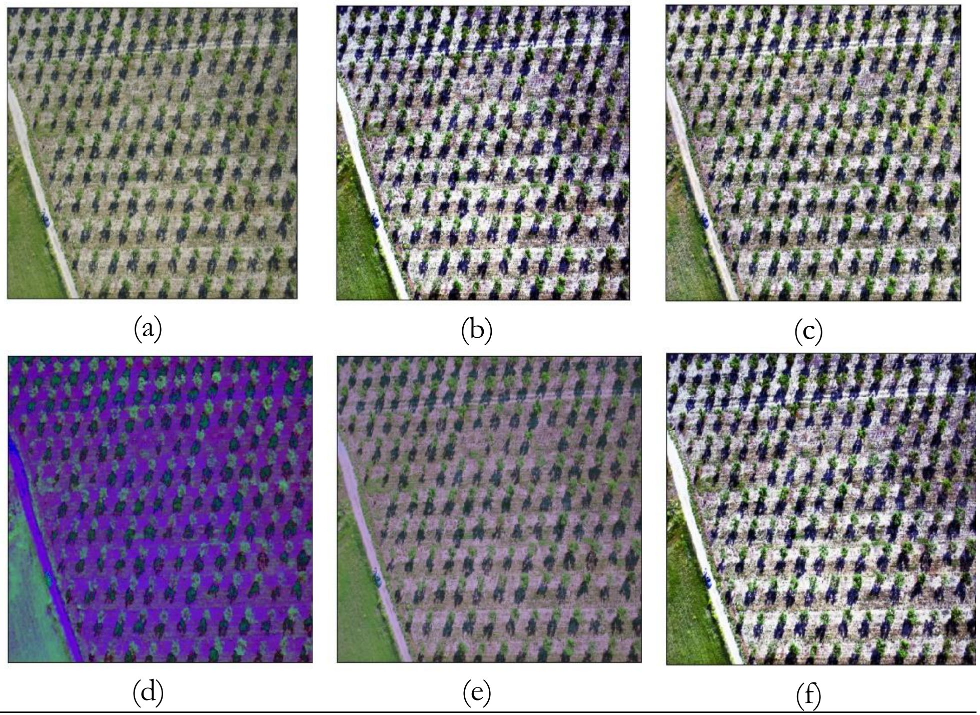


Figure 40. Image colour transformations used in for the proposed approach: (a) RGB, (b) EQ, (c) CLAHE, (d) HSV colourspace, (e) 6-channel RGB and HSV fusion, and (f) 6-channel CLAHE and HSV fusion.

4.4.3 Performance Metric

The Sørensen–Dice coefficient [190] was selected as the performance metric for the segmentation of trees against their background. It was preferred over the intersection over union (IoU, also known as the Jaccard index [191]) because the IoU penalizes bad classifications harder [192] and, in the case of tree foliage, the exact details of the foliage shape is not of high importance. As a loss function, the negative value of the dice coeffsicient was used, as is common in image segmentation tasks [193].

5 RESULTS

5.1 LEAF-BASED IMAGE CLASSIFICATION FOR DISEASE DETECTION WITH CONVOLUTIONAL NEURAL NETWORKS

Prior to the exploratory analysis of the optimal CNN setup, it is necessary to see where the other famous classification ML algorithms stand in the particular problem. In a previous work [32], it was shown that neural networks outperform other algorithms in a similar dataset, thus pointing to investigating the best neural network implementation. However, some of the most famous classical ML algorithms were tested on this new dataset for the comparison. The results are shown in Table 10.

Table 10. Comparison of classical (ML) algorithms.

	DT	**RF**	**Ada-Boost**	**SVM**	**ANN (Perceprton)**
Accuracy	64.59	79.55	77.45	81.37	83.38
Precision	65.02	80.17	77.82	81.66	84.28
Recall	64.78	79.83	77.78	81.21	84.11
F1 score	64.32	79.39	77.42	80.74	83.22

It is clear that ANNs perform better in a particular problem, and therefore the choice in focusing on CNNs, a direct derivative of ANNs, is justified and validated.

The first comparative analysis was dedicated to measure the accuracy as well as the loss of both the validation and the testing dataset for images containing background information. Measuring accuracies allows us to see if the predictions made to the unknown set match the performance of the model while training. In Table 11 the accuracy percentage and the loss for each method used are listed.

Table 11. Accuracy and loss for the three preprocessing approaches, for the validation and testing sets on images with background information.

	Accuracy		**Loss**	
	Validation	*Testing*	*Validation*	*Testing*
Grayscale	92.869	92.469	0.192	0.197
Fast Fourier	93.153	92.938	0.198	0.176
RGB	96.847	95.969	0.086	0.111

The second comparative analysis was to measure the accuracy and loss for each method, but with additionally applying the background removal method. This way, the images contain less relevant-to-the-infection information, which should lead to increased performance. Table 12 show that by removing unnecessary information, the performance of each model has increased.

Table 12. Accuracy and loss for the three preprocessing approaches, for the validation and testing sets on images without background information.

	Accuracy		**Loss**	
	Validation	*Testing*	*Validation*	*Testing*
Grayscale	95.710	95.469	0.116	0.130
Fast Fourier	96.591	96.531	0.108	0.105
RGB	99.006	98.719	0.031	0.049

Finally, the best performing model is selected (RGB images with background removal) and compared it to state-of-the-art CNNs for image classification. The afforementioned CNNs have set accuracy records when originally trained on the Imagenet [33] database, and thus, are selected for comparison. In detail, DenseNet121 [34] a version of DenseNet (Densely Connected Convolutional Networks) that is 121 layers deep, was used. DenseNet solved the vanishing gradients problem, despite its depth, by having in its architecture both 1×1 and

3×3 convolutional layers, as well as batch normalization. Another architecture that was used is VGG16 [35], a version of VGG (Visual Geometry Group). As a general rule, VGGs only use 3×3 convolutional layers, stacked on top of each other. ResNet50 [36] is the 50-layer deep version of ResNet (Residual neural Network), an exotic architecture that is based on "network-in-network" micro-architectures. Due to global average pooling, instead of fully connected layers, the size of ResNet50 is significantly smaller than VGG16. Finally, Inception V3 [37] is 48-layers deep and is Inception's third instalment. Inception V3 incorporated RMSProp (Root Mean Square Propagation) optimizer and 7×7 convolution in the 1×1, 3×3 and 5×5 convolutions which were already presented within the same module of the network.

For the sake of comparison, the networks are used along with their weights, and only the last layers are retrained on the specific dataset, also known as transfer learning. Thus, each algorithm is re-trained/fine-tuned on the desired dataset, with the same preprocessing functions, in order to compare only the models' performance. Additionally, the training times of each algorithm have been measured with the %%time function of Python, set to measure the execution time of only the training of the model and none of the preprocessing or the processing of the predictions. The results are shown in **Table 13**.

Table 13. Comparison of most common CNNs with the proposed CNN.

	VGG16	**DenseNet121**	**ResNet50**	**Inception V3**	**Proposed CNN**
Validation accuracy	96.88	99.049	98.505	99.290	99.006
Validation loss	0.10	0.033	0.050	0.027	0.031
Testing accuracy	95.78	96.577	98.363	99.375	98.719
Testing loss	0.12	0.071	0.067	0.013	0.049
Execution time (s)	1.39×10^3	1.87×10^3	1.29×10^3	1.71×10^3	0.99×10^3

It is noticed that the proposed CNN architecture has performed better than the rest of the state-of-the-art architectures, except Inception V3, which achieved better accuracy both in the validation dataset, as well as in the testing dataset. However, a key point difference is that the proposed CNN architecture is significantly shallower compared to the other architectures, therefore it can be trained a lot faster on the same dataset.

It is clear that the proposed network achieves a similar accuracy to the state-of-the-art Inception V3 network, in less time due to its shallower architecture. The tradeoff is in favor of the proposed network, since the difference in time is significant, while the accuracy difference is minimum. Given that in this particular problem the useful features are not so complex, the proposed CNN performs satisfactorily.

5.2 Object detection with single-shot detector algorithm for tree-level disease classification in orchards

5.2.1 Object detector training performance

Two different training datasets were derived from the initial image dataset, depending on the segmentation dimensions (640x640 or 128x1280 pixels), and were used for training the object detectors. These datasets do not derive directly from the results of the segmentation of the original images. A number of sub-images containing unnecessary information, irrelevant to the target (leaves), such as soil, sky and other unnecessary information, were discarded. Detailed information on the number of images used in every step of the analysis is shown in Table 14.

Table 14. The number of images that were used for training and validation of the object detector and for the final classification of the trained model**.**

Description	*Total*	*SSD Training*	*SSD Validation*	*Classification Validation*
Original images	379×4=1,516	80×4 = 320	20×4=80	279×4=1,116
Total sub-images (1,280x1,280) (x12)	18,192	3,840	960	13,392
Total sub-images (640x640) (x48)	72,768	15,360	3,840	53,568
Total sub-images after removing soil/sky (1,280x1,280)	5,597	1,187	277	4,133

Total sub-images after removing soil/sky (640x640)	11,094	2,361	492	8,241

Three different CNN classifiers were evaluated and compared based on their overall performance; Resnet50, Inception v2, and Mobilenet v2. Direct comparison between these methods was conducted on the same dataset, at two different image sizes. Examining the performance metrics of the classifiers the application of Resnet50 CNN in the SSD architecture produced the best results in terms of accuracy and speed since it reached higher AP within less training steps. The configuration parameters alongside with the performance metrics for each approach are presented in Table 15.

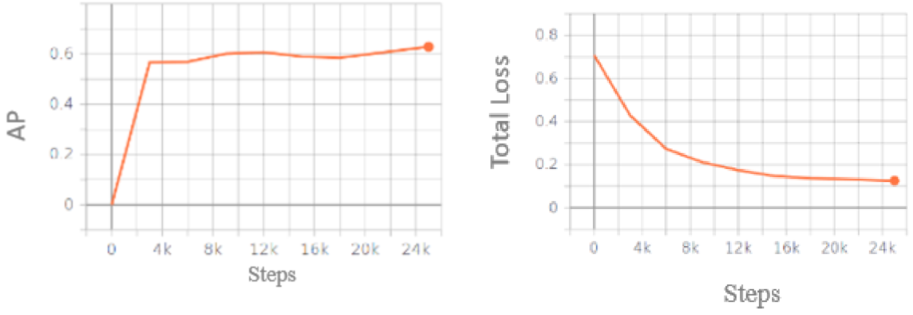
Table 15. Parameters and performance metrics for the models evaluated using the training dataset of the methodology.

	Model					
Training parameters	**Resnet50**		**Inception v2**		**Mobilenet v2**	
# train images	2.361	1.187	2.361	1.187	2.361	1.187
# test images	492	277	492	277	492	277
width	640	1,280	640	1,280	640	1,280
height	640	1,280	640	1,280	640	1,280
Total steps	20,000	15,000	50,000	50,000	20,000	20,000
Vertical flip	yes	yes	yes	yes	yes	yes
Horizontal flip	yes	yes	yes	yes	yes	yes
Batch size	32	32	12	12	12	12
Learning rate	1.18^{-9}	1.6^{-2}	3.0^{-3}	3.76^{-5}	4.17^{-3}	3.0^{-3}
Training time (hours)	15	14	4	7	5	8
Performance *metrics*						

AP (IOU = 0.50:0.95)	**0.629**	0.35	0.57	0.391	0.475	0.285
AP (IOU = 0.50)	**0.926**	0.601	0.931	0.574	0.824	0.428
AP (IOU = 0.75)	**0.706**	0.37	0.618	0.456	0.505	0.336
AR MaxDets = 1	0.621	0.296	0.567	0.252	0.496	0.198
AR MaxDets = 10	0.71	0.515	0.65	0.545	0.594	0.417
AR MaxDets = 100	0.73	0.576	0.663	0.601	0.622	0.472
Classification loss	0.32	0.568	3.65	10.768	0.494	1.193
Localization loss	0.109	0.182	0.5	0.632	0.17	0.262
Regularization loss	0.125	0.151	0.59	0.5407	0.344	0.364
Total loss	0.556	0.901	4.74	11.941	1.099	1.819

The experimentation with the CNN classifiers was performed at different image sizes to investigate the appropriate level of segmentation of the initial images in order to achieve the highest possible accuracy with the lowest computational time. Direct result of this was to obtain different number of images used for the training phase; 2.361 sub-images sized 640×640 pixels and 1.187 sub-images sized 1,280×1,280. Regardless the segmentation size, the annotated boxes were the same in both datasets.

Based on the results that are presented in Table 15, Resnet50 and Inception v2 performed better than Mobilenet v2 for both sub-image sizes in terms of prediction performance. This was expected since Mobilenet v2 aims at producing fast predictions during deployment. Inception v2 outperformed Mobilenet v2, however, by requiring significantly more iterations for reaching the highest AP through all examples (steps), meaning that, in general, it was struggling to learn the desired features underlying in the images. Resnet50 achieved the best accuracy, reaching approximately 63% AP, in spite of being the slowest one, due to its extensive depth of convolutional layers. All three CNNs showed similar behavior considering the size of the input images, performing better with the images of 640×640 pixels. The selected model for the testing in real conditions was the 640×640 pixel size Resnet50. The AP and total loss regarding the training steps of the selected model is shown in Figure 41.



(a) (b)

Figure 41. Average precision (AP) (a) and total loss (b) for the Resnet50 classifier training, on sub-images with 640×640 pixels size.

According to the analysis, the selected model starts reaching a minimum plateau in the total gain loss after 16,000 runs, and in the AP value after 9,000 runs.

5.2.2 Expert knowledge classification

Expert agronomists categorized the trees in the two classes (severely infected or healthy/lightly infected), after a detailed manual inspection of the orchard's images. A total of 243 out of the 379 trees were classified as severely infected, while the remaining 136 were classified as lightly infected or healthy. The number of tree images for each class, that was selected for training, was close to equal, in order to be able to capture a good amount of both severe and light infection of anthracnose on leaves, for a balanced training and proper generalization of the model. Thus, the validation dataset ended up having 194 severely infected and 85 lightly infected / healthy trees. The class distributions for the total number of trees and the classification-validation dataset are shown in Figure 42.

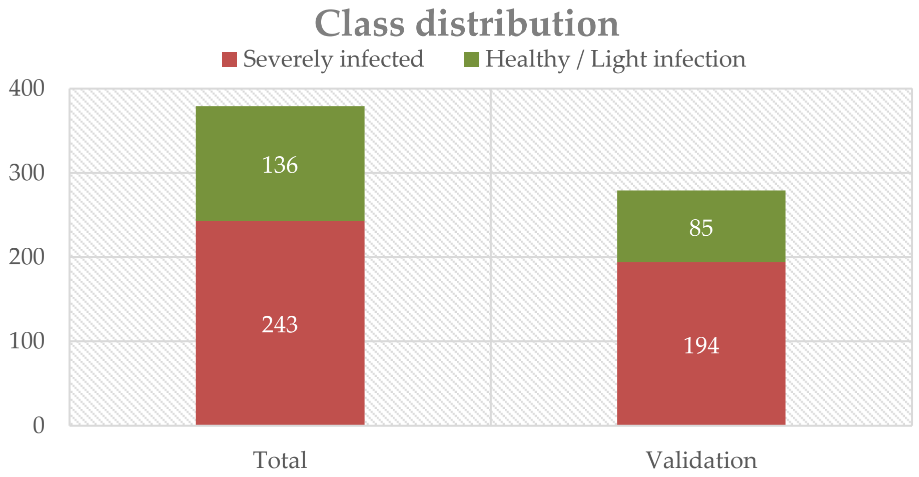


Figure 42. Class distribution for the total number of tree images, as labelled in the orchard, and for the tree images used for classification-validation datasets.

5.2.3 Object detector classification and validation

Once the best trained model was selected, it was applied on the remaining 279 trees of the orchard from which the corresponding images were not used during the training phase. A predicted class was appointed and cross-referenced for each tree, with a ground truth class, given by the on-field experts. The maps deriving from these classifications are shown in Figure 43. Figure 43(a) shows the classification map as it was formed by the expert's labelling, and Figure 43(b) shows the classification map as it was formed by the trained model's predictions. The two maps are very similar depicting the spatial distribution of healthy and infected trees throughout the orchard. These two maps, when paced side-by-side, can visually demonstrate the proposed method's efficacy, as well as the feasibility and applicability of an automated, high-accuracy, anthracnose-detection system. The aim of developing such a system is to be used as an assistive tool in precision agriculture. After further geostatistical analysis, this georeferenced information can lead to the production of variable rate fungicide application map as part of a decision-making system.

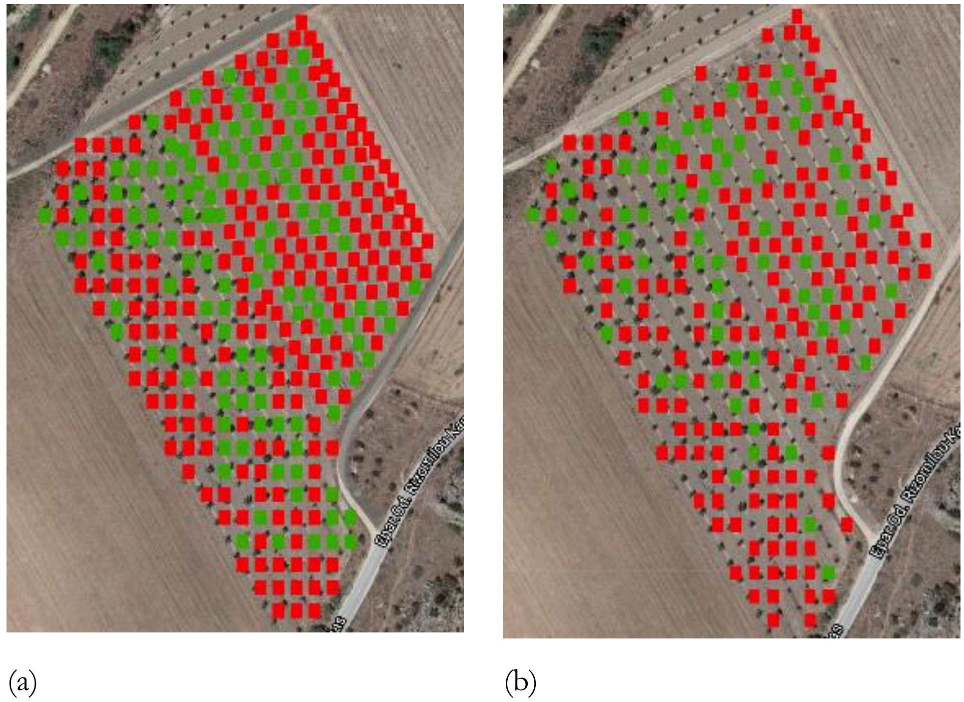


Figure 43. Experts' knowledge tree classification map (a) and the predicted classification of the trees according to the object detector results (b) of the studied orchard. The red squares indicate the infected and the green the healthy classified trees. Map (b) contains fewer points because the trees used for model training were not included in the classification analysis.

The results were inserted in a confusion matrix in order to observe the ratio of correctly predicted classes (Table 16). The confusion matrix shows the sums of properly and improperly predicted classes over the real classes, in a tabular form. Confusion matrices are equally useful to the performance metrics, because they offer visual interpretation of the results and a detailed distribution of the misclassified cases.

Table 16. Performance of the object detector during the validation phase.

		Expert classification		
		Severe infection	*Healthy / Light infection*	***Total***
Predictor classification	*Severe infection*	166	28	**194**
	Healthy / Light infection	22	63	**85**
	Total	**188**	**91**	**279**

The accuracy, F1 score, precision, and recall, are the most common performance metrics for classification purposes. Their descriptions and mathematical formulas are described in [194]. The different performance metrics show the success of the tree classification under various scopes. In detail, accuracy considers how the number of predictions that was correct regardless of the class; the harmonic mean, or F1 score, calculates the classification accuracy by considering class imbalance that might be present, which fits the proposed approach. Precision shows how many of the positive predictions were actually correct considering all the predicted positives (correct and incorrect) and recall shows how many of the positive predictions were correct considering all true positives (regardless of whether they were properly classified). This approach aims to tackle the detection of anthracnose infected leaves, it is valuable to take into consideration both precision and recall, in order to be able to bring forth any weaknesses the train model has in predicting false negatives (predict healthy when the tree is infected).

The classification of the test trees reached 82.1% accuracy. However, due to the classes' imbalance, the F1 score, as the harmonic mean of precision and recall, is more unbiased and therefore a more appropriate metric [195]. The formulation of the F1 score is for finding equal balance between precision and recall for a class, which makes it significantly useful, especially when classes within datasets are imbalanced. The trained model application on unknown, test data (trees), achieved 86.9% with regards to the F1 score, and on the same time, precision and recall achieved 88.3% and 85.6% respectively (Table 17).

Table 17. Performance metrics for classification of the trees' classes.

Performance metric	Value
Accuracy	0.821
F1 score	0.869
Precision	0.883
Recall	0.856

Type 1 classification errors, referring to the false negative classification, were 27.3% more than type 2 classification errors, referring to the false positive classification. This signifies that the trees that have been misclassified as severely infected, and consequently would be unnecessarily treated, are more than the severely infected misclassified as healthy trees, which would not be treated at all and spread the virus.

The properly classified, along with the misclassified predicted trees, were mapped. The aim was to visualize the distribution, density, and spatial characteristics of the misclassified trees in order to understand if there were any location-related variables that affected the performance of the trained object detector (Figure 44). The colour coding of this mapping consists of correctly classified trees (blue), false negatives i.e. anthracnose infected trees that were labelled as healthy (yellow), and false positives i.e. healthy trees that were incorrectly labelled as infected (pink). According to the resulted map, the vast majority of the predictions were correct in accordance with the expert classification. The false predictions were fairly balanced between the two classes (false positives and false negatives). In real in-field applications, the false positive predictions are not as important, compared to the false negatives. Unnecessary treatment with fungicide application would not be an issue for these trees since it would not affect their overall health and the general yield and production of the orchard.

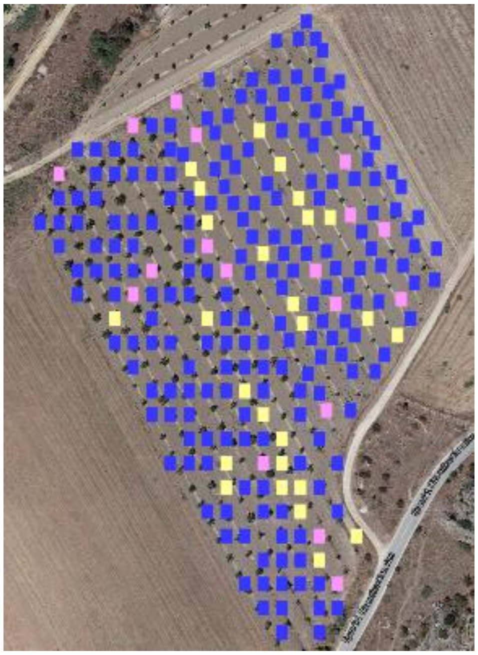


Figure 44. Point map of the properly classified (blue), false negative classified (yellow) and false positive classified (pink) trees, based on the trained object detector as compared to the ground truth classification (expert knowledge).

5.3 Orchard mapping in complex environments with deep learning semantic segmentation

5.3.1 Validation on Dataset

All models were trained between 40 and 100 epochs, a visualization of which is seen in Figure 45. Early stopping was used for preventing overfitting of the models. The models were trained and tested on 96 and 10 images respectively, which were randomly selected from the 106 images of the dataset, including all seven use cases (use cases presented in Table 8 and Table 9). In this way, the generalization of the model was ensured. The accuracy achieved by the models under the differently pre-processed datasets is shown in Table 18.

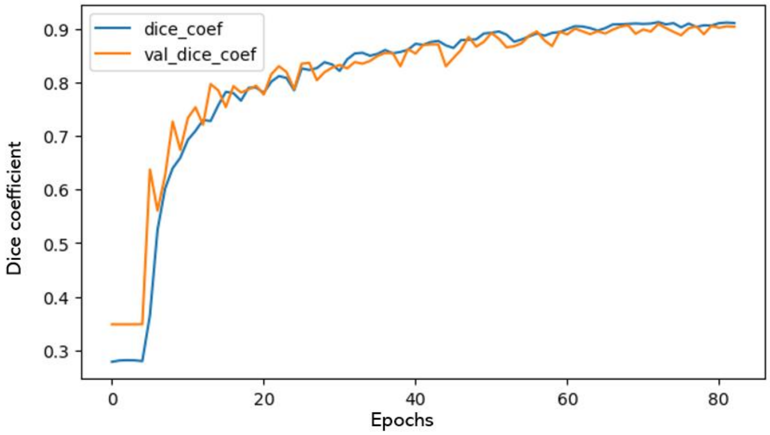


Figure 45. Learning plot with training and validation accuracy.

Table 18. Accuracy (dice coefficient) for investigated methods of pre-processing.

Image Colourspace	*RGB*	*EQ*	*CLAHE*	*HSV*	*RGB + HSV*	*CLAHE + HSV*
Channels	**3**				**6**	
Training accuracy	0.91	0.90	0.90	0.92	0.91	0.91
Validation accuracy	0.90	0.88	0.89	0.90	0.89	0.90

Testing accuracy	0.87	0.77	0.86	0.86	0.85	0.86

As mentioned previously, the dice coefficient is used for benchmarking the performance of trained models. However, the ability of a trained model to properly segment trees is measured by visual inspection. The system was validated by applying the trained models to never-before-seen images of entirely different use cases and comparing the results to the identification of a human expert. The false positives (FPs), i.e., incorrectly identifying trees at locations where there were none, and false negatives (FNs), i.e., failing to identify trees, could thus be registered. On top of the tree canopy segmentation, the exact location of a tree's trunk was computed based on the predicted masks. The method for computing this location was based on the centroids of the image moments, i.e., the weighted average of the predicted masks. Therefore, for each mask representing a tree canopy, and with the condition that it was isolated and in no way connected to an adjacent mask, a single point was calculated to signify the position of the tree trunk, considering a fairly symmetrical canopy shape. A visual example of the predicted segmentation (left) and the real annotation (right), both overlaid on the original images, is given in **Figure 46**.

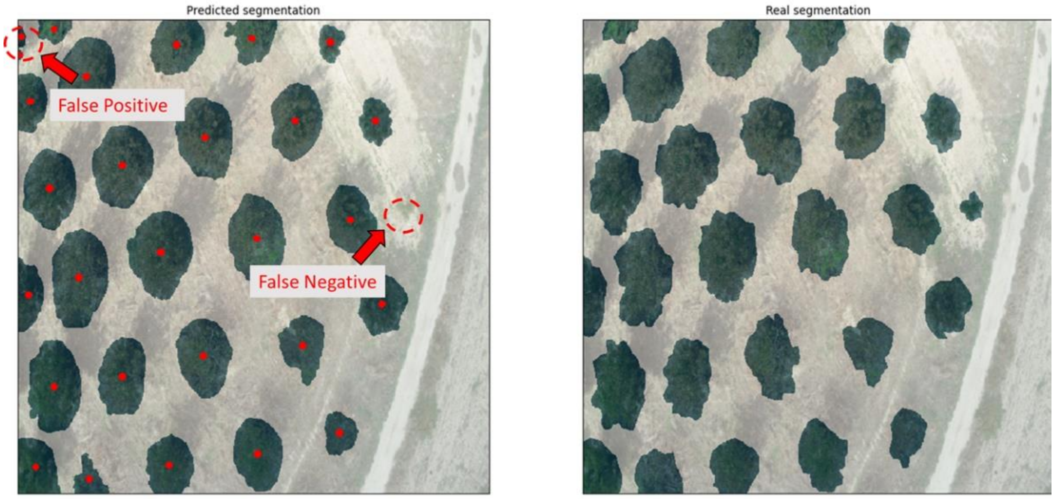


Figure 46. Examples of false positive and false negative segmentation predicted by the developed system (left) as compared to the real segmentation (right).

Since the primary aim of this approach is to solve the problem of the accurate mapping of trees' locations within orchards, the absolute intersection between all pixels was mainly considered for the training phase. The rough shape and size of a properly identified tree canopy was what would lead to a correct computation of the trunk location and the estimation of the tree's age. Therefore, in order to choose the best-trained model for the application, the test set was manually investigated across the predicted segmentations from each approach. Based on this premise, FPs and FNs were identified and each model ultimately received a score based on the ratio of FPs, FNs, and their sum, over the total amount of trees in each image, as seen in Table 19.

Table 19. Overall performance evaluation, expressed as percentages (%), of the models examined in the test set of the methodology, in terms of false positives (FPs), false negatives (FNs), and their sum ratios over the total number of trees in the test set.

Image Colourspace	***RGB***	***EQ***	***CLAHE***	***HSV***	***RGB + HSV***	***CLAHE + HSV***
FPs (%)	7.49	9.41	16.17	7.57	7.49	4.99
FNs (%)	5.81	8.73	15.17	6.48	10.66	16.22
Total misidentifications (%)	13.30	18.14	31.34	14.05	18.16	21.21

From the overall evaluation of the models' performance, the RGB model was identified as the simplest and provided the best results. Therefore, it was selected as the primary model to be investigated further. In the next step, the performance of the RGB model was investigated for each use case separately. In this way, the strengths and weaknesses of the selected approach could be identified and therefore tackled in future work. The results of the RGB method were further broken down per test image, covering all use cases that were included in this level of the methodology, as shown in Table 20.

Table 20. Performance evaluation of the RGB model (best performing) applied to the separate test images for each use case, expressed as percentages (%) of false positives, false negatives, and their sum total.

Test Image	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***Mean***
Use Case	**2**	**1**	**5**	**4**	**4**	**6**	**6**	**5**	**7**	**3**	
FPs (%)	7.69	8.33	16.67	9.09	2.08	1.82	2.33	12.64	14.29	0.00	7.49
FNs (%)	0.00	4.17	4.17	18.18	0.00	1.82	0.00	3.45	2.38	23.94	5.81
Total (%)	7.69	12.50	20.83	27.27	2.08	3.64	2.33	16.09	16.67	23.94	13.30

The accuracy achieved for all use cases using the RGB model ranged between 72.7% and 97.9%, which can be considered as a satisfactory result. Comparing images 6 and 7 with 9, the effect of the presence of weeds' on the accuracy of the model is evident, since the first two images, which belong to use case 6 (large trees; few weeds), performed considerably better compared to image 9, which belongs to use case 7 (large trees; many weeds). In the latter, the FPs were the primary reason for limiting the system's performance. This signifies that the developed weeds within the image frame led to increased FP misclassifications (weeds classified as trees). Interestingly, when running test images from use cases 1 and 2 (i.e., images captured during autumn when the canopy was turning brown), accuracy was notably high, albeit with a low level of weeds coverage.

With regard to common characteristics between use cases, three indicative results from the RGB model are presented in Figure 47. These three categories cover the most contrasting situations; (a) ideal conditions with medium/large tree canopies and ground with only a small amount of weeds, (b) intermediate conditions with large tree canopies but weed-infested ground, and (c) unfavorable conditions with small tree canopies and some weeds present. The first image belongs to use case 4, containing clear green canopies and ground covered by only a few weeds. The second image, which represents use case 7, shows large green canopies; however, the ground is almost entirely covered with weeds of a similar shade of green. The third image is from an orchard free of weeds (use case 3); however, the canopies are particularly small in size due to the young age of the trees. Use cases 4 and 6 are the most ideal, considering canopy and background colour contrast due to the season and the lack of weeds. A noteworthy outcome is that even though use cases 4 and 5 both had medium-sized canopies, the trained model's accuracy was completely different due to the presence of weeds. Additionally, use cases 1 and 2 demonstrated similar behavior as use cases 3 and 4, since all of them were almost free of weeds, with the only difference being the more brownish colour, making it slightly harder to identify all canopies. In all images, a mask overlay of 50% transparency was applied in order to visualize the segmentations; therefore, the real shades of the images were altered.

Conditions	**Test Images**

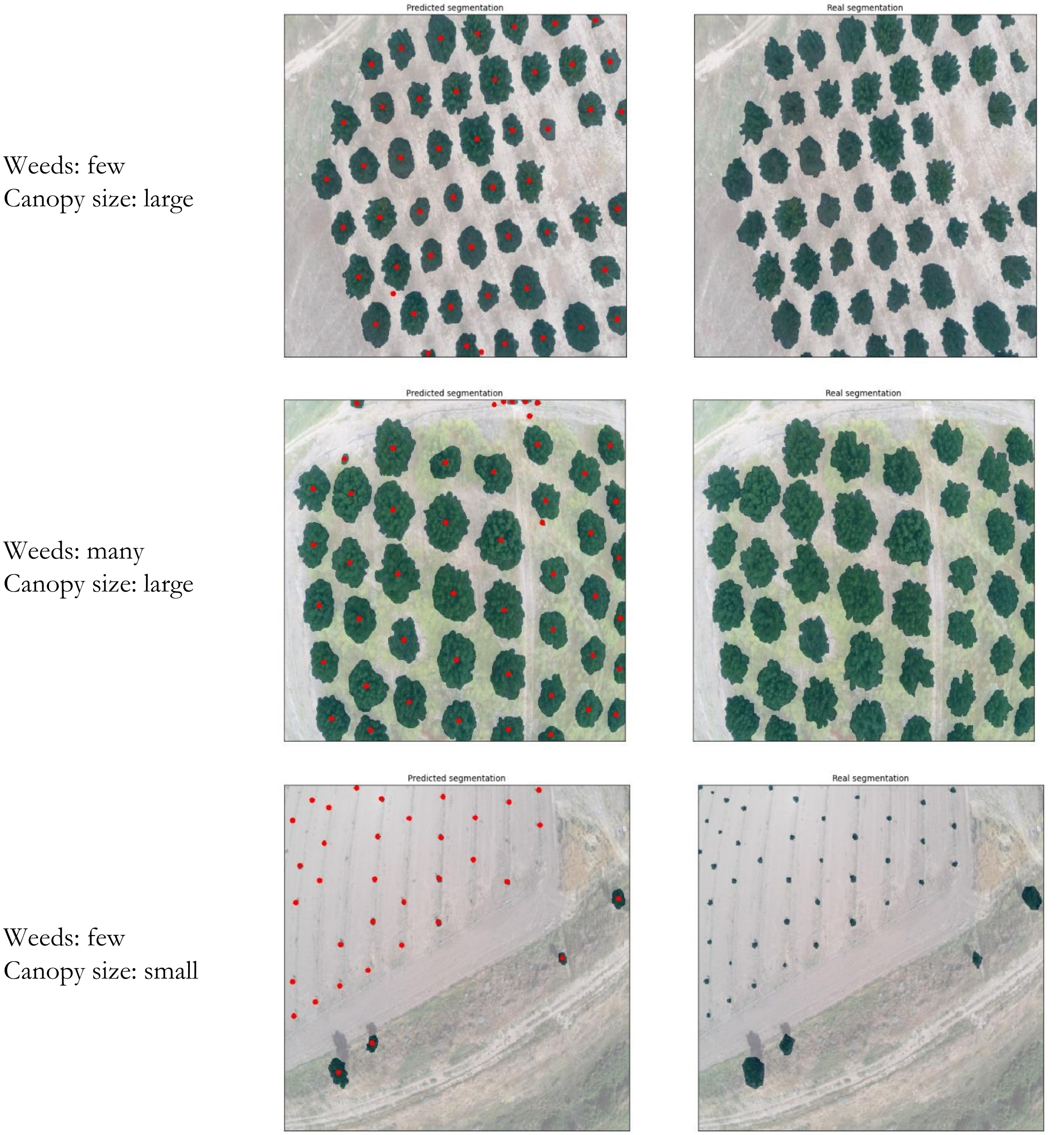


Figure 47. Results of indicative RGB images covering a range of different conditions.

5.3.2 Validation on Orthomosaics

The system as presented above showed its ability to recognize tree canopies with high accuracy when applied to high resolution images of certain dimensions. However, investigating the performance of the system with orthomosaics covering the entirety or a large part of the orchard area was also considered to be of great interest. Therefore, in a further analysis, the trained models were applied to orthomosaics captured from orchards with pixel resolution considerably lower than the original training dataset. The aim of this test was to examine the extent of the trained models' capabilities considering the pixel resolution range of all canopies. Applying the models directly to the orthomosaics produced errors due to the presence of "transparent" pixels that denote areas outside the bounds of the appointed orchard. Two methods were used to overcome this inconvenience: "oversampling", i.e., filling the transparent pixels with the dominant ground colour; or "undersampling", i.e., cropping the largest area possible that did not contain "out-of-borders" areas.

The test included (a) analysis of orthomosaics treated as a whole (i.e., as one image) and (b) analysis of sub-images clipped from the orthomosaic. It is important to note that these were never-before-seen images that had not been a part of the original dataset. Similarly to the training phase, orthomosaics of three different use cases were selected.

Case A. The first case displayed an orchard with large- to medium-sized canopies. As mentioned above, the pixel resolution was smaller than that of the training dataset. The accuracy reached 99%, with only a small FP segmentation on the right section of the middle of the image detected, visible in Figure 48.

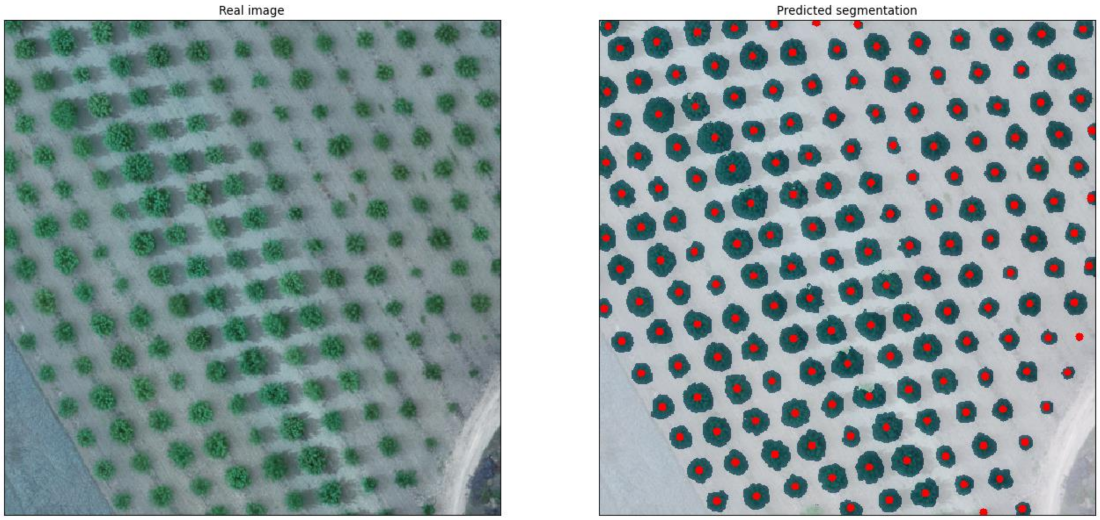


Figure 48. Undersampled orthomosaic of an orchard with large- to medium-sized canopies (left) and the segmentation predicted by the model (right).

Case B. The second use case was an undersampled orthomosaic of an orchard with young trees, shown in Figure 49. It was observed that even though the canopies were significantly small, the trained model was able to achieve a high accuracy of 90.5% with only 5.3% FNs and 4.3% FPs.

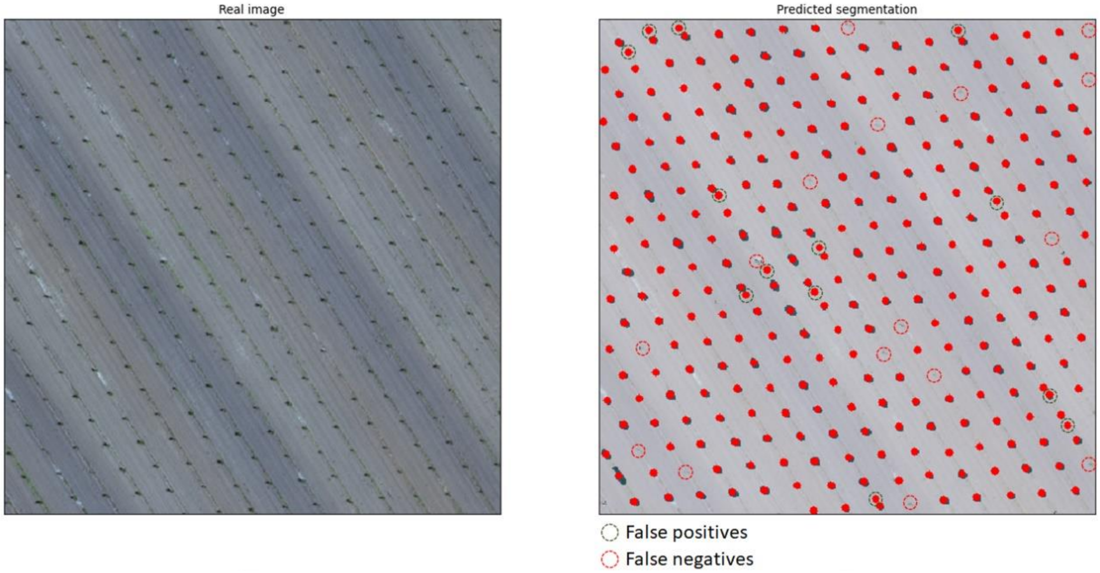


Figure 49. Undersampled orthomosaic of an orchard with young trees featuring small-sized canopies (left) and the segmentation predicted by the model (right).

Case C. Finally, an orthomosaic with a higher resolution compared to the previous case of an orchard with small-sized canopies was undersampled and tested. However, the presence of developed weeds dispersed throughout the orchard produced many FPs in the segmentation, as seen in Figure 50.

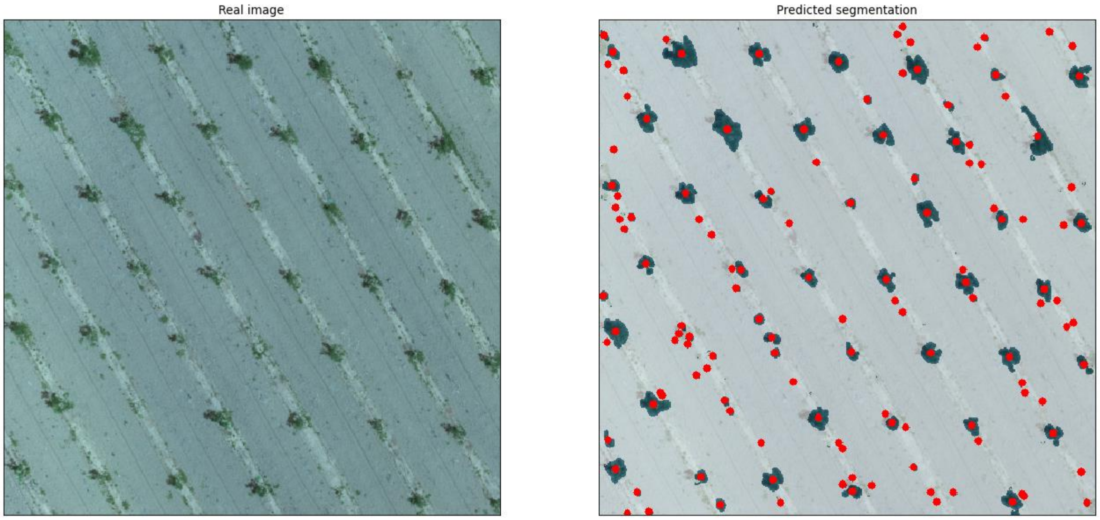


Figure 50. Undersampled orthomosaic of an orchard with small canopies, not treated for weeds (left), and the segmentation predicted by the model (right).

Even though a rule-based condition could eliminate such small segmentations, this could be counterproductive for cases with young-aged trees with small canopies. However, the original orthomosaic, as seen in Figure 51, produced significantly fewer FPs compared to the undersampled one above.

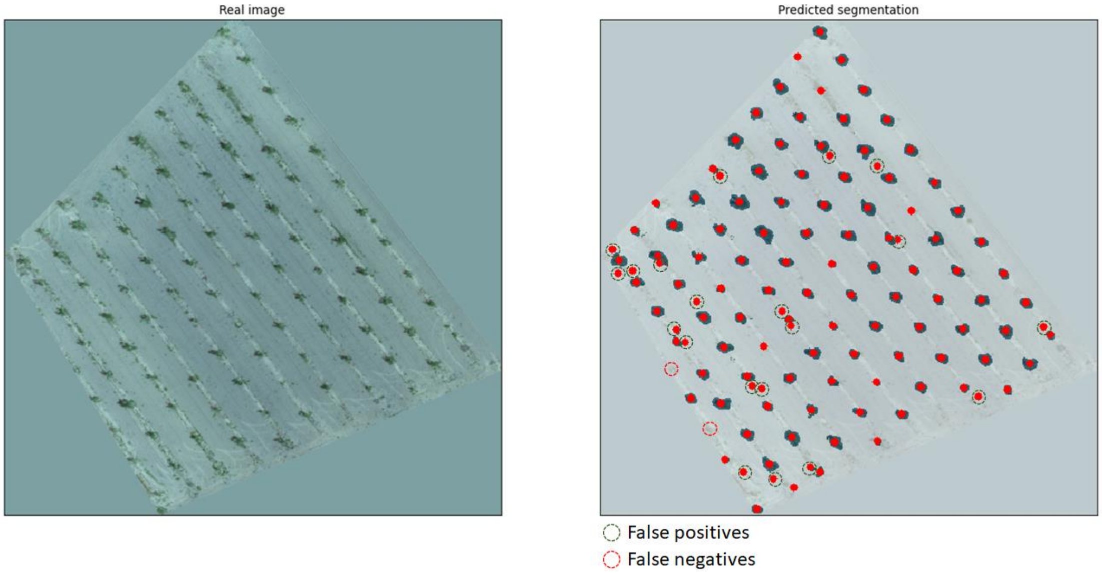


Figure 51. Complete orthomosaic of one of the orchards used for the tree segmentation task, with trees with small-sized canopies, not treated for weeds (left), and the segmentation predicted by the model (right).

The accuracy achieved for the orthomosaic was notably high, reaching 82%, and the segmentation prediction showed 16.4% FPs and only 1.6% FNs. It is worth mentioning that all the FPs were recognized as trees due to the presence of large surfaces covered by weeds, simulating the size and the shape of the top view of the tree canopy. This indicates that the model can be expected to demonstrate excellent performance with weed-free orchards. Furthermore, the FNs were located at the edges of the orthomosaic where part of the canopy of the respective trees was missing.

5.3.3 Comparison with Baselines and Other Methods

A comparison of the proposed approach with other traditional computer vision techniques, unsupervised machine learning methods, object detection approaches, and other image segmentation deep learning techniques is presented in this section. For all methods, baseline versions were used with minor tuning of parameters. For the traditional computer vision techniques, blob, feature, and colour detection were implemented with the assistance of OpenCV Python library [196]. Specifically, for the feature detection, oriented FAST and rotated BRIEF (ORB) was used as a baseline. With regard to the unsupervised machine learning approach, a K-means algorithm [197] was implemented from Python's SciKit-Learn library [198]. For the object detection approach, the single shot detection (SSD) algorithm [184] with a ResNet50 [199] backbone was used, and for the segmentation approach, the Mask R-CNN algorithm with a ResNet101 [199] backbone, both implemented with the Keras library [200] with the Tensorflow backend [201]. Since all methods have different ways to extract information from images, the characterization of FPs and FNs was conducted by a domain expert agronomist. The total percentage of both FP and FN instances was used as a metric of comparison, and all methods were tested on the same test images from segmentation task. The supervised learning algorithms were trained with the default parameters and with early stopping on the same training dataset. The results for all methods are presented in Table 21.

Table 21. Comparison of the proposed approach (in bold) with other computer vision baselines and machine learning methods using total percentage of misidentifications as a metric (sum of false positives and false negatives).

Test Image	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Mean**
Use Case	**2**	**1**	**5**	**4**	**4**	**6**	**6**	**5**	**7**	**3**	
Blob detection	63.65	56.81	34.35	34.57	31.73	28.00	25.75	28.54	65.39	39.57	40.84

Feature detection (ORB)	65.68	59.56	49.72	46.85	48.38	50.24	47.90	48.81	63.40	43.74	52.43
Colour detection	53.88	52.96	35.32	32.27	31.12	29.62	29.03	27.51	55.88	27.45	37.50
Clustering (K-means)	52.25	54.17	40.19	39.12	38.69	36.47	36.16	36.20	53.55	42.97	42.98
Object detection (SSD)	12.34	15.28	21.68	29.16	5.92	7.03	7.05	19.39	21.01	27.10	16.60
Mask R-CNN	8.31	13.01	19.80	27.21	3.45	3.98	2.80	16.59	17.98	23.00	13.61
Proposed U-net	**7.69**	**12.50**	**20.83**	**27.27**	**2.08**	**3.64**	**2.33**	**16.09**	**16.67**	**23.94**	**13.30**

Blob detection performed poorly on use cases 1 and 2 due to the canopies being brown or leafless, on 4 and 5 due to the canopies' shadows, and on 7 due to the matching green colour on the weed-rich ground. On use case 3, no significant drawbacks were noted. Feature detection resulted in too many FP identifications in all cases because of the leaf-like appearances of most objects present in the aerial orchard photos. Colour detection achieved better performance on use cases 3–6 compared to the previous two methods, but with manual tweaking of the colour values for each image separately; however, when foliage and ground colour bore a resemblance, there were almost no identifications. When K-means was tuned to create two clusters, for trees and backgrounds, it took into account all pixels that belonged to weeds or similar fauna. The algorithm trained with SSD was able to find most trees; however, the locations of the tree trunks, which were computed as the center of the bounding box, had noticeable deviations from the ground truth. Finally, Mask R-CNN is a two-stage approach but, even though it performed similarly to the proposed U-net approach, the generated model was five to ten times larger (the size of the proposed U-net-based model was ~22 MB), thus rendering the lightweight implementation prerequisite as null. All methods offer benefits and drawbacks; however, it is evident that, to meet all requirements needed to tackle the problem at hand, the proposed U-net approach appears to be the optimal one.

6 DISCUSSION

6.1 A CONVOLUTIONAL NEURAL NETWORKS BASED METHOD FOR ANTHRACNOSE INFECTED WALNUT TREE LEAVES IDENTIFICATION

The problem of the automatic identification of anthracnose on walnut tree leaves has been tackled with the use of deep learning algorithms, specifically convolutional neural networks. A total of 4.491 images was acquired, balanced in terms of healthy and infected leaves depictions. A number of preprocessing techniques, such as fast Fourier transform and background removal, were tested in order to evaluate their contribution to the increase of performance.

Several CNN architectures were tested, with accuracies ranging from 92.4% to 98.7%, leading to the one that performed best under all preprocessing scenarios.

The proposed methodology was designed from the ground up in order to address the specific issues of the anthracnose–detection problem. Initially, it was needed to address the background issue, and to investigate whether the background plays any role, and in what level, in the accuracy of the developed classifier. Another issue that needed addressing was the type of images to be processed, meaning if they would be coloured or monochromatic. Coloured images contained the colour information, which is important in this specific use case, since anthracnose discolours areas of the green leaf into brown spots. However, there is value in the monochromatic approach, since the images can be taken in different times within a day (or night), and the colour variations might change. Monochromatic images can create a more generalized classifier, even if the accuracy appears to be lower, because it diminishes this colour dependency. Because of the reduced accuracy of the monochromatic approach, a feature extractor was selected and tested on its performance. The fast Fourier transform indeed improved the results of the monochromatic approach by extracting edge information with a high pass filter, leading to clearer view of the leaves' abrupt changes on their surface.

The best performing algorithm was then compared with a series of state-of-the-art CNNs commonly used for image classification problems. It was noted that the proposed algorithm for the particular problem performed equally, and in some cases even better, compared to these algorithms. This first level of the methodology validates the premise that CNNs are algorithms that can offer high accuracy in image classification-based problems. This has a direct application on precision agriculture where the automatic identification of diseases is crucial for the crops.

The main outcomes of the image classification task, as part of the methodology, can be summarized as follows:

The proposed CNN method exhibits outstanding performance when RGB analysis is performed for the examined images of the anthracnose-on-walnut-leaves case. This can be emanated from the fact that the results produced from the application of the CNN architecture are based on distinct features that appear specifically on the anthracnose-infected leaves, compared to the healthy leaves, i.e. brown spots and areas.

The fast Fourier transform method seems to be of major significance in feature extraction in the case of the grayscale images, because it accentuates the abrupt changes and edges of the leaves. This denotes that the infected leaves have more edgy features than the healthy ones.

The proposed CNN architecture exhibits high performance in all scenarios in which it has been tested considering the case of anthracnose disease identification on walnut tree leaves. As it is observed from Table 11 and Table 12, the accuracies range from 92.4% to 98.7%.

The proposed CNN architecture exhibits better, or similar, performance to well-known CNN architectures (i.e., DenseNet121, VGG16, ResNet50 and InceptionV3) which have been efficiently used as benchmarks in image processing problems for the past years.

Overall, for the purpose of image analysis and classification, the CNN methodology is proved to be proper for complex image classification tasks, such as the one under hand here, when a large number of images is considered, outweighing the popular CNN architectures for image analysis, such as DenseNet121, VGG16, ResNet50 and InceptionV3.

6.2 A DEEP LEARNING APPROACH FOR ANTHRACNOSE INFECTED TREES CLASSIFICATION IN WALNUT ORCHARDS

The problem of object detection in precision agriculture lies to the fact that agricultural environments are extremely rich in information and highly complex regarding visual aspects. Given these circumstances, the aim of the tree-level anthracnose detection task was to prove that an approach based on deep learning algorithms for disease identification on tree-level is attainable and accurate. This object-detection approach was the first step towards identifying the gap, proving that the problem can be tackled, paving the road for more studies and alternative approaches for improving its accuracy and applicability.

Some focal points that were derived from the tree-level anthracnose detection methodology are given below:

- SSD offers great trade-off balance between accuracy and speed, therefore allowing the proposed approach to be able to run in real-time.
- Resnet50 was the best performing CNN classifier and was able to reach ~63% AP.
- Sub-images sized 640×640 pixels produced better results than the ones sized 1,280×1,280 pixels, signifying that at this level, the training requires less clearly detected boxes per image in order to perform well.
- Optimal threshold for the proper classification between healthy/lightly infected and infected was set to 10 predicted bounding boxes per side, or 40 per tree.
- The tree canopy size in relation to the image size, varies for all cases, however some trees are larger with denser canopy, therefore having more leaves than others. This was not taken into consideration during the training phase, since the aim was for the model to be invariant to such changes.
- Type 1 errors (false positives: appointed healthy, classified infected) were investigated after concluding the training and testing phases, revealing that the majority of these cases were small trees. The size of these trees affected the classification results probably because it affected the ratio of the tree surface over the entire image, the leaf density, and the total number of leaves included in one shot meaning that there were less leaves sparsely located inside the canopy.
- In some cases, larger trees that were initially classified as healthy from the expert agronomists, contained small areas of densely populated infected

leaves. The proposed model detected those areas and predicted bounding boxes for infected leaves. For example, the tree with ID A.1.2. was appointed as "healthy" by the expert agronomists, however the trained model detected areas of densely populated anthracnose infected leaves signifying infection (Figure 52). In such cases the misclassification is considered as human error and strengthens the case why systems based on artificial intelligence could assist human experts.

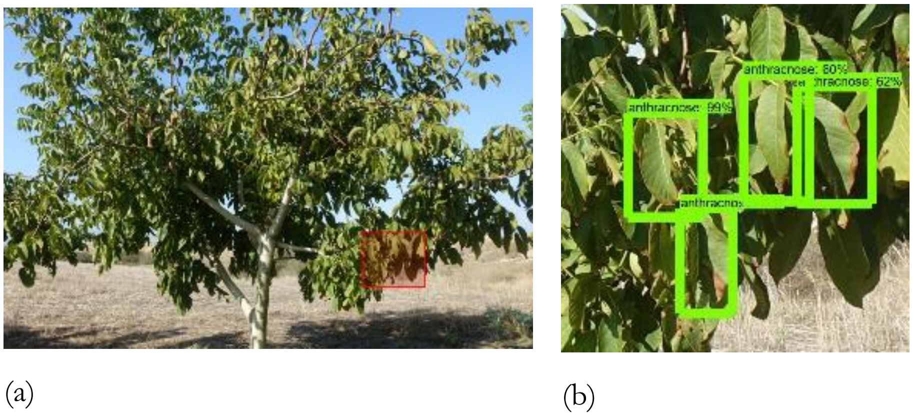


Figure 52. Tree A.1.2. classified by the experts as "healthy" (a), and the infected leaves detected by the trained model (b).

The main outcomes of this tree-level disease detection methodology are summarized in the following statements:

- Proof of concept: A proposed novel approach of identifying anthracnose disease on walnut trees has been developed and evaluated. An object detector was trained on tree-level images and was able to identify and locate anthracnose-infected walnut leaves on images depicting walnut canopies.
- Applicability on real-life conditions: This detector was applied on 279 trees of a walnut orchard in real-field conditions and successfully classified them to infected or healthy trees, as compared with the ground truth, the expert agronomist classification.
- Open-air conditions applicability: The model of choice for object detection was the SSD, an algorithm that utilizes deep learning techniques to predict the location and the class of the object in a single step. This

allows fast and accurate detection and classification, important characteristics for deployment of real-time image capturing systems.

- Low risk misidentification due to the multiple leaves on the canopy: Given the nature of the problem, slight errors in the prediction process can be "smoothed out" due to the fact that the tree canopies are dense and the symptoms are evident on several leaves.
- Relatively high accuracy for object localization: The complexity for locating and identifying anthracnose-infected leaves is significantly high. The objects-to-be-detected are leaves with brown spots and the detection of such target in real field environments is particularly challenging. This target must be properly distinguished from the background, containing a) leaves that in some cases are shaded and in others directly illuminated by the sun, b) leaves at different angles and shapes, c) branches and soil which in many cases are similarly coloured with the infection spots. Still, the object detector was tested on never-before-seen images, reaching a ~63% average precision.
- High accuracy for classification of walnut trees' class: The application of the trained model on previously unseen trees was used as the evaluation process for the proposed methodology. An expert agronomist had priorly classified these trees, however, they were not used for the model training process. Prediction accuracy reached 82.1%, precision 88.3%, recall 85.2% and most importantly, due to the imbalance of classes, the F1 score was 86.9%.
- The classification accuracy of the classifier was considerably high with only a few cases where the model failed to classify the trees in the appropriate category. The vast majority of the trees that were misclassified by the trained model, were false positives, which means that they were healthy but were classified as infected. This implies that these healthy trees would be treated for the disease, which is not a significant issue since the farmer would uniformly apply the fungicide to all trees if following the traditional practices. The real problem is the false-negative misclassifications which would be infected trees that would not be treated. This would pose a risk of inefficient treatment of the disease with possible spreading of the disease at a later point in time. However, these cases were extremely rare and in the proposed system the scouting for diseases can be performed on regular basis limiting the risk of misclassified and mistreated infected trees.
- After further investigation of the misclassified trees, some false-positives were proved to be true-positives, which means that the experts

misclassified those trees as healthy. It is implied that a properly trained, robust AI model, can achieve equally good performance in some cases, if not better, when compared to human experts. This means that such AI models can serve as useful tools for disease detection in a synergetic interaction with expert agronomists.

Pathogens of the genus Marssonina infect other tree crops as well such as apples, with the species Marssonina colonaria, and strawberries with the species Marssonina fragariae, showing the same symptoms with the ones occurred in walnuts [202]. Therefore, the tools developed it this methodology can have a broader application to these crops as well after the required calibration. In addition, the fungus overwinters in fallen leaves on the ground, therefore a disease map can aid in taking protective measurements to contain the disease for the following growing seasons.

6.3 ORCHARD MAPPING WITH DEEP LEARNING SEMANTIC SEGMENTATION

The proposed methodology is a steppingstone used to address a common problem in agricultural environments; the accurate mapping of orchards via UAS (Unmanned Aerial Systems). The primary focus was to construct a methodology of tree segmentation and mapping of orchards. During the testing phase of the models, useful insights were produced, along with some outcomes that showed both FP and FN misidentifications. In general, the FPs in the proposed methodology were the result of the following:

- Weeds and shrubs misidentification as tree canopies; and
- Single-canopy splits that were segmented into multiple smaller high-density instances.

On the other hand, the FNs referred to:

- Circumstantial inadequacy in identifying small canopies; and
- Limitations in identifying trees with leafless canopies.

Considering the preprocessing method that was used, more outcomes can be discussed. For example, the simple EQ, according to the original image brightness and the size of the trees, either produced FPs next to canopies, most of them being weeds, or failed to find the trees entirely, especially if their canopy was small in size. The CLAHE methodology, a valuable tool that can sustain high performance under diverse brightness conditions, reduced and transformed the canopy sizes at a high degree, which lead to shapes and sizes which were different that the canopies in the raw images. A variety of cases has been identified, where the slimming caused by the CLAHE method, is splitting some canopies in parts, resulting to the incorrect calculation of the tree size and consequently the location of its trunk. The model that was trained on images which were transformed into the HSV colourspace, performed well especially in the identification of rough shapes. However, some clearly visible canopies, which were not missed by other methods, where missed by this approach, which resulted to a high number of FNs. The fused approach demonstrated that the shortcomings of each method affected the predicted segmentations, therefore leading to models with worse performance than their best-performing counterparts. Nevertheless, the RGB model achieved the highest training and validation accuracy, the best testing accuracy, and the best performance considering FPs and FNs. This approach demonstrated robustness with all types of orchards and all seasons and for all different sizes, proving that it was the best approach for the problem at hand. Another factor that mostly affected the presence of FNs was the reshaping that images underwent in order to be fed into the training algorithm and consequently to the trained model. Resizing can compress information and in some cases this compression made small canopies “disappear”. However, even though some vital information could have been lost due to resizing, the FN errors remained at a low ratio.

The results of the tree segmentation methodology also demonstrated that the majority of FP segmentations were either a) trees or bushes that were outside of the orchard, b) developed weeds dispersed throughout the field area, or c) split canopies resulting in two separate masks. The first category is easy to handle since the coordinates of the orchard are known and therefore any masks outside of it can be disregarded. Since the tree trunks can be calculated based on the shape of the canopy, their distances can be measured, and a set of rules applied to the orchard’s structure could identify such misidentifications. The latter could serve as a good solution to address the misidentification problems caused by weeds. The third category can also be addressed by applying methods that identify the lines on which each tree is planted, therefore deducting whether the calculated coordinates of a trunk fall within an acceptable limit. All the above indicate future research directions for the continuation of this work.

The second misidentification factor can also be addressed by changing the resolution of the processed images. According to the results of the model performance evaluation on orthomosaics, in orchards with young trees featuring small canopies and filled with developed weeds, the performance was rather poor. This was attributed to the fact that the top view of the weeds was similarly coloured, shaped, and sized as the very small trees within the image. This led to the identification of a large number of FPs. The resolution of the images used in the procedure played an important role in the accuracy. Running the same model on the complete orthomosaic, the results were remarkably improved, reaching 82% accuracy. This was attributed to the fact that the lower pixel resolution resulted in smoothing of the image, merging the pixels that included small weeds with the surroundings, thus making the trees stand out in the image.

Higher accuracy with regard to the overlapping area of pixels may be desired as this is a confident performance metric for model training. However, since the annotation was conducted with high detail on the canopy while the prediction was not required to outline fine details, the metric based on FP and FN predictions was additionally used to identify which method achieved the best results. Regarding the accuracy metric, the best model achieved 91% for training, 90% for validation, and 87% for testing accuracy. Considering the false predictions ratio, 13.3% was achieved for both positive and negative misidentifications of segmented canopies.

In general, image segmentation has been used in many areas; however, this is the first time, based on the authors' knowledge, that it has been applied to UAV images of orchards. Image segmentation was selected over object detection due to a number of benefits, some of which can be summarized in the following bullet points:

- The trees' canopy size can be distinguished,
- The trees' canopy shape can be identified,
- Gaps in the planting scheme due to missing or defoliated and diseased trees can be identified,
- The 2D surface of the imaged canopies can be computed,
- The 3D surface and volume of the trees' canopy can be computed,
- The trees' ages can be approximated,
- The amount of pesticide/water needed for individual trees can be reduced by assigning proportionate amounts,
- The orchard's yield potential can be calculated based on UAV imagery.

There are diverse possibilities for applying image segmentation to orchards and it can cover multiple aspects of operational activities in agriculture. This can be achieved with the use of deep learning, as it has proven its use in multiple occasions [203]. Additionally, semantic segmentation is an active domain with novel approaches being proposed systematically [204], some of which have direct associations with the specific shortcomings of remote sensing [205].

For the proposed tree segmentation approach, U-net was utilized and tweaked to match the addressed problem and the available dataset. U-net might be considered as a relatively basic neural network considering the existence of autoencoders; however, several benefits of its use are apparent from the derived results:

- It achieved consistent performance >85% with all image datasets even if they had not been enhanced,
- High performance could be obtained even with a small number (~100) of images and even without image augmentation,
- The trained model could produce masks instantaneously.

These outcomes render the selection of U-net as optimal for free field deployment on UAV images. The lightness of the architecture leads to trained models which can run with on-board devices using low-power processors. This ease of application, combined with the high performance for the selected RGB model and the fact that this performance was achieved with a small dataset, leads to the conclusion that the proposed methodology is a promising start in the development of a highly sophisticated system that can identify trees in orchards and extrapolate a multitude of information useful for a variety of related operations.

The segmentation methodology could be further advanced by investigating the use of other sensing tools with different capabilities and functions. These sensors might include hyperspectral or multispectral cameras, stereo/depth cameras, or thermal cameras. Each of these sensing tools has different pros and cons:

- Hyper/multispectral cameras. These cameras have multiple applications in agriculture, especially for crop monitoring. The main advantage is the high-value data related to crop and soil status. The disadvantages of this type of camera are the high computational cost that is required to transform the raw data, the high purchase cost, and the operational constraints due to various calibrations that have to take place before each flight and their dependence on weather conditions since cloud coverage greatly affects their measurements.
- Stereo/depth cameras. These are a type of camera commonly used in UGV applications due to their accurate depth perception in tandem with

RGB depiction. There are two major disadvantages that constrain the use of these sensors; their low range of operational distance (most cameras have a 20 m range) and increased onboard computational requirements.

- Thermal cameras. These cameras provide high-value data, similar to the hyper- and multispectral cameras. However, they have high computational and operational costs.

However, using one of these sensors, or a combination of them, would increase the complexity of the system, adding computational costs. Since the goal is to develop a widely acceptable rapid system for on-the-go applications, the proposed methodology was strictly based on using RGB camera, thus making it accessible to the majority of UAS users. An initial approach for developing a simple tree segmentation system that provides instant and accurate results was proposed and developed. Evaluating the use of the abovementioned sensors is part of research future plans for further development.

The proposed system can serve as a tool for identifying the locations of trees and obstacles within orchards and can be used as part of situation awareness and path planning for agricultural robots and autonomous vehicles. In future work, this model could serve as a UAV-based scouting tool in a UAV–UGV synergetic scheme for autonomous UGV operations within orchards. Additionally, this system can identify gaps within tree rows, thus serving as a subsystem of a farm management information system (FMIS).

7 VALUE

Agricultural environments are by their nature highly complex environments with a plethora of information. On top of that, the spatial and temporal variability of said environments change the operational parameters in such a level that one task that could be performed successfully under a particular set of conditions, it could fail when the conditions change. One example is the tree canopy segmentation were, depending on the time of the year, the canopy/ground prevailing colours might be completely different. There is a dire need for solutions that deal with such issues and overcome obstacles. The agricultural domain was and still is one of the most challenging in terms of automating tasks, and therefore, every step towards improvements is valuable.

7.1 OUTCOMES

7.1.1 Scientific soundness

The application of self-learning, data-driven methodologies have proven to be highly effective in the past two decades [206]. Specifically, machine and deep learning algorithms have been constantly breaking performance thresholds and improving the state-of-the-art of a large variety of domains, at an extreme rate [207]. Due to its immense popularity, the AI community has expanded proportionally, and alongside of it, the level of competition. Each and every novel algorithm or application is thoroughly validated, evaluated, cross-checked or peer-reviewed by scientists and practitioners across the world, leading to unbiased systematic filtration. Thus, scientific soundness is preserved, since the proposed solutions are valid in terms of their application real-world scenarios, either as single solutions or as suite of solutions in an integrated system of disease detection and control. Data-driven methodologies, and especially deep learning, are still dealt with scepticism. However, the work presented in this thesis, structured as a tri-modal approach, managed to produce three peer-reviewed papers, all published in globally known scientific journals, and in conferences, where the scientific soundness of the methods was discussed and appraised.

7.1.2 Benefits

Machine and deep learning algorithms have proven again and again that they offer an excellent solution to complex problems. Their benefits are well documented throughout the literature, and are presented in short here:

Performance:

The application of self-learning methodologies for complex vision tasks has numerous benefits. First and foremost, the performance of models built on deep learning algorithms, is by far superior to any other machine learning models. In the past decade, deep learning has proven in many cases and in various applications that it can outperform any other approaches, including well-known machine learning algorithms such as ANNs and SVMs [208]. As explained above, performance is directly related to the availability of large volumes of data, however, this is easily overcome with the acquisition of images with low-cost equipment, and with large temporal variability (seasonal, daily) and abundant spatial variability. On top of that, such models can be trained on a plethora of publicly available datasets that contain annotated images for agricultural-related applications, and then retrained for a more specific task on an acquired smaller dataset.

Continuous improvement:

Data-driven models can only improve with addition of data, given that this data is well curated and properly annotated. This means that even if a model's performance is initially below the desired threshold, it can be improved by acquiring more data during time. For the specific case of disease detection on tree canopies, and related vision-related tasks in agricultural environments, proper data collection can increase the accuracy of predictions since the models will be trained on data that contain additional information. Such examples could include the collection of images during different times, weather conditions, capturing angles, and during different seasons. Additionally, disease progression can be taken into consideration and image collection can be organized based on this. However, if time is not an impacting factor, it means that additional data/images, would contribute nothing to the performance or robustness, and would only decrease training time.

Versatility:

Prediction models that rely on analytical functions or solely on expert knowledge suffer greatly from versatility issues. For example, a rule-based model that is developed specifically to recognize anthracnose on walnut trees based on distinct features of the disease and the leaves, would need to be designed from scratch in order to be able to detect a different disease on a different type of tree. On the contrary, data-driven algorithms, once properly selected, can be used for similar approaches given that the data are properly curated and used for training. As mentioned above, there is also a technique, called transfer learning, that utilizes models that have been trained in similar but different tasks, and uses them as a starting point instead of training the desired model from scratch. Therefore, data-driven models are ideal for applications that need versatility via the utilization of gained knowledge and improvement via acquired data.

7.1.3 Issues

The application of machine and deep learning algorithms besides its tremendous benefits, has on its own some issues and drawbacks. These issues can originate from the nature of the algorithms themselves, or from the nature of the specific problem at hand. An overview of these issues is given below:

Black box:

Deep learning has been an undeniable breakthrough in the AI domain, however it suffers from the black box issue [209]. The black box implies that the algorithms and trained models are so deep and complex that the internal processes become “opaque” to interpretation. This is a nuisance for many applications, however, when interpretability of predictions is of essence i.e. medical applications, it can be a valid obstacle. In the domain of agriculture, and specifically the area of visual inspections, interpretability falls second to results since reasoning is not used for decision making at this level.

Complexity:

Another issue is the complexity and variability of the agricultural environment where the proposed methodology operates on. Seasonal variation can alter the environment almost completely, in terms of visual characteristics, which can render a trained model almost worthless. This is a well-known issue, both for the domain of agricultural environments in terms of visual complexity and variability, as well as in the domain of artificial intelligence in terms of model robustness and invariance to altered input or adversarial examples. The issue of model robustness and resistance to adversarial attacks is at the moment at the centre of extensive research, with new solutions emerging constantly.

Volume (data):

Finally, another issue is the dependability of performance on the amount of data. Deep learning approaches are heavily dependent on large amounts of data, in order to perform above average, due to the size and complexity of their architecture. Specifically for computer vision tasks, DL algorithms use intricate mathematical operations such as tensor multiplications, convolutions and pooling, and other "tricks" such as skip connections or feature propagation, which increase the number of "internal parts" needed to process input images. Given a small number of training images, DL algorithms underfit and fail to make successful predictions, therefore, large volumes of data are necessary for such algorithms to perform as intended.

Annotation:

Data collection can be a relatively easy task with the availability of different types of sensors, the variety of automation with which these sensors can collect data at specific periods or certain events, and the ease of storage, where storing devices have become widely available and cheap to obtain, locally or on the cloud. A well-known issue is the annotation or labelling of the data. This is a time-consuming procedure where each data entry gets appointed to one or several characteristics, usually conducted manually by humans. In the case of disease detection, expert agronomists are required to determine and characterise if the signs on the leaves are a disease or another issue, and if it is, which disease is it. This requires expert knowledge and years in training, and cannot be achieved by untrained personnel, which is commonly used for annotating more generic tasks.

7.2 APPLICABILITY

Main aim of this thesis is to investigate the applicability of the proposed methodology in real operational conditions, in agricultural environments. The main points the present thesis covers, are presented here:

In-field applicability:

As described earlier in the manuscript, several studies attempted to tackle the issue of disease identification on plants and tree leaves. These studies created datasets of leaf images placed on high contrast backgrounds (white/black) or utilized techniques that segmented the leaf from the background. The results of these studies demonstrated the ability of the algorithms to achieve high accuracies in classification tasks, however, they suffered from in-field applicability since in agricultural environments there are almost zero chances to find a single leaf against monochromatic background. The potential for the effective application of such data-driven approaches in real-life conditions was apparent, however, not easily attainable.

The present thesis, addressed this exact issue, developing a methodology based on in-field images, selecting and adapting the proper deep learning algorithms and applying them in a series of vision-related tasks for smart farming. Theoretical approaches for disease detection techniques are valuable in research, however, the development of methodologies for in-field application can be equally challenging if not more. Specifically for agriculture, the operational environments possess such levels of complexity, that can systematically push the limits of such applied methodologies and engage additional research on the matter.

Temporal variability:

As thoroughly stated throughout the manuscript, temporal variability is a key element for the success of vision-related tasks in operational agricultural environments. Temporal variability vastly affects the visual characteristics of an agricultural environment, especially when operations take place throughout the year. The temporal variability is potentially an obstacle in a methodology's robustness and performance. However, by implementing deep learning algorithms that are highly dependent on large volumes of data, this obstacle can turn out to be the tool for building models that not only overcome the visual variability of daily or seasonal effects, but also to build a more robust methodology that is invariant to features that are related to background information, and focus on features which are directly linked to the task at hand, i.e. canopy segmentation or disease detection and classification.

Robustness of performance:

Robustness is the notion that a model's prediction is stable to minor variations in the input, aiming to the fact that the predictions are based on reliable feature abstractions of the task at hand, the same way as a human would perform the said task. Robustness is a difficult concept to define and interpret in machine learning applications, usually associated with system safety and security. However, for applications that do not have immediate "life or death" safety and security concerns, such as the agricultural tasks that are tackled in the present thesis, robustness means stability of performance within the range of predictable temporal and spatial variability, as well as, to minor unforeseeable changes. As described above, temporal variability can increase a model's robustness and performance. Moreover, when dealing with images, there are techniques that can increase robustness by manipulating the existing dataset such as applying geometrical variations on the input images of the training dataset, for example flipping, zooming, tilting and spectral variations, for example shifting a colourspace's values. As described, this technique has been used throughout in all three steps of the presented methodology. Another measure for robustness in trained models, is their resilience against adversarial attacks. Adversarial attacks are well known machine learning techniques that attempt to exploit vulnerabilities in models via obtainable information. Purpose of adversarial attacks is to force a model into producing profoundly false predictions, without changing important aspects of the input image. The type of erroneous prediction the attack aims at, can be classified into two categories: targeted when forcing towards a specific prediction, and untargeted when forcing the misclassification of the correct one. Smart agricultural applications are not unaffected by adversarial attacks, however, it is easier to monitor the performance of models such as the ones used in the presented methodology, since they are built on the premise of Narrow AI, thus serving an exclusive purpose/task. Additionally, operational agricultural environments are "protected" spaces, not in terms of security, but in terms of inaccessibility and general indifference by the public, thus minimizing the risk of unforeseen adversarial inputs. Under such a premise, model robustness can be achieved, albeit it is essential that there are safety measures taken within the operations so that it is preserved.

7.3 FUTURE RESEARCH DIRECTIONS

The proposed methodologies presented in this thesis, are developed with in-field deployment as a basic goal. Proof-of-concept is the first step towards achieving such a goal, however, the following steps can be equally challenging and scientifically intriguing. Some of these future steps are presented below.

- Scalability: One of the most important aspects of data-driven approaches is their ability to optimize their functions in order to effectively address the increasing number of data that accumulate over time. Systems based on deep-learning technologies are highly reliable on the quantity and quality of the data they utilize. A simplistic pipeline for non-methodological data collection could increase the number of data and cover the "quantity" aspect of requirements, however, unless quality is also increased, redundant data could be considered "dead weight" to a model's performance. The applicability and sustainability of such systems rely heavily on their ability to scale and adapt in various environments.

The present thesis proposes a methodology that will be directly integrated and applied in a variety of real agricultural environments; therefore, it needs to be easily scalable, otherwise it will be used narrowly and will be short-lived.

- Robustness: A critical aspect for the models' performance consistency when operating in different environments, is robustness. Ensuring robustness in such a large variety of agricultural environments and external conditions is a challenging task, particularly when the models are built for in-field deployment. Systematic research on the potential conditions that could affect a model's performance (such as fog) need to take place in order to identify such threats and develop solutions. On top of that, research is also necessary for identifying potential adversarial attacks that can impede a model's performance drastically.
- Custom variants of algorithms: The AI domain is one of (if not) the most popular research domains all over the world. Novel algorithms are developed weekly, pushing the limits and capabilities of the state-of-the-art, and on the same time past concepts are re-evaluated for their usefulness. Together with ML and DL algorithms, soft computing techniques such as fuzzy-logic systems, can offer additional value to the models. On top of that, model explainability is highly sought after, together with general AI, where researchers aim to create algorithms that can simultaneously handle multiple types of input in order to extract an output, just like human brains do. It is impossible to know beforehand which of these tools will provide added value to the models and their respective tasks, therefore rigorous research needs to be conducted.
- Decision support systems: Finally, the final aim of the canopy segmentation, the disease detection, and its proper classification, is to provide input for decision support systems (DSS). This is an inherently exigent task, even for human experts, since there are a lot of factors that need to be taken into consideration before reaching a decision. Considering that these vision-based tasks are built without any additional information such as geographical or temporal information and metadata, further research is in place for properly and seamlessly integrating heterogenous data for enhancing the produced outcome. This will lead to a better input for the DSS, which in turn would require to take into consideration a large range of inputs such as weather conditions or operational costs.

8 CONCLUSIONS

The present thesis focuses on the application of machine learning techniques for utilizing large volumes of data in order to extract valuable information. Main scope of this thesis is to investigate the effectiveness of machine learning algorithms in complex and demanding tasks and develop methodologies that solve important problems with the use of large volume of data.

Initially, a broad range of algorithms [145] and applications [210][211][212] has been investigated, leading to the selection of smart agriculture as the domain on which more extensive research will take place. In particular, vision tasks related to tree segmentation and disease detection are of high value for Agriculture 4.0, albeit inherently complex and difficult to conduct in real-life conditions. A number of studies addressed similar problems; however, the images were either taken in laboratories or have been properly preprocessed beforehand with background and noise removal techniques. A gap has been identified regarding the application of machine learning techniques for computer vision problem in real operational agricultural environments. This thesis aimed to fill this gap by developing a methodology that would be able to tackle a series of tasks related to crop management, based on machine/deep learning algorithms and large volumes of data. Target is the identification of tree canopies and their positions within an orchard from images taken from UAVs, which will consequently be used to guide UGVs to their exact positions and with visual inspection, detect the presence of disease on tree level, and finally classify the disease on the leaves. This methodology is driven by the quantity and quality of data collected from in-field measurements, alongside expert annotation of diseases. In particular, the problem this thesis focused upon, is walnut orchards located in Greece, where anthracnose, a viral disease, affects the yearly yield, which in turn can pose a serious economic impact to producers since walnut is a high value crop.

The developed methodology comprises of three sequential tasks, all of them based on visual information. The tasks have distinct scopes between them; however, the ultimate goal is to identify the location of trees within an orchard and classify each tree based on disease presence on them, in order to create a variability map of the orchard, which consequently can be used for precise treatment with minimum waste and costs. The main requirement which makes the present methodology challenging, is its in-field deployment with low-cost imaging equipment, meaning that it is developed to work in agricultural environments under real conditions instead of labs, and is capable of doing so, only with RGB cameras instead of expensive hyperspectral or infrared cameras. Expensive equipment can be useful, but the added cost would incur a steep increase which would have negative impact to the marketability of such a solution. Instead, the use of powerful machine and deep learning algorithms was preferred to tackle the problems of the specific tasks.

Extensive research was conducted for the selection of the proper algorithm for each task. A plethora of data was available for the investigation, development, and application of the proposed methodology. The data, exclusively RGB images, where collected from multiple in-field image collections, at times where anthracnose could be clearly identified by expert agronomist, and thus, all data could be properly labelled. Initially, a variety of machine learning algorithms were tested for their performance on image classification on diseased leaves, however, it was evident that neural networks outperformed the rest, which was anticipated since neural networks, including their derivatives, have been systematically the state-of-the-art for the past decade.

The final results for each level of the methodology showed significant promise in terms of accuracy and robustness, as well as in-field deployment. For the task of tree canopy segmentation, a U-net architecture has been modified in order to fit best to the problem's requirements and the available input data. The resulted trained model is lightweight and can be implemented with low computational cost on edge devices, such as UAVs, resulting to close to real-time segmentation. The validation of the approach was conducted on test images similar to the ones used for training, however, full orthomosaics were tested as well, with the model achieving up to 99% accuracy on slightly different type of input (camera image vs orthomosaic). For the task of disease detection on tree level, arguably the most complex of all, an SSD object detector was trained on in-field images of trees infected with anthracnose. Detecting anthracnose infected leaves proved to be challenging on that level due to the images' resolution, therefore a sliding-window-type algorithm was developed in order to apply inference to parts of the tree. The results would then be aggregated and combined for all sides of the tree, and a final classification for the tree would derive. The trained model was applied on a real orchard, initially inspected by an expert agronomist, and managed to correctly classify 87% of previously unknown walnut trees. Finally, for the task of image classification on leaf level, a custom CNN architecture was designed and developed for the proper classification of anthracnose infected and healthy walnut leaves. Different types of preprocessing were tested on the input images, with the trained models ranging between 92.4% to 98.7% in accuracy.

Notable issues for applying data-driven techniques in computer vision tasks for agricultural environments can be related to the volume and quality of the data, the demanding task of labelling, as well as the models' performance in different conditions and their lack of explainability. However, these are all known issues and subject to continuous research by many researchers and scientists. The proposed methodology can and will act as a proof-of-concept for the effective in-field computer vision tasks in the complex operational agricultural environments. The natural outcome of this thesis will be to use the proposed methodology as a steppingstone, and further develop it in order to create a solid prototype ready for in-field deployment.

Future plans include, but are not limited to, further development of the methodology towards its applicability on in field operations. Aim of the proposed methodology is to be applied and perform successfully in real conditions, which is a challenge on its own. In field deployment will present new challenges and obstacles concerning the performance of the models in various environments, under different light conditions, seasonal variations, and many more unexpected variables that can affect the models' performance. Additionally, the integration of such a methodology to production level with low consumption requirements, fast execution, increased security mechanisms and adaptability features is a mandatory task. In the meantime, the expected growth in the ML and AI domain in the short-term future, will introduce new algorithms and methods that will offer increased performance and explainability through the models. This fact will allow all subsequent applications to research and re-evaluate the application of new algorithms for existing solutions.

9 BIBLIOGRAPHY

[1] K. A. Zimmermann, “History of Computers - A Brief Timeline of Their Evolution | Live Science,” *Live Science*, 2017. .

[2] D. A. Trinkle, D. Auchter, S. A. Merriman, and T. E. Larson, “History of Computers,” in *The History Highway*, 2021.

[3] P. N. Stearns, *The Industrial Revolution in World History*. 2020.

[4] K. Kumar and M. Castells, “The Information Age: Economy, Society and Culture. Volume I. The Rise of the Network Society,” *Br. J. Sociol.*, 1997.

[5] J. L. Abu-Lughod and M. Castells, “The Information Age: Economy, Society and Culture, Vol. 2: The Power of Identity,” *Contemp. Sociol.*, 1998.

[6] Jamie Woodcock, “Work in The Age of Data,” 2019.

[7] V. Shobana and N. Kumar, “Big data - A review,” *Int. J. Appl. Eng. Res.*, 2015.

[8] “International Data Corporation (IDC).” [Online]. Available: https://www.idc.com/.

[9] G. Dhillon, “From The Age Of Computing To The Age Of Data: Are You Ready?,” *Forbes*, Mar-2019.

[10] S. Hagan, “Digital Economy Has Been Growing at Triple the Pace of U.S. GDP,” *Bloomberg*, 2018.

[11] B.-T. S. Alliance, “2013 BSA global cloud computing scorecard,” 2013.

[12] M. Javornik, N. Nadoh, and D. Lange, “Data Is the New Oil,” 2019.

[13] Deloitte, “2020 Global Life Sciences Outlook,” *Deloitte Insights*, 2020.

[14] C. Petrov, “25+ Impressive Big Data Statistics for 2022,” *techjury*, February, 2022.

[15] A. Hoist, “Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025.” Statista, 2021.

[16] G. Press, “54 Predictions About The State Of Data In 2021,” *Forbes*, 2020. .

[17] S. Liu, “Big data - Statistics & Facts,” *Statista*, 2020. .

[18] D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice Hall, 1982.

[19] T. S. Huang, "Computer Vision: Evolution and Promise," *Report*, 1997.

[20] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*. 1993.

[21] R. Klette, *Concise Computer Vision - An Introduction into Theory and | Reinhard Klette | Springer*. 2014.

[22] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Pearson, 2001.

[23] M. K. Bhuyan, *Computer Vision and Image Processing*. Macmillan Education UK, 2019.

[24] B. Ja¨hne, "Computer Vision and Applications: A Guide for Students and Practitioners," *J. Electron. Imaging*, vol. 11, no. 1, p. 115, 2002.

[25] D. Forsyth and J. Ponce, *Computer vision: a modern approach*. 2003.

[26] F. Magdoff and B. Tokar, "Agriculture and food in crisis an overview," *Monthly Review*, vol. 61, no. 3. 2009.

[27] A. McBratney, B. Whelan, T. Ancev, and J. Bouma, "Future directions of precision agriculture," in *Precision Agriculture*, 2005.

[28] B. Whelan and A. McBratney, "Definition and interpretation of potential management zones in Australia," *Proc. 11th Aust. Agron. Conf.*, 2003.

[29] W. Borg, "Measuring Vegetation," *Educational research: an introduction. America*. 2017.

[30] R. G. Trevisan, D. S. Bullock, and N. F. Martin, "Spatial variability of crop responses to agronomic inputs in on-farm precision experimentation," *Precis. Agric.*, 2021.

[31] M. Sardogan, A. Tuncer, and Y. Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm," in *UBMK 2018 - 3rd International Conference on Computer Science and Engineering*, 2018.

[32] P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," in *Conference on Advances in Signal Processing, CASP 2016*, 2016.

[33] P. Tm, A. Pranathi, K. Saiashritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato Leaf Disease Detection Using Convolutional Neural Networks," in *2018 11th International Conference on Contemporary Computing, IC3 2018*, 2018.

[34] B. C. Heidman and U. A. Rosa, "Real-Time Tree Localization in

Orchards," *Appl. Eng. Agric.*, 2008.

[35] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*. 2018.

[36] A. M. Turing, "Computing machinery and intelligence," in *Machine Intelligence: Perspectives on the Computational Model*, 2012, pp. 1–28.

[37] R. J. Solomonoff, "The time scale of artificial intelligence: Reflections on social effects," *Hum. Syst. Manag.*, vol. 5, no. 2, pp. 149–153, 1985.

[38] E. Stern, "Individual differences in the learning potential of human beings," *npj Sci. Learn.*, vol. 2, no. 1, 2017.

[39] V. Marinoudi, C. G. Sørensen, S. Pearson, and D. Bochtis, "Robotics and labour in agriculture. A context consideration," *Biosyst. Eng.*, vol. 184, pp. 111–121, Aug. 2019.

[40] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-Inspired Artificial Intelligence," *Neuron*, vol. 95, no. 2. pp. 245–258, 2017.

[41] N. Kriegeskorte and P. K. Douglas, "Cognitive computational neuroscience," *Nature Neuroscience*, vol. 21, no. 9. pp. 1148–1160, 2018.

[42] S. Makridakis, "The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms," *Futures*, vol. 90. pp. 46–60, 2017.

[43] A. Agrawal, J. Gans, and A. Goldfarb, "The Impact of Artificial Intelligence on Innovation," in *The Economics of Artificial Intelligence*, 2019, pp. 115–148.

[44] L. Skyttner, "Artificial Intelligence and Life," in *General Systems Theory*, 2006, pp. 319–351.

[45] I. Farkas, "Artificial intelligence in agriculture," in *Computers and Electronics in Agriculture*, 2003, vol. 40, no. 1–3, pp. 1–3.

[46] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism.*, vol. 69, pp. S36–S40, 2017.

[47] L. Zhang, Y. Pan, X. Wu, and M. J. Skibniewski, "Introduction to Artificial Intelligence," in *Lecture Notes in Civil Engineering*, vol. 163, 2021, pp. 1–15.

[48] B. hu Li, B. cun Hou, W. tao Yu, X. bing Lu, and C. wei Yang, "Applications of artificial intelligence in intelligent manufacturing: a review," *Frontiers of Information Technology and Electronic Engineering*, vol. 18,

no. 1. pp. 86–96, 2017.

[49] D. J. Cook, J. C. Augusto, and V. R. Jakkula, “Ambient intelligence: Technologies, applications, and opportunities,” *Pervasive and Mobile Computing*, vol. 5, no. 4. pp. 277–298, 2009.

[50] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245. pp. 255–260, 2015.

[51] J. A. Campbell, “On artificial intelligence,” *Artif. Intell. Rev.*, vol. 1, no. 1, pp. 3–9, 1986.

[52] D. E. Goldberg and J. H. Holland, “Genetic Algorithms and Machine Learning,” *Machine Learning*, vol. 3, no. 2. pp. 95–99, 1988.

[53] A. K. Tiwari, “Introduction to machine learning,” *Ubiquitous Machine Learning and Its Applications*. pp. 1–14, 2017.

[54] D. Roth, “Learning based programming,” *Stud. Fuzziness Soft Comput.*, vol. 194, pp. 73–95, 2006.

[55] T. Goertzel, “The path to more general artificial intelligence,” in *Journal of Experimental and Theoretical Artificial Intelligence*, 2014, vol. 26, no. 3, pp. 343–354.

[56] B. Goertzel and C. Pennachin, “Artificial General Intelligence,” *Cognitive Technologies*, vol. 8. 2007.

[57] P. Cunningham, M. Cord, and S. J. Delany, “Supervised learning,” in *Cognitive Technologies*, 2008, pp. 21–49.

[58] L. Francis, “Unsupervised learning,” in *Predictive Modeling Applications in Actuarial Science: Volume I: Predictive Modeling Techniques*, 2014, pp. 280–312.

[59] C. C. Aggarwal, “Educational and software resources for data classification,” in *Data Classification: Algorithms and Applications*, 2014, pp. 657–665.

[60] D. Silver *et al.*, “Mastering the game of Go without human knowledge,” *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[61] Y. Liao, K. Yi, and Z. Yang, “CS229 Final Report Reinforcement Learning to Play Mario,” *Stanford.Edu*, 2012.

[62] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, “Learning to Walk Via Deep Reinforcement Learning,” 2019.

[63] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, “Large-scale parallel

collaborative filtering for the netflix prize," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008, vol. 5034 LNCS, pp. 337–348.

[64] K. Jacobson, V. Murali, E. Newett, B. Whitman, and R. Yon, "Music Personalization at Spotify," 2016, pp. 373–373.

[65] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, 2003.

[66] Y. Chen, F. S. Tsai, and K. L. Chan, "Machine learning techniques for business blog search and mining," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 581–590, 2008.

[67] A. Anagnostis, E. Papageorgiou, and D. Bochtis, "Application of artificial neural networks for natural gas consumption forecasting," *Sustain.*, vol. 12, no. 16, 2020.

[68] A. Anagnostis, G. Asiminari, E. Papageorgiou, and D. Bochtis, "A convolutional neural networks based method for anthracnose infected walnut tree leaves identification," *Appl. Sci.*, vol. 10, no. 2, 2020.

[69] N. Papandrianos, E. Papageorgiou, A. Anagnostis, and K. Papageorgiou, "Bone metastasis classification using whole body images from prostate cancer patients based on convolutional neural networks application," *PLoS One*, vol. 15, no. 8, 2020.

[70] N. Papandrianos, E. Papageorgiou, A. Anagnostis, and K. Papageorgiou, "Efficient bone metastasis diagnosis in bone scintigraphy using a fast convolutional neural network architecture," *Diagnostics*, 2020.

[71] N. Papandrianos, E. I. Papageorgiou, and A. Anagnostis, "Development of Convolutional Neural Networks to identify bone metastasis for prostate cancer patients in bone scintigraphy," *Ann. Nucl. Med.*, vol. 34, no. 11, pp. 824–832, 2020.

[72] N. Papandrianos, E. Papageorgiou, A. Anagnostis, and K. Papageorgiou, "Efficient bone metastasis diagnosis in bone scintigraphy using a fast convolutional neural network architecture," *Diagnostics*, vol. 10, no. 8, 2020.

[73] D. J. Olive, *Linear regression*. 2017.

[74] G. Grégoire, "Multiple linear regression," in *EAS Publications Series*, 2015, vol. 66, pp. 45–72.

[75] L. J. Davis and K. P. Offord, “Logistic regression,” in *Emerging Issues and Methods in Personality Assessment*, 2013, pp. 273–283.

[76] J. W. Gooch, “Stepwise Regression,” in *Encyclopedic Dictionary of Polymers*, 2011, pp. 998–998.

[77] L. Moutinho, G. Hutcheson, G. Hutcheson, and G. Hutcheson, “Ordinary Least-Squares Regression,” in *The SAGE Dictionary of Quantitative Management Research*, 2014, pp. 225–228.

[78] J. H. Friedman, “Multivariate Adaptive Regression Splines,” *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.

[79] W. S. Cleveland, “Robust locally weighted regression and smoothing scatterplots,” *J. Am. Stat. Assoc.*, vol. 74, no. 368, pp. 829–836, 1979.

[80] B. Efron *et al.*, “Least angle regression,” *Ann. Stat.*, vol. 32, no. 2, pp. 407–499, 2004.

[81] G. C. McDonald, “Ridge regression,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 1, no. 1, pp. 93–100, 2009.

[82] C. De Mol, E. De Vito, and L. Rosasco, “Elastic-net regularization in learning theory,” *J. Complex.*, vol. 25, no. 2, pp. 201–230, 2009.

[83] S. L. Kukreja, J. Löfberg, and M. J. Brenner, “a Least Absolute Shrinkage and Selection Operator (Lasso) for Nonlinear System Identification,” *IFAC Proc. Vol.*, vol. 39, no. 1, pp. 814–819, 2006.

[84] P. Hartono, “Bayes theorem,” *Kyokai Joho Imeji Zasshi/Journal Inst. Image Inf. Telev. Eng.*, vol. 63, no. 1, pp. 52–54, 2009.

[85] H. S. Stern, “Bayesian Statistics,” in *International Encyclopedia of the Social & Behavioral Sciences: Second Edition*, 2015, pp. 373–377.

[86] N. Ye and N. Ye, “Naïve Bayes Classifier,” in *Data Mining*, 2020, pp. 31–36.

[87] H. Zhang, “The optimality of Naive Bayes,” in *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*, 2004, vol. 2, pp. 562–567.

[88] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian Network Classifiers,” *Mach. Learn.*, vol. 29, no. 2–3, pp. 131–163, 1997.

[89] G. F. Cooper and E. Herskovits, “A Bayesian method for the induction of probabilistic networks from data,” *Mach. Learn.*, vol. 9, no. 4, pp. 309–347, 1992.

[90] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[91] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *Am. Stat.*, vol. 46, no. 3, pp. 175–185, 1992.

[92] T. Kohonen, “The Self-Organizing Map,” *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[93] S. Seo and K. Obermayer, “Soft learning vector quantization,” *Neural Comput.*, vol. 15, no. 7, pp. 1589–1604, 2003.

[94] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally Weighted Learning,” *Artif. Intell. Rev.*, vol. 11, no. 1–5, pp. 11–73, 1997.

[95] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. 2017.

[96] T. Hothorn, K. Hornik, and A. Zeileis, “ctree: Conditional Inference Trees,” *Compr. R Arch. Netw.*, no. Quinlan 1993, pp. 1–34, 2015.

[97] J. R. Quinlan, “Induction of Decision Trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[98] S. L. Salzberg, “C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993,” *Mach. Learn.*, vol. 16, no. 3, pp. 235–240, 1994.

[99] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[100] Y. Freund and R. E. Schapire, “Experiments with a New Boosting Algorithm,” *Proc. 13th Int. Conf. Mach. Learn.*, pp. 148–156, 1996.

[101] L. Breiman, “Arcing the edge,” *Statistics (Ber).*, 1997.

[102] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[103] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[104] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A K-Means Clustering Algorithm,” *Appl. Stat.*, vol. 28, no. 1, p. 100, 1979.

[105] C. G. Small, “A Survey of Multidimensional Medians,” *Int. Stat. Rev. / Rev. Int. Stat.*, vol. 58, no. 3, p. 263, 1990.

[106] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4. pp. 433–459,

2010.

[107] A. De Cheveigné, "Quadratic component analysis," *Neuroimage*, vol. 59, no. 4, pp. 3838–3844, 2012.

[108] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, 1999.

[109] T. Hastie, R. Tibshirani, and A. Buja, "Flexible discriminant analysis by optimal scoring," *J. Am. Stat. Assoc.*, vol. 89, no. 428, pp. 1255–1270, 1994.

[110] P. Geladi and B. R. Kowalski, "Partial least-squares regression: a tutorial," *Anal. Chim. Acta*, vol. 185, no. C, pp. 1–17, 1986.

[111] R. Kramer, "Principal Component Regression," in *Chemometric Techniques for Quantitative Analysis*, 1998, pp. 99–110.

[112] W. M. Bowen, "Multidimensional Scaling," in *International Encyclopedia of Human Geography*, 2009, pp. 216–221.

[113] J. H. Friedman and W. Stuetzle, "Projection Pursuit Regression," *J. Am. Stat. Assoc.*, vol. 76, no. 376, p. 817, 1981.

[114] R. Agrawal and R. Srikant, "Fast Algorithms For Mining Association Rules In Datamining," in *International Journal of Scientific & Technology Research*, 2013, vol. 2, no. 12, pp. 13–24.

[115] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 3, pp. 372–390, 2000.

[116] Y. Y. Chen, Y. H. Lin, C. C. Kung, M. H. Chung, and I. H. Yen, "Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes," *Sensors (Switzerland)*, vol. 19, no. 9, 2019.

[117] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[118] T. Hastie, R. Tibshirani, and J. Friedman, "Springer Series in Statistics," *Elem. Stat. Learn.*, vol. 27, no. 2, pp. 83–85, 2009.

[119] M. A. Zinkevich, M. Weimer, A. Smola, and L. Li, "Parallelized stochastic gradient descent," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010.

[120] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[121] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Syst.*, 1988.

[122] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities (associative memory/parallel processing/categorization/content-addressable memory/fail-soft devices)," *Proc. NatL Acad. Sci. USA*, 1982.

[123] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[124] G. Hinton, "Where do features come from?," *Cogn. Sci.*, vol. 38, no. 6, pp. 1078–1101, 2014.

[125] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*. 2015.

[126] A. L. Caterini and D. E. Chang, "Recurrent neural networks," in *SpringerBriefs in Computer Science*, no. 9783319753034, 2018, pp. 59–79.

[127] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, 1997.

[128] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 1724–1734.

[129] J. Günther, P. M. Pilarski, G. Helfrich, H. Shen, and K. Diepold, "First Steps Towards an Intelligent Laser Welding Architecture Using Deep Neural Networks and Reinforcement Learning," *Procedia Technol.*, vol. 15, pp. 474–483, 2014.

[130] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[131] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*. 2019.

[132] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings - 30th IEEE Conference on Computer*

Vision and Pattern Recognition, CVPR 2017, 2017, vol. 2017-Janua, pp. 4438–4446.

[133] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[134] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, vol. 3, no. January, pp. 2672–2680.

[135] G. Hinton, "Deep belief networks," *Scholarpedia*, vol. 4, no. 5, p. 5947, 2009.

[136] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Journal of Machine Learning Research*, 2009, vol. 5, pp. 448–455.

[137] T. Bunde *et al.*, "Munich Security Report 2022: Breaking the Tide – Unlearning Helplessness," Feb. 2022.

[138] J. A. Aznar-Sánchez, M. Piquer-Rodríguez, J. F. Velasco-Muñoz, and F. Manzano-Agugliaro, "Worldwide research trends on sustainable land use in agriculture," *Land use policy*, vol. 87, 2019.

[139] M. Lampridi, D. Kateris, C. G. Sørensen, and D. Bochtis, "Energy footprint of mechanized agricultural operations," *Energies*, vol. 13, no. 3, p. 769, Feb. 2020.

[140] M. G. Lampridi, C. G. Sørensen, and D. Bochtis, "Agricultural sustainability: A review of concepts and methods," *Sustainability (Switzerland)*, vol. 11, no. 18. p. 5120, Sep-2019.

[141] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. February, pp. 70–90, 2018.

[142] D. D. Bochtis, C. G. C. Sørensen, and P. Busato, "Advances in agricultural machinery management: A review," *Biosystems Engineering*, vol. 126. Academic Press, pp. 69–81, Oct-2014.

[143] A. Chlingaryan, S. Sukkarieh, and B. Whelan, "Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: A review," *Computers and Electronics in Agriculture*, vol. 151. pp. 61–69, 2018.

[144] A. Anagnostis, L. Benos, D. Tsaopoulos, A. Tagarakis, N. Tsolakis, and D. Bochtis, "Human activity recognition through recurrent neural networks for human–robot interaction in agriculture," *Appl. Sci.*, vol. 11, no. 5, pp. 1–21, Mar. 2021.

[145] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors (Switzerland)*, vol. 18, no. 8. Multidisciplinary Digital Publishing Institute, p. 2674, Aug-2018.

[146] A. Anagnostis *et al.*, "A deep learning approach for anthracnose infected trees classification in walnut orchards," *Comput. Electron. Agric.*, vol. 182, p. 105998, 2021.

[147] X. E. Pantazi, D. Moshou, and A. A. Tamouridou, "Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers," *Comput. Electron. Agric.*, vol. 156, pp. 96–104, Jan. 2019.

[148] M. Kerkech, A. Hafiane, and R. Canals, "Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images," *Comput. Electron. Agric.*, vol. 155, pp. 237–243, 2018.

[149] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access*, vol. 6, pp. 30370–30377, 2018.

[150] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Electron. Agric.*, vol. 150, pp. 220–234, 2018.

[151] M. T. Habib, A. Majumder, A. Z. M. Jakaria, M. Akter, M. S. Uddin, and F. Ahmed, "Machine vision based papaya disease recognition," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 3, pp. 300–309, 2020.

[152] J. Abdulridha, O. Batuman, and Y. Ampatzidis, "UAV-based remote sensing technique to detect citrus canker disease utilizing hyperspectral imaging and machine learning," *Remote Sens.*, vol. 11, no. 11, 2019.

[153] K. Simonyan and A. Zisserman, "VGG-16," *arXiv Prepr.*, 2014.

[154] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Comput. Ind.*, vol. 108, pp. 115–120, 2019.

[155] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Comput. Electron. Agric.*, 2018.

[156] S. Khaki and L. Wang, "Crop yield prediction using deep neural networks," *Front. Plant Sci.*, vol. 10, 2019.

[157] P. Nevavuori, N. Narra, and T. Lipping, “Crop yield prediction with deep convolutional neural networks,” *Comput. Electron. Agric.*, vol. 163, Aug. 2019.

[158] Q. Yang, L. Shi, J. Han, Y. Zha, and P. Zhu, “Deep convolutional neural networks for rice grain yield estimation at the ripening stage using UAV-based remotely sensed images,” *F. Crop. Res.*, vol. 235, pp. 142–153, 2019.

[159] Y. Chen *et al.*, “Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimages,” *Remote Sens.*, vol. 11, no. 13, 2019.

[160] Y. Cai *et al.*, “Integrating satellite and climate data to predict wheat yield in Australia using machine learning approaches,” *Agric. For. Meteorol.*, 2019.

[161] J. Han *et al.*, “Prediction of winter wheat yield based on multi-source data and machine learning in China,” *Remote Sens.*, vol. 12, no. 2, 2020.

[162] C. Folberth, A. Baklanov, J. Balkovič, R. Skalský, N. Khabarov, and M. Obersteiner, “Spatio-temporal downscaling of gridded crop model yield estimates based on machine learning,” *Agric. For. Meteorol.*, vol. 264, pp. 1–15, 2019.

[163] J. Yu, S. M. Sharpe, A. W. Schumann, and N. S. Boyd, “Deep learning for image-based weed detection in turfgrass,” *Eur. J. Agron.*, vol. 104, pp. 78–84, 2019.

[164] A. Bakhshipour and A. Jafari, “Evaluation of support vector machine and artificial neural networks in weed detection using shape features,” *Comput. Electron. Agric.*, vol. 145, pp. 153–160, 2018.

[165] M. Dian Bah, A. Hafiane, and R. Canals, “Deep learning with unsupervised data labeling for weed detection in line crops in UAV images,” *Remote Sens.*, vol. 10, no. 11, 2018.

[166] J. Gao *et al.*, “Fusion of pixel and object-based features for weed mapping using unmanned aerial vehicle imagery,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 67, pp. 43–53, 2018.

[167] A. I. de Castro, J. Torres-Sánchez, J. M. Peña, F. M. Jiménez-Brenes, O. Csillik, and F. López-Granados, “An automatic random forest-OBIA algorithm for early weed mapping between and within crop rows using UAV imagery,” *Remote Sens.*, vol. 10, no. 2, 2018.

[168] P. Lottes, J. Behley, A. Milioto, and C. Stachniss, “Fully convolutional networks with sequential information for robust crop and weed detection in precision farming,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2870–

2877, 2018.

[169] H. Huang, J. Deng, Y. Lan, A. Yang, X. Deng, and L. Zhang, "A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery," *PLoS One*, vol. 13, no. 4, 2018.

[170] K. ling TU, L. juan LI, L. ming YANG, J. hua WANG, and Q. SUN, "Selection for high quality pepper seeds by machine vision and classifiers," *J. Integr. Agric.*, vol. 17, no. 9, pp. 1999–2006, 2018.

[171] K. Tan, R. Wang, M. Li, and Z. Gong, "Discriminating soybean seed varieties using hyperspectral imaging and machine learning," *J. Comput. Methods Sci. Eng.*, vol. 19, no. 4, pp. 1001–1015, 2019.

[172] C. Gonzalez Viejo, S. Fuentes, D. Torrico, K. Howell, and F. R. Dunshea, "Assessment of beer quality based on foamability and chemical composition using computer vision algorithms, near infrared spectroscopy and machine learning algorithms," *J. Sci. Food Agric.*, vol. 98, no. 2, pp. 618–627, 2018.

[173] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 770–778.

[174] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[175] H. A. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," in *Computer Vision and Pattern Recognition*, 2009.

[176] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[177] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[178] A. Soni, R. Koner, and V. G. K. Villuri, "M-unet: Modified u-net segmentation framework with satellite imagery," in *Advances in Intelligent Systems and Computing*, 2020.

[179] Tzutalin, "LabelImg." Git code, 2015.

[180] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, 2010.

[181] D. Drozdov, M. Kolomeychenko, and Y. Borisov, "Supervise.ly." [Online]. Available: https://supervise.ly/.

[182] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, 1989.

[183] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector BT - Computer Vision – ECCV 2016," in *Computer Vision – ECCV 2016*, 2016.

[184] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9905 LNCS, pp. 21–37.

[185] S. Chen and X. Wang, "Single-Shot Detector with Multiple Inference Paths," *2019 IEEE Int. Conf. Image Process.*, no. 2, pp. 2005–2009, 2019.

[186] G. Salton and M. J. McGill, "Introduction to modern information retrieval," *Inf. Process. Manag.*, vol. 19, no. 6, pp. 402–403, 1983.

[187] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What Makes for Effective Detection Proposals?," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[188] S. Der Chen and A. R. Ramli, "Minimum mean brightness error bi-histogram equalization in contrast enhancement," *IEEE Trans. Consum. Electron.*, 2003.

[189] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," *J. VLSI Signal Process. Syst. Signal Image. Video Technol.*, 2004.

[190] A. Carass *et al.*, "Evaluating White Matter Lesion Segmentations with Refined Sørensen-Dice Analysis," *Sci. Rep.*, 2020.

[191] Paul Jaccard, "The Distribution of the Flora in the Alpine Zone," *New Phytol.*, 1912.

[192] Y. Zhang, S. Mehta, and A. Caspi, "Rethinking Semantic Segmentation Evaluation for Explainability and Model Selection," Jan. 2021.

[193] V. Khryashchev and R. Larionov, "Wildfire Segmentation on Satellite Images using Deep Learning," in *Moscow Workshop on Electronic and Networking Technologies, MWENT 2020 - Proceedings*, 2020.

[194] A. Anagnostis, G. Asiminari, E. Papageorgiou, and D. Bochtis, "A Convolutional Neural Networks Based Method for Anthracnose Infected Walnut Tree Leaves Identification," *Appl. Sci.*, 2020.

[195] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data - Recommendations for the use of performance metrics," in *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, 2013.

[196] G. Bradski, "The OpenCV Library," *Dr. Dobb's J. Softw. Tools*, 2000.

[197] N. A. Peppas and J. R. Robinson, "Bioadhesives for optimization of drug delivery," *J. Drug Target.*, vol. 3, no. 3, pp. 183–184, 1995.

[198] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, 2011.

[199] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[200] F. Chollet and others, "Keras." 2015.

[201] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." 2016.

[202] P. Sharma, "An Overview of the Field of Family Business Studies: Current Status and Directions for the Future," *Fam. Bus. Rev.*, vol. 17, no. 1, pp. 1–36, 2004.

[203] A. Garcia-Garcia, S. Orts-Escolano, S. O. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv*. 2017.

[204] Y. Yuan, X. Chen, and J. Wang, "Object-Contextual Representations for Semantic Segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2020.

[205] Y. Li, T. Shi, Y. Zhang, W. Chen, Z. Wang, and H. Li, "Learning deep semantic segmentation network under multiple weakly-supervised constraints for cross-domain remote sensing image semantic segmentation," *ISPRS J. Photogramm. Remote Sens.*, 2021.

[206] I. Arel, D. Rose, and T. Karnowski, "Deep machine learning-A new frontier in artificial intelligence research," *IEEE Comput. Intell. Mag.*, 2010.

[207] R. Elshawi and S. Sakr, “Automated Machine Learning: Techniques and Frameworks,” in *Lecture Notes in Business Information Processing*, 2020.

[208] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electron. Mark.*, 2021.

[209] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*. 2019.

[210] K. Shailaja, B. Seetharamulu, and M. A. Jabbar, “Machine Learning in Healthcare: A Review,” in *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, 2018.

[211] A. Mosavi, M. Salimi, S. F. Ardabili, T. Rabczuk, S. Shamshirband, and A. R. Varkonyi-Koczy, “State of the art of machine learning models in energy systems, a systematic review,” *Energies*. 2019.

[212] T. Wuest, D. Weimer, C. Irgens, and K. D. Thoben, “Machine learning in manufacturing: Advantages, challenges, and applications,” *Prod. Manuf. Res.*, 2016.

10 APPENDIX

10.1 APPENDIX A

Model performance (classification)

Grayscale without background removal

Table A1: Confusion matrix of grayscale images with background information.

	Confusion Matrix	Predicted	
		Anthracnose	Healthy
True	Anthracnose	331	14
	Healthy	27	302

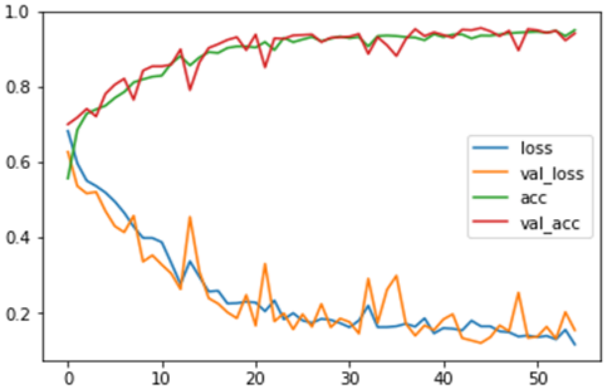


Figure A1: Training and validation loss and accuracy for grayscale images with background information.

Fast Fourier without background removal

Table A2: Confusion matrix of grayscale images applied with FFT and background information.

Confusion	Predicted

	Matrix	Anthracnose	Healthy
True	Anthracnose	358	6
	Healthy	33	277

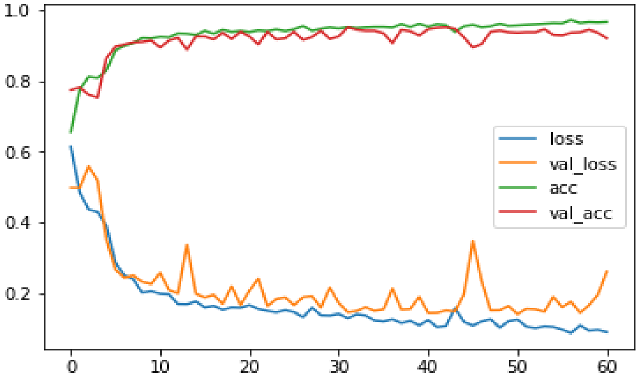


Figure A2: Training and validation loss and accuracy for grayscale images applied with FFT and background information.

RGB without background removal

Table A3: Confusion matrix of RGB images with background information.

	Confusion Matrix	Predicted	
		Anthracnose	Healthy
	Anthracnose	352	7
	Healthy	7	
True			308

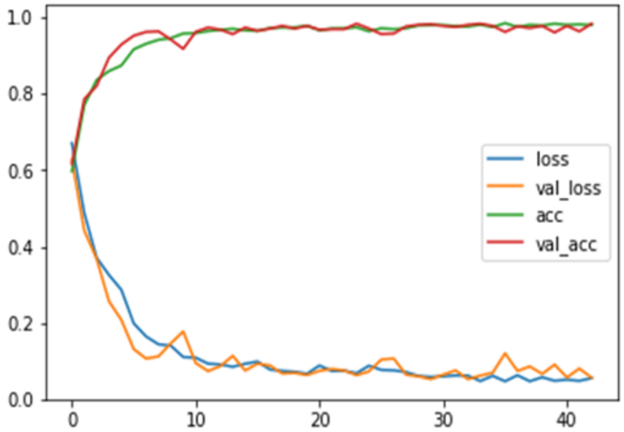


Figure A3: Training and validation loss and accuracy for RGB images with background information.

Grayscale with background removal

Table A4: Confusion matrix of grayscale images with background information.

Confusion Matrix		Predicted	
		Anthracnose	Healthy
True	Anthracnose	343	18
	Healthy	9	304

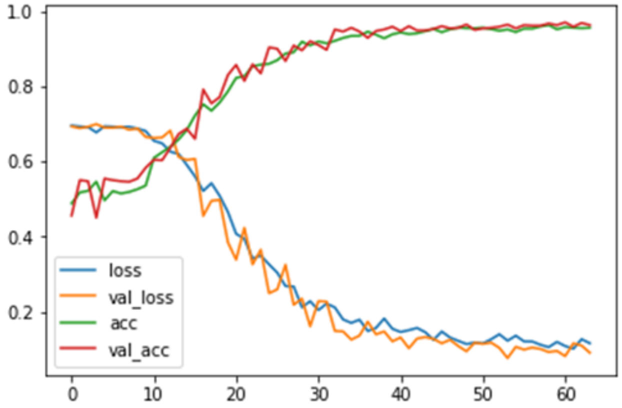


Figure A4: Training and validation loss and accuracy for grayscale images without background information.

Fast Fourier with background removal

Table A5: Confusion matrix of grayscale images applied with FFT and background information.

Confusion Matrix		Predicted	
		Anthracnose	Healthy
True	Anthracnose	344	5
	Healthy	15	310

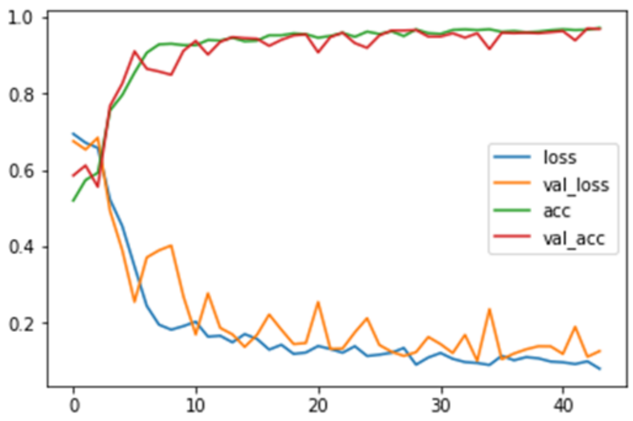


Figure A5: Training and validation loss and accuracy for grayscale images applied with FFT and no background information.

RGB with background removal

Table A6: Confusion matrix of RGB images with background information.

	Confusion Matrix	Predicted	
		Anthracnose	Healthy
True	Anthracnose	366	2
	Healthy	1	305

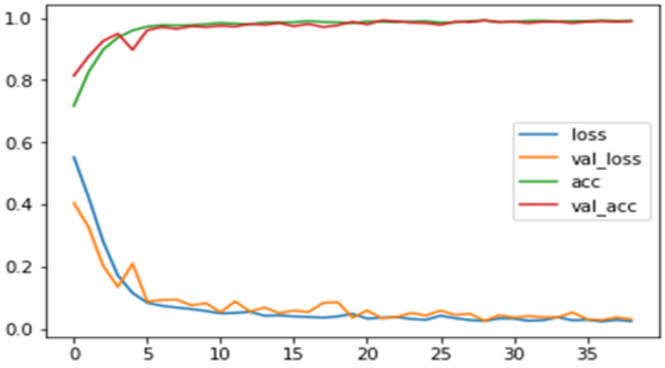


Figure A6: Training and validation loss and accuracy for RGB images with background information.

10.2 APPENDIX B

Proposed CNN Architecture

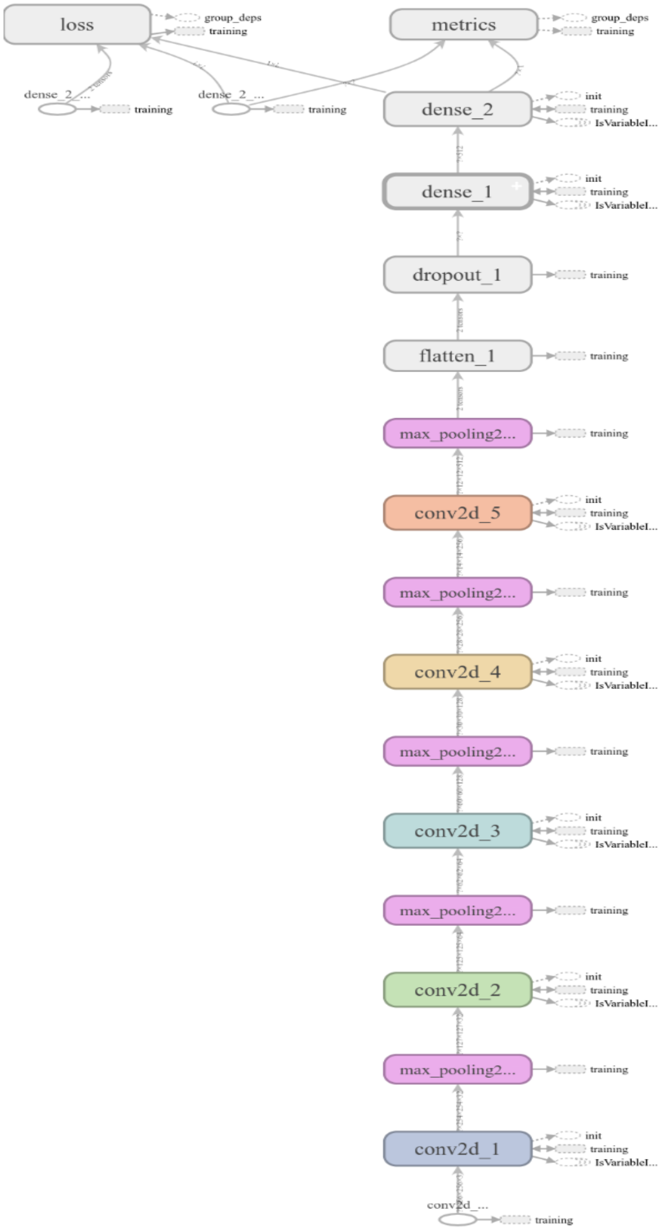
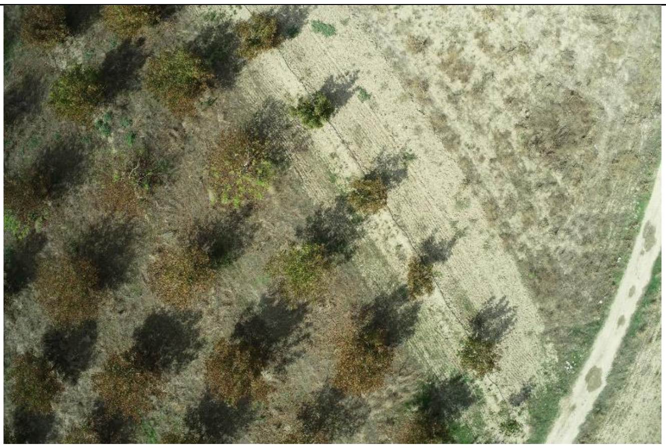
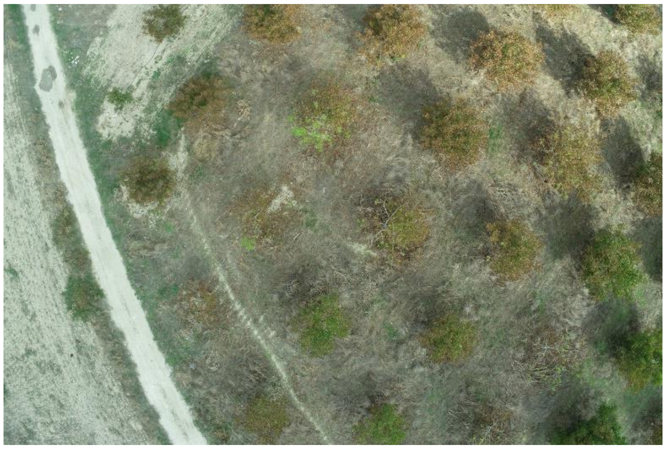


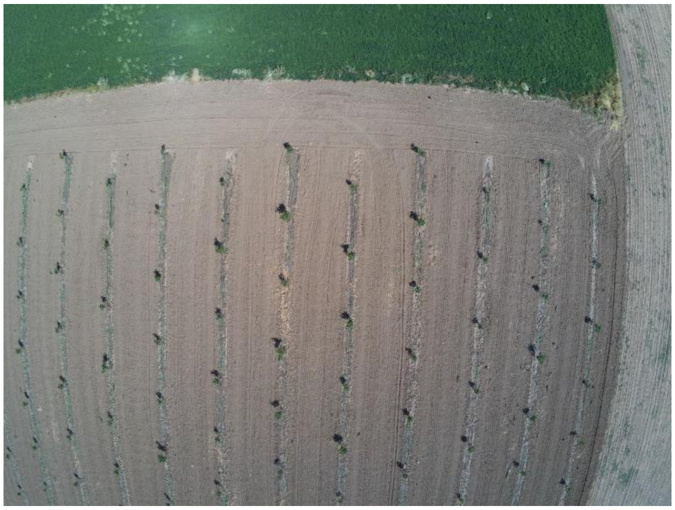
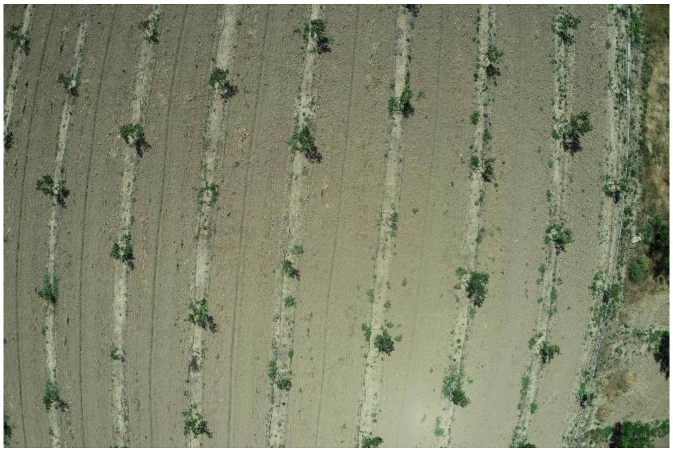
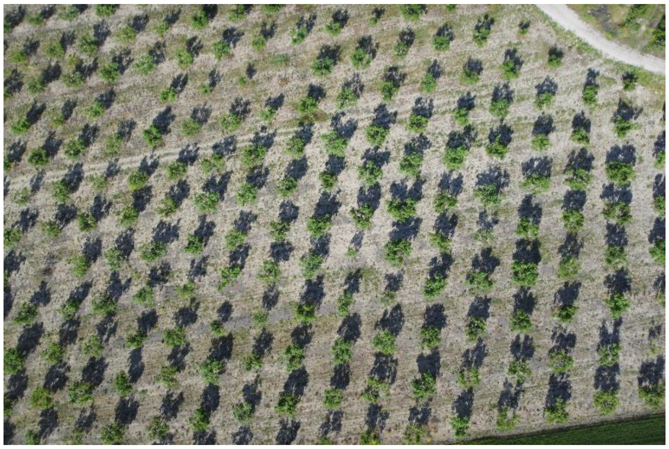
Figure B1: Proposed CNN's architecture.

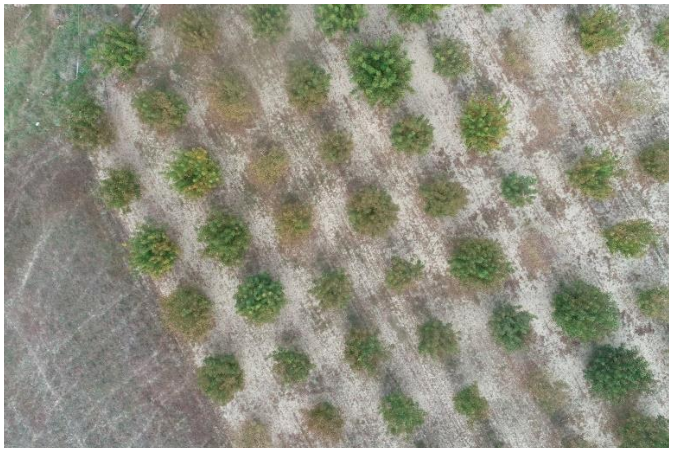
10.3 APPENDIX C

Orchard variability images

Table C1. Sample images of the seven use cases included in the study.

Use Case No.	Conditions	Sample Image
1	Yearly season: Autumn Weeds coverage: Low Canopy size: - Foliage color: Brown Ground color: Brown	
2	Yearly season: Autumn Weeds coverage: Low Canopy size: - Foliage color: Mixed Ground color: Brown	

3	Yearly season: Summer Weeds coverage: Low Canopy size: Small Foliage color: Green Ground color: Brown	
4	Yearly season: Summer Weeds coverage: Low Canopy size: Medium Foliage color: Green Ground color: Brown	
5	Yearly season: Summer Weeds coverage: Low Canopy size: Medium Foliage color: Green Ground color: Mixed	

6	Yearly season: Summer Weeds coverage: Low Canopy size: Large Foliage color: Green Ground color: Brown	
7	Yearly season: Summer Weeds coverage: High Canopy size: Large Foliage color: Green Ground color: Green