REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

PLANT DISEASES DETECTION USING DENSE NET

Munaf Mudheher KHALID

Master's Thesis

Supervisor

Asst. Prof. Dr. Oğuz KARAN

İstanbul, 2023

PLANT DISEASES DETECTION USING DENSE NET

Munaf Mudheher KHALID

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2023

This thesis titled PLANT DISEASES DETECTION USING DENSENET prepared by MUNAF MUDHEHER KAHLID and submitted on 29/12/2023 has been **accepted unanimously** for the degree of Master's in Information Technologies.

Asst. Prof. Dr. Oğuz KARAN
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Oğuz KARAN	Department of Software Engıneerıng. Altınbas University	________________
Assoc. Prof. Dr.sefer KURNAZ	Department of Computer Engıneerıng. Altınbas University	________________
Asst. Prof. Dr.serdar KARGIN	Department Bıomedıcal Engıneerıng. Istanbul Arel University	________________

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

I solemnly affirm that all information and data included in this graduation project were gathered and presented in complete alignment with academic norms and ethical standards. Furthermore, I ensure that any material or conclusions derived from external sources have been properly acknowledged within the text, and all such references cited in the text are correspondingly listed in the Reference List. This declaration is in strict adherence to the requisite academic rules and ethical conduct.

Munaf Mudheher KHALID

Signature

ABSTRACT

PLANT DISEASES DETECTION USING DENSE NET

KHALID, Munaf Mudheher
M.Sc., Information Technologies, Altınbaş University,
Supervisor: Asst. Prof. Dr. Oğuz KARAN
Date: December/2023
Pages: 80

The precise and prompt identification of plant diseases (PD) is paramount in safeguarding crop productivity and fortifying global food security. As deep learning (DL) methodologies continue to evolve, the prominence of automated PD prediction mechanisms has witnessed a significant upswing. This study delves into the formulation of such an advanced system. Here, contemporary architectures, such as CNNs, DenseNet, ResNet, MobileNet, and VGG, are meticulously implemented, evaluated, and then juxtaposed against one another. Amidst these, DenseNet stands out, registering an exemplary ACC of 98%, thereby outstripping other state-of-the-art models. However, beyond mere ACC, our study extends to ensure the interpretability of the model's predictions. By integrating Explainable Artificial Intelligence (XAI) techniques, specifically the Grad-CAM approach, we strive to illuminate the decision-making pathways of the model, granting users a transparent view into its diagnostic process. This not only fortifies trust in the predictions but also provides valuable insights for potential real-world applications. In essence, this research underscores the pivotal role of automation in PD detection and demonstrates the compelling potential of blending DL with XAI for enhanced transparency and efficacy.

Keywords: Plant Diseases, Dataset, Classification, Deep Learning, DenseNet, Explainable Artificial Intelligence, ACC.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

AI	:	Artificial Intelligence
DL	:	Deep Learning
ML	:	Machine Learning
SVM	:	Support Vector Machines
RF	:	Random Forest
TL	:	Transfer Learning
ACC	:	Accuracy
PREC	:	Precision
REC	:	Recall
F1-S	:	F1-Score
PD	:	Plant Diseases
XAI	:	Explainable Artificial Intelligence
SL	:	Supervised Learning
UL	:	Unsupervised learning
RL	:	Reinforcement learning
LR	:	Logistic Regression
KNN	:	K-Nearest Neighbors
NN	:	Neural Network
DT	:	Decision Tree
RF	:	Random Forest
CNN	:	Convolutional Neural Network
CL	:	Convolutional Layer

ReLU	:	Rectified Linear Unit
PL	:	Pooling Layer
FCL	:	Fully Connected Layer
VGG	:	Visual Geometry Group
DenseNet	:	Densely Connected Convolutional Networks
ResNet	:	Residual Network
LRP	:	Layer-wise Relevance Propagation
DNN	:	Deep Neural Network

1. INTRODUCTION

1.1 MOTIVATION

Agriculture has been practiced since the dawn of human civilization, and with the advancement of science, it became apparent that plants were living organisms that could be affected by various diseases caused by microorganisms such as bacteria, viruses, and fungi. These PDs can have a severe impact on crops, and in turn, on human livelihoods and health. The Irish famine of 1840 serves as a stark reminder of the devastating effects of PDs. The late blight of potato, caused by a fungus-like oomycete pathogen called Phytophthora infestans, resulted in the loss of a significant amount of potato crops, leading to the deaths of approximately 1 million people and the immigration of 2 million others [1].

The widespread occurrence of PDs presents a critical challenge to global food security, a fundamental aspect of fulfilling the nutritional needs of the world's population. As highlighted by the Food and Agriculture Organization of the United Nations [2], plants form the cornerstone of human diet, making up 80% of our food consumption. Disturbingly, plant-related pests and diseases are responsible for an enormous reduction of 20 to 40% in worldwide food production. This substantial loss is not just a matter of decreased food availability but also has severe economic implications, resulting in financial losses exceeding $220 billion. This issue is especially acute in areas such as sub-Saharan Africa, where diseases like cassava mosaic disease (CMD) and cassava brown streak disease (CBSD) lead to annual cassava yield losses valued at over 1 billion US dollars, further exacerbating the region's food security concerns [3].

The impact of these agricultural losses extends significantly, especially affecting economies reliant on small-scale and subsistence farming. Plant pests and diseases not only diminish the quantity of available crops but also compromise their quality and taste, which in turn escalates food prices and limits accessibility. Consequently, implementing effective measures for the prevention and control of PDs is crucial for promoting robust plant health and optimizing agricultural yields. To combat PDs, a variety of management strategies can be employed, encompassing cultural, biological, and chemical approaches. Cultural practices, such as rotating crops, maintaining field sanitation, and adhering to proper irrigation and fertilization techniques, play a key role in mitigating disease risks. Biological control, on the other hand, involves leveraging natural predators and parasites to curb PDs.

Chemical methods, including the application of fungicides, bactericides, and other pesticides, are also common. An illustrative example of PD is presented in Figure 1-1. However, it's important to note that excessive reliance on chemical methods can result in environmental harm and the development of resistance in pathogens.

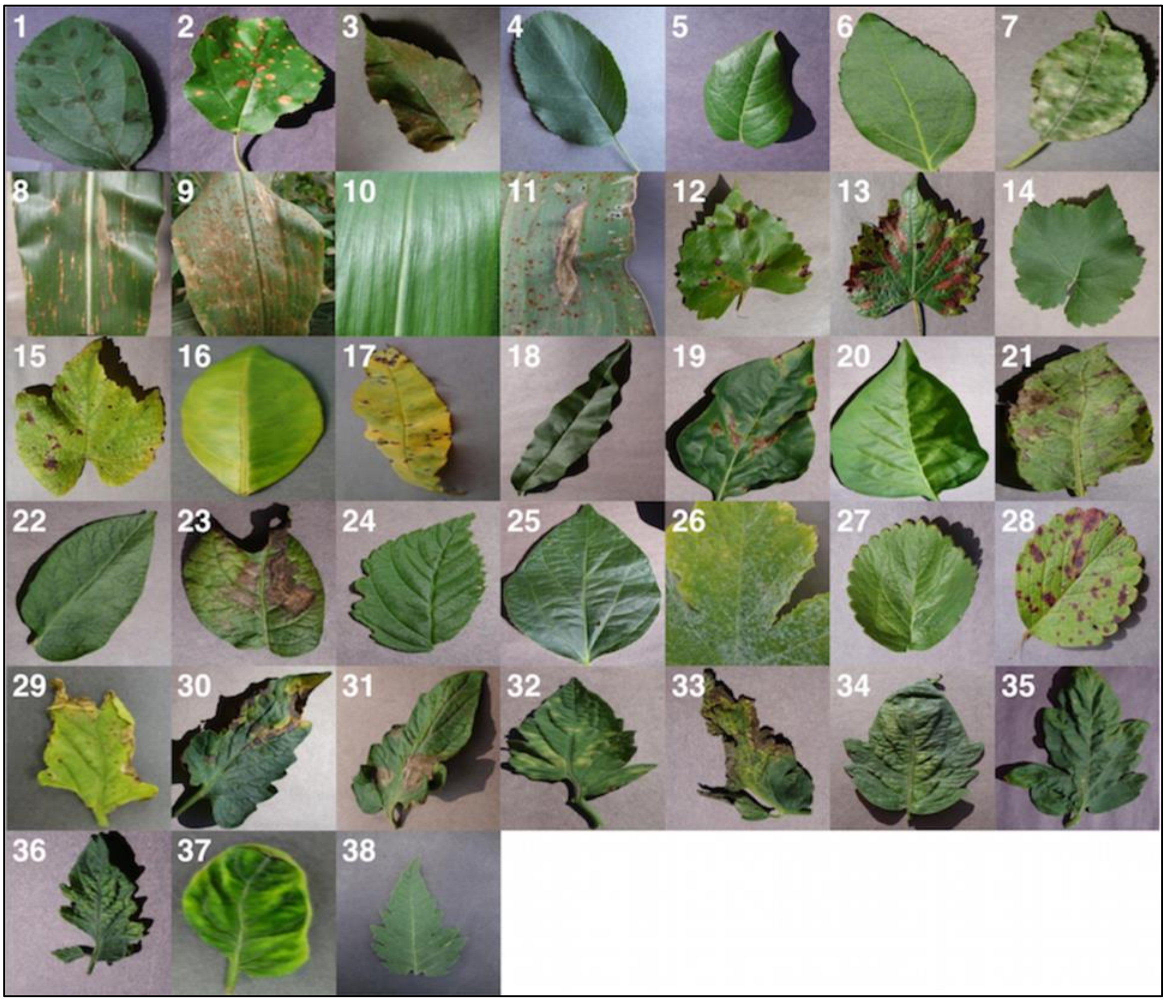


Figure 1.1: An Example of Plant Diseases [4].

In developed nations, the prevalent practice for observing and tracking plant health and field conditions hinges on resource-intensive ground surveys and monitoring endeavors [5]. These initiatives typically encompass yearly collection and monitoring phases, where they gather information about the current state of, for instance, forest ecosystems and scrutinize any deviations spotted. However, the efficiency of such programs can be questionable due to their limited spatial and temporal scope. The challenges are accentuated when considering the manual nature of detecting and diagnosing PDs; it's not only tedious and costly but also mandates specialized expertise [6]. Consequently, less affluent nations grapple with resource constraints and technical limitations, rendering them unable to deploy similar monitoring mechanisms.

In light of the burgeoning global population, there exists an imperative for the agricultural sector to increase food production by a staggering 70% from present benchmarks. The contemporary reliance on chemical agents, encompassing bactericides, fungicides, and nematicides, to counteract plant afflictions poses detrimental repercussions on both the agro-ecosystem and the broader environment. Given these concerns, it becomes paramount to pioneer eco-conscious strategies that facilitate the timely recognition and diagnosis of crop diseases, ensuring that food security remains uncompromised.

Developing cutting-edge solutions for early detection, characterization, and spatial representation of crop diseases promises multifaceted advantages, encompassing cost reduction, damage containment, and streamlined disease management. Proactive identification paves the way for bolstered yield, by ushering in prompt preventative interventions, precluding the disease's proliferation. This preemptive approach is demonstrably superior to curative methods, whose efficacy wanes once the maladies manifest overtly in the crops. Such avant-garde tools empower farmers to institute judicious disease control strategies, sidestepping the rampant employment of pesticides and chemicals which pose threats of groundwater contamination and persistent toxic residues in crop outputs. While techniques for object detection and identification are prevalent for conventional imagery, a conspicuous void persists with respect to automated techniques for discerning and delineating crop diseases via high-definition multispectral aerial imagery, especially within the context of sub-Saharan Africa.

Machine learning (ML) and DL have numerous advantages over traditional methods for diagnosing PDs [7]. These techniques are capable of analyzing vast amounts of data quickly and effectively and can detect even the slightest changes in plant morphology that may be missed by human specialists. Additionally, ML models can be trained to recognize patterns in plant images, allowing for the detection of multiple diseases simultaneously. Despite the potential of ML and DL in PD diagnostics, several challenges must be addressed. One significant obstacle is the limited availability of large-scale labeled datasets that are necessary for training and testing ML algorithms. Furthermore, there is a need to enhance the generalizability of ML models, particularly when dealing with unknown or newly emerged diseases. This research utilizes a DL approach to develop a framework that can ingest multispectral aerial and satellite imagery in near-real time for detecting, identifying, monitoring, and mapping crop diseases in agricultural fields at regional levels.

1.2 PROBLEM STATEMENT

The Food and Agriculture Organization estimates that PDs, insects, and weeds account for roughly 25% of the world's crop losses. In the United States, PDs are responsible for an annual loss of approximately $40 billion in crop yields. In developing countries, smallholder farmers play a crucial role in food production. However, traditional methods of PD detection are costly and time-consuming, making it impractical for smallholder farmers to implement them. As a result, there is a critical need for an accurate and affordable PD diagnosis system that can identify diseases at an early stage, enabling farmers to take preventive measures to control the spread of the disease. With recent advancements in DL technologies and the availability of large-scale datasets, there is potential for developing cost-effective and efficient PD diagnosis systems to achieve this goal.

1.3 RESEARCH HYPOTHESES

Possible research hypotheses for the master's thesis could include:

a. ML algorithms trained on large-scale labeled datasets can accurately detect and identify crop diseases in multispectral aerial and satellite imagery.

This hypothesis suggests that ML can be used to accurately identify crop diseases using multispectral aerial and satellite imagery. The idea is that with a large enough dataset of labeled images, the ML can learn to recognize patterns and accurately classify different types of crop diseases. This research could involve developing and testing different ML algorithms to determine which is most effective in identifying and classifying crop diseases.

b. The use of ML for early detection and mapping of crop diseases can improve agricultural productivity and reduce the need for excessive use of pesticides and chemicals.

This hypothesis suggests that using ML algorithms for early detection and mapping of crop diseases can help farmers identify problems before they become widespread, thus reducing the need for excessive use of pesticides and chemicals. This research could involve testing the effectiveness of different ML on real-world farms and measuring the impact on agricultural productivity and chemical use.

1.4 RESEARCH OBJECTIVES

Our study focuses on the detection and classification of PDs, with an emphasis on not only developing an accurate prediction model but also ensuring that the model's decision-making process is interpretable and transparent. The main goal is to develop an accurate and efficient prediction model that utilizes a combination of ML and DL techniques, while also integrating XAI techniques to make the results comprehensible for end-users, especially for farmers and other non-experts.

To achieve this objective, we intend to construct a varied dataset through the implementation of data augmentation methods, including rotations, translations, and distortions. Such methods are anticipated to enhance the model's ability to generalize and its resilience. Utilizing this enriched dataset, our plan is to both train and evaluate the proposed model. This model will be developed using the Python programming language and will leverage a comprehensive database of plant images. This database will include a range of plant conditions, encompassing both healthy specimens and those impacted by various pests and diseases.

As part of our evaluation, we will compare the performance of our proposed model with traditional methods for PD detection. Beyond accuracy and efficiency, we will also assess the model's interpretability and the clarity of its decision-making process. This will help us identify the key factors that impact not only the model's predictive performance but also its usability in real-world scenarios.

Our study further aims to underscore the importance of using advanced ML techniques in PD detection, specifically highlighting the potential of ensemble DL, ML models, and XAI in the agriculture sector. Ultimately, our goal is to provide a practical, reliable, and transparent tool for farmers and agriculture industry stakeholders. This tool will improve the efficiency and effectiveness of PD detection, reducing losses caused by these diseases, and empowering users with a deeper understanding of the model's predictions.

1.5 THESES ORGANIZATION

This thesis presents a comprehensive approach to utilizing ML methods for PD classification, including the methodologies used in the study. The organization of this report is outlined as follows:

Chapter 1: provides a brief history of PDs and the motivation behind this research.

Chapter 2: previous accomplishments in the field and relevant experiences are discussed.

Chapter 3: covers fundamental concepts of ML and DL technologies, as well as the state-of-the-art our architecture used in this study.

Chapter 4: presents the results of the study, as well as significant discoveries that may benefit other researchers.

Finally, Chapter 5 encapsulates the principal discoveries of the study and proposes directions for prospective research endeavors.

2. LITERATURE REVIEW

2.1 INTRODUCTION

In recent years, there has been significant progress in image processing and ML, leading to the inception of automated mechanisms for detecting and diagnosing PDs. These systems harness digital images of vegetation to discern and diagnose ailments. Features extracted from these images through image processing are subsequently used to feed ML models.

This chapter seeks to delineate prior research associated with PD identification using image processing and ML methodologies. The chapter is structured into five segments: The Prior Research segment delves into existing literature on PD identification leveraging image processing and ML methods. PDs offers insight into various PDs. Image Processing expounds on the methods employed to handle digital plant images. The ML segment elucidates diverse ML strategies tailored for PD recognition. Lastly, the DL and TL section explores the adoption of DL and transfer learning (TL) methods in the realm of PD detection.

2.2 PREVIOUS STUDIES

In contemporary times, ML and DL have demonstrated impressive achievements across diverse domains, spanning from image classification and speech recognition to natural language understanding and medical diagnostics. The agricultural domain is not far behind, tapping into the capabilities of these technologies to address multifaceted challenges, notably in the realm of PD identification. Numerous scholarly endeavors have been directed towards forging ML and DL models adept at PD recognition, harnessing a plethora of image processing methodologies. The primary intent behind these models is to offer an economical and proficient means to pinpoint plant ailments at nascent stages, thereby empowering agriculturists to preemptively combat disease proliferation. This chapter offers a comprehensive overview of the cutting-edge research centered on ML and DL frameworks designed for plant ailment recognition.

[8], proposed a method for identifying brown spot and blast diseases in rice plants using geometrical features such as area, and perimeter of the infected area of the leaves. The study aimed to address the issue of significant yield losses for farmers caused by various diseases that rice plants are susceptible to. The authors utilized a global threshold method and applied a kNN classifier to classify the data containing 330 images of paddy leaves. While the

proposed work achieved a result of 76.59%, the study used ACC as its only metric for evaluating the KNN classifier.

[9], suggested a study with the goal of identifying PDs early on in order to decrease crop loss and boost production efficiency. A deep CNN model trained to classify various PDs according to their symptoms was used in the proposed work. The plant village collection provided 54,323 images of PD and 38 distinct illness categories, which made up the dataset used in this study. The study classified the various PDs using the well-known AlexNet architecture, which has eight layers of learnable characteristics. The model's effectiveness as an early warning or advice tool for PDs is validated by the proposed strategy, which produced an ACC of 96.50%.

[10], aimed to classify and predict PDs using various ML models, including SVM, KNN, RFC, and CNN. Image features such as contrast, correlation, entropy, and inverse difference moments were extracted using the Haralick texture features algorithm, and the extracted features were fed into the different algorithms. CNN directly processed the images as input. The study evaluated the performance of the models on sixteen image categories and found that CNN produced the highest ACC of 97.89%, followed by RFC with 87.43%, SVM with 78.61%, and finally KNN with 76.96%. These results demonstrate the effectiveness of using ML models for detecting and classifying PDs, with CNN performing the best in terms of accuracy.

A CNN-based DL-based method for identifying and categorizing illnesses in tomato leaves was presented by [11]. Three Convolution layers, three Max PLs, and two FCLs make up the suggested model. The study assessed the suggested model's performance and contrasted it with CNN models that have already been trained, including VGG16, InceptionV3, and MobileNet. According to the experimental findings, the pre-trained models scored 77.2%, 63.4%, and 63.75%, respectively, whereas the suggested CNN model outperformed them, achieving a high ACC of 91.2% for the 9 disease and 1 healthy class. Furthermore, the suggested model only needed 1.5 MB of storage space, underscoring the advantages of avoiding pre-trained models.

In this study, [12] put up a methodology for identifying and classifying leaf diseases on tomato plants. The model that is being given combines the CNN method with the Learning Vector Quantization (LVQ) algorithm. 500 images of tomato leaves with four different disease symptoms make up the dataset for the study. The CNN is used to extract significant elements from the images, such as color information, which is crucial for the study of plant leaf diseases. The three channels in this model are subjected to filters based on RGB components. The LVQ is then given the output feature vector from the convolutional component to train the network. The experiment's findings demonstrate that the four different forms of tomato leaf diseases are successfully recognized using the suggested method.

[13], presented a novel method for PD recognition based on deep neural networks and leaf image categorization. With PRE ranging from 91% to 98%, the suggested model was able to identify 13 distinct types of PDs after being trained on the DL architecture known as Caffe. This study's methodology is a novel approach to training, and the publication includes a detailed description of every step needed to put this illness recognition model into practice.

In this study, [14] suggested a method for the automated identification and categorization of different ailments in mango trees. Among the 1200 images of both healthy and diseased mango leaves in the dataset used in this framework, five significant leaf diseases were identified: Anthracnose, Alternaria leaf spots, Leaf Gall, Leaf webber, and Leaf burn. With a 96.67% ACC, the suggested CNN model demonstrated that real-time applications are possible.

In their study, [15] embarked on a comprehensive exploration of the primary determinants shaping the architecture and performance of deep neural networks in plant pathology. Their methodological approach involved an exhaustive review of existing literature coupled with empirical tests on an image database encompassing close to 50,000 images. This database was meticulously curated to mirror authentic field scenarios. Through their investigation, the authors pinpointed multiple elements influencing the proficiency of deep neural networks in discerning and categorizing various plant ailments. Their insights offer a grounded perspective, fostering more informed conclusions on the topic. [5], concentrated on creating a framework for the detection and categorization of various diseases in rice plants using CNN architecture and SVM. The dataset, which includes 619 images of sick rice plant leaves in each of the four categories, was processed using the framework. When the accuracies were evaluated for various ratios of training and testing datasets, the highest ACC of 91.37% was attained.

In their research, [16] evaluated TL strategies, such as VGG-16 and VGG-19, in addition to proposing specific CNN configurations for Olive PDs. Their framework utilized a dataset comprising 3,400 images of olive plant leaves. To enhance the breadth of the dataset, they incorporated a data augmentation technique. Preliminary ACC hovered around 88%, but post-data augmentation, it witnessed a significant rise to roughly 95%.

The proposed study by [17] uses ML algorithms to categorize diseases; it mainly applies to data and automatically ranks the ACC of all those activities. This proposed work demonstrates the various stages where general identification of common PDs and use of ML algorithms on comparative research work are for PD identification and classification, including: acquiring image dataset, processing and cleaning the obtained data which is present in the image dataset, segmenting of image data in the image dataset, extraction of essential features, and classification of input images. The proposed work shows that CNN is highly accurate when there are numerous diseases present.

Aiming to amplify crop yields, [18] devised a system rooted in DL paradigms which provides consistent monitoring across various plant growth stages, ranging from the initial seedling phase to the final yield stage. Their proposed mechanism was tested using a dataset containing 571 images, capturing tomato leaves and the fruit in diverse developmental stages. The performance metrics revealed that the inception-ResNet v2 achieved an ACC of 87.27%, while the autoencoder recorded an ACC of 79.09%. This research underscores the transformative potential of TL in discerning and classifying PDs through leaf imagery.

[19] put forth a novel approach to disease diagnosis based on an artificial intelligence-based system that gathers a variety of biomedical data from homogeneous distributed sensors. To assess the relationships between various biomedical data for disease identification and diagnosis, the system combines a variety of DL architectures with ensemble learning and attention processes. The effectiveness of the suggested approach was thoroughly tested using biological data. The findings show that using DL technologies to medical artificial intelligence for disease diagnosis in healthcare decision-making processes has considerable advantages. For instance, the proposed methodology significantly outperformed higher-level illness detection models with a disease detection rate of 92%.

[19] introduced an innovative automatic technique for detecting PDs by harnessing deep ensemble neural networks (DENN). This approach capitalizes on refining pre-existing models through TL while simultaneously employing data augmentation strategies to mitigate

overfitting issues. When subjected to evaluation using the widely recognized plant village dataset, the proposed model showcased superior performance in comparison to established pre-trained architectures, including ResNet 50 and 101, InceptionV3, DenseNet 121 and 201, MobileNetV3, and NasNet. The assessment underscored the model's proficiency in distinguishing diverse PDs, consistently surpassing the capabilities of the aforementioned pre-trained frameworks.

[20], presented a framework that is capable of categorizing a wide range of leaf diseases that might affect plants and fruits, particularly in the critical feature extraction stage. This suggested approach extracts characteristics from an improved deep TL model by means of model engineering. They use several SVM models in order to improve feature differentiation and maximize processing performance. In addition, the RBF kernel values are carefully chosen depending on the model that is being used during training. They made use of the UCI and PlantVillage datasets to verify the efficacy of their approach. Six leaf picture sets showing both healthy and sick stages in apple, maize, cotton, grape, pepper, and rice make up the combined 90,000 images in these databases.

In their research, [21] advocated for the ensemble CNN approach as an effective tool in addressing the intricacies of plant leaf identification. They implemented and trained five distinct CNN models. These models underwent ensemble strategies based on each model's CNN output probabilities. Three different datasets were used to examine this methodology: the mulberry leaf, tomato leaf disease, and corn leaf disease datasets. MobileNetV2 performed the best of the separate models, achieving 91.19% ACC on the mulberry leaf dataset. In contrast, the Xception model achieved an amazing ACC of 91.77% when used with data augmentation methods like Vertical Flip, Height Shift, and Fill Mode. Surprisingly, when tested on the leaf disease datasets, DenseNet121 and Xception both had ACC rates higher than 99%. By using the weighted average ensemble strategy, the researchers were able to forecast the output of each CNN by selecting only the best-performing models from the ensemble.

Table 2.1: Related Works.

Refe	Methods	Datasets	Results
Suresha et al. (2017) [8]	Geometrical features, global threshold method, kNN classifier	330 images of paddy leaves	76.59% ACC
Madhulatha et al. (2020) [9]	Deep CNN model, AlexNet architecture	54,323 images of PDs, 38 categories from plant village dataset	96.50% ACC
Hatuwal et al. (2020) [10]	SVM, KNN, RFC, CNN, Haralick texture features algorithm	16 image categories (specifics not detailed)	CNN: 97.89%, RFC: 87.43%, SVM: 78.61%, KNN: 76.96%
Agarwal et al. (2020) [11]	CNN with three Convolution layers, Max PLs, FCLs, VGG16, InceptionV3, MobileNet	N/A	Proposed CNN: 91.2%
Sardogan et al. (2020) [12]	CNN and Learning Vector Quantization (LVQ) algorithm	500 images of tomato leaves	Successful identification of four different diseases
Sladojevic et al. (2016) [13]	Deep CNN using Caffe DL architecture	N/A (specifics not detailed)	91% - 98% ACC for 13 types of PDs
Arivazhagan et al (2018) [14]	CNN	1200 images of healthy and diseased mango leaves	96.67% ACC
Barbedo et al (2018) [15]	Review and empirical tests, CNN and SVM	50,000 images and 619 images of diseased rice plant leaves	91.37% ACC (for rice plant dataset)
Uğuz et al. (2021) [16]	CNN, VGG-16, VGG-19, Data augmentation	3,400 images of olive plant leaves	Pre-augmentation: ~88%, Post-augmentation: ~95%

Table 2.2: Related Works 'Table Continued'.

Shruthi et al. (2019) [17]	ML algorithms, CNN	N/A	High ACC for multiple diseases (specifics not detailed)
Tran et al. (2019) [18]	Inception-ResNet v2, Autoencoder	571 images of tomato leaves and fruits	Inception-ResNet v2: 87.27%, Autoencoder: 79.09%
Djenouri et al. (2022) [19]	AI-based system with ensemble learning and attention processes, DENN	Biological data (specifics not detailed) and plant village dataset	92% disease detection rate; superior to pre-trained models
Saberi Anari et al. (2022) [20]	Deep TL model, SVM with RBF	PlantVillage and UCI datasets with approx. 90,000 images	N/A (success in classification but specifics not detailed)
Chompookham et al (2021) [21]	Ensemble CNN, five distinct CNN models with strategies like unweighted average, weighted average	Mulberry leaf, tomato leaf disease, and corn leaf disease datasets; MobileNetV2 and Xception models	MobileNetV2: 91.19%, Xception with augmentation: 91.77%, DenseNet121 and Xception > 99%

2.3 PLANT DISEASE

2.3.1 Phytopathology

Phytopathology is a multifaceted field that delves into the study of PDs, merging insights from a plethora of scientific disciplines. This domain draws knowledge from subjects like biology, chemistry, ecology, genetics, among others. Phytopathologists delve into the exploration, pinpointing, and management of plant ailments stemming from a variety of causes, be it infectious agents like fungi, bacteria, viruses, and nematodes, environmental stressors like drought, heat, and pollution, or genetic irregularities.

Phytopathology researchers and practitioners employ a comprehensive approach that combines traditional and modern techniques, including molecular biology, genetic engineering, and immunology, to better understand the complex interplay between plants, pathogens, and the environment [22], as shown in figure 2-1.

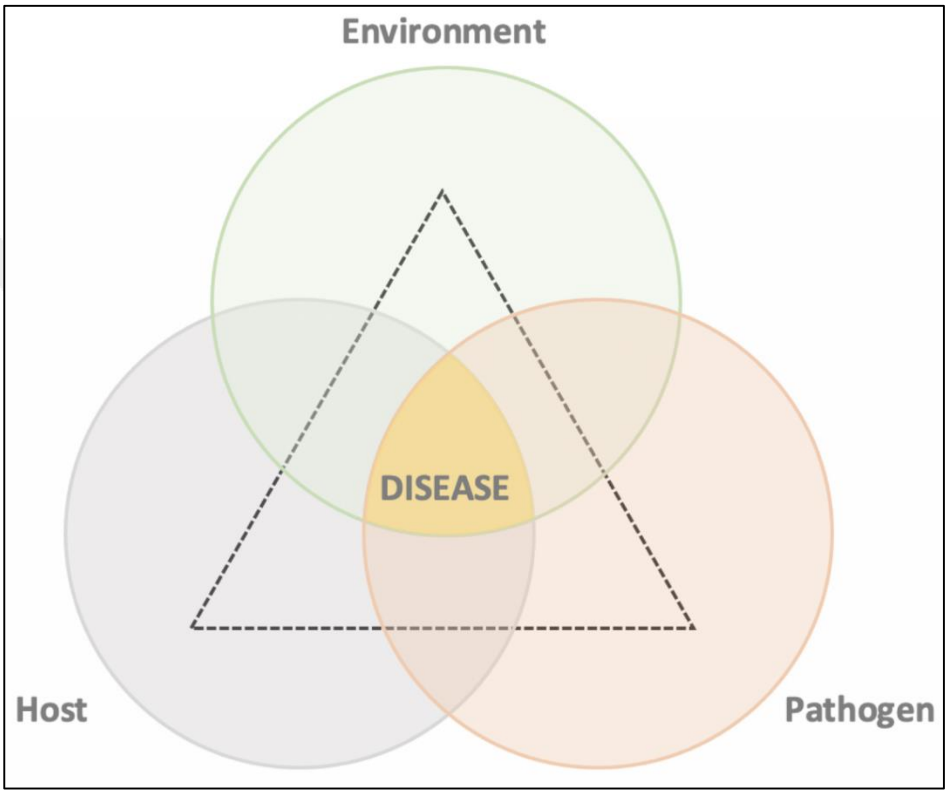


Figure 2.1: Plant Disease Triangle [23].

The PD Triangle is a conceptual framework used to illustrate the complex interactions that occur between three components that are necessary for the occurrence of a PD. These three components are the host plant, the pathogen, and the environment.

The host plant is the organism that is being infected, and it can be any plant species or cultivar. The pathogen is the disease-causing organism, which can be a virus, bacterium, fungus, or nematode, among others. The environment refers to the physical and biological factors that influence the interaction between the host and the pathogen, such as temperature, humidity, soil conditions, and the presence of other organisms.

According to the PD Triangle, the occurrence of a disease requires the presence of all three components in a favorable arrangement. If any of the three components are missing or not optimal, the disease will not occur or will be less severe. For instance, if the environment is unfavorable for the pathogen's growth and reproduction, even if the host is present and susceptible, the disease may not develop. Similarly, if the pathogen is absent or not virulent, the host may not develop the disease, even in a favorable environment. The PD Triangle is a useful tool for understanding the complex interactions that occur between plants, pathogens, and the environment.

Phytopathologists work to develop techniques to prevent, manage, and mitigate the consequences of PDs on crops, forests, and other plant-based systems. They are vital in ensuring the health and productivity of plants and ecosystems and have become increasingly important in recent years due to globalization, emergence of new diseases, and the need for sustainable agricultural practices to feed a growing global population while preserving natural resources.

The demand for skilled plant pathologists is on the rise, with opportunities in academia, government organizations, and the commercial sector. As a result, the field of phytopathology has grown in significance, and its importance will continue to increase as we face new challenges and seek innovative solutions to safeguard the health of plants and ecosystems.

2.3.2 The Impact of Plant Diseases on Agricultural Production and Food Security

PDs can significantly damage crops and agricultural production. The extent of damage, varying from mild to severe, depends on the disease's severity and the affected plant type.

In some cases, PDs can cause complete crop failure, resulting in significant economic losses for farmers and food shortages for consumers [24]. For example, a disease outbreak in a major crop such as wheat, rice, or corn can lead to global shortages and increased prices for food products.

Furthermore, plant ailments can diminish both the quality and quantity of crop outputs, which can subsequently drive down their market value, impacting farmers' earnings. This can influence the accessibility and cost-effectiveness of food for consumers, particularly in developing nations where farming is a primary livelihood and vital for food provision.

Furthermore, managing PDs often requires significant resources, including labor, equipment, and the use of pesticides or other control methods. These costs can add up quickly, and may be a major financial burden for farmers, especially small-scale farmers with limited resources.

2.3.3 Plant Pathogens

Plant pathogens are organisms that cause diseases in plants. These pathogens can be viruses, bacteria, fungi, nematodes, or other microorganisms that infect and damage various parts of plants such as roots, stems, leaves, and fruits [25]. PDs caused by these pathogens can reduce crop yields, lower the quality of produce, and cause economic losses to farmers and agricultural industries.

Fungi:

Fungi are eukaryotic organisms, which include yeasts, molds, and mushrooms. They obtain their nutrients by absorbing organic matter from the environment, as they lack chlorophyll and cannot perform photosynthesis. Fungi have a unique chitin cell wall and reproduce through spores, both sexually and asexually. They play a crucial role in the ecosystem by decomposing dead organic matter and forming symbiotic relationships. Fungi can be found in various habitats and have medicinal properties, but can also cause diseases in plants, animals, and humans.

Figure 2.2: Fungi [26].

Bacteria:

Bacteria are single-celled microorganisms found in various environments and can adapt to their surroundings. They reproduce through binary fission and can be spherical, rod-shaped, or spiral-shaped. Bacteria are crucial to the carbon and nitrogen cycles, aiding in decomposing organic matter and recycling nutrients. They have applications in biotechnology and medicine, including genetic engineering. However, some bacteria can cause diseases in plants, animals, and humans. Bacteria can be classified into groups based on their characteristics, such as their cell wall structure and ability to survive with or without oxygen.

Figure 2.3: Bacteria.

Viruses:

Plant viruses are small, infectious agents that require living host cells to reproduce and spread, causing a range of symptoms that reduce crop yield or quality. They can be transmitted through insect vectors, contaminated tools or equipment, and infected plant material. Strategies for managing plant viruses include cultural practices, chemical control measures, and plant breeding for resistance.

Figure 2.4: Viruses.

2.4 IMAGE PROCESSING

Digital image processing refers to the use of computer algorithms to manipulate digital images by performing various operations such as enhancement, transformation, or analysis. It involves the application of mathematical operations to digital images to extract valuable information or modify the image in some way.

Digital image processing encompasses a sequence of standard procedures including image capture, pre-processing, image enhancement, feature extraction, and image categorization. In the image acquisition phase, a digital snapshot is taken using cameras or other related imaging devices. The pre-processing step aims to elevate the image's quality by eliminating noise, rectifying distortions, and modifying the contrast and brightness of the image.

In order to facilitate the interpretation of images, image enhancement techniques are employed to sharpen edges, reduce blur, and enhance contrast. Feature extraction is the process of identifying and isolating relevant features in an image, while picture classification involves categorizing images based on their features or other criteria.

2.4.1 Types of Digital Images

Digital images come in many different types, each with its own unique characteristics and applications. Four common types of digital images are indexed images, grayscale images, RGB images, and binary images.

Indexed images: are images that use a limited number of colors, usually 256 or fewer, from a pre-defined color palette. This makes them efficient to store and transmit, making them useful for graphics in web design, computer icons, and simple images.

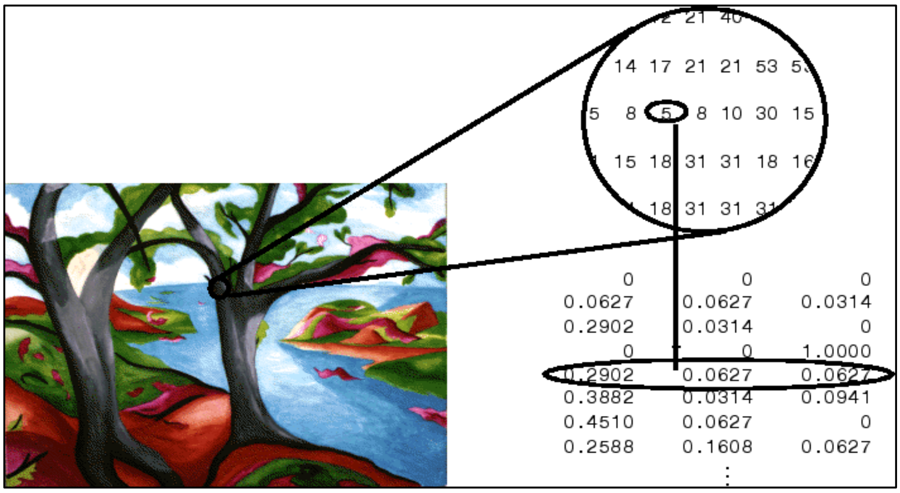


Figure 2.5: Indexed Images.

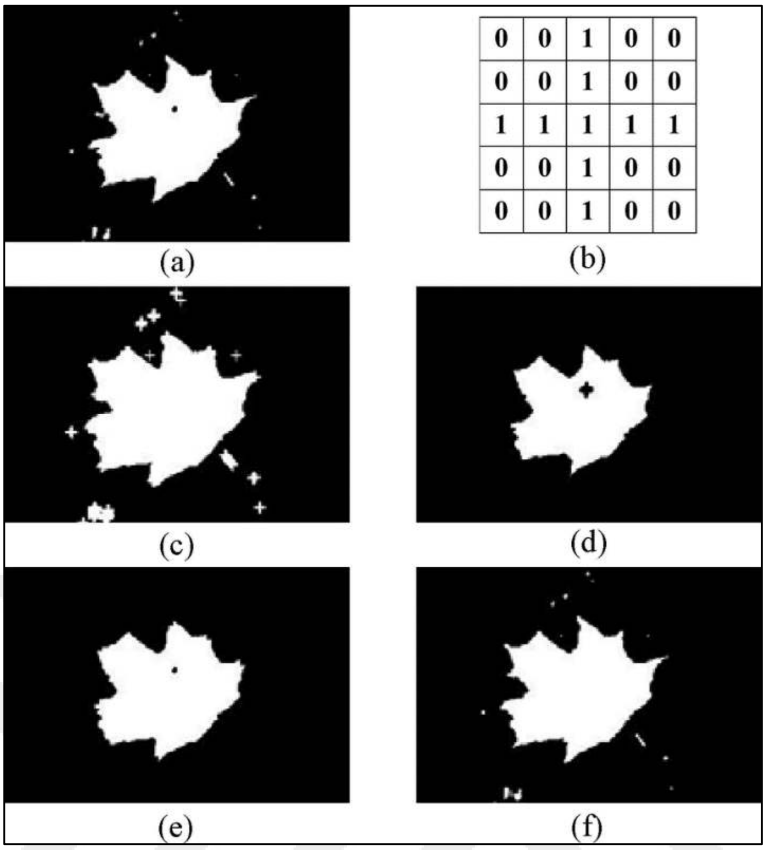
Grayscale images: are composed of shades of gray, ranging from white to black, and are often used for medical imaging, scientific visualization, and artistic photography. These images can show more detail in bright and dark areas compared to color images.

Figure 2.6: Grayscale Images.

RGB images: are composed of three primary colors (red, green, and blue) and are used for digital photography and color graphics. RGB images can create millions of color variations, providing greater color detail and accuracy.

Figure 2.7: RGB Images.

Binary images: are images that only have two colors - black and white - and are often used in image processing applications, such as edge detection or image segmentation. They are also commonly used in optical character recognition (OCR) and barcode scanning.



0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

Figure 2.8: Binary Images.

Understanding the different types of digital images is important for selecting the appropriate format for a particular application, as well as for understanding the processing and analysis techniques that can be applied to them.

2.4.2 Digital Images Techniques

Digital image processing techniques involve a variety of algorithms and methods used to enhance, transform, or analyze digital images. Digital image processing techniques can be broadly classified into three groups: image enhancement, image restoration, and image compression.

Image enhancement: is a digital image processing technique used to improve the quality of an image by adjusting its brightness, contrast, sharpness, or color balance. Image enhancement methods are frequently used to improve the visibility of important aspects in an image or to correct issues such as poor lighting, low contrast, or color distortion [51]. There are several image enhancement techniques available, each designed to address specific flaws or qualities of an image.

One common image enhancement technique is histogram equalization, which stretches the contrast throughout the entire range of intensity levels by adjusting the image's histogram. This can help to enhance the visibility of previously difficult-to-see features due to poor contrast.

Contrast stretching is another technique that stretches the intensity range of a picture to increase contrast, often by adjusting the brightness and contrast of specific portions of the image.

Sharpening is a technique that enhances the perceived sharpness of an image by boosting the image's edges and fine details.

Noise reduction is another image enhancement technique that eliminates or decreases the amount of noise in an image, which can help to improve image quality, especially in photographs taken in low light or at high ISO.

Color correction is a technique used to adjust the color balance of an image to correct flaws such as color casts or inaccurate white balance.

Image blending is a technique used to combine multiple photographs of the same scene to produce a composite image with increased clarity and detail.

Figure 2.9: Image Enhancement.

Image restoration: is a digital image processing technique that aims to enhance the quality of an image by reducing noise, blur, or any other distortions. The main goal of image restoration is to recover as much of the original image as possible while minimizing any degradation that might have occurred during image capture, transmission, or storage [27].

Various approaches are used to restore images, each tailored to address specific types of distortions.

One of the most common techniques for image restoration is filtering, which utilizes a filter to remove or enhance certain visual elements in an image, such as a median filter that can eliminate salt and pepper noise or a Gaussian filter that can remove blurriness.

Deconvolution is another approach that aims to mitigate the effects of blur or other distortions using a mathematical model of the degradation process. This technique can estimate the original image if the image has been blurred by a known point-spread function (PSF).

Superresolution is an approach that combines multiple low-resolution images of the same scene to enhance the spatial resolution of the image beyond its original resolution. This can be achieved using techniques like interpolation or neural networks to estimate a high-resolution image from low-resolution inputs.

Inpainting is yet another approach that involves filling in missing or damaged regions of an image, such as scratches, holes, or occlusions. This is done using methods such as diffusion or patch-based techniques to approximate the missing information.

DL is a technology that employs neural networks to learn complex image restoration tasks such as noise removal and information restoration. DL algorithms can be trained on vast datasets of damaged and restored images, producing state-of-the-art results on a variety of image restoration tasks.

Image compression: is a vital digital image processing technique that reduces the size of an image file while preserving its quality as much as possible. This technique is commonly used in various applications such as image storage, transmission, and display, where large image files can be costly due to storage or bandwidth constraints [28].

Image compression techniques are classified into two main types:

a. Lossless compression allows the compressed image to be perfectly reconstructed from the compressed data, without any loss of information. This type of compression is frequently used in situations where preserving the original image data is essential, such as in medical imaging or scientific data processing. Lossless compression techniques include Run-length Encoding (RLE), Huffman coding, and Arithmetic coding, among others.

b. Lossy compression involves losing some information during the compression process to achieve a higher compression ratio. This type of compression is often used in situations where reducing the image size is more critical than retaining all of the original image details. Typical lossy compression algorithms include Discrete Cosine Transform (DCT), Wavelet Transform, and Fractal compression. These algorithms work by identifying and removing redundancies or irrelevant information in the image, resulting in a smaller file size with some loss of detail.

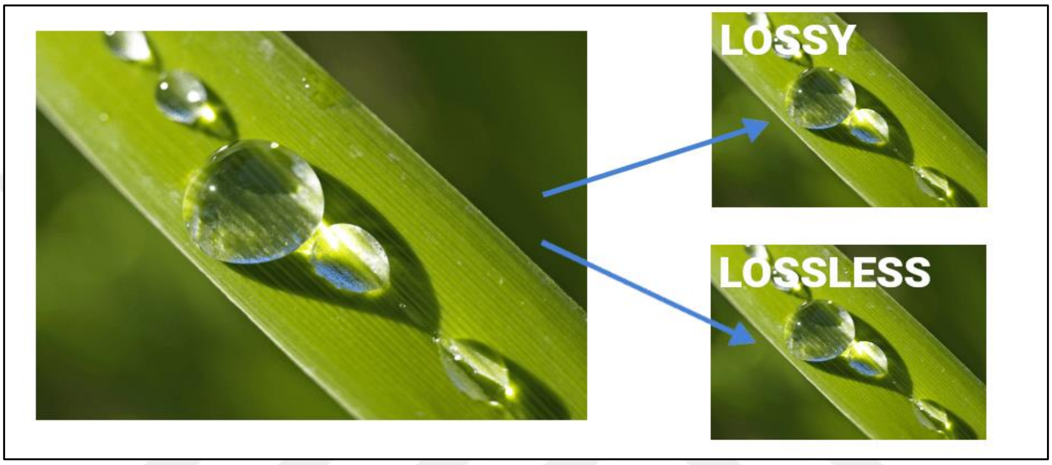


Figure 2.10: Lossless Image vs Lossy Image Compression.

2.5 MACHINE LEARNING (ML)

ML is a field of computer science and AI that focuses on building systems that can learn and improve from experience without being explicitly programmed. It involves developing algorithms and statistical models that enable computers to automatically improve their performance on a specific task by learning from data. ML can be broadly classified into three main types (Figure 2.11).

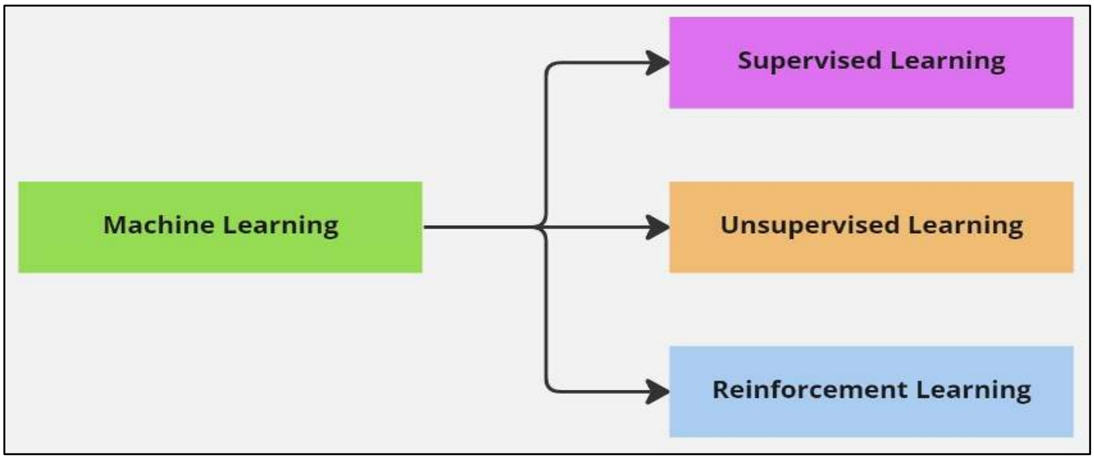


Figure 2.11: Machine Learning Types.

2.5.1 Machine Learning Types

Supervised learning (SL): is an ML approach in which a model learns from a dataset with predefined labels, associating specific outputs with each corresponding input. The model's main aim is to discern patterns within the training data, creating a linkage between the input and output variables. After the training phase, the model can make predictions on novel, unseen data [29].

The essence of SL is to utilize labeled data to effectively train the model, enabling it to accurately predict outcomes for new data. This process involves modifying the model's parameters based on discrepancies between its predictions and the actual labels in the training set. The goal is to reduce this discrepancy, which is often accomplished by minimizing a defined cost function. Figure 2-12 illustrates the supervised learning procedure.

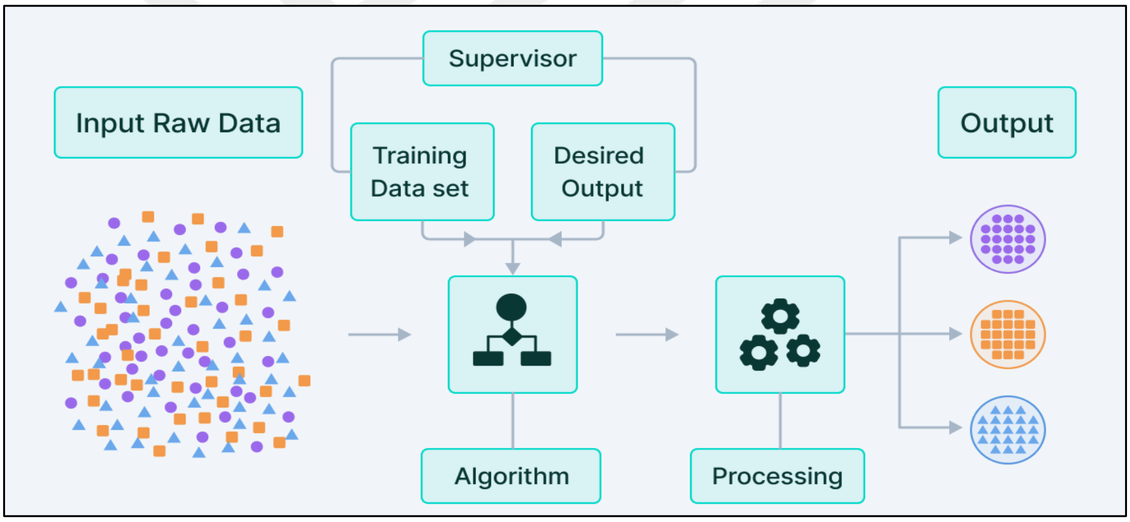


Figure 2.12: Supervised Learning.

Unsupervised learning (UL): Training a model on an unlabeled dataset without a defined output or objective value for each input is known as UL in ML [30]. In contrast to supervised learning, UL focuses on finding connections, patterns, or organization in data without knowing ahead of time what the expected outcomes should be. UL usually entails giving the model a dataset and giving it instructions to look for hidden structures or patterns in the data. UL frequently uses two methods: dimensionality reduction, which lowers the number of features or variables in the dataset while maintaining pertinent information, and clustering, which groups together comparable data points.

UL is especially useful when working with large, complex datasets with no predetermined outputs or when manual labeling of the data is impractical or impossible. For instance, UL can be used for anomaly detection, market segmentation, and image segmentation, among other applications. Clustering is a key process in UL, where the unsupervised algorithm works with unlabeled data to create groups or clusters, allowing the model to categorize incoming input data into a given cluster. Figure 2.13 illustrates the UL procedure.

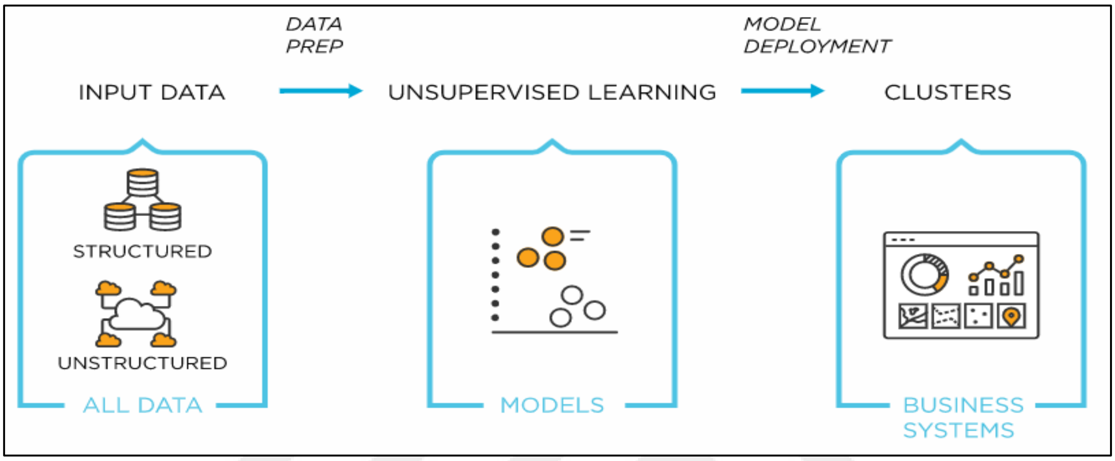


Figure 2.13: Unsupervised Learning.

Reinforcement learning (RL): is a ML approach that allows an agent to make decisions based on the feedback it receives from its actions in the form of rewards or penalties [31]. The goal of RL is to maximize the cumulative reward over time by making the best possible judgments through trial and error. The model interacts with its environment by executing actions, which affect its state, and the environment provides feedback in the form of incentives or penalties that reflect the quality of the agent's activities.

To create a policy that connects the present state of the environment to the action that maximizes the cumulative reward over time, the RL model must learn by trial and error, which can be time-consuming and inefficient. To solve this, RL algorithms use tactics like exploration and exploitation to balance the discovery of new behaviors with the application of prior information.

When there are numerous feasible actions in a broad state space or when it is challenging to create a clear target, RL is especially helpful. RL is utilized in many industries, including gaming, robotics, and self-driving automobiles. Figure 2-14 shows how the RL process works.

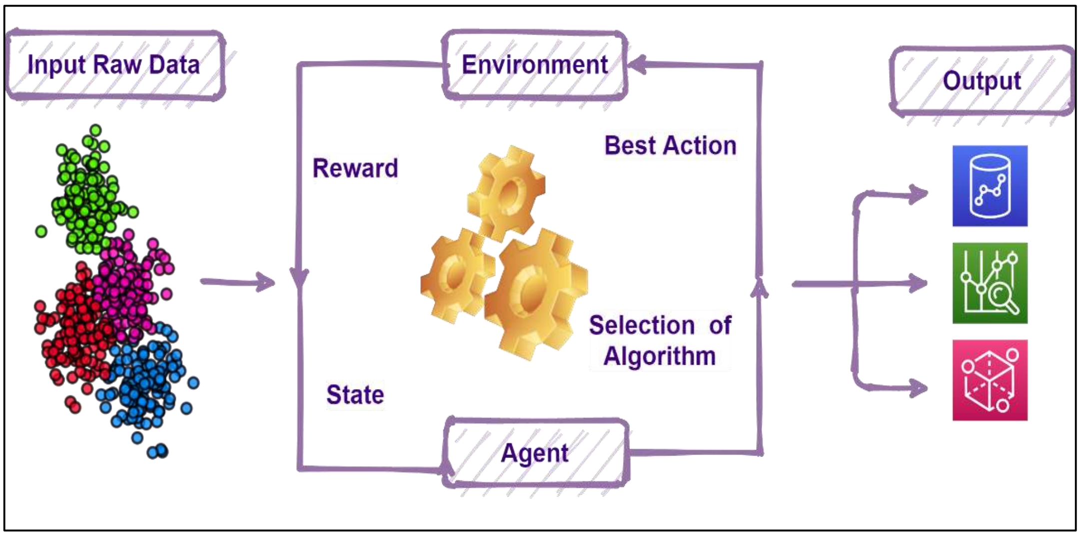


Figure 2.14: Reinforcement Learning.

2.5.2 Machine Learning Techniques

Logistic regression (LR): is a widely-used statistical method primarily tailored for binary classification tasks. It falls under the generalized linear model category and estimates the likelihood of a particular event based on various input predictors.

LR works by taking a set of input variables and applying a linear transformation to them. The resulting value is then passed through a sigmoid function, which maps the value to a probability between 0 and 1. The sigmoid function has an S-shaped curve, which ensures that the output is always between 0 and 1 [32].

The equation for logistic regression is as follows:

$$P(y=1 \mid x) = 1 / (1 + e^{-(\beta 0 + \beta 1x1 + \beta 2x2 + ... + \beta n*xn)}) \tag{2.1}$$

Where: $P(y=1 \mid x)$ is the probability of the event occurring given the input variables.

x1, x2, ..., xn are the input variables.

β0, β1, β2, ..., βn are the coefficients of the linear transformation.

The coefficients are estimated using maximum likelihood estimation, which involves finding the values of β0, β1, β2, ... βn that maximize the likelihood of the observed data. This is done by iteratively adjusting the coefficients until the model converges to a set of values that provide the best fit to the data.

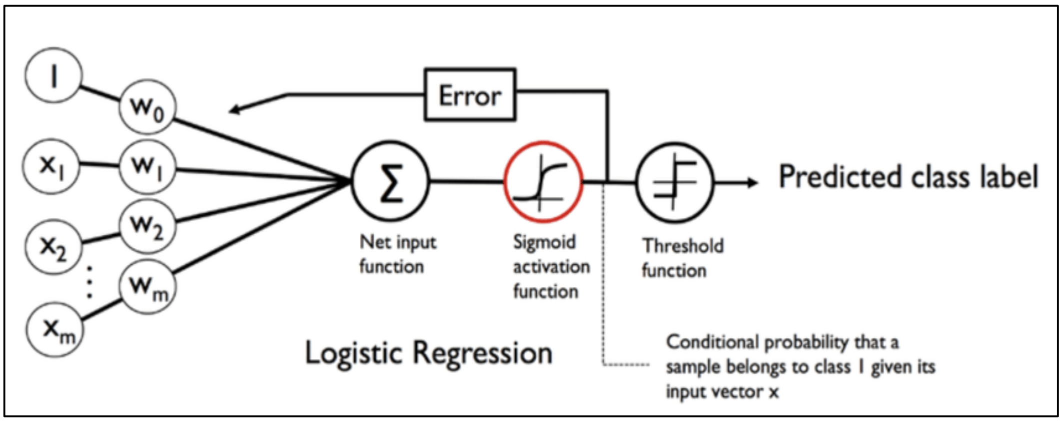


Figure 2.15: Logistic Regression [33].

Support Vector Machines (SVMs): stand out as a robust ML method commonly employed for both classification and regression [34]. The fundamental principle behind SVM is identifying an optimal boundary, known as the hyperplane, to distinguish data into distinct categories. The selected hyperplane seeks to maximize the margin, which is the distance between the nearest data points from contrasting classes. A larger margin enhances the model's generalization capability and its resilience to data noise.

SVM predominantly features two core elements: the kernel function and the optimization technique. The kernel's role is to project input data into a higher-dimensional space, making the task of discovering a separating hyperplane more feasible. Linear, polynomial, and radial basis function (RBF) kernels are among the most prevalent choices.

SVM's adaptability allows it to cater to both linear and non-linear datasets. Its efficacy is evident in diverse applications, including image recognition, text categorization, and bioinformatics. Nonetheless, SVM can be resource-intensive, particularly with vast datasets or intricate kernel methods. Hence, several optimization strategies and approximation approaches have been introduced to mitigate these limitations.

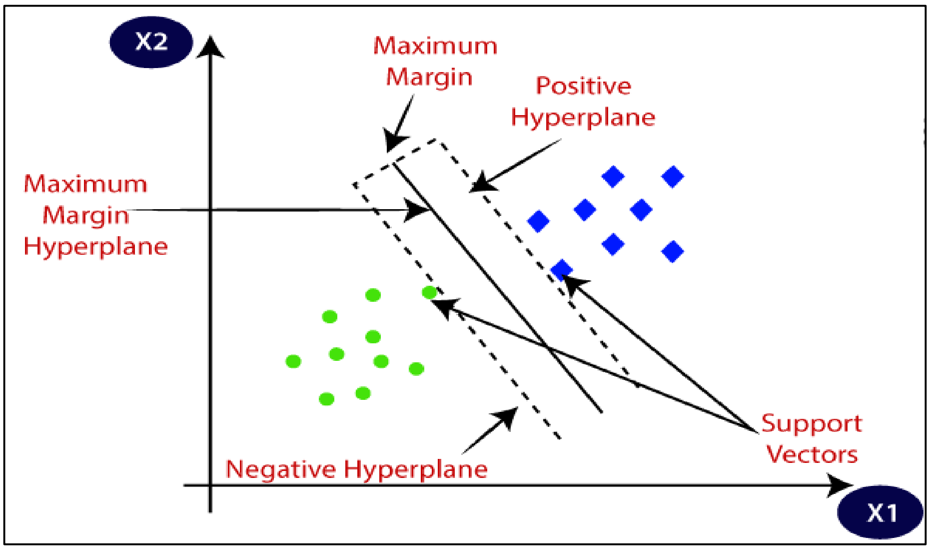


Figure 2.16: Support Vectors Machines.

Decision Tree (DT): is a graphical representation of all possible outcomes and decisions that can be made in a specific scenario [35]. At the root of the tree is the starting point, which represents the entire dataset. The tree then splits into nodes, where each node represents a feature or attribute in the dataset that has the most significant impact on the target variable.

At each node, a decision is made based on a threshold value or condition, which divides the dataset into smaller subsets that are more homogenous in terms of the target variable. This process of dividing the dataset continues until a leaf node is reached, where the final prediction or decision is made.

The architecture of a DT can be divided into two main components: the tree structure and the decision rules. The tree structure consists of the nodes, branches, and leaf nodes, while the decision rules specify the conditions and thresholds that determine the split at each node.

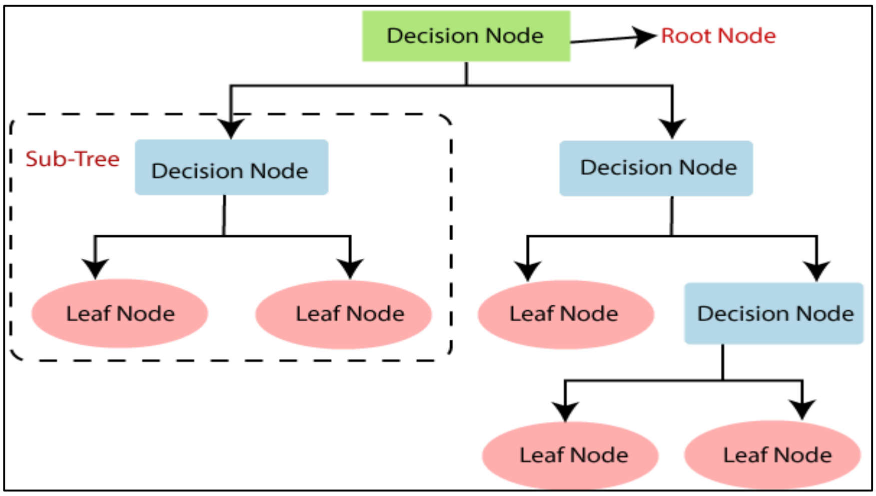


Figure 2.17 : Decision Tree Architecture.

Random forest (RF): is built on top of DTs, where multiple DTs are trained on random subsets of the dataset, and the final prediction is made by aggregating the predictions of all the trees [36].

The architecture of a RF is composed of two main components: the forest of DT and the mechanism for aggregating their predictions. Each DT in the forest is trained on a random subset of the dataset, and the features are also sampled randomly at each node, which helps to reduce overfitting and increase the model's generalization ability.

The mechanism for aggregating the predictions of the DT can be done by either taking the majority vote (in the case of classification tasks) or averaging the predicted values (in the case of regression tasks). Additionally, RF can be further optimized by tuning hyperparameters such as the number of trees in the forest, the maximum depth of the trees, and the minimum number of samples required to split a node. These hyperparameters can affect the accuracy and computational complexity of the model and should be carefully tuned to achieve the best performance.

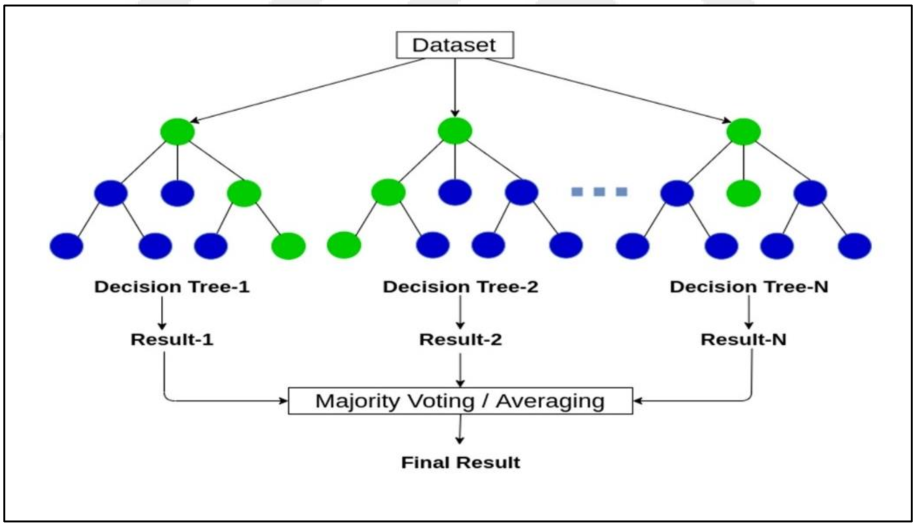


Figure 2.18: Random Forest Architecture.

2.6 DEEP LEARNING

DL is a subset of ML which is based on artificial neural network (ANN) that is characterized by its multiple hidden layers between the input and output layers [37]. It is inspired by the structure and function of the human brain and is particularly well-suited for tasks such as image.

Artificial neurons, which are layers of interconnected nodes with the ability to learn representations of the data at various levels of abstraction, are the building blocks of DL systems. The layers at the start of the network, referred to as the input layer, process unprocessed data, while the layers at the conclusion, referred to as the output layer, generate the final forecast or choice. One or more hidden layers exist in the interim, extracting progressively complicated aspects from the data.

2.6.1 Convolutional Neural Networks (CNNs)

CNNs are a kind of DL model that are mostly applied to video analysis, pattern identification, and image processing. According to [38], they are made to automatically and adaptively learn the spatial hierarchies of features from data, most of which is 3D. Multiple layers of synthetic neurons, or nodes, that imitate human brain neurons make up CNNs. The spatial hierarchies of features are intended to be automatically and adaptively learned by these layers. A CNN's architecture is made to capitalize on an input image's (2D) structure, as well as other 2D inputs like speech signals. Local connections and tied weights are used to do this, and then some kind of pooling produces characteristics that are translation invariant [39].

Here are the main layers that make up a CNN:

a. Input Layer: This is where image data is entered into the network to be processed. The input layer receives the image's raw pixel data.
b. Convolutional Layer (CL): This layer, which forms the basis of a CNN, is made up of a group of trainable filters, also known as kernels, that are all located in the input layer's depth and have a small receptive field. Each filter performs a dot product with the entries of the filter and the input during the feed-forward phase by convolution over the width and height of the input. For every filter, this procedure produces a 2D activation map. As a result, the network is trained to find filters that activate when specified features are detected at designated spatial positions in the input [40].
c. ReLU (Rectified Linear Unit) Layer: This layer introduces an elementwise activation function, specifically the max(0,x) thresholding function at zero, thereby preserving the volume's dimensions. Its primary role is to infuse nonlinearity into the CNN since the preceding layers are predominantly linear in nature [41].
d. Pooling Layer (PL): This layer is put sporadically between CLs that come after it. Its purpose is to gradually reduce the representation's spatial dimension, which in turn lowers the number of parameters and computation in the network and, as a result, also

controls overfitting. According to Cao et al. (2018) [42], the PL resizes the input spatially and functions independently on each depth slice.

e. Fully Connected Layer (FCL): As in ordinary neural networks, neurons in a FCL have complete connections to every activation in the layer above them. As a result, their activations can be calculated using matrix multiplication and bias offset.

f. Output Layer: This is CNN's last layer. The neurons in question control the network's ultimate output. The output layer, for instance, will provide the likelihoods that each class the input image belongs to in a classification issue.

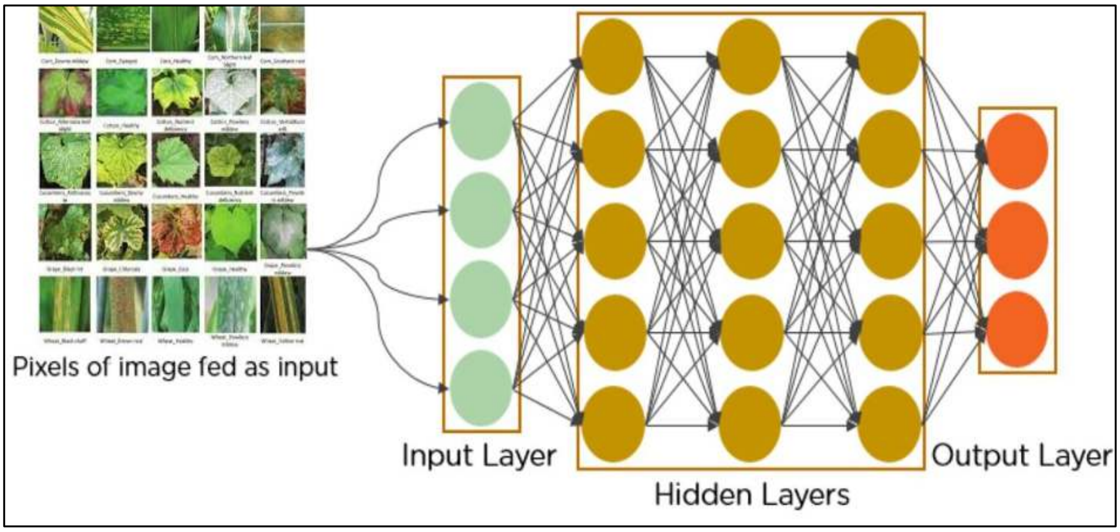


Figure 2.19: Convolutional Neural Networks Architecture.

2.6.2 Visual Geometry Group

The Oxford University group Visual Geometry Group is the source of the VGG model, a convolutional neural network that was highly praised for its simple architecture and outstanding results in the 2014 ImageNet Large Scale Visual Recognition Challenge [43]. Consistent architecture is one of VGG's key characteristics. A RELU activation function follows each CL, which uses 3x3 filters with a stride of 1. In the meantime, the simplicity and adaptability of the model are increased by having all PLs adopt a 2x2 size with a stride of 2.

VGG offers multiple versions, distinguished by their layer counts and the structure of CLs within each block. For instance, the VGG19 variant consists of 19 layers, encompassing 16 CLs distributed across five blocks and three dense layers. In the VGG16 model, the initial two blocks comprise two CLs each, whereas the subsequent three blocks each contain three.

Following each block is a max PL. Concluding the architecture are the three dense layers, succeeded by a softmax activation for the final categorization [44].

The VGG models, including VGG16, are known for their excellent performance on object recognition tasks. However, they are also quite large, which can make them slow to train and use. They also require a lot of memory, both for training and inference. Despite these drawbacks, the VGG models remain popular due to their strong performance and simplicity.

2.6.3 Densely Connected Convolutional Networks

DenseNet, an acronym for Densely Connected Convolutional Networks, is a DL architecture known for its efficacy across a variety of tasks. Developed to address challenges such as the vanishing gradient and overfitting, especially in scenarios with limited labeled data, DenseNet utilizes dense layer-to-layer connections. This promotes gradient flow, facilitates more nuanced feature extraction, and enhances model stability against minor input variations [45].

In the DenseNet structure, each layer receives input from all prior layers, ensuring a continuous feed-forward connectivity. This design offers several benefits. It promotes feature recycling, enhancing the model's efficiency. Additionally, it curtails the overall parameter count, thus mitigating the overfitting risk. Moreover, this connectivity augments information and gradient dissemination, potentially easing the training process [46]. Across domains such as remote sensing image categorization, medical image processing, and high-resolution road imagery extraction, DenseNet has consistently demonstrated top-tier performance, often surpassing other DL models [47].

A salient attribute of DenseNet is its capacity to achieve stellar accuracy with a relatively low parameter count. This is realized via function-preserving transformations, enabling the model to emulate other networks' functions with a distinct parameterization, facilitating additional training and performance enhancement. This approach can notably speed up training, especially for expansive networks.

2.6.4 Residual Network

ResNet, short for Residual Network, is a kind of CNN that uses "skip connections" or shortcuts to get over some layers. This structure was designed to solve the vanishing gradient issue that frequently arises during the training of extremely deep neural networks. Deeper network training is made easier by these skip connections, which allow the gradient to be directly backpropagated to older layers [48]. Layer count is a distinguishing factor amongst ResNet variations. ResNet50, for instance, has 50 layers total: 48 CLs, one input layer, and a FCL. These CLs are organized into four phases, a batch normalization layer and a ReLU activation function come next, each with unique filter amounts and sizes. [48] concluded that the structure is a completely linked layer and that the final categorization is achieved using a softmax activation. ResNet models, like ResNet50, have acquired popularity in a variety of fields, including PD prediction, because of their competitive performance, which frequently outperforms other DL models. In one study, the early detection of Septoria, Tan Spot, and Rust three common wheat diseases in Europe was the main focus. This study demonstrated the ResNet approach's strong performance under real-world acquisition [49].

2.6.5 MobileNet

A class of effective models called MobileNet is designed for embedded and mobile vision applications. MobileNets are built on a simplified design that builds lightweight deep neural networks using depth-wise separable convolutions. Since its initial introduction by Howard et al. in 2017, this architecture has served as the foundation for other models aimed at mobile devices. One sort of factorized convolution that lowers the computational cost that MobileNets use to attain their efficiency is called depthwise separable convolutions. They are therefore perfect for applications requiring lightweight models that can operate on low-processor devices.

The adaptability of MobileNets is one of their main benefits. They are easily modifiable to fit the needs of a certain application. Two hyperparameters the width multiplier and the resolution multiplier can be changed to achieve this. The width multiplier lowers the input/output channel sizes, which lowers the network's computational cost. In contrast, the resolution multiplier shrinks the size of the input image, which lowers the amount of computations the network must perform.

2.7 TRANSFER LEARNING (TL)

TL is a popular ML technique that involves leveraging knowledge and representations learned in one domain or task to improve performance in another domain or task. TL can be especially useful when data is scarce or when the target domain is significantly different from the source domain [50].

TL approaches generally start by training a model on a vast dataset from a related field or task. This model is then refined on a more specific dataset pertaining to the desired domain or task. The initial training phase can utilize either UL or SL, contingent on the presence of labeled data in the originating domain.

Once the model has been pretrained, the weights can be frozen or partially unfrozen, and the model can be fine-tuned on the target domain using supervised learning. During fine-tuning, the model learns to adapt its representations to the target domain by adjusting the weights of the final layers while preserving the knowledge learned in the pretraining phase.

2.8 EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI)

XAI represents a paradigm shift in how we understand and interact with advanced AI systems. Traditionally, many AI models, especially those based in DL, have been regarded as "black boxes" due to their complex and opaque decision-making processes. This lack of transparency can be a significant barrier in fields that require trust and accountability, such as healthcare, finance, and law. XAI aims to make the workings of AI models more understandable to humans, thereby fostering trust, facilitating debugging, enhancing the learning process, and ensuring ethical and fair use of AI.

One of the core methods in XAI involves creating models that are inherently interpretable. These models are designed to provide insights into their decision-making process. For example, DTs or rule-based systems allow users to trace the path taken to reach a conclusion. However, these simpler models often lack the power and scalability of more complex systems like neural networks. Another approach in XAI is to develop post-hoc explanation methods, which attempt to explain the decisions of already trained models. This can involve techniques like feature importance, which identifies which inputs most significantly affect the model's output. Another post-hoc method is the use of surrogate models, where a simpler, more interpretable model is used to approximate the behavior of a complex model.

In the realm of image processing, XAI takes on additional significance and complexity. DL models, have achieved remarkable success in image recognition tasks. However, their decision-making process is often incomprehensible to humans. XAI methods for images aim to reveal how these models perceive and interpret visual information.

2.8.1 Layer-wise Relevance Propagation (LRP)

LRP is an innovative technique in the field of XAI, particularly relevant for deep neural networks. It addresses a fundamental challenge in AI: making the decision-making process of complex models understandable to humans. DL models, especially those used in image and speech recognition, are known for their high performance but are often criticized for their lack of transparency. LRP offers a method to visually and quantitatively interpret what a neural network is "thinking" or "seeing" when it makes a decision. The core idea behind LRP is to decompose the output prediction of a neural network back into the input space, providing a detailed map of how each input feature contributed to the final decision. This is done in a layer-by-layer manner, hence the term "layer-wise." Starting from the output layer, LRP attributes the prediction of the network back through the layers, assigning a relevance score to each neuron in each layer. These scores represent the contribution of each neuron to the final decision, effectively tracing the reasoning of the network in reverse. This backtracking is not a simple task, as it involves navigating the complex web of nonlinear operations and transformations that occur within a neural network. LRP achieves this by using specific rules to propagate relevance. These rules are designed to maintain the conservation of relevance, meaning the total amount of relevance attributed to the input layer equals the output score of the network. Various rules have been proposed to handle different types of layers and operations within neural networks, each with its strengths and suitability for different kinds of networks and tasks.

LRP can offer informative visualizations in the context of image recognition. It might draw attention to portions of an input image that were essential to the classification or prediction process of the network. For instance, in a facial recognition task, LRP could indicate which features of the face (like eyes, nose, or mouth) were most relevant for identifying a person. Such visualizations are not only useful for understanding and trusting AI decisions but also for identifying potential biases or weaknesses in the model. For instance, if a model consistently focuses on irrelevant features for its decisions, it might indicate a need for retraining or restructuring. Moreover, LRP's application is not limited to image data. It can be applied to any domain where DL is used, including text, audio, and complex structured data. This flexibility makes LRP a valuable tool for a wide range of applications, from healthcare diagnostics to financial forecasting.

2.8.2 Grad-CAM

Grad-CAM, is a technique in the realm of XAI that offers visual explanations for decisions made by CNNs, particularly in the context of vision tasks such as image classification. This technique has gained significant attention due to its ability to provide insights into the inner workings of complex CNN models, which are often considered opaque or "black box" systems.

The essence of Grad-CAM lies in its ability to produce "heatmaps" or visualizations that highlight the regions of the input image that were most influential in the neural network's decision-making process. This is particularly valuable for understanding and interpreting the predictions of CNNs in tasks where visual data is crucial. For instance, in medical imaging, Grad-CAM can help highlight the areas in an X-ray or MRI scan that led to a particular diagnostic prediction, thereby providing valuable insights for clinicians.

Grad-CAM operates by taking advantage of the gradients of the target idea that flow into the CNN's last CL. This method is predicated on the notion that gradients can reveal the relative significance of individual neurons for a given choice. Grad-CAM calculates the gradient of the score with respect to the feature maps of a CL for the class of interest (e.g, "dog" in a dog vs. cat classifier). The neuron significance weights are then calculated by taking the global average of these gradients.

Following their acquisition, the significance weights are mixed with the CL's feature maps to create a coarse heatmap with the same spatial dimensions as the layer. The feature maps are effectively combined into a weighted heatmap, where the weights represent the relative importance of each map for the class of interest. After then, the heatmap is upscaled to match the dimensions of the input image, creating a visual overlay that draws attention to the regions that are crucial for the forecast.

One of the key strengths of Grad-CAM is its applicability to a wide variety of CNN architectures without any changes in the network or the need for retraining. This makes it a versatile and powerful tool for interpreting the predictions of pre-trained models. Additionally, Grad-CAM's visual explanations are intuitive and easily understandable, even for individuals without a deep technical background in AI, facilitating better communication and trust between AI practitioners and end-users.

2.9 CONCLUSION

This chapter offers a synopsis of prior research concerning PD identification via image processing and ML methodologies. The literature underscores a growing inclination towards automated systems for PD detection, attributed to their precision and effectiveness. Image processing remains pivotal in discerning and gleaning features from plant images, subsequently feeding them into diverse ML models. Algorithms like SVM and DT are prominently employed for disease classification and diagnosis in plants. Additionally, DL and TL have surfaced as potent methods in this domain, delivering enhanced accuracy while minimizing intensive feature engineering.

3. METHODOLOGY

3.1 INTRODUCTION

PDs pose a significant threat to crop yield and food security worldwide. Accurate and timely detection of these diseases is crucial for effective disease management and prevention. In recent years, advancements in computer vision and DL techniques have opened up new possibilities for automated PD prediction. This chapter focuses on the development of a robust and efficient PD prediction system using DL models.

3.2 THE PROPOSED FRAMEWORK

Our proposed method for PD prediction involves multiple steps, as depicted in Figure 3.1. The process begins with the collection of a dataset containing images of both healthy and diseased plants. This dataset then undergoes preprocessing, where images are selected based on relevance, resized for consistency, and their pixel values normalized between 0 and 1. After preprocessing, the dataset is divided into training and testing groups. The next stage involves the implementation of DL architectures, including CNNs, DenseNet, ResNet, MobileNet, and VGG. A key feature of our approach is the integration of XAI through the Grad-CAM technique, which provides insightful visual explanations for the model's predictions. It does this by identifying key areas in the images that significantly influence the model's decisions, thereby adding a layer of interpretability. The different models are then trained using the prepared data, and their performance is subsequently assessed on the testing set. The model that demonstrates the best performance metrics is selected as the primary tool for PD prediction, ensuring not only high accuracy but also a transparent decision-making process.

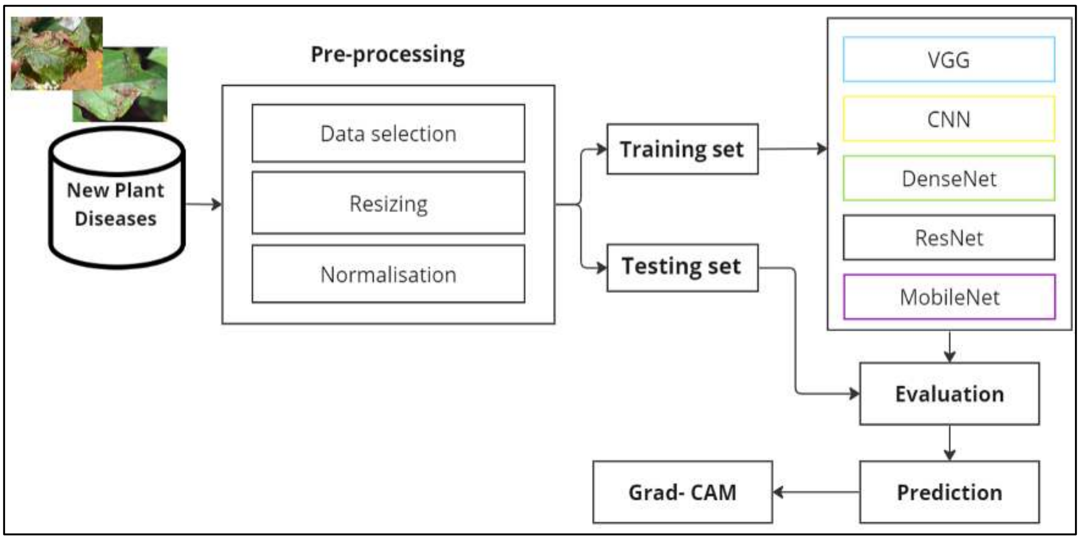


Figure 3.1: The Proposed Framework.

3.2.1 Dataset

This dataset includes over 87,000 RGB images of crop leaves that have been divided into 38 classifications to depict both healthy and unhealthy plants. With the original directory structure intact, the dataset is split into an 80/20 training and validation set. Furthermore, 33 test images are prepared in a different directory for prediction purposes. Researchers and practitioners interested in PD detection and classification tasks can obtain this dataset through Kaggle, which is publically available.

3.2.2 Pre-Processing

The preprocessing steps help to ensure that the data is appropriately prepared for subsequent stages such as training and evaluation of the models.

3.2.2.1 Data selection

In this step, a subset of labels is chosen for analysis. Specifically, only 20 labels are selected from each folder. This means that out of the available classes in the dataset, only 20 specific classes are considered for further processing and analysis in this study.

3.2.2.2 Resizing

The images are resized to a consistent size of 224 x 224 pixels. Resizing the images to a uniform dimension is important for compatibility with many DL models that require fixed input sizes. By resizing the images to 224 x 224 pixels, all the images in the dataset are brought to the same size, ensuring consistency during subsequent stages of the analysis

3.2.2.3 Normalization

Normalization is applied to the pixel values of the images. Each image is divided by 255, resulting in pixel values ranging from 0 to 1. This normalization process is often performed in ML tasks to standardize the data and ensure that the pixel values are within a consistent range. Dividing by 255 scales the pixel values to a normalized range, making them suitable for further processing and analysis with DL models.

3.2.3 The Proposed Models

3.2.3.1 Convolution neural network (CNN)

In this study, the initial model employed was a CNN developed using TensorFlow. The architecture starts with a 2D CL with 32 filters and 'relu' activation, succeeded by a max PL that diminishes spatial dimensions. This sequence is replicated once before introducing a dropout layer, which counters overfitting by sporadically zeroing out input units during training sessions. Subsequently, the model transitions the 2D spatial information into a 1D format, directing it through a densely connected layer comprising 128 nodes with 'relu' activation. The concluding layer is a dense one with 20 nodes, equipped with a 'softmax' activation, tailored for multi-category classification. The design of this model is apt for tasks related to image classification, presenting adaptability in layer and parameter adjustments relative to the task's intricacy and data quantity.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 222, 222, 32)      896

 max_pooling2d_2 (MaxPooling  (None, 111, 111, 32)     0
 2D)

 conv2d_3 (Conv2D)           (None, 109, 109, 32)      9248

 max_pooling2d_3 (MaxPooling  (None, 54, 54, 32)       0
 2D)

 dropout_1 (Dropout)         (None, 54, 54, 32)        0

 flatten_7 (Flatten)         (None, 93312)             0

 dense_6 (Dense)             (None, 128)               11944064

 dense_7 (Dense)             (None, 20)                2580

=================================================================
Total params: 11,956,788
Trainable params: 11,956,788
Non-trainable params: 0
```

Figure 3.2: The CNN Model Architecture.

3.2.3.2 VGG model

The given script outlines a model rooted in the VGG19 structure, which consists of 19 layers: 16 convolutional and 3 fully connected. It has been pre-trained on the ImageNet dataset, which houses a vast array of images spanning numerous classes. This pre-training's weights are integrated into the subsequent model, thus furnishing it with a proficient feature-detection capability already accustomed to discerning diverse image features.

To adapt to potential new classification tasks distinct from the ImageNet's objectives, the top layers, particularly the final FCL geared for ImageNet classes, are removed. The output from the foundational VGG19 is then transformed from 3D to 1D through a flattening process. Subsequently, this streamlined output navigates through a quartet of FCLs. The trio of initial Dense layers adopt the 'relu' activation, injecting non-linearity and capacitating the model to assimilate intricate patterns. Sequentially, the nodes in these layers taper from 1024 to 512, finally settling at 248, acting as a dimensionality compressor.

The terminal Dense layer, furnished with 20 nodes and empowered by the 'softmax' activation, is devised for a classification task with 20 distinct classes. This 'softmax' operation ensures the layer's output offers probabilities, collectively amounting to 1, each signifying the model's assurance in classifying the input image under a specific category.

Following this setup, the model is structured for training via the TensorFlow's Keras interface. Accepting input images of dimensions (224, 224, 3), the model yields a probability distribution spanning 20 classes. It's pivotal to acknowledge that the concluding duo of layers in the VGG19 base remain unalterable during training, embodying a TL technique. This approach utilizes the acumen from its foundational training (ImageNet categorization) to benefit a novel objective. While these static layers persist in feature extraction as honed by ImageNet, the freshly introduced layers evolve with the latest data.

```
Model: "model"

 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 1024)              25691136

 dense_1 (Dense)             (None, 1024)              1049600

 dense_2 (Dense)             (None, 512)               524800

 dense_3 (Dense)             (None, 248)               127224

 dense_4 (Dense)             (None, 20)                4980

=================================================================
Total params: 47,422,124
Trainable params: 29,757,548
Non-trainable params: 17,664,576
_________________________________________________________________
```

Figure 3.3: VGG Model Architecture.

3.2.3.3 DenseNet model

The DenseNet201 structure is a derivative of the DenseNet design, encompassing 201 layers. This convolutional neural network (CNN) is pre-trained on the extensive ImageNet dataset, encompassing numerous distinct image classes. As it has been trained on over a million

ImageNet images, which include a vast array of objects like keyboards, mice, and pencils, the network has cultivated a profound ability to identify an assortment of features in images.

To suit potential new classification tasks differing from ImageNet's objectives, the uppermost layers, especially the ultimate fully connected classification layer, are excised. The output from the foundational DenseNet201 is subsequently relayed through a series of four FCLs. The initial three of these layers deploy the ReLU activation function, facilitating non-linear learning and enabling the assimilation of intricate patterns. The nodes within these layers sequentially reduce from 2048 to 1024 and finally to 512, acting as a dimensionality condenser. Post each layer, a Dropout layer is introduced, intermittently nullifying certain input units during training iterations, bolstering the model's resistance to overfitting.

The terminal FCL, fitted with 20 nodes, employs the 'softmax' activation, gearing it towards a 20-class categorization challenge. This activation ensures the layer's output morphs into probabilities, collectively equating to 1, each denoting the model's certainty regarding the image's class alignment.

Upon configuration, the model is primed for training through TensorFlow's Keras module, accepting images of dimensions (224, 224, 3) and rendering a probability distribution across the 20 classes. Significantly, the core DenseNet201 remains immutable during training, maintaining its pre-trained feature extraction abilities, while the freshly appended layers adapt to the new dataset. This methodology embodies TL, capitalizing on the insights from its original ImageNet training for diverse tasks.

3.2.3.4 ResNet model

The ResNet50 structure is a 50-layer variation of the ResNet design. Trained originally on the expansive ImageNet dataset, which spans a multitude of image categories, its weights are transitioned into a fresh model, serving as a proficient feature identification tool.

To adapt to potentially distinct classification endeavors from the ImageNet goals, the upper layers, particularly the concluding FCL dedicated to ImageNet categorizations, are eliminated. Following this, the output from the core ResNet50 undergoes a GlobalAveragePooling2D layer, averaging the feature maps spatially and streamlining the output dimensions. Successively, this output is channeled through four Dense layers. The initial trio of these layers employs the 'relu' activation, endowing the model with non-linear learning capacities. As nodes in these layers sequentially decrease from 1024 to 512 and ultimately 248, they facilitate a reduction in dimensionality.

The concluding Dense layer, equipped with 20 nodes, integrates the 'softmax' activation, tailor-made for 20-class categorization. This activation guarantees the output morphs into a collective probability of 1, with each value signifying the model's prediction confidence for the respective class.

Post-configuration, this model is prepped for training via TensorFlow's Keras framework, processing images of (224, 224, 3) dimensions and producing a probability distribution across 20 classes.

Notably, the penultimate duo of layers in the foundational ResNet50 remains unmodifiable during the training phase, preserving their ImageNet-trained feature extraction capabilities. This strategy epitomizes TL, where pre-trained insights are harnessed for new challenges, letting the appended layers evolve based on the fresh dataset.

3.2.3.5 MobileNet model

The given script establishes a model rooted in the MobileNetV2 framework. This version of MobileNet incorporates two novel elements: linear bottlenecks interspersed among the layers and shortcut links connecting these bottlenecks. Initially trained on the expansive ImageNet dataset, comprising diverse image categories, the model's learned weights are imported into our new structure, creating an adept feature detector.

To better align with a potentially distinct classification objective compared to the original ImageNet task, the terminal layers, especially the final fully connected classification layer, are excised.

Subsequently, the core MobileNetV2's output is channeled through four Dense layers. The leading trio utilizes the 'relu' activation, infusing the model with the capability to discern intricate patterns. Node counts in these layers sequentially drop from 2024 to 1024 and finally to 512, exemplifying a dimensionality curtailment strategy. Post each of these layers, a Dropout layer is integrated, sporadically nullifying a portion of input units during training iterations to counteract overfitting.

The terminal Dense layer, furnished with 20 nodes, adopts the 'softmax' activation, gearing it for a 20-class distinction task. This activation ensures the layer's output crystallizes into probabilities that collectively equal 1, each denoting the model's prediction assurance for each class.

Upon configuration, this model is primed for training using TensorFlow's Keras module, processing (224, 224, 3) dimensional images and outputting a probability spread over 20 classifications.

3.2.4 Explaining AI with Grad-CAM Techniques

we harness the power of the renowned Gradient-Weighted Class Activation Mapping technique, commonly referred to as Grad-CAM [52]. This innovative approach facilitates a deeper understanding of decision-making processes within ML models by offering precise explanations tailored to each specific input. One of the most distinguishing features of Grad-CAM is its capacity to produce vivid visualizations, elucidating the relevance of specific pixels as perceived by deeply trained algorithms. Its reputation as a prominent tool within the XAI domain is well-deserved, considering its wide acceptance and utilization in numerous computer vision tasks. Considering the nature of our research, which predominantly relies on image-centric data harvested from three separate experiments, the reliability and efficacy of Grad-CAM compelled us to select it as our XAI technique of choice. While the world of XAI boasts a plethora of methods, our focus remains undistracted by exclusively concentrating on Grad-CAM, purposefully avoiding comparative discussions or explorations into the nuances of alternative explanatory techniques. It's pivotal to acknowledge that the foundational pillars upon which Grad-CAM rests are deeply rooted in CAM methodologies [53]. A hallmark of Grad-CAM is its adeptness at leveraging gradient data accrued throughout the model's training journey. This unique capability paves the way for discerning the hierarchy of importance among neurons in the decision-making lattice of the model, with neurons manifesting higher gradient magnitudes signifying a more central role in shaping the model's determinations.

4. RESULTS AND DISCUSSION

4.1 INTRODUCTION

In this chapter, we delve into a comprehensive analysis of the classification outcomes derived from various advanced deep learning models, specifically focusing on disease detection. This chapter is meticulously structured to not only present the quantitative results but also to offer a qualitative interpretation of these results, allowing for a deeper understanding of the models' performance and their practical implications.

4.2 CLASSIFICATION RESULTS

4.2.1 Training parameters

Table 4.1 provides details on the training specifications for the DL models. The model is tailored to distinguish among 20 unique categories, as indicated by the 20 classes. Over the training period, the model underwent 20 epochs, iterating over the dataset entirely 20 times. The chosen loss function was Sparse Categorical Cross Entropy, frequently utilized for tasks with multiple classifications. ADAM served as the optimizer, a widely-recognized method for optimizing stochastic objective functions based on first-order gradients. Each update of the model parameters utilized 64 data samples, as denoted by the batch size. To assess the model's efficacy during its training, 20% of the dataset was earmarked for validation, represented by a 0.2 validation split. Lastly, the designated size for input images was fixed at 224 x 224 pixels.

Table 4.1: Models Training Parameters.

Parameter	**values**
Classes number	20
Epochs	20
Loss	Sparce_categorical_cross_entropy
Optimizer	ADAM
Batch size	64
Validation split	0.2
Image input size	224 x 224

4.2.2 CNN Results

Figure 4.1 illustrates the CNN model's evolution over 20 epochs. An epoch reflects a full cycle through the training dataset, followed by a performance evaluation on a distinct validation set.

Initially, in epoch one, the model registers a substantial training loss at 0.6185, signifying a marked deviation between its forecasts and the true outcomes. Still, its training ACC stands at approximately 80.46%, indicating its efficiency in predicting most cases correctly. The validation metrics are even more promising, with a reduced loss of 0.2106 and an enhanced accuracy of 92.82%, showcasing the model's competence with unfamiliar data.

By the time the model reaches its 20th epoch, notable progress is evident. The training loss shrinks to a mere 0.0448, signifying the model's successful learning curve and its capacity to narrow the gap between predictions and actual outcomes. Furthermore, its training ACC ascends to an impressive 98.74%, denoting near-perfect predictions for the vast majority of training cases.

Regarding the validation dataset in the 20th epoch, the model posts a loss of 0.1324 and an ACC of 97.47%. Although these metrics mark an ascent from the starting point, the slightly elevated validation loss compared to the training loss might hint at potential overfitting, where the model exhibits superior performance on training data relative to new data. However, given the substantial validation accuracy, the model demonstrates its adaptability and proficiency in correctly predicting a vast majority of the validation cases.

Figure 4.1: CNN Training Process.

The confusion matrix (CM) provided in Figure 4.2 is a 20x20 matrix, which corresponds to a classification model trained to distinguish between 20 different classes. These classes represent various plant health conditions, including healthy states and specific diseases for different types of plants such as peach, raspberry, blueberry, apple, tomato, pepper, potato, grape, corn, cherry, soybean, and orange. The diagonal elements of the matrix represent the number of instances where the predicted class is the same as the actual class, indicating correct predictions made by the model. For instance, the first element of the matrix (130) signifies that the model correctly identified 'Peach_healthy' 130 times.

Off-diagonal elements in the matrix represent instances where the model's predictions did not match the actual class, indicating misclassifications. For example, the element at the 5th row and 20th column (28) indicates that the model predicted the class 'Tomato_Early_blight' 28 times, while the actual class was 'Peach_healthy'.

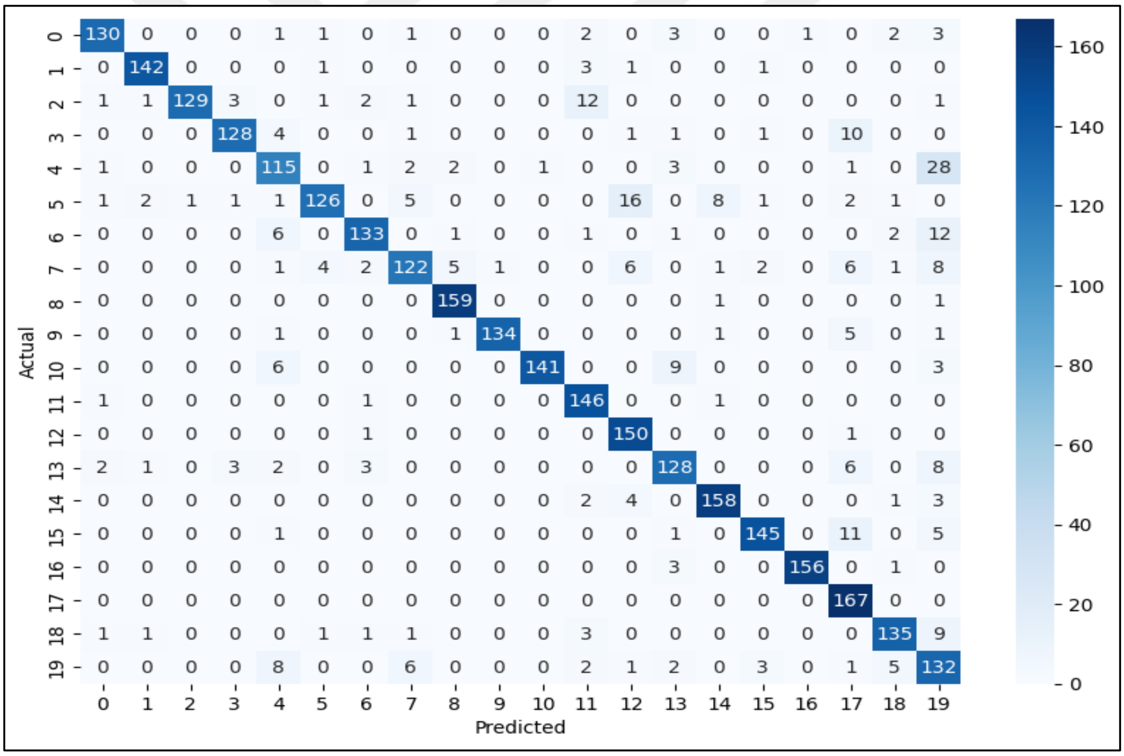


Figure 4.2: CNN Confusion Matrix.

Figure 4.3, the classification report (CR), offers an in-depth assessment of the CNN model's ACC across all dataset classes, employing four primary metrics:

a. Precision (PRE): Defined as the proportion of true positives (TP) to the combined total of true positives and false positives (FP), precision gauges the model's efficiency in accurately classifying positive instances among those it identified as positive. Notably, the classes 'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)' and 'Potato___Early_blight' both boast a peak PRE score of 1.00, suggesting flawless identification.

b. Recall (REC): Calculated as the ratio of true positives to the combined number of true positives and false negatives (FN), REC measures the model's prowess in accurately distinguishing positive cases from all the genuine positive instances. The class 'Grape___Esca_(Black_Measles)' stands out with a top REC rate of 1.00, showing the model's capability to flawlessly detect all instances of this category.

c. F1-Score (F1-S): Representing the harmonic average of precision and REC, the F1-S offers a comprehensive view of the model's efficacy. A score nearing 1 denotes superior performance. In this report, 'Grape___Esca_(Black_Measles)' achieves the zenith F1-S of 1.00, showcasing an optimal blend of PRE and REC.

d. Support: This indicates the class's actual count within the dataset. For instance, the 'Peach___healthy' class comprises 144 instances.

At the report's conclusion, 'ACC' sheds light on the model's overarching accuracy, reflecting the ratio of correct predictions (including both true positives and true negatives) to the entirety of the evaluated instances. For this model, the achieved ACC stands at 91%.

	precision	recall	f1-score	support
Peach___healthy	0.94	0.95	0.95	144
Raspberry___healthy	0.99	0.97	0.98	148
Blueberry___healthy	0.88	0.99	0.93	151
Apple___Cedar_apple_rust	0.92	0.86	0.89	146
Tomato___Late_blight	0.85	0.79	0.81	154
Pepper,_bell___healthy	0.95	0.79	0.86	165
Tomato___Leaf_Mold	0.96	0.65	0.77	156
Pepper,_bell___Bacterial_spot	0.95	0.90	0.92	159
Potato___Early_blight	0.99	0.94	0.97	161
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.00	0.99	1.00	143
Corn_(maize)___Northern_Leaf_Blight	0.93	0.99	0.96	159
Tomato___Tomato_mosaic_virus	0.90	0.97	0.94	149
Cherry_(including_sour)___healthy	0.96	0.98	0.97	152
Peach___Bacterial_spot	0.99	0.79	0.88	153
Soybean___healthy	0.79	0.99	0.88	168
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.93	0.96	0.95	163
Grape___Esca_(Black_Measles)	0.99	1.00	1.00	160
Orange___Haunglongbing_(Citrus_greening)	0.93	0.98	0.96	167
Tomato___Target_Spot	0.74	0.92	0.82	152
Tomato___Early_blight	0.75	0.79	0.77	160
accuracy			0.91	3110
macro avg	0.92	0.91	0.91	3110
weighted avg	0.92	0.91	0.91	3110

Figure 4.3: CNN Classification Report.

4.2.3 VGG Results

The VGG19 model underwent training for 20 epochs, where each epoch signifies a full pass through the entire training dataset. Here's a breakdown of its progression:

Epoch 1: Starting off, the model reported a training loss of 0.8909 and an ACC rate of 0.7119. The loss denotes the model's prediction error, with lower figures indicating superior performance. Meanwhile, the model's validation metrics – a loss of 0.4955 and an ACC of 0.8568 – provide insights into how the model fares with data it hasn't seen before.

Epoch 2 to Epoch 19: Over this span, notable improvements were observed. Training loss dipped from 0.3450 to 0.0774, while training ACC surged from 0.8854 to 0.9861. The declining validation loss and rising validation ACC signify the model's apt learning and adept generalization capabilities.

Epoch 20: Culminating its training, the model posted a training loss of 0.0774 and an ACC of 0.9861. The validation metrics displayed a loss of 0.4749 and an ACC of 0.9093.

While the training outcomes showed consistent enhancements, the fluctuating validation loss hinted at potential overfitting a scenario where a model overly adapts to the training set, hampering its performance with new data.

Ideally, the most effective model couples the highest validation ACC with the lowest validation loss. In this context, Epoch 14's model emerges as the frontrunner, boasting a validation ACC of 0.9402 and a loss of 0.2756, indicating its strong generalization capability during that phase of training.

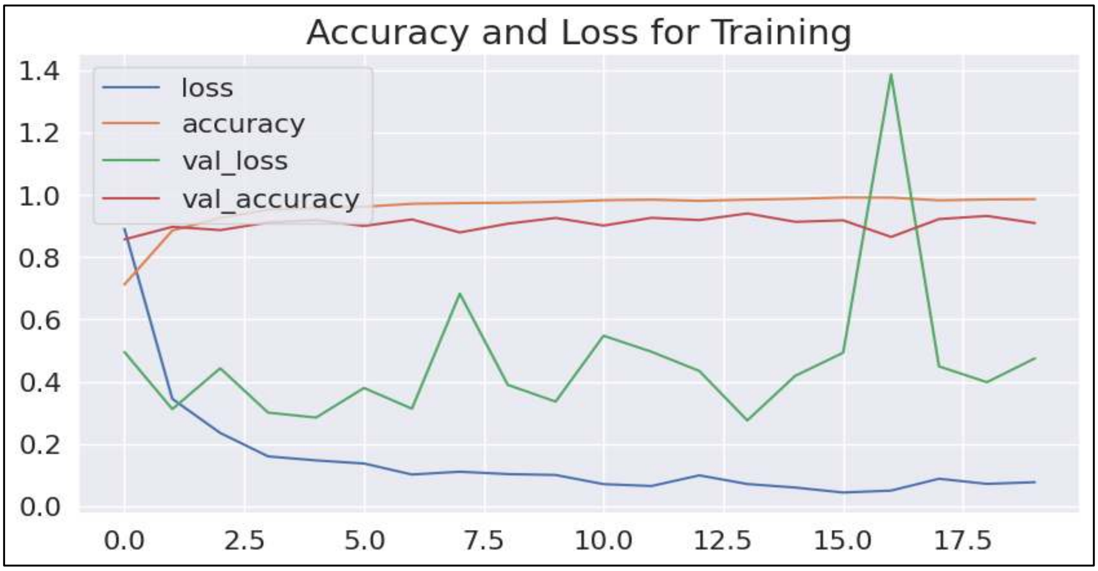


Figure 4.4: VGG Training Process.

Figure 4.5 represents the VGG19 CM. The diagonal elements represent the number of points for which the predicted label is equal to the true label, i.e., the correct predictions. For instance, the first row, first column value of 137 indicates that 137 instances of 'Peach___healthy' were correctly predicted.

The off-diagonal elements provide the instances where the prediction was incorrect. For example, the value of 1 at the first row, third column indicates that 1 instance of 'Peach___healthy' was incorrectly predicted as 'Blueberry___healthy'.

The model seems to perform best in predicting 'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)' and 'Grape___Esca_(Black_Measles)', as indicated by the perfect scores of 142 and 160 respectively in the CM. These classes have no instances that were misclassified.

On the other hand, the model seems to struggle with 'Tomato___Leaf_Mold' and 'Tomato___Target_Spot', as indicated by the relatively high number of misclassifications in these rows. For instance, 55 instances of 'Tomato___Leaf_Mold' were misclassified as other classes, and 12 instances of 'Tomato___Target_Spot' were misclassified.

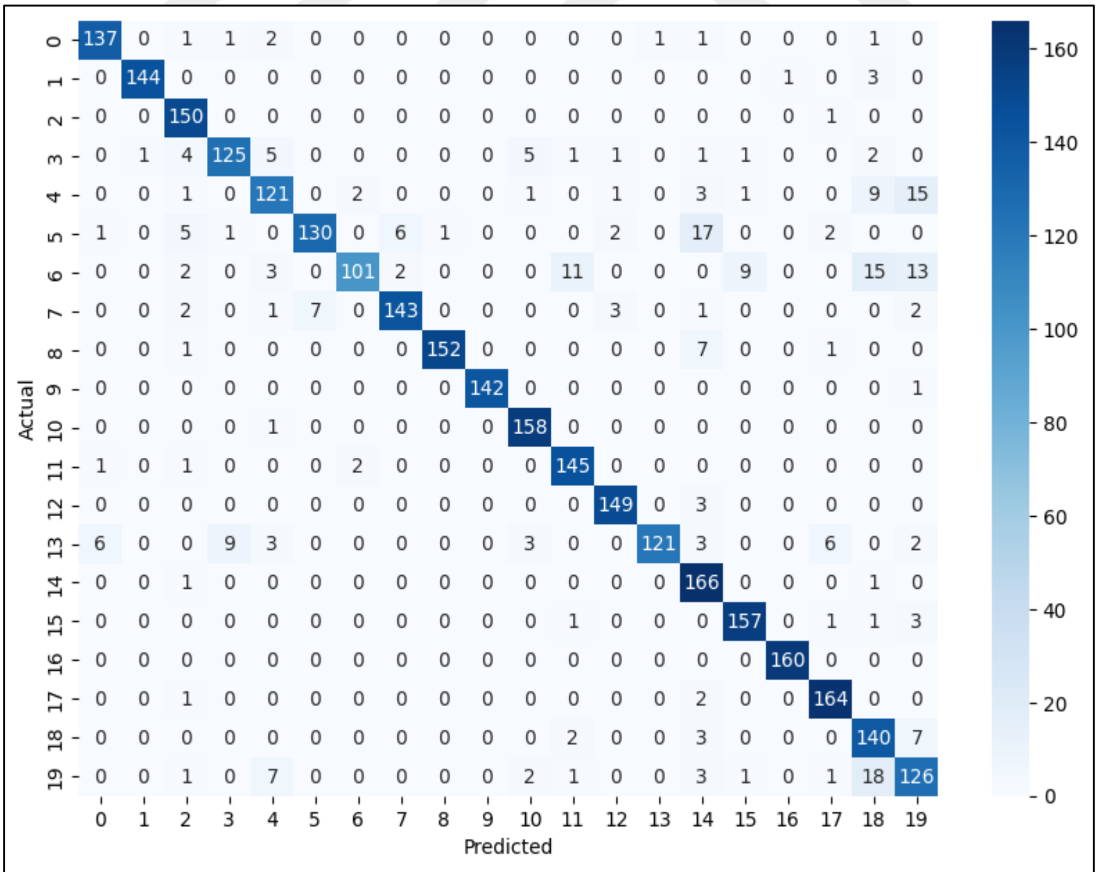


Figure 4.5: VGG Confusion Matrix.

PRE is the ability of the classifier not to label as positive a sample that is negative. For example, 'Raspberry___healthy' has the highest PRE of 0.99, meaning that when the model predicts an instance to be 'Raspberry___healthy', it is correct 99% of the time.

REC is the ability of the classifier to find all the positive samples. For instance, 'Blueberry___healthy' has a REC of 0.99, indicating that it correctly identified 99% of all 'Blueberry___healthy' instances.

The F1 score is a weighted harmonic mean of PRE and REC such that the best score is 1.0 and the worst is 0.0. F1 scores are lower than ACC measures as they embed PRE and REC into their computation. The 'Grape___Esca_(Black_Measles)' class has a perfect F1 score of 1.00, indicating excellent PRE and REC balance for this class.

Support is the number of actual occurrences of the class in the specified dataset. For example, 'Soybean___healthy' has the highest support of 168, meaning there are 168 instances of 'Soybean___healthy' in the test dataset. The ACC of the model is 91% of the time across all classes.

	precision	recall	f1-score	support
Peach___healthy	0.94	0.95	0.95	144
Raspberry___healthy	0.99	0.97	0.98	148
Blueberry___healthy	0.88	0.99	0.93	151
Apple___Cedar_apple_rust	0.92	0.86	0.89	146
Tomato___Late_blight	0.85	0.79	0.81	154
Pepper,_bell___healthy	0.95	0.79	0.86	165
Tomato___Leaf_Mold	0.96	0.65	0.77	156
Pepper,_bell___Bacterial_spot	0.95	0.90	0.92	159
Potato___Early_blight	0.99	0.94	0.97	161
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.00	0.99	1.00	143
Corn_(maize)___Northern_Leaf_Blight	0.93	0.99	0.96	159
Tomato___Tomato_mosaic_virus	0.90	0.97	0.94	149
Cherry_(including_sour)___healthy	0.96	0.98	0.97	152
Peach___Bacterial_spot	0.99	0.79	0.88	153
Soybean___healthy	0.79	0.99	0.88	168
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.93	0.96	0.95	163
Grape___Esca_(Black_Measles)	0.99	1.00	1.00	160
Orange___Haunglongbing_(Citrus_greening)	0.93	0.98	0.96	167
Tomato___Target_Spot	0.74	0.92	0.82	152
Tomato___Early_blight	0.75	0.79	0.77	160
accuracy			0.91	3110
macro avg	0.92	0.91	0.91	3110
weighted avg	0.92	0.91	0.91	3110

Figure 4.6: VGG Classification Report.

4.2.4 ResNet Results

In the first epoch, the model started with a high loss of 2.7833 on the training set and 2.5942 on the validation set, indicating a high error rate. The ACC was also low, at 0.1014 for the training set and 0.1488 for the validation set, indicating that the model was only correctly predicting about 10% and 15% of the training and validation sets, respectively. As the epochs progressed, both the loss decreased and the ACC increased for both the training and validation sets, indicating that the model was learning and improving its predictions. For example, by the 20th epoch, the loss had decreased to 1.3633 on the training set and 1.5063 on the validation set. The ACC had also increased to 0.5424 on the training set and 0.5022 on the validation set.

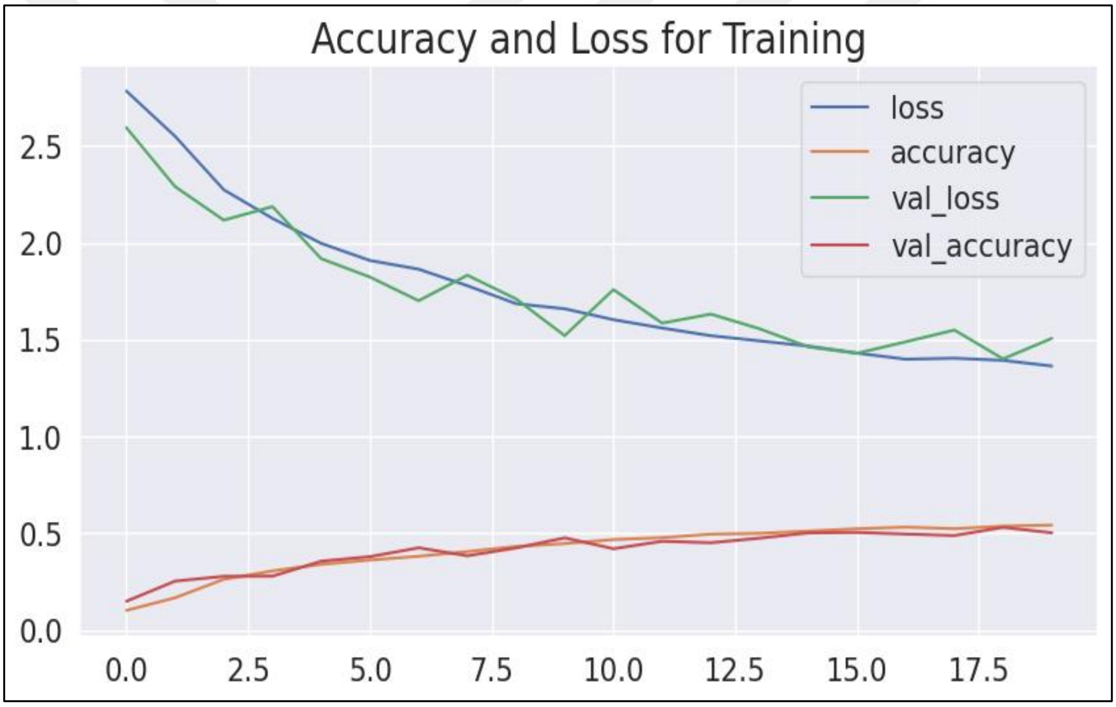


Figure 4.7: ResNet Training Process.

The diagonal cells represent instances where the predicted class is the same as the actual class, i.e., the model made correct predictions. For example, the first cell in the matrix (top-left corner) shows that the model correctly identified 105 instances of the first class (Peach___healthy). Off-diagonal cells indicate the instances where the model made incorrect predictions. For instance, the cell at the second row and first column shows that the model predicted 41 instances of the second class (Raspberry___healthy) as the first class (Peach___healthy). In general, the higher the diagonal values of the CM the better, indicating many correct predictions. The values outside the diagonal should be as low as possible, indicating fewer misclassifications.

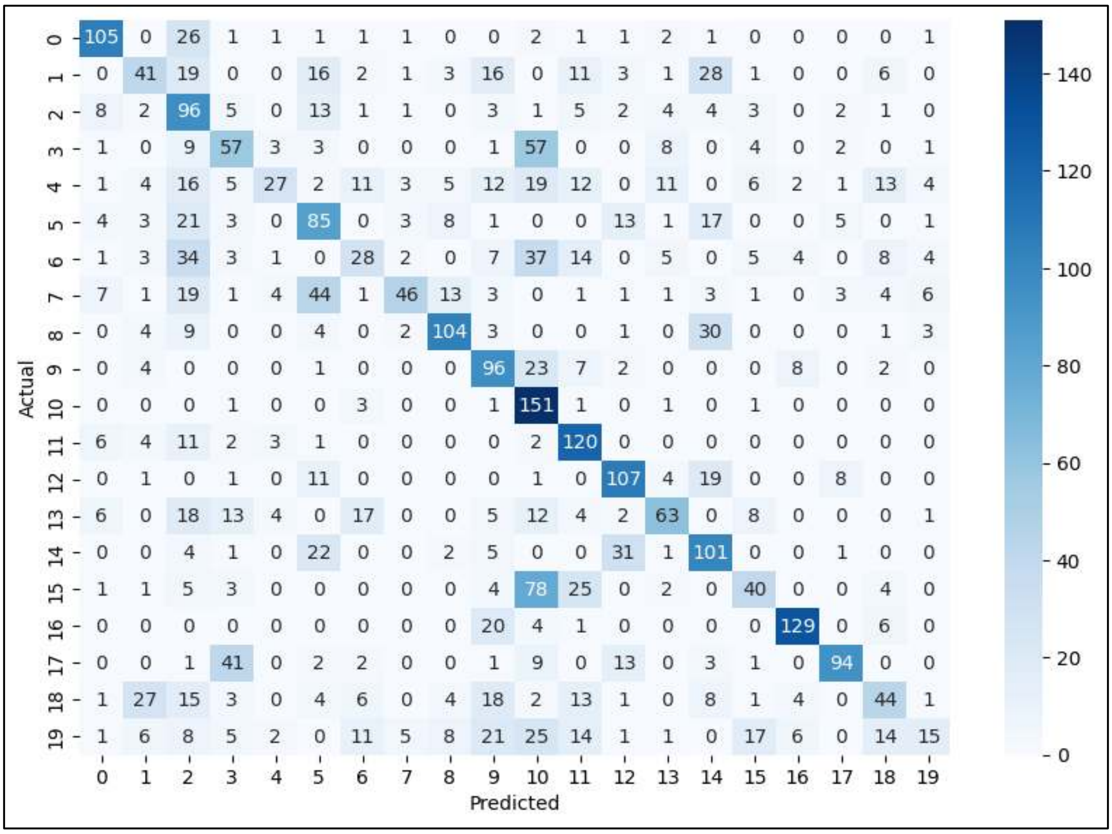


Figure 4.8: ResNet Confusion Matrix.

From the report in Figure 4.9, it appears that the model has varying performance across different classes. For instance, the model performs well on classes like "Grape___Esca_(Black_Measles)" with a PRE of 0.84 and REC of 0.81, but struggles with classes like "Tomato___Early_blight" with a PRE of 0.41 and REC of 0.09. The overall ACC of the model is 0.50, indicating that the model correctly predicts the class 50% of the time.

	precision	recall	f1-score	support
Peach___healthy	0.74	0.73	0.73	144
Raspberry___healthy	0.41	0.28	0.33	148
Blueberry___healthy	0.31	0.64	0.42	151
Apple___Cedar_apple_rust	0.39	0.39	0.39	146
Tomato___Late_blight	0.60	0.18	0.27	154
Pepper,_bell___healthy	0.41	0.52	0.45	165
Tomato___Leaf_Mold	0.34	0.18	0.23	156
Pepper,_bell___Bacterial_spot	0.72	0.29	0.41	159
Potato___Early_blight	0.71	0.65	0.68	161
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	0.44	0.67	0.53	143
Corn_(maize)___Northern_Leaf_Blight	0.36	0.95	0.52	159
Tomato___Tomato_mosaic_virus	0.52	0.81	0.63	149
Cherry_(including_sour)___healthy	0.60	0.70	0.65	152
Peach___Bacterial_spot	0.60	0.41	0.49	153
Soybean___healthy	0.47	0.60	0.53	168
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.45	0.25	0.32	163
Grape___Esca_(Black_Measles)	0.84	0.81	0.82	160
Orange___Haunglongbing_(Citrus_greening)	0.81	0.56	0.66	167
Tomato___Target_Spot	0.43	0.29	0.35	152
Tomato___Early_blight	0.41	0.09	0.15	160
accuracy			0.50	3110
macro avg	0.53	0.50	0.48	3110
weighted avg	0.53	0.50	0.48	3110

Figure 4.9: ResNet Classification Report.

4.2.5 MobileNet Results

The training process in Figure 4.10 Epoch 1 indicates that the model started with a loss of 0.6946 and an ACC of 0.7770 on the training set. On the validation set, the loss was 0.2470 and the ACC was 0.9166. Over the next epochs, both the training loss and validation loss generally decreased, which is a good sign indicating that the model is learning and improving its predictions. The ACC on both the training and validation sets generally increased, which is also a good sign. By the final epoch (Epoch 20), the model achieved a training loss of 0.0608 and a training ACC of 0.9871. On the validation set, the model achieved a loss of 0.2776 and an ACC of 0.9563.

It's worth noting that the validation loss started to increase after Epoch 10, while the validation ACC remained relatively stable. This could be a sign of overfitting, where the model is learning the training data too well and is not generalizing effectively to unseen data in the validation set.

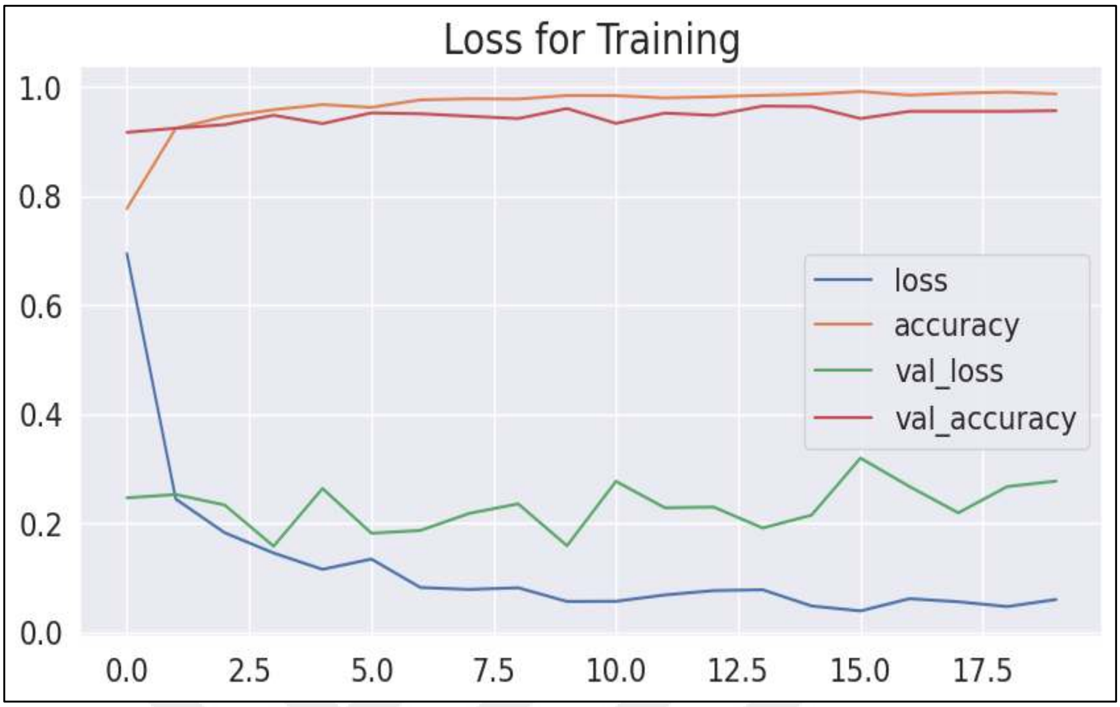


Figure 4.10: MobileNet Training Process.

The first row in figure 4.11 shows that 143 instances were correctly predicted as Class 1 (true positives), and 1 instance was predicted as Class 1 but actually belongs to Class 7 (false positive).

The second row shows that all 148 instances were correctly predicted as Class 2.

The third row shows that 146 instances were correctly predicted as Class 3, 1 instance was predicted as Class 3 but actually belongs to Class 1, 2 instances were predicted as Class 3 but actually belong to Class 4, 1 instance was predicted as Class 3 but actually belongs to Class 7, and 1 instance was predicted as Class 3 but actually belongs to Class 16.

The last row shows that 118 instances were correctly predicted as Class 20, 2 instances were predicted as Class 20 but actually belong to Class 4, 9 instances were predicted as Class 20 but actually belong to Class 7, 2 instances were predicted as Class 20 but actually belong to Class 8, 1 instance was predicted as Class 20 but actually belongs to Class 9, and 13 instances were predicted as Class 20 but actually belong to Class 19.

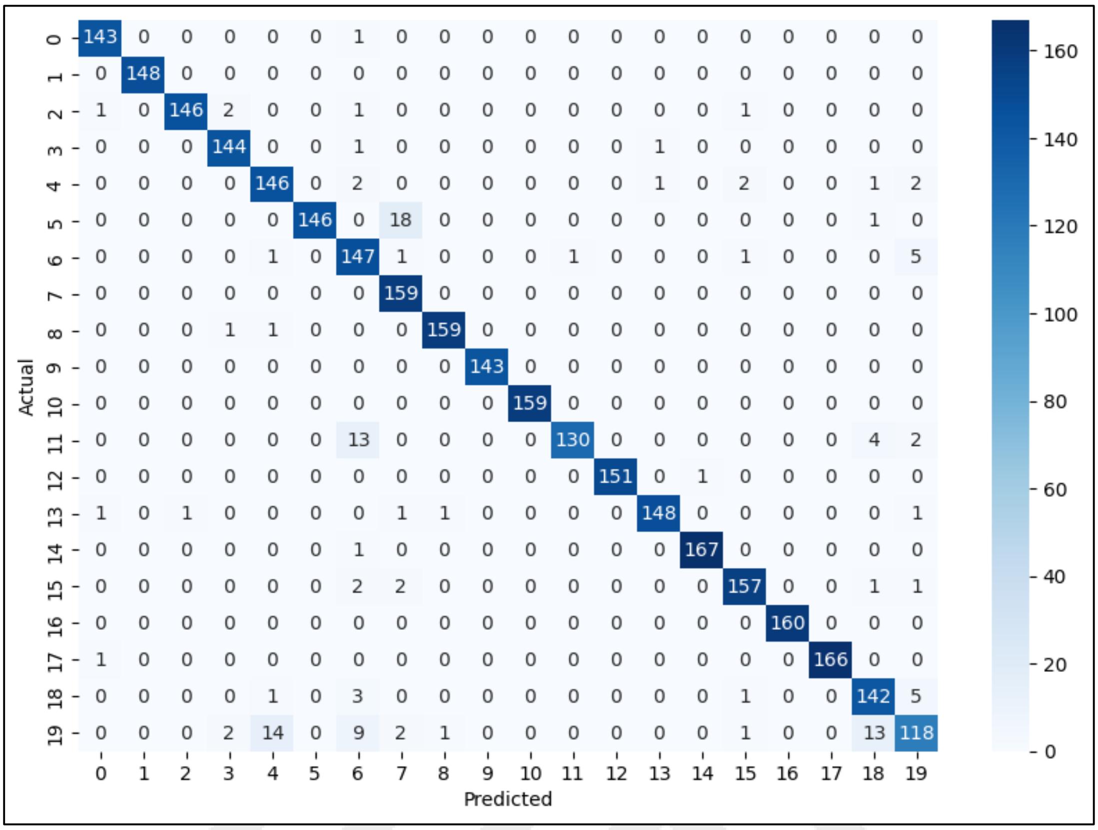


Figure 4.11: MobileNet Confusion Matrix.

The model performs best on "Raspberry___healthy", "Grape___Leaf_blight_(Isariopsis_Leaf_Spot)", "Corn_(maize)_Northern_Leaf_Blight", "Grape___Esca(Black_Measles)", and "Orange___Haunglongbing(Citrus_greening)" classes with a F1-S of 1.00.

The model performs worst on "Tomato___Leaf_Mold" and "Tomato___Early_blight" classes with F1-Ss of 0.87 and 0.80 respectively.

The ACC of the model is 0.96, which means the model correctly predicted the class 96% of the time on the test data.

	precision	recall	f1-score	support
Peach___healthy	0.98	0.99	0.99	144
Raspberry___healthy	1.00	1.00	1.00	148
Blueberry___healthy	0.99	0.97	0.98	151
Apple___Cedar_apple_rust	0.97	0.99	0.98	146
Tomato___Late_blight	0.90	0.95	0.92	154
Pepper,_bell___healthy	1.00	0.88	0.94	165
Tomato___Leaf_Mold	0.82	0.94	0.87	156
Pepper,_bell___Bacterial_spot	0.87	1.00	0.93	159
Potato___Early_blight	0.99	0.99	0.99	161
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1.00	1.00	1.00	143
Corn_(maize)___Northern_Leaf_Blight	1.00	1.00	1.00	159
Tomato___Tomato_mosaic_virus	0.99	0.87	0.93	149
Cherry_(including_sour)___healthy	1.00	0.99	1.00	152
Peach___Bacterial_spot	0.99	0.97	0.98	153
Soybean___healthy	0.99	0.99	0.99	168
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.96	0.96	0.96	163
Grape___Esca_(Black_Measles)	1.00	1.00	1.00	160
Orange___Haunglongbing_(Citrus_greening)	1.00	0.99	1.00	167
Tomato___Target_Spot	0.88	0.93	0.90	152
Tomato___Early_blight	0.88	0.74	0.80	160
accuracy			0.96	3110
macro avg	0.96	0.96	0.96	3110
weighted avg	0.96	0.96	0.96	3110

Figure 4.12: MobileNet Classification Report.

4.2.6 DenseNet Results

Epoch 1: The model started with a relatively high loss of 0.6185 on the training set, but it quickly learned to classify the images with an ACC of 80.46%. On the validation set, the model achieved a loss of 0.2106 and an ACC of 92.82%.

Epoch 2-5: The model continued to improve, reducing the loss on the training set and increasing the ACC. However, on the validation set, while the ACC increased, the loss fluctuated, indicating that the model might be overfitting to the training data. Epoch 6-10: The model achieved its highest validation ACC of 96.87% at epoch 6. The loss on the validation set decreased to its lowest at epoch 10, indicating that the model was still learning useful patterns from the data.

Epoch 11-15: During these epochs, the model's performance on the validation set fluctuated. This could be due to the model starting to overfit the training data, learning patterns that do not generalize well to unseen data. Epoch 16-20: The model's performance continued to fluctuate. The highest validation ACC was achieved at epoch 18 with 97.51%. Final Epoch

(Epoch 20): The model ended with a training ACC of 98.74% and a validation ACC of 97.47%. The validation loss at this point was 0.1324.

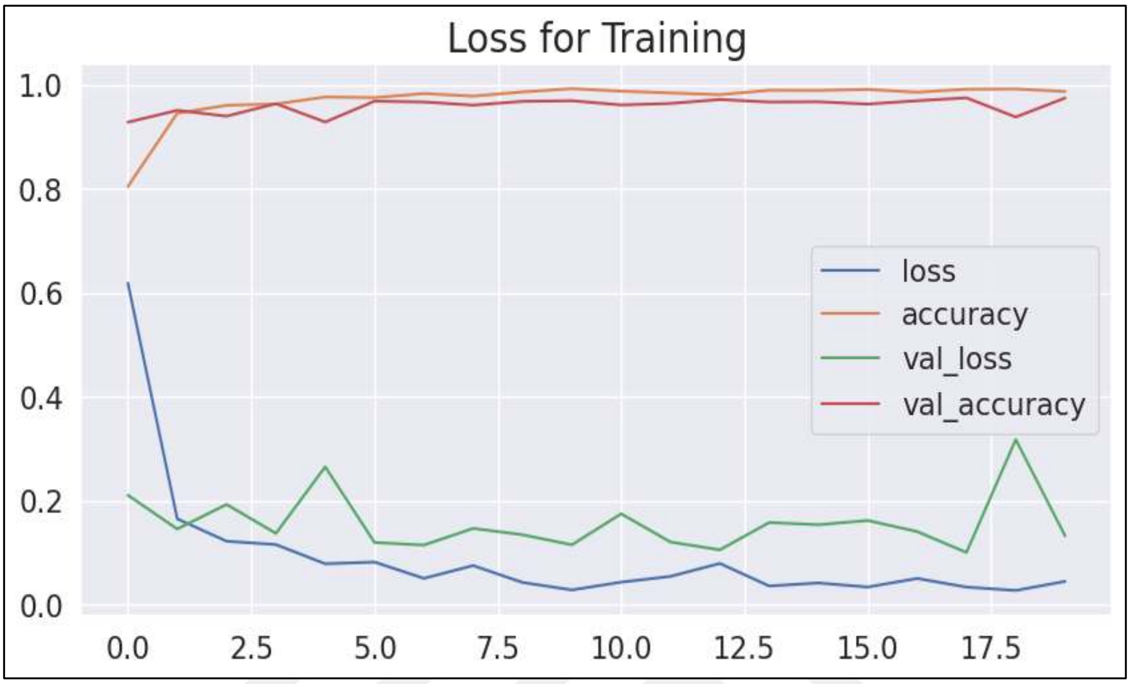


Figure 4.13: DenseNet Training Process.

The CM of DenseNet201 model in figure 4.14 indicates the following results. The model achieved its highest validation ACC of 97.51% in the 18th epoch, with a corresponding validation loss of 0.1004. However, the model's performance on the validation set fluctuated somewhat across epochs, with the validation ACC ranging from a low of 92.82% in the first epoch to the aforementioned high of 97.51%. Similarly, the validation loss ranged from a high of 0.3178 in the 19th epoch to a low of 0.1004 in the 18th epoch. Despite these fluctuations, the overall trend was one of improving performance over time.

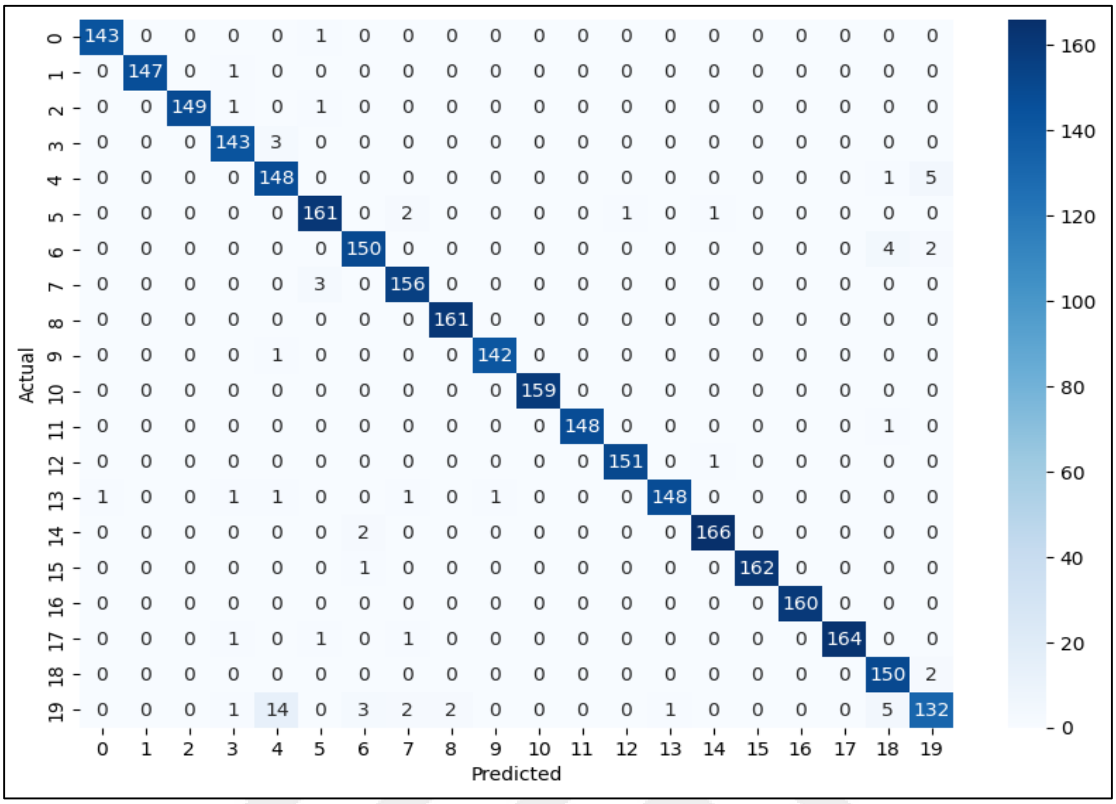


Figure 4.14: DenseNet Confusion Matrix.

In terms of the model's performance on individual classes, the model achieved a PRE, REC, and F1-S of 1.00 for several classes, including Raspberry___healthy, Corn_(maize)__Northern_Leaf_Blight, and Grape___Esca(Black_Measles). This indicates that the model was able to perfectly identify and predict these classes. However, the model's performance was slightly lower for some classes, such as Tomato___Late_blight, where the precision was 0.89, REC was 0.96, and F1-S was 0.92, and Tomato___Early_blight, where the precision was 0.94, REC was 0.82, and F1-S was 0.88. Despite these lower scores, the model's overall performance was quite high, as evidenced by the overall ACC of 98%.

	precision	recall	f1-score	support
Peach___healthy	0.99	0.99	0.99	144
Raspberry___healthy	1.00	0.99	1.00	148
Blueberry___healthy	1.00	0.99	0.99	151
Apple___Cedar_apple_rust	0.97	0.98	0.97	146
Tomato___Late_blight	0.89	0.96	0.92	154
Pepper,_bell___healthy	0.96	0.98	0.97	165
Tomato___Leaf_Mold	0.96	0.96	0.96	156
Pepper,_bell___Bacterial_spot	0.96	0.98	0.97	159
Potato___Early_blight	0.99	1.00	0.99	161
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	0.99	0.99	0.99	143
Corn_(maize)___Northern_Leaf_Blight	1.00	1.00	1.00	159
Tomato___Tomato_mosaic_virus	1.00	0.99	1.00	149
Cherry_(including_sour)___healthy	0.99	0.99	0.99	152
Peach___Bacterial_spot	0.99	0.97	0.98	153
Soybean___healthy	0.99	0.99	0.99	168
Tomato___Tomato_Yellow_Leaf_Curl_Virus	1.00	0.99	1.00	163
Grape___Esca_(Black_Measles)	1.00	1.00	1.00	160
Orange___Haunglongbing_(Citrus_greening)	1.00	0.98	0.99	167
Tomato___Target_Spot	0.93	0.99	0.96	152
Tomato___Early_blight	0.94	0.82	0.88	160
accuracy			0.98	3110
macro avg	0.98	0.98	0.98	3110
weighted avg	0.98	0.98	0.98	3110

Figure 4.15: DenseNet Classification Report.

4.3 VISUAL INTERPRETATION OF DISEASE DETECTION

Within the intricacies of DL, GradCAM stands as a beacon of transparency, meticulously crafted to shed light on class differentiation. It unravels the model's cognitive process, demystifying the specific regions in the input that the layers of the network prioritize during its deliberation and verdict phases. This in-depth perspective is instrumental in demarcating not just the regions the model observes, but also the significance it bestows upon diverse portions of the input. Figure 4.16 provides a compelling visual representation of this principle, displaying the heatmaps constructed via the GradCAM methodology. These heatmaps employ a spectrum of hues, oscillating between red and blue, to convey the model's inferred relevance or weights for different regions. It is the more intense shades, whether deep red or profound blue, that spotlight the locales within the image that the network discerns as repositories of pivotal data. These particular sections predominantly sway the model's final judgment. Embedding this insight within the ambit of our research, centered on identifying ailments in plants, the accentuated zones in these images insinuate the probable segments of the flora exhibiting telltale signs of an affliction. Figure 4.16, by pairing the pristine images alongside their corresponding GradCAM heatmaps, furnishes a visual roadmap. This not only assists researchers and agronomy experts but also equips a broader readership to discern regions that the models earmark as hallmarks of potential disease. Such elucidation fosters a layer of trust in the model's predictive prowess, laying the foundation for enlightened interventions in practical agricultural scenarios.

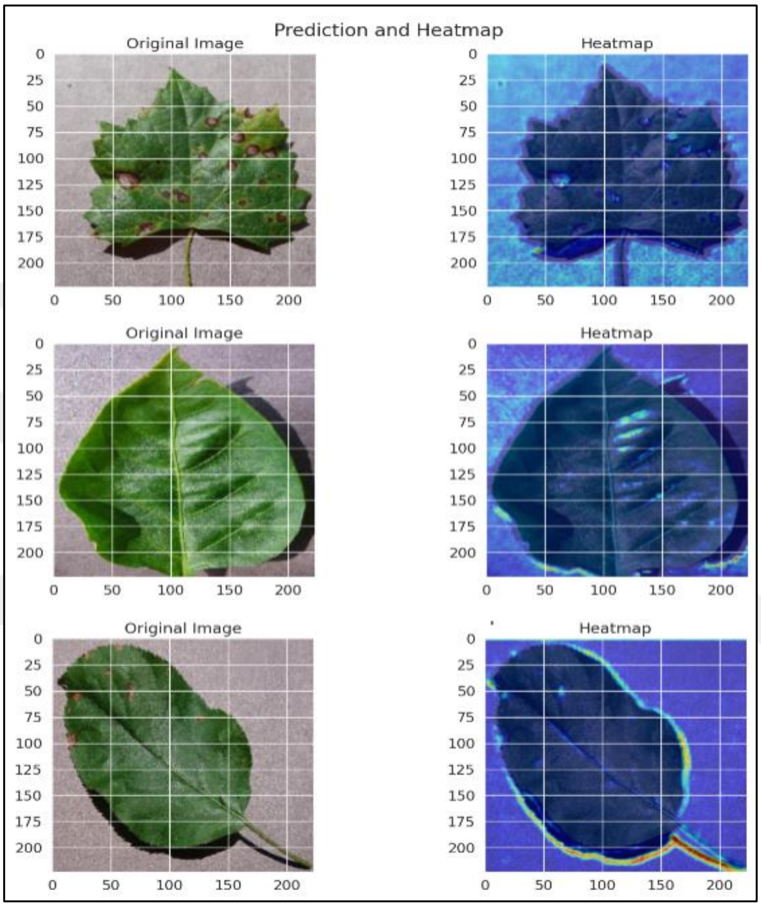


Figure 4.16: Comparative Analysis of Original Plant Images and Corresponding GradCAM Heatmaps Highlighting Disease Indicators.

4.4 RESULTS COMPARISON

The five models you've used are all types of Convolutional Neural Networks (CNNs), but they have different architectures and, as a result, have achieved different levels of ACC on your task.

a. CNN: This is a basic Convolutional Neural Network model. It achieved an ACC of 91%, which is quite good. CNNs are the foundation of most modern image processing models, and this result shows that even a basic CNN can perform well on image classification tasks.

b. VGG: This is a specific architecture of CNN achieved an ACC of 91%, the same as the basic CNN. This might be surprising, as one might expect the more complex model to perform better. However, it's possible that the added complexity of VGG doesn't provide any benefits for this specific task, or it might even be detrimental if it leads to overfitting.
c. ResNet: This is another specific architecture of CNN, known for its "residual learning" approach. It's designed to solve the problem of vanishing gradients, which can occur in very deep networks. However, ResNet only achieved an ACC of 50% on your task, which is significantly lower than the other models. This could be due to a variety of reasons, such as the model not being a good fit for the task, the model not being trained properly, or the model's complexity leading to overfitting.
d. DenseNet: DenseNet is another type of CNN, known for its densely connected layers, which can improve performance and reduce the number of parameters. DenseNet201 achieved the highest ACC of all the models, at 98%. This suggests that MobileNet's architecture is particularly well-suited to this task.
e. MobileNet: known for being lightweight and efficient, while still achieving good performance. MobileNet achieved an ACC of 96%, which is higher than both the basic CNN and VGG. This suggests that the densely connected architecture of DenseNet201 might be beneficial for this task.

In conclusion, while all these models are types of CNNs, their different architectures can lead to different levels of performance. For this particular task, DenseNet201 was the most accurate model with an ACC of 98%.

Table 4.2: Classifiers Results Comparison.

Classifier	ACC
CNN	91%
VGG	91%
ResNet	50%
DenseNet	**98%**
MobileNet	96%

4.5 COMPARISON RESULTS WITH THE LITERATURE

The comparison results with the literature indicate how the proposed DenseNet model's performance stacks up against other models reported in various studies. Table 4.3 indicate these results.

Proposed DenseNet model: The DenseNet model proposed in your study achieved an ACC of 98%. This is the highest ACC among the models compared here, indicating that your DenseNet model performed exceptionally well on the given task.

Agarwal et al. (2020) [11]: The model proposed by Agarwal et al. in 2020 achieved an ACC of 92%. While this is a good ACC, it is lower than the ACC achieved by your DenseNet model, suggesting that your model may have some advantages over the model proposed by Agarwal et al.

Arivazhagan et al. (2018) [14]: The model proposed by Arivazhagan et al. in 2018 achieved an ACC of 95%. This is quite close to the ACC of your DenseNet model, indicating that their model also performed well on the task. However, your DenseNet model still outperformed it by 3 percentage points.

Djenouri et al. (2022) [19]: The model proposed by Djenouri et al. in 2022 also achieved an ACC of 92%, the same as the model by Agarwal et al. Again, while this is a good ACC, it is lower than the ACC achieved by your DenseNet model.

Chompookham et al. (2021) [21]: The model proposed by Chompookham et al. in 2021 achieved an ACC of 91.19%. This is the lowest ACC among the models compared here, and significantly lower than the ACC of your DenseNet model.

In conclusion, the proposed DenseNet model outperformed the models proposed in the compared literature, achieving the highest ACC. This suggests that the DenseNet architecture and/or the specific implementation and training process used in your study may be particularly effective for this task.

Table 4.3: Comparison With the Literature.

Classifier	ACC
Proposed DenseNet model	98%
[Agarwal et al.(2020)]	92%
[Arivazhagan et al (2018)]	95%
[Djenouri et al.(2022)]	92%
[Chompookham et al (2021)]	91.19%

5. CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

PDs pose substantial threats to crop yields and, consequently, to global food security. Therefore, timely and precise identification of these diseases is of paramount importance to ensure effective mitigation strategies. Automation in the sphere of PD detection, leveraging DL models, stands out as a potentially transformative solution to this pressing issue. In our investigation, we not only developed a formidable PD prediction system using cutting-edge DL architectures like CNNs, DenseNet, ResNet, MobileNet, and VGG, but we also integrated XAI methodologies, specifically the Grad-CAM technique, to ensure transparency and interpretability of the model's decisions. This blend of advanced modeling with XAI provides a clear visualization of the model's focus areas in the input images, offering insights into which features the model deems crucial for its predictions. Among the deployed models, DenseNet emerged as the frontrunner, boasting an outstanding ACC rate of 98%. Beyond sheer performance metrics, the integration of XAI empowers users with a clearer understanding of the model's reasoning, enhancing trust in its predictions. Our study reaffirms that, when coupled with XAI, DL models have the potential to set new benchmarks in the realm of PD detection.

5.2 FUTURE WORK

There are still opportunities for advancements and additional research in the field of PD diagnosis and prediction, even with the success of this study. Real-time illness identification in the field can be facilitated by expanding the current system, which could lead to important practical consequences in the future. Farmers can get on-site disease diagnostics and fast advice for managing diseases by having a mobile application or portable device with a trained model. To improve the system's ability to identify things on time and accurately, real-time data collecting could be incorporated through the use of sensors or drone-based imagery.

REFERENCES

[1] Lucas, G. B., Campbell, C. L., & Lucas, L. T. (1992). Introduction to plant diseases: identification and management. Springer Science & Business Media.

[2] World Health Organization. (2020). The state of food security and nutrition in the world 2020: transforming food systems for affordable healthy diets (Vol. 2020). Food & Agriculture Org..

[3] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. Frontiers in plant science, 8, 1852.

[4] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in plant science, 7, 1419.

[5] Xiao, Q., & McPherson, E. G. (2005). Tree health mapping with multispectral remote sensing data at UC Davis, California. Urban Ecosystems, 8, 349-361.

[6] Dubey, S. R., & Jalal, A. S. (2014). Adapted Approach for Fruit Disease Identification Using Images. arXiv preprint arXiv:1405.4930.

[7] Yang, X., & Guo, T. (2017). Machine learning in plant disease research. March, 31, 1.

[8] Suresha, M., Shreekanth, K. N., & Thirumalesh, B. V. (2017, April). Recognition of diseases in paddy leaves using knn classifier. In 2017 2nd International Conference for Convergence in Technology (I2CT) (pp. 663-666). IEEE.

[9] Madhulatha, G. and Ramadevi, O. (2020) 'Recognition of Plant Diseases using Convolutional Neural Network', in 2020 Fourth International Conference on ISMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 738–743. doi: 10.1109/I-SMAC49090.2020.9243422

[10] Hatuwal, B. K., Shakya, A. and Joshi, B. (2020) 'Plant Leaf Disease Recognition Using Random Forest, KNN, SVM and CNN', POLIBITS, 62, p. 7.

[11] Agarwal, M. et al. (2020) 'ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network', Procedia Computer Science, 167, pp. 293–301. doi: 10.1016/j.procs.2020.03.225.

[12] Sardogan, M., Tuncer, A., & Ozen, Y. (2018, September). Plant leaf disease detection and classification based on CNN with LVQ algorithm. In 2018 3rd international conference on computer science and engineering (UBMK) (pp. 382-385). IEEE.

[13] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.

[14] Arivazhagan, S., & Ligi, S. V. (2018). Mango leaf diseases identification using convolutional neural network. International Journal of Pure and Applied Mathematics, 120(6), 11067-11079.

[15] Barbedo, J. G. (2018). Factors influencing the use of deep learning for plant disease recognition. Biosystems engineering, 172, 84-91.

[16] Uğuz, S., & Uysal, N. (2021). Classification of olive leaf diseases using deep convolutional neural networks. Neural computing and applications, 33(9), 4133-4149.

[17] Shruthi, U., Nagaveni, V., & Raghavendra, B. K. (2019, March). A review on machine learning classification techniques for plant disease detection. In 2019 5th International conference on advanced computing & communication systems (ICACCS) (pp. 281-284). IEEE.

[18] Tran, T. T., Choi, J. W., Le, T. T. H., & Kim, J. W. (2019). A comparative study of deep CNN in forecasting and classifying the macronutrient deficiencies on development of tomato plant. Applied Sciences, 9(8), 1601.

[19] Djenouri, Y., Belhadi, A., Yazidi, A., Srivastava, G., & Lin, J. C. W. (2022). Artificial intelligence of medical things for disease detection using ensemble deep learning and attention mechanism. Expert Systems, e13093.

[20] Saberi Anari, M. (2022). A hybrid model for leaf diseases classification based on the modified deep transfer learning and ensemble approach for agricultural aiot-based monitoring. Computational Intelligence and Neuroscience, 2022.

[21] Chompookham, T., & Surinta, O. (2021). Ensemble methods with deep convolutional neural networks for plant leaf recognition. ICIC Express Letters, 15(6), 553-565.

[22] Sam, A. (2020). Phytopathology – What it is, history and importance https://notesread.com/phytopathology-history-importance /

[23] Fenu, G., & Malloci, F. M. (2021). Forecasting plant and crop disease: an explorative study on current algorithms. Big Data and Cognitive Computing, 5(1), 2.

[24] Van der Plank, J. E. (1963). Plant diseases. Elsevier Science.

[25] Dangl, J. L., & Jones, J. D. (2001). Plant pathogens and integrated defence responses to infection. nature, 411(6839), 826-833.

[26] Maritan-Thomson, D. (2023, February 27). The rise of fungi: The last of Us. MDPI Blog. https://blog.mdpi.com/2023/02/27/rise-of-fungi/

[27] Gunturk, B., & Li, X. (2018). Image restoration. CRC Press.

[28] Barnsley, M. F., & Hurd, L. P. (1993). Fractal image compression. AK Peters, Ltd.

[29] Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. Machine learning techniques for multimedia: case studies on organization and retrieval, 21-49.

[30] Barlow, H. B. (1989). Unsupervised learning. Neural computation, 1(3), 295-311.

[31] Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of artificial intelligence research, 4, 237-285.

[32] Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

[33] Torres, R., Ohashi, O., & Pessin, G. (2019). A machine-learning approach to distinguish passengers and drivers reading while driving. Sensors, 19(14), 3174.

[34] Noble, W. S. (2006). What is a support vector machine?. Nature biotechnology, 24(12), 1565-1567.

[35] Quinlan, J. R. (1987). Simplifying decision trees. International journal of man-machine studies, 27(3), 221-234.

[36] Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. ISPRS journal of photogrammetry and remote sensing, 114, 24-31.

[37] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[38] Sa, Inkyu et al. 2016. “DeepFruits: A Fruit Detection System Using Deep Neural Networks.” *Sensors* 16(8): 1222. http://www.mdpi.com/1424-8220/16/8/1222.

[39] McQuin, Claire et al. 2018. “CellProfiler 3.0: Next-Generation Image Processing for Biology” ed. Tom Misteli. *PLOS Biology* 16(7): e2005970. https://dx.plos.org/10.1371/journal.pbio.2005970.

[40] Xin, Yang et al. 2018. “Machine Learning and Deep Learning Methods for Cybersecurity.” *IEEE Access* 6: 35365–81. https://ieeexplore.ieee.org/document/8359287/.

[41] Levie, Ron, Federico Monti, Xavier Bresson, and Michael M. Bronstein. 2019. “CayleyNets: Graph Convolutional Neural Networks With Complex Rational Spectral Filters.” *IEEE Transactions on Signal Processing* 67(1): 97–109. https://ieeexplore.ieee.org/document/8521593/.

[42] Cao, Chensi et al. 2018. “Deep Learning and Its Applications in Biomedicine.” *Genomics, Proteomics & Bioinformatics* 16(1): 17–32. https://linkinghub.elsevier.com/retrieve/pii/S1672022918300020.

[43] Simonyan, Karen and Zisserman, Andrew. 2014. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” *arXiv preprint arXiv:1409.1556*.

[44] Jiang, Zhi-Peng, Yi-Yang Liu, Zhen-En Shao, and Ko-Wei Huang. 2021. “An Improved VGG16 Model for Pneumonia Image Classification.” *Applied Sciences* 11(23): 11185. https://www.mdpi.com/2076-3417/11/23/11185.

[45] Tao, Yiting, Miaozhong Xu, Zhongyuan Lu, and Yanfei Zhong. 2018. “Dense Net-Based Depth-Width Double Reinforced Deep Learning Neural Network for High-Resolution Remote Sensing Image per-Pixel Classification.” *Remote Sensing* 10(5).

[46] Zhang, Zhicheng et al. 2018. “A Sparse-View CT Reconstruction Method Based on Combination of DenseNet and Deconvolution.” *IEEE Transactions on Medical Imaging* 37(6): 1407–17. https://ieeexplore.ieee.org/document/8331861/.

[47] Xu, Yongyang, Zhong Xie, Yaxing Feng, and Zhanlong Chen. 2018. “Road Extraction from High-Resolution Remote Sensing Imagery Using Deep Learning.” *Remote Sensing* 10(9): 1461. http://www.mdpi.com/2072-4292/10/9/1461.

[48] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. "Deep Residual Learning for Image Recognition." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem: 770–78.

[49] Picon, Artzai et al. 2019. "Deep Convolutional Neural Networks for Mobile Capture Device-Based Crop Disease Classification in the Wild." *Computers and Electronics in Agriculture* 161: 280–90. https://linkinghub.elsevier.com/retrieve/pii/S0168169917312619.

[50] Torrey, L., & Shavlik, J. (2010). Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242-264). IGI global.

[51] Maini, R., & Aggarwal, H. (2010). A comprehensive review of image enhancement techniques. arXiv preprint arXiv:1003.4053.

[52] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).

[53] Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2921-2929).

[54]

[55] Shrivastava, V. K., Pradhan, M. K., Minz, S., & Thakur, M. P. (2019). Rice plant disease classification using transfer learning of deep convolution neural network. International archives of the photogrammetry, remote sensing & spatial information sciences, 3(6), 631-635.

[56] Sunil, C. K., Jaidhar, C. D., & Patil, N. (2022). Binary class and multi-class plant disease detection using ensemble deep learning-based approach. International Journal of Sustainable Agricultural Management and Informatics, 8(4), 385-407.

[57] Vallabhajosyula, S., Sistla, V., & Kolli, V. K. K. (2022). Transfer learning-based deep ensemble neural network for plant leaf disease detection. Journal of Plant Diseases and Protection, 129(3), 545-558.

[58] Vats, S., & Chivukula, V. N. A. D. (2022, December). Plant Disease Detection Using DeepNets and Ensemble Technique. In 2022 IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT) (pp. 1-6). IEEE.

[59] O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.