COMPARATIVE ANALYSIS OF MACHINE LEARNING AND DEEP LEARNING MODELS FOR PLANT DISEASE DETECTION

Thesis Submitted

in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

Bioinformatics

by

Elesweta Sahoo
(Roll No. 23/BIO/10)

Under the Supervision of
Dr. NAVNEETA BHARADVAJA

Department of Biotechnology

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-110042. India

May, 2025

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE’S DECLARATION

I Elesweta Sahoo hereby certify that the work which is being presented in the thesis entitled Comparative Analysis of Machine learning and Deep learning Models for Plant disease detection in partial fulfilment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of Biotechnology, Delhi Technological University is an authentic record of my own work carried out during the period from January to May 2025 under the supervision of Dr. Navneeta Bharadvaja.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that **Elesweta Sahoo (23/BIO/10)** has carried out their search work presented in this thesis entitled **"Comparative Analysis of Machine Learning and Deep Learning for Plant disease detection"** for the award of **Master of Technology** from Department of Biotechnology, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Dr. Navneet Bharadvaja

Associate Professor

Department of Biotechnology

Delhi Technological University

Shahbad Daulatpur, Main Bawana Road,

Delhi-110042. India

Date:

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis EVALUATION OF PHYTOCHEMICALS FROM TINOSPORA CORDIFOLIA AGAINST ONCOGENIC AND RESISTANCE-ASSOCIATED TARGETS IN LOW-GRADE SEROUS OVARIAN CARCINOMA

Total Pages: 105 Name of the Scholar: Elesweta Sahoo

Supervisor

(1) Dr Navneeta Bharadvaja
(2)____
(3)____

Department

This is to report that the above thesis was scanned for similarity detection. Process and outcome are given below:

Software used: Similarity Index: _______, Total Word Count: _________

Date: ____________

Candidate's Signature **Signature of Supervisor**

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Dr. Navneeta Bharadvaja, for her exceptional guidance, support, and encouragement throughout the course of this research. Her expertise, valuable insights, and thoughtful feedback have been instrumental in the successful completion of this thesis. I am especially thankful for her patience, constructive criticism, and consistent motivation, which have helped me stay focused and improve the quality of my work at every stage.

Her mentorship has not only enriched my academic growth but has also inspired me on a personal level. I truly appreciate the time and effort she dedicated to our discussions, and the clarity she brought to complex concepts through her thorough understanding of the subject matter.

I am also thankful to the faculty members and staff of Department of Biotechnology, Delhi Technological University, for their support and for providing a positive and resourceful academic environment. Their assistance, both direct and indirect, contributed significantly to the progress of this work.

Elesweta Sahoo

23/BIO/10

ABSTRACT

Diseases in plants remain a serious hazard to the world’s food supply, farming results and environmental protection, especially in farming-reliant areas. Diagnosing plant diseases using old methods mostly involves looking at plants and this can be slow, subjective and easily incorrect. Because more people are being born, there is more demand for food which makes crop health and lower losses to diseases extremely important. Using pesticides as pest controls endangers health and damage to the environment and the growing resistance of pathogens has weakened their effectiveness. So, fast and accurate methods for detecting plant diseases are urgently needed.

In modern developments in AI, ML and DL, it is now possible to use computers to identify plant diseases through digital image processing. SVM, RF, DT and GB are promising at classifying plants from their leaf images. These simple models depend on color, texture and shape brought out by HOG and LBP algorithms. However, although these methods are easy to use and explain, they need domain specialists to create features by hand and often fail with complex visual patterns.

CNNs and similar models deliver results regardless of visual differences, since they can discover needed patterns right from the unprocessed images. Examples of these architectures such as VGGNet, ResNet, AlexNet and EfficientNet, have proven better at detecting and naming many plant diseases on datasets like Plant Village. With these models, it’s possible to reuse networks already trained without having much data in your domain. The use of flipping, rotation and brightness adjustment makes models work better and helps them avoid overlearning.

In this study different models of DL & ML approaches are tested and assessed for infected plant detection using the Plant Village data in this research. The Objective was to discover the model that best and generally recognized plant leaf diseases through image data. Test accuracy was highest with 99% for the Random Forest classifier and was followed by the DT with 96% and GB with 95%. While the training accuracy for the SVM was high at 96%, its test accuracy fell to just 85%. As part of DL, popular training models were transferred and used in the field of deep learning. ResNet50 performed the best, reaching 96.8% test accuracy, while VGG16 had 95.2% and AlexNet came in at 94.1%. They worked well even with new, unseen data, as a result of using extra data and fine-tuning.

Generally, the findings indicated that Ensemble Classifiers and CNN-based CNNs offer the best accuracy when using ML and DL, respectively. Using both DL and ML together will play a major role in the future of agriculture. As a result, they help to automatically recognize and identify plant diseases, aiding the practice of precision farming. Thanks to new developments in data gathering, understanding models and running them on phones, these technologies will greatly help with real-time monitoring and crop health care.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

List of Abbreviations

AI	Artificial intelligence
ML	Machine Learning
DL	Deep Learning
SVM	Support Vector Machine
RF	Random Forest
DT	Decision Tree
GB	Gradient Boosting
CNN	Convolutional neural network
RestNet	Residual Network
HSV	Hue Saturation Value
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
RGB	Red Green Blue
DBNs	Deep Belief Network
EV	Expected Value
GLAM	Gray Level Co- Occurrence
RMS	Root Mean Square
SML	Supervised Machine Learning
USML	Unsupervised Machine Learning
RL	Reinforcement Learning
TL	Transfer Learning
LDD	Leaf Disease Detection
PD	Plant Disease
RestNet50	Residual Network 50
RestNet-101	Residual Network 101
RestNet - 152	Residual Network 152
IP	Image Preprocessing
FE	Feature Extraction
DA	Data Augmentation

RC	Roc Curve
CR	Classification Report
AS	Accuracy Score
AUC	Area under the curve
ReLU	Rectified linear unit
CM	Confusion Matrix
TA	Training Accuracy
TA	Testing accuracy
VA	Validation Accuracy
DBMs	Deep Boltzmann Machine
EL	Entropy Loss
GAP	Global Average pooling
PDD	Plant Disease Detection

CHAPTER I
INTRODUCTION

In world record Plant diseased outbreak have large scale of suggestive challenges where the world facing many challenges where Plant disease causing problems in agriculture for farmers. For decades, plant disease outbreaks have been a serious problem for agriculture endangering food security and representing an average global yield loss. The minimisation of damage caused to crops with pests and pathogens is important for future demand (Upadhyaya et al., 2024). Conventional chemical methods used in plant protection against disease have their environmental and health concerns. In addition, pesticide-resistant pathogens have intensified the ineffectiveness of these measures. More sustainable crop disease management is possible by knowledge and exploitation of the plant's intrinsic disease resistance mechanisms. Resistance to disease is of the two types below:- quantitative and qualitative.This disease is influenced by genes that interact with the other genes and the environment to lower disease intensity possibly influencing other traits. On the other hand qualitative disease resistance otherwise called monogenic disease resistance is controlled by single genes. Knowledge of the mechanisms of plant disease resistance and the genetic determinants will give enough information to breeders to formulate strategies of breeding resistant plants hence increasing productivity and sustainable agriculture. Methods are suggested with molecular breeding that includes marker which assist the individuals that can be selected the identification of individuals that carried with beneficial alleles that are chosen for breeding. (Upadhyaya et al., 2024)

Agriculture has always been of great need to humans since human existence because plants were the first source of food. Nowadays, agriculture is still perceived as one of the key food resources in people's lives and is the core of several spheres of human life. The truth is even in the developing countries; the economy of many countries depends on agriculture as a pillar. Agriculture matters a great deal because about 70% of people find employment in plant cultivation and similar fields. (Upadhyaya et al., 2024).

The more people for food every day. In nations reliant on farming, it is recognized that farm crops require protection from any leaf diseases. The largest reasons behind decreases in both quantity and quality of production in agriculture. Such losses lead to higher production costs for agricultural companies and lower revenue profits. True, we have not yet produced enough accurate and speedy methods of identification. As plant diseases go unnoticed, food shortages will continue to increase. The faster plant infections can be identified, the better can we fight and cure them. A number of investigations are in progress to support pest management methods and reinforce traditional pesticide use. An automatic method of classifying plant diseases is useful for identification. The management and control of agriculture often consider plant diseases as important factors. Lately, it has grown essential to quickly identify plant diseases using automatic methods. Disease in crops endangers the world’s food supply, it is detrimental to plants and speedy recognition of multiple issues is difficult. Lesions or distinct marks will often be seen on leaf/ stem and flowers, stalks and fruits of affected plants. If you step back, it’s usually possible to tell anomalies apart from each other by their unique visual patterns. Plant leaves usually show a variety of symptoms when a plant is ill, making them a key part of figuring out what disease it may have (Upadhyaya et al., 2024).

Agriculturalists have some reasons best known to them relied on their eyes to determine type of illnesses and form opinion based on their previous topic of plant. The variety of the plants means that various crops also will have a variety of disease characteristics which further complicates the categorization of plant ills. Also, knowledge of agriculturists and farmers must also be handed down from father to son. Nevertheless, both shape and color of leaf crown are still predominant aids in screening of plants for disease. Human can only identify plant diseases relying on their experience and close analysis of the symptoms for the disease on plant leaves after spending time and applying effort, and direct knowledge. With the broad year plants are reflected by more varied crop diseases and therefore further complication of plant diseases classification arises. In addition, the false prediction of plant diseases leads to excessive use of pesticides, which increases the price of manufacturing. On the basis of such facts, it is important to create a reliable disease dependable disease identification connected to an own database to help farmers, particularly young inexperienced farmers. Researchers have come up with new methods of establishing plant diseases using the image process in order to solve these problems. This is now the first priority according to the research agenda. Pests and Diseases can destroy the crops or part of the plant hence low amount of food generated hence food insecurity (Jafar et al., 2024).

Machine learning is a new area that can improve performance and interpretation of a trait association. This has been made possible with technological developments that allow us to obtain larger and varied datasets that can be used in terms of trait association. Machine learning can analyze all genetic data like alleles, identify Genes associated with traits like disease resistance.
It can not only predict the efficacy of genes in plants, which are key to defense against pathogens, but can also reveal the complex interaction between plants and pathogens. This review lists the disease resistance prediction in plants via machine learning including examples on applications of machine learning for predicting agronomic traits. It also involves the very latest progress and future options for increasing plant disease resistance via methodologies of genomics and machine learning. Recent methods including ML and DL algorithms have been used to boost the recognition rate as well as the accuracy of the results. (Mazumder et al., 2024). There are numerous steps conducted through machine learning under the heading of plant disease detection and diagnosis with ML and DL types and models as the traditional machine learning approach. The field of machine learning is an emerging one to boost the performance as well as the interpretation of trait association. This has been made possible with technological developments that allow us to obtain larger and varied datasets that can be used in terms of trait association. It allows us to see how well

certain genes work in plants that are vital to fighting pathogens and the close links between plants and pathogens. (Upadhyaya et al., 2024).

This research aims at transforming plant health monitoring using AI and ML technologies to make plant disease detection and classification more efficient, accurate and automatic. This research shows the necessary image-based strategies, investigating the sophisticated algorithms and computational models which can apply analysis of plant leaf images to determine the disease with exactitude. Artificial Intelligence has a type which is ML and DL fields where it became quite easy, cost friendly and effective for infected plant detection the occupation with accuracy, reliability, scalability, efficiency and efficacy. AI-driven plant detection is the identification and classifying of plant species, parts (like leaves, or flowers) or health based on image or sensor data that is automated. This technology is applicable in precision agriculture, biodiversity monitoring and plant pathology (Demilie, 2024).

Advances in digital technology and the emergence of both deep and machine learning presented itself as a blessing across many areas, most significantly medicine (Iniyan et al., 2020). AI-based plant detection has challenges associated with data imbalance, generalization across species and understanding of DL models. These demands the use of enhanced architectures, besides better transfer learning strategies, domain specific augmentation, as well as explainable ai. An aim here is to find and devise high-performance and lightweight yet high-precision plant detection systems that can be used in a wide variety of agricultural applications with low computation cost. Dice coefficients, for instance, are calculated based on foundational processes such as image preprocessing, segmentation, and feature extraction, which support both shallow and deep learning architectures. (Jafar et al., 2024).

Through the use of publicly available datasets such as Plant Village where there are various plant species and diseases, this study conducts data augmentation, transfer learning, and hyperparameter optimization in an attempt to increase model robustness and scalability. Pre-trained networks like DL types are being fine-tuned to enhance classification accuracy while reducing computational costs. Pre-processing methods such as noise reduction, normalization, and transformations of the color space provide for the best value of input quality for training models, Y-state segmentation methods such as edge detection and clustering allow to draw the outline of the diseased areas precisely. While AI refers to the wide gamut of techniques that mimics intelligence that of human, both reasoning and problem solving, machine learning is particularized to computers learning and acting on data. The splicing of a dataset happens when the model is getting trained and test sets. The dataset for training is trained by the model to make predictions and the performance of the model in turn evaluated using test dataset. When it comes to supervised learning models, they require labelled training datasets in order to figure out the mapping from inputs to outputs, while unsupervised ones analyses unlabeled data in search of hidden patterns.

The performance and efficiency of model are optimized by tuning the parameter and structure of model and the how the model is behaving on the test set was evaluated with several metrics such as all scores of Dl models. MLs algorithms are an excellent choice for the analysis of complex agronomic datasets which can include many variables. The action of models of these algorithms can be used to extract patterns, learn from previous data to develop models that may predict agronomic traits. Many recent studies have given testimony to the ability of ML in field of agriculture. Introduced deep learning method named deep neural network genomic prediction (DNNGP), which surpassed both traditional linear regression models and other methods of machine learning for the prediction of agronomic traits based on multi-omics data in plants. Jointly developed crop modelling and machine learning, but generated an RF model using normalized difference vegetation index and climate variables to enhance yield forecasts for winter wheat and oilseed (Maniyath et al., 2018). No predictive model whether or not it incorporates machine learning is not based on good datasets for training. These datasets can be harvested from varied sources like field observation, weather data and remote sensing, soil nutritional profiles and genetic information. The important part of the preprocessing is when the data is cleaned, missing values are managed and some feature extensions and more importantly when the model are performed and optimised and reducing the bias. By extension, splitting of data between training and validation sets, together with techniques like cross-validation and hold- out validation will help validate model accuracy as well as mitigate risk of over-fitting or under-fitting model. Various pieces have focused on classifying plant diseases using machine learning. The bulk of the ML research has been centered around categorizing diseases of plant based on its leaf image features; type, color, and texture. There are three major methods of ML types and methods (Maniyath et al., 2018).

Shallow Architectures which is Limited architectures which fall under traditional machine learning like RF, SVM, GB, and DT are very much reliant on the pre created feature extraction methods. There is manual feature creation that can quite well capture the data set by use of these architectures. Generally, the features used are as follows: HSV captures changes in color; HOG is used to encode shape and texture while LBP calculates differential texture attributes at the pixel level and Red-Green-Blue (RGB) then basic color distributions are deciphered from the above. Steps of workflow for implementing shallow learning models of ML for PDD include the following: Emergency Room is an example of an unsupervised learning application. (a) Data Acquisition and Labeling: Pictures are taken in and segregated into diseased and healthy sets. (b) Pre-Processing: Other pre-processing techniques such as Noise filtering and Contrast stretching provide an accurate batch of data. Feature Extraction: Features are manually selected and then extracted based on the knowledge of the field in question. (d) Model Training and Testing: A classifier for patterns and validation for noise is developed based on the extracted features (Jafar et al., 2024). Shallow techniques are computationally effective compared to deep architectures, and designing the appropriate features is not always easy whereas their performance is proportional to both data size and data’s complexity. The deep Architectures where we learn the models of DL namely (CNNs) do not only learn the features of the images but also learn features from raw images themselves on the other hand. This ensures no manual feature design is required and this makes deep architectures superior at picking complex patterns and variations for example: Ambiguity and unsteady lighting, structures and gradients. Small differences may have signs of some variety of diseases in various plants. In Artificial intelligence and machine learning courses various tools like various types of models and training and testing sets will be used for plant detection which is used for pathogen free agriculture to provide for effective observation for productivity and sustainable agriculture.

CHAPTER II
LITERATURE REVIEW

Agricultural biodiversity is critical to the production of food and biotic materials by mankind and an integral part of the human civilization. Fungi, Bacteria and nematodes are major pathogenic organisms which can affect plant with diseases where environmental factors like checking soil PH, where the temperature is extremes are continually damaging a plant. That functioning and structure, their cultivators. still rely on hand methods of detecting ailments and classifying them because they can hardly can this early and hence, they waste their outputs. How much produce is generated in Agriculture determines the economy. Therefore, the disease in plants is crucial needed because it may affect plant and it will decrease the productivity, quality and quantity of the respective products. When symptoms manifest on plant leaves, automatic disease detection and categorization detects and classifies them, reducing the quantity of labor required for monitoring huge number of crops. From disease in the leaf is a big problem in rice generation, and this disease can damage the crop leading to the decrease of the products. It is hard to detect the type of disease in a leaf. There are many ways to detecting the plant diseases but there is no accurate lead for sustainable decreasing in output of agriculture which is based on physical observation. Plant diseases first attack the leaf then spread the plant infecting it completely thus leaving it poor in quality, poor in yield. New developments in DL have led to innumerable approaches for analyzing the infected plants based on images made from infected plants. Identifying and classifying infected plants at the very early stage is key to improving agricultural productivity (Elaraby et al., 2021).

AI helps greatly increase crop yield because it allows for the early detection of leaf diseases in plants. It raises the crop yield and leads to better diagnoses of plant diseases which help in betterment of cultivation. Farming is crucial for every country and a problem with crop health will affect both food and the nation’s economy.

At the same time, deciding on the appropriate image processing method is complicated due to the variety found in data. It is typically not possible to reach accurate results in plant disease detection without using numerous and diverse image datasets, so sophisticated methods are needed, including CNN. When dealing with significant and complicated data, these models work effectively and increase the accuracy of classifying input records.

Using image processing methods allows you to improve an image and pull-out important details, so they are important for many areas. In agriculture, image processing helps carry out jobs like color analysis, pattern identification and remote sensing. If used well, these approaches can help discover plant diseases early, avoiding serious crop damage. By using computer technology to manage images, agriculture can help secure environmental and social health. (Iniyan et al., 2020).

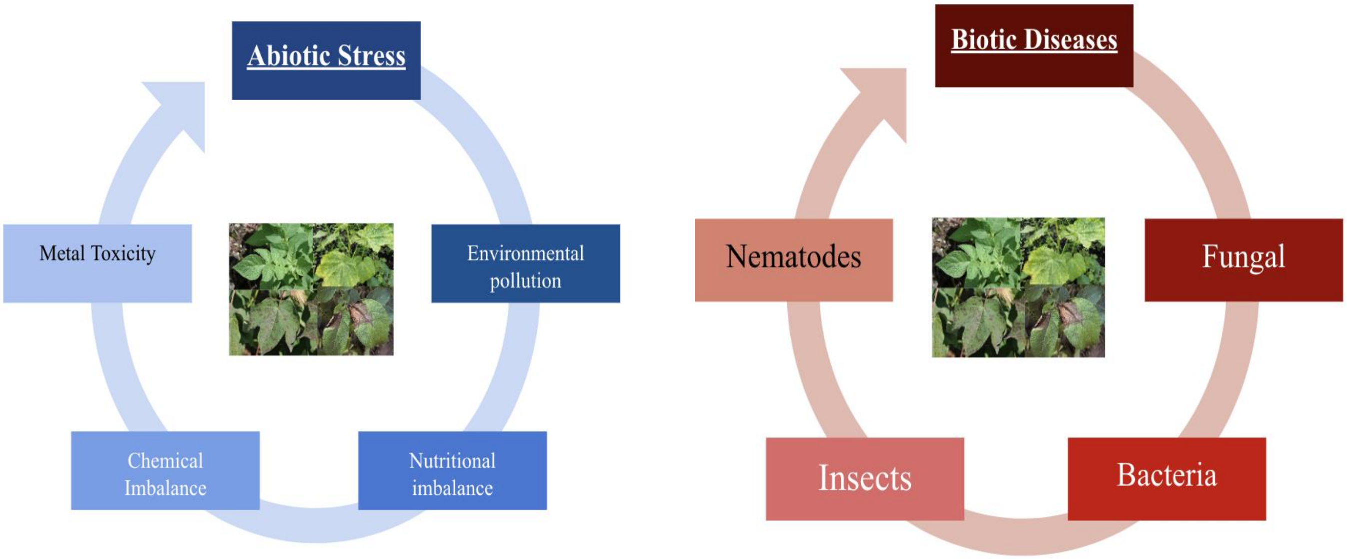


Figure 1.1 Comparison of Abiotic Stress and Biotic Stress

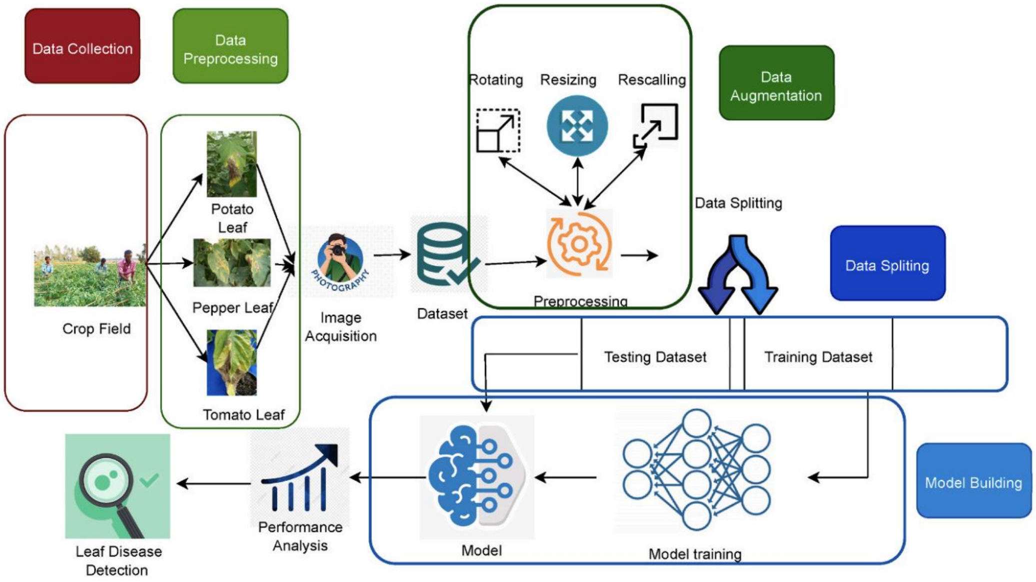
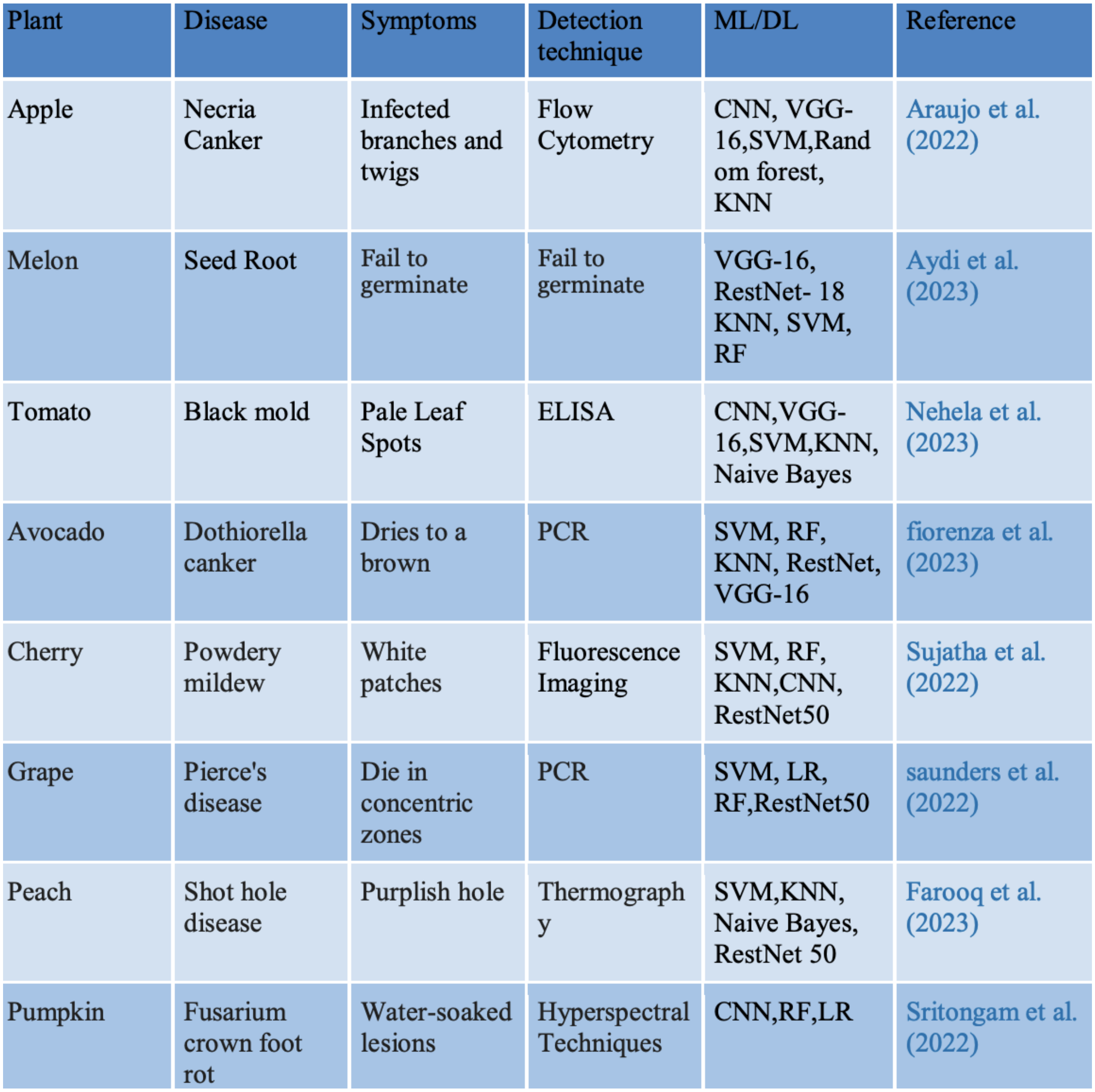


Figure 1.2 Crop AI detection

Table No-1 "Plant Diseases, Symptoms, Detection Methods, and ML/DL Approaches"



Plant	Disease	Symptoms	Detection technique	ML/DL	Reference
Apple	Necria Canker	Infected branches and twigs	Flow Cytometry	CNN, VGG-16,SVM,Random forest, KNN	Araujo et al. (2022)
Melon	Seed Root	Fail to germinate	Fail to germinate	VGG-16, RestNet- 18 KNN, SVM, RF	Aydi et al. (2023)
Tomato	Black mold	Pale Leaf Spots	ELISA	CNN,VGG-16,SVM,KNN, Naive Bayes	Nehela et al. (2023)
Avocado	Dothiorella canker	Dries to a brown	PCR	SVM, RF, KNN, RestNet, VGG-16	fiorenza et al. (2023)
Cherry	Powdery mildew	White patches	Fluorescence Imaging	SVM, RF, KNN,CNN, RestNet50	Sujatha et al. (2022)
Grape	Pierce's disease	Die in concentric zones	PCR	SVM, LR, RF,RestNet50	saunders et al. (2022)
Peach	Shot hole disease	Purplish hole	Thermography	SVM,KNN, Naive Bayes, RestNet 50	Farooq et al. (2023)
Pumpkin	Fusarium crown foot rot	Water-soaked lesions	Hyperspectral Techniques	CNN,RF,LR	Sritongam et al. (2022)

2.1 AI

AI branch of CS is a creation to develop machines that can enable other machines to do things which generally require human intelligence. AI is a collective name of different techniques including ML, DL and NLP. Large Language Models (LLMs) are an AI algorithm that utilizes deep learning powers and very massive data sets in order to comprehend, summarize, create new, and predict new text-based content. LLMs is basically text-based context which is designed for summary, translation, rewriting, classification and analyses the statement. It is versatile to different ways of NLPs. A subpart of AI, NLP main focus is on the computer includes different methods including: reading of texts, detection of the sentiment of the text, recognition, and transformation of the speech, and machine translation. In recent years AI has had a rapid change since the advent of rule-based systems way back to the new era of ML and deep learning algorithms.

Now, AI is changing healthcare, finance, transportation, and various other realms and the impact of AI is only just beginning to increase. In academia basically use of AI is like tutoring the system which is for personalized benefits and the computer program is developing new way to gain knowledges. In research AI has been used to analyze large datasets, finding patterns, which would be very hard to spot for people. This has resulted in breakthroughs in such areas as genomics and drug discovery. AI has been utilized in healthcare institutions to create diagnostic tools and individualized treatment plans. As AI develops further, it will become of utmost importance to develop it responsibly for the good of everyone. In spite of medical progress, the well-designed disease diagnosis is still a problem worldwide because of the complexity of symptoms and underlying mechanisms.

AI, specifically ML and DL, offers enormous various strength to transform diagnostics of health care. The ML algorithms can control workflows, support decision making, and automations by learning from big data sets. Using complex medical data, deep learning models such as CNNs, are good at detecting patterns of various diseases (Iniyan et al., 2020). AI based applications have shown heights of promising results in different kinds of disease. For example, a study proved that AI minimized false positives and negatives in breast cancer diagnosis on a mammogram. More importantly, AI performed better than radiologists in diagnosing breast cancer in early stages. Similarly, dermatologist level accuracy was obtained from CNNs in diagnosing skin cancer. Some other successful applications are detection of diabetic retinopathy, prediction of cardiovascular disease and detection of pneumonia from chest X-rays. AI has also been applied in predicting acute appendicitis where analogs such as Random Forest were used and got over 83% accuracy. These results may imply possible application in recognizing infectious diseases, such as COVID-19 from blood or image data. The benefits of AI are higher accuracy, cost-effectiveness, lower human error and real time support.

In clinical laboratories, AI adds diagnostic antennae, therefore, accelerating diagnostic activities. ML is used for identification of microorganisms, for analysis of genomic data and gram stain classifications with high accuracy. AI also automates processes such as the blood culture and the susceptibility test thereby making early treatment decisions, particularly in infectious diseases such as Malaria. AI is reshaping emergency room through the optimization of patient triage, optimization of use of resources and facilitating quicker and more accurate diagnoses. AI tools are capable of detecting high-risk case, prevent unnecessary visits, and provide treatment decisions using real time data. They help control the surging demand, enhance quality of patient care, reduce diagnostic errors; these errors frequently correlate with higher mortality and extended hospitalization. Altogether, AI-based diagnostic systems are very promising in contemporary healthcare environment.

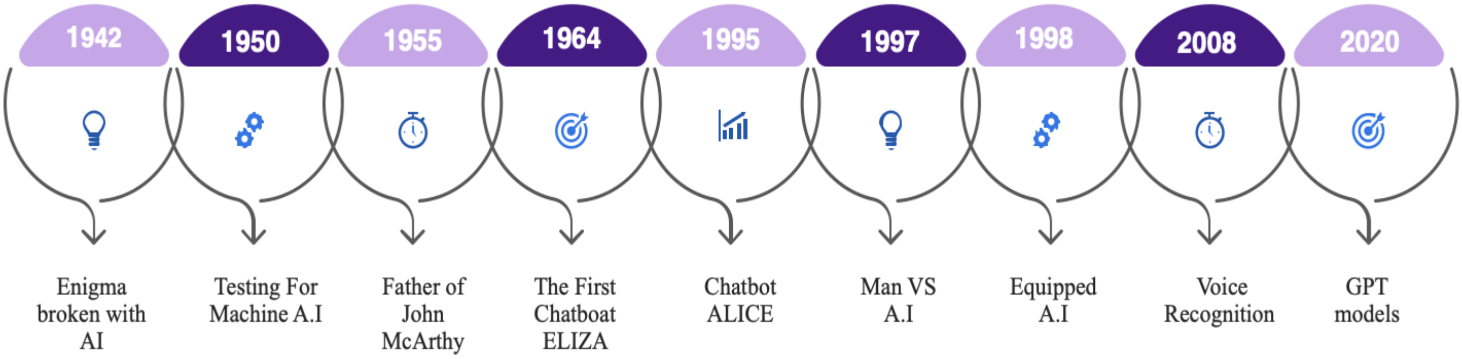


Figure 21. Timeline in the History of AI

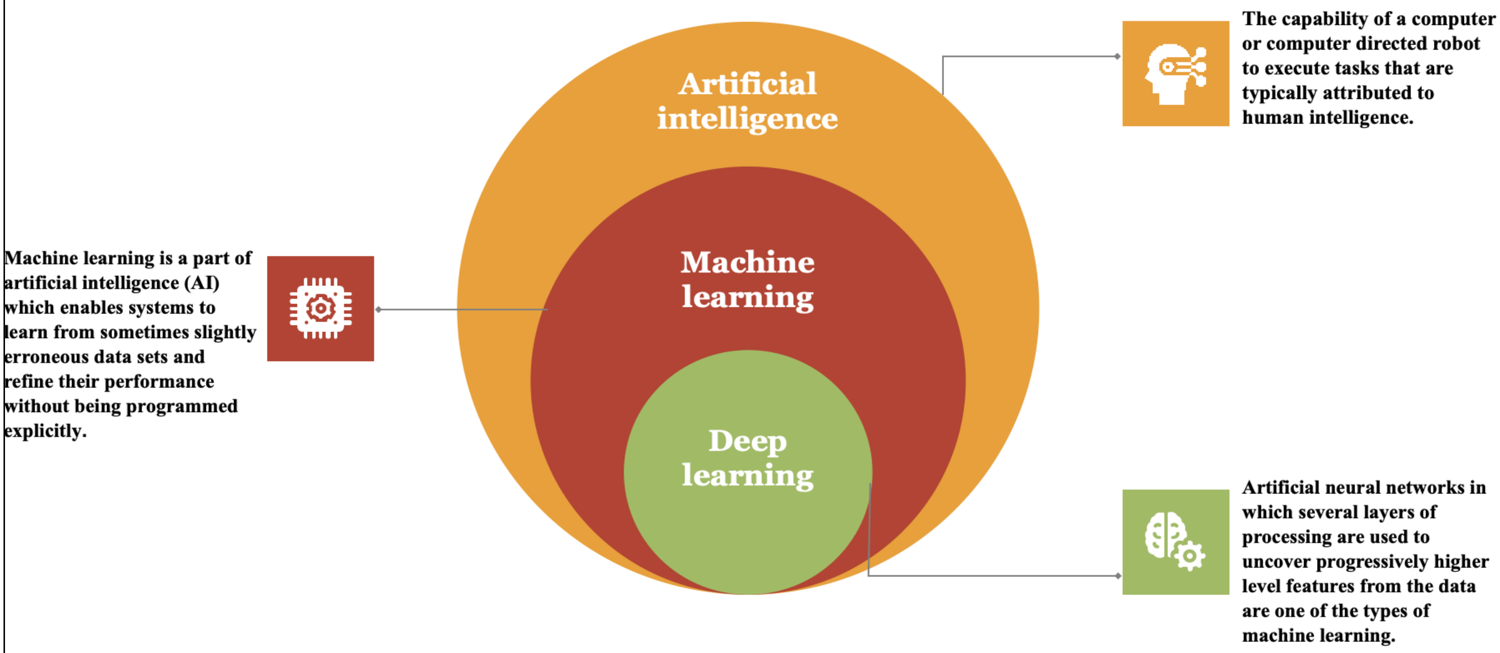

Artificial intelligence
The capability of a computer or computer directed robot to execute tasks that are typically attributed to human intelligence.
Machine learning is a part of artificial intelligence (AI) which enables systems to learn from sometimes slightly erroneous data sets and refine their performance without being programmed explicitly.
Machine learning
Deep learning
Artificial neural networks in which several layers of processing are used to uncover progressively higher level features from the data are one of the types of machine learning.

Figure 2.2 The Pyramid of AI, ML, and DL

Biotic and abiotic diseases are divided in plant pathology according to plant diseases. Environmental changes and climate conditions that are some of the non-living causes that cause abiotic disorders. Infect plant can manifest indicators of abiotic diseases under adverse acidic or low, soils or conditions of gas greenhouse. It can be very hard to detect plant infections, which makes identification as well as classification a huge problem. Ts note that lots of plant diseases present with similar symptoms. Due to close resemblance, identification of which plant disease it is causing harm to is not easy. Defected plants and types of plant issues some few signs, that can be hard to describe and identify. The signs, like poor leaves, are identified in order to recognize the disease. Most important for agriculture we see the maximize the production of plants like vegetables and fruits in economically ways. This study is to establish the cause of the leaf diseases. Earlier research has indicated consistently of the leaves of a plant is directly linked of the in the same plant. If a plants leaves are healthy, then its immune system becomes stronger and is better able to fend off infections in various parts of the plant. These are very dangerous diseases because they can be transmitted rapidly and cause serious destruction (Jafar et al., 2024).

AI-Driven Disease Classification: Several researches apply DL and ML models to detect plant diseases in an image classification task. For example, one can apply such networks that has neural like as CNNs to differentiate between healthy and non-healthy leaves, and some architectures with ResNet and Xception fine-tuning amazed by their high (up to 99.95%) classification accuracy.

Dataset and Feature Extraction: The datasets used, used to train the models, contain ~1000 of images of healthy and infected plant leaves. Merging feature extraction approaches including texture, color and shape (application of K-means clustering) with sophisticated neural networks (BPNN, and other CNNs) are used for the categorizing of diseases. **High Accuracy in Disease Detection:** The accuracy rates of several models have proved to be quite impressive. For example, a Random Forest (RF) model demonstrated 94% accuracy, a CNN-based model – 87%. To determine how the different algorithm or doing for multiple approaches have been consider ML algorithms and their types are concerned, and some algorithms outperform others. **Transfer Learning and Fine-Tuning:** Even transfer learning methods supported by architecturesthat are already trained before, such as VGG16, EfficientNet and Xception have been examined. The result of the research showed that VGG16 performed better than others since its classification accuracy was 99.25 %. **Comprehensive Model Training:** Optimized hybrid models combining traditional ML methods with DL which helped to achieve high classification accuracy as with the 99.4% accuracy achieved using K-means clustering combined with neural networks in some cases, were used (Jafar et al., 2024). AI and image datasets which has processing methods present considerable advantages concerning PDD and categorizing but have major hardships. Such techniques in image processing can divide diseased regions in plant images, but noise management and background interference remain as bottlenecks to accuracy. The need for new methods to increase the reliability of these techniques is part especially in the Agricultural field. Important AI-based systems in real-time for disease identification are few, a big need for machine learning models that offer real-time analysis to the farmers. Further, the AI can help with optimization of chemical formulation, which will help with putting the right measure of chemicals to mitigate disease (Jafar et al., 2024). Through analysis of leaf images, AI models can detect the early indicators of these problems, which is helpful in the precision agriculture.

Creating room for AI based models, there is a gap in practical, an example being mobile and web platforms specializing for general users. Existing models proved useful but are demanding in terms of testing and verification for wider implementation. Drones, as expensive as they may be, have attracted attention precisely for their agricultural utility, in terms of crop health monitoring and spraying. Drone and sensors using concept of AI can form an integrated system for real time detection and intervention of disease. Such kind of hybrid systems would enhance the agricultural practices by automating the monitoring and control of crops. In a proposed framework, plant disease data sets are used to train AI models and transfer learning is used in model validation in the framework. These trained models can them be launched on mobiles or drones which will allow real-time disease detection. The integration of AI with IoT and real time image capture would enhance further optimization of plant disease identification posing a strong instrument for modern agriculture (Iniyan et al., 2020).

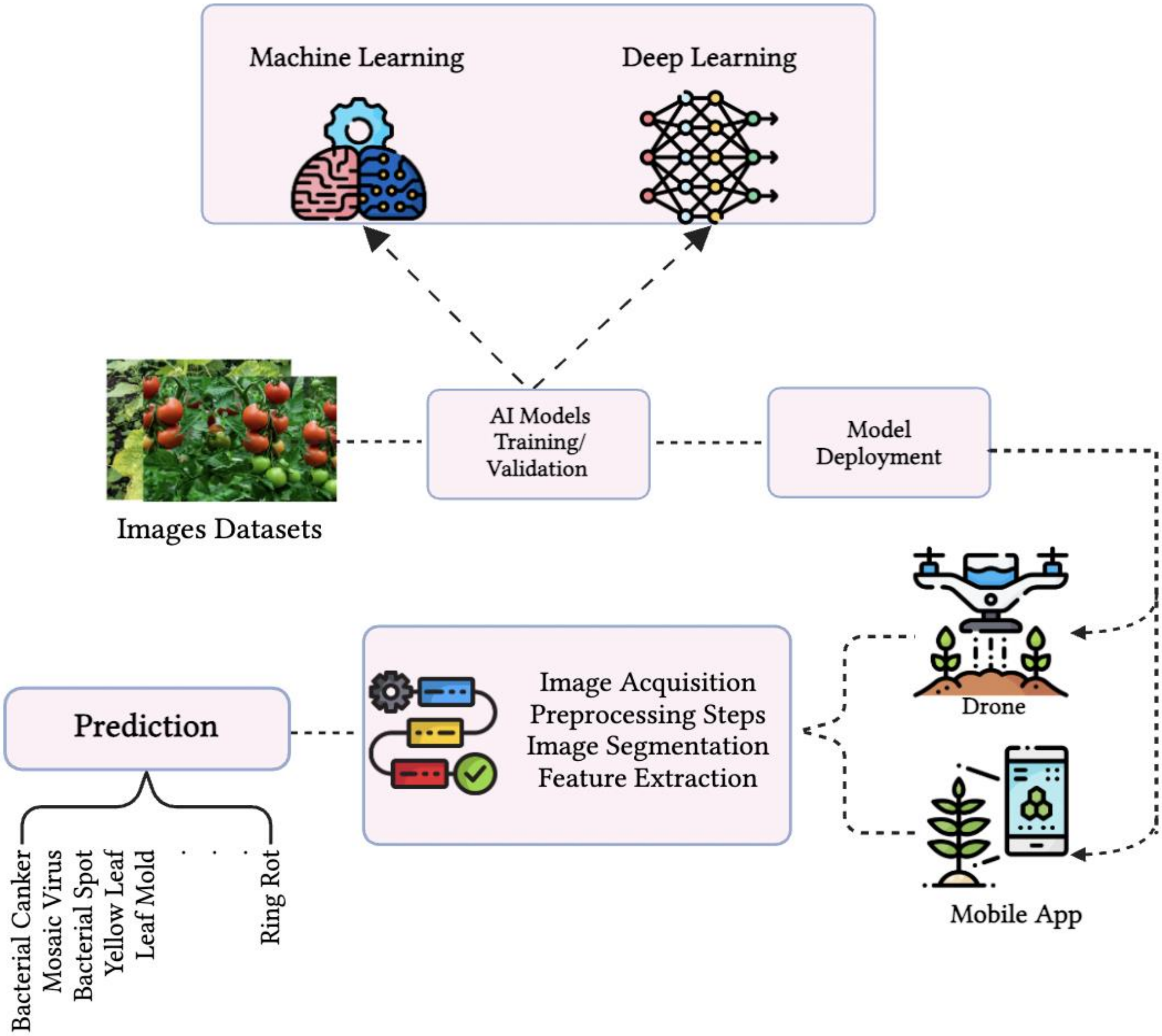


Figure 3.1 Dataset based Planet Disease detection using AI

2.2 MACHINE LEARNING (ML)

The ML is a set of various statistical methods, which enables the PC to learn dynamically without explicit programming. Such learning is typically manifested in altered functioning of an algorithm. This system can acknowledge faces by learning a set of photos representing different people. There are 3 types of ML they are SML, USML and RF. Life expectancy on the average has improved significantly in the last decades due to technological developments. While technology has advanced a lot these days, many are still hoping that advances like AI and ML will bring a renaissance to medicine. Of course, even minor and unimportant tasks within any operation can be perfected with the aid of computing. Healthcare has included machine learning for a while and it looks set to have a significant role in the future. AI and ML are used in many ways in healthcare, in addition to business and e-commerce. We can do almost anything with Blockchain technology. ML is playing a key role in making the healthcare sector more successful. Having to keep Electronic Medical Records in use has made healthcare systems choose Big Data tools for data analysis. The usefulness of this approach will rise through the use of ML tools. These tools help to bring automation and smart choices into the main sections of health care. One of the biggest effects of ML could be the ability to help billions of people worldwide live more comfortably. (Jafar et al., 2024).

ML technologies are a broad area of application for the improvement of clinical trial research. Relying on advanced techniques of predictive analysis on Using clinical trial data, medical professionals could save time and money in conducting a wide variety of medical tests. Different fields and clinical trials use machine learning regularly. Resorting to predictive work search based on ML allows researchers to use data from meetings with doctors and information shared on the internet to find potential candidates. Real-time access to data and continuous monitoring of staff enables investigators to determine the perfect sample for analysis and use technology to decrease mistakes from data. Today, there is a wealth of medical imaging data recorded electronically, and a number of different algorithms can be employed to search and discover patterns and anomalies with this dataset. Algorithms under ML can inspect the data just like a radiologist can also, identifying patches on skin that are not normal, any type of lesions, tumors and bleeding of organs like brain or tumors. Consequently, the number of the use of these setups to support radiologists is likely to rise (Jafar et al., 2024).

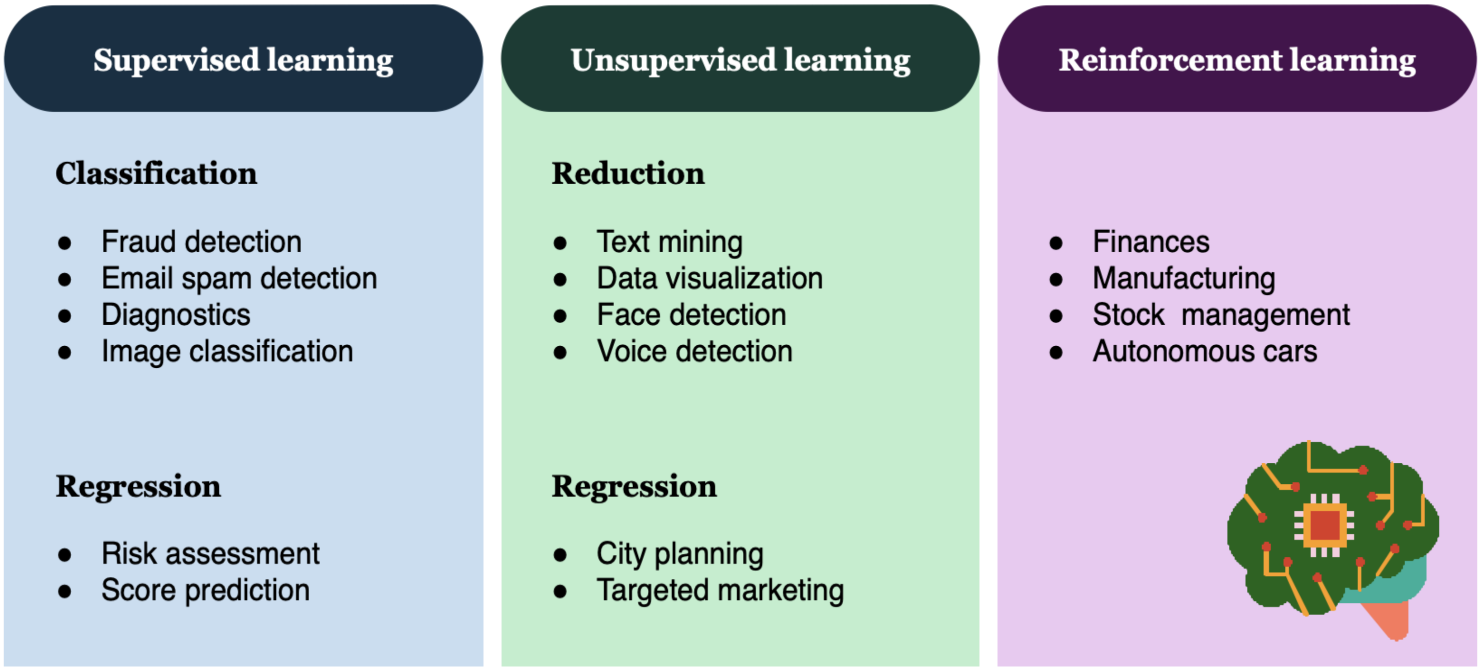
ML is also being strongly applied to the area of research. The time and cost needed to complete clinical studies are extremely high. With ML, researchers can reduce unsuitable participants for clinical trials, since it allows them to use several available data points such as the patients' social media, medical history and visits to different doctors. We can also use ML by watching participants during actual trials. They can further help researchers choose the best number of samples to study and rely on electronic records to avoid making mistakes in the database. The purpose here is to study how machine learning might transform healthcare. (Maniyath et al., 2018).

Level of medical care and the discussion of complex diseases should be enhanced further but there are issues especially addressing the proper dosage and length of therapy for the patients or group with little clinical research, like children. Over the past years, machine learning has grown in use for pediatric care, helping to forecast personalized curing plan for children. The COVID19 pandemic has however projected the importance to ML, since organizations have resorted to it for a competitive advantage in their operations internally, and for research and product development in dynamic settings. ML as the key of AI is becoming a key tool in healthcare. ML can support a number of applications, whether through improving the quality of patient care or optimizing the hospital's operation owing to the use of algorithms to enable data-driven learning. As health care changes as a result of ongoing technological changes, ML gives medical practitioners the tools to make better, personalized treatment decisions. Sizing up the most appropriate therapy options depending on medical history, genetic profiles, lifestyle factors and the ever-evolving test results is one of the greatest challenges. AI has many sophisticated approaches including deep neural network, the research paper covers reinforcement learning, probability based graphical models and learning which is semi supervised. These techniques help the people in medical services to analyze the past data such as family history data as well as genetic data thereby making quick as well as accurate decisions that improve patients' care as well as outcomes. It assists to have treatments personalized as it identifies through the analysis of medical past record, genetic information and test results. ML techniques such as deep neural networks and reinforcement learning help on making forecasts regarding best therapies as well as improving outcomes and simplifying hospital administration and operations (Maniyath et al., 2018).

Based on their learning approach, the type of data they accept and require as input and the problem they are meant to solve, machine algorithms can be broadly classified. The three major divide categories include SML, USML, RL. In addition to the above, there are other hybrid and advanced techniques that build these ML concepts to match complex problem solving demands. In studies SML is a type of ML where datasets are trained with inputs and some target which given labelled output. Our aim is to learn a mapping function that is able to forecast the result generated. This strategy works well when great volumes of labeled data can be found. Although in domains such labeled data are scarce or expensive, it becomes costly and not feasible.

Unsupervised learning can be used when there are only inputs data without even one output label available. The algorithm tries to find latent features, trends or arrangements within the given information. Clustering is one of the most common among this category, where the algorithm clusters similar data points basing on inherent attributes. Later they can support activities like predicting behavior of users or customer demographics segmentation without knowing any particular labels in advance (Maniyath et al., 2018).

Reinforcement learning (RL) is interested in decision making problems where the agent must act in an environment in order to achieve a goal. The agent learns by trial-and-error learning; it gets rewards and penalty for its actions. The purpose is to maximize the overall cumulative reward over time. RL has far ranged applicability including game-playing AI and robotic control systems. To sum up, there is supervised learning that predicts known results, unsupervised learning that finds concealed data patterns and reinforcement learning that is concerned with learning of optimal actions based upon interaction. Every approach has a set of unique strengths and chosen according to how the data is organized and what the problem happens to be. (Iniyan et al., 2020).



Supervised learning

Classification

- Fraud detection
- Email spam detection
- Diagnostics
- Image classification

Regression

- Risk assessment
- Score prediction

Unsupervised learning

Reduction

- Text mining
- Data visualization
- Face detection
- Voice detection

Regression

- City planning
- Targeted marketing

Reinforcement learning

- Finances
- Manufacturing
- Stock management
- Autonomous cars

Figure 4.1 Types of ML and their Classification

2.2.1 APPROACHES OF MACHINE LEARNING

2.2.1.1 SVM:

A SVM relies on ML data and can be used for either identifying groups or finding exact values. Nevertheless, classification problems are the main domain of use for unsupervised learning. We show a lot of data as points in a space with as many dimensions as you have features and each feature's value corresponds to its own coordinate. Davidson then identifies the plane that can differentiate the two classes in the best way possible. SVM lines up each example as a point in space and makes sure that the examples in different classes are far from each other is its main feature. SVM can do more than linear classification; it can easily perform non-linear classification using an implicit conversion to features in higher dimensions. The objective of

The goal of method is to obtain the optimal decision boundary or line so as to classify the dimensional space by which we can classify additional points more easily in future. This optimal boundary is named a hyperplane. In order to facilitate in creation of the hyperplane, SVM identifies the extreme vectors and even the points (Iniyan et al., 2020).

Step one involves collecting sample pictures of plant, healthy and affected (e.g., Bacterial spot, Tomato Mosaic Virus, Rust, Northern Blight) using the digital camera in JPG format, with RGB color. These images are the training dataset. During the preprocessing stage we create separate directories for the healthy and diseased images, and load the images into arrays. Noise and distortion due to changes in lighting or camera settings—are decreased when changing RGB images to grayscale leading to increased accuracy. The second is image segmentation where a given image is partitioned to form a pixel region based on intensity into pixel regions. The third is feature extraction where we see images size, color, shape and texture and this will be analyzed by SVM. The last technique is HOG (histogram of Oriented Gradients which is use for image which will give accuracy and it will retain meaningless data and remove that data which we stored in machine (Iniyan et al., 2020).

SVM is harnessed in the presented strategy for the identifying elements of leaf parts, overcoming the intricacies of identifying diseases from human observation. The reason for choosing SVM is its capability to deal with complex problems of feature extraction and classification (Demilie, 2024). Due to recent advances in machine learning, identification of the feature has become more effective, and SVM is excellent in distinguishing among classes of diseases. Support Vector Machine, because of its robustness in supervised learning, can be of special value for image-based classification and leaf diseases accurate prediction (Iniyan et al., 2020).

SVM which is linear is appropriate for linearly separated models or data because through it, two dimensions is employed to dived a data which is linear in nature by a straight line that will determine different classes, or in higher-dimensional spaces, a hyperplane is used to separate data. When the data is able to be segmented by a straight line, the straight line becomes the effective segmenting boundary for Linear SVM to classify data properly (Iniyan et al., 2020).

If the data separation demands the use of complex boundaries that could not be a straight line, then a Non-Linear SVM classifier is used. By means of the use of kernel functions, the data is mapped to a HD space where there is a possibility of separation linearly, leading to effective classification (Iniyan et al., 2020).

Data can be divided by various lines or decision borders in n-dimensional space, but it is important to find the right decision boundary to organize data points properly. This best-limit is called the SVM hyperplane. As the hyperplane's size relies on the amount of characteristic in give datasets in the dataset, it will be a straight line if the dataset of two features, as shown in the example image. In addition, when there are three features, the existence of the hyperplane in the dataset is that of a two-dimensional plane.

Support Vectors:SV are the model when the data point or vector, which is hyperplane is closest and which plays a critical role in its positioning. Support vectors are referred to so because they take a central role in preserving the hyperplane's position.

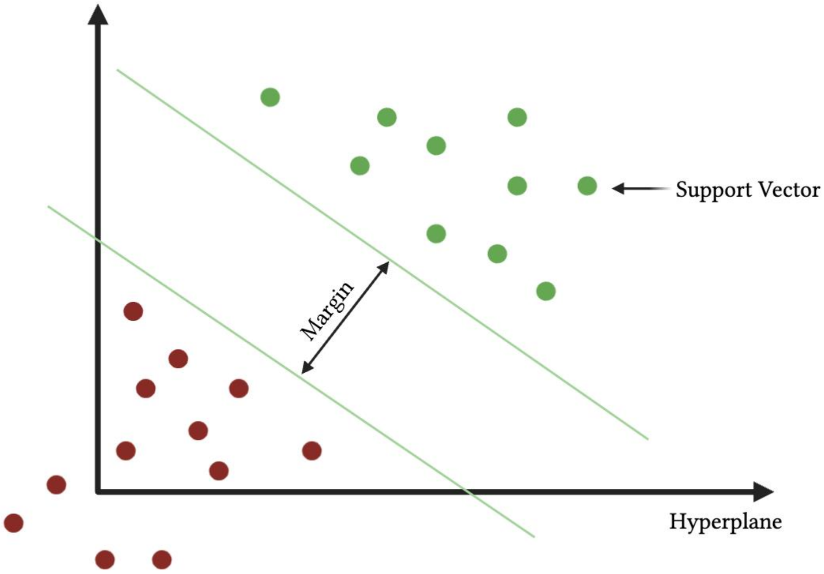


Figure 5.1 SVM Margin and Hyperplane Representation

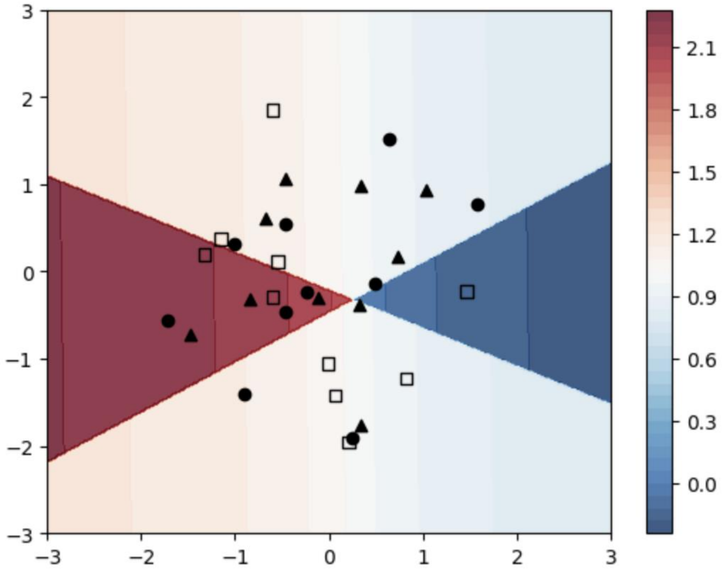


Figure 5.2 SVM Decision Boundary Visualization

2.2.1.2 RANDOM FOREST

There are so many diseases that plants can acquire due to environmental factors like humidity as well as biological pathogens including bacteria viruses, and fungi. Leaf diseases often lead to visible colors and shapes changes, but it is difficult to render them because various diseases have similar visual features. Prompt leaf diseases identification is critical in contamination of crops and prompt measures of treatment. To overcome this challenge, a ML system based on random forest has been suggested for efficient spotting and grouping plant leaf diseases (Gupta et al., 2022).

1. Preprocessing Stage:

- Initially, the pre-processing processes sharpen the details and reduce noise interference with the leaf images. By changing the intensities of the pixels, contrast enhancement is intended to emphasize important features accordingly and thereby intensify the visible features of disease areas in the images.

2. Feature Extraction:

- Focus is shifting to feature extraction only from the infected areas of the leaves, thus removing irrelevant information and enhancing the system's disease classification capability. Texture characteristics such as Contrast, Correlation, Energy, and Homogeneity are obtained by means of a GLCM, and this highlights the spatial patterns between the intensities of pixels. Statistical features and their types are used to extract out through

MATLAB tools to describe more image characteristics relevant to disease symptoms (Gupta et al., 2022).

3. Dataset and Training:

- This set of data included pictures of the leaves of plants, from both healthy plants and diseased ones, using several online archives. Such data set consists of labeled images, and it is known what specific disease each of them is related to in advance. The prepared labeled data is then used as the input to train the models which is classifier.

Classification Using Random Forest:

Such method is called RF which relies on a combination of several decision trees. In contrast to the individual decision trees which are so prone to overfit nowadays, Random Forest improves generalization by combining results of multiple trees. Each tree now uses the entire training dataset to produce its own prediction of the class label.

During prediction time, each tree produces a class, and the last result is based on higher voting among the trees. The conglomerate nature of the Random Forest makes it more reliable, accurate, and adaptable for complex high dimensional classification problems as demonstrated by the analysis of plant disease pictures. This ability of processing both the numeric and categorical data makes it especially appropriate for dealing with the heterogeneous feature sets produced in this system (Wójtowicz et al., 2021).

This approach demonstrates that because of the robustness of Random Forest to input variability and noise, it has a good classification ability for plant leaf diseases. By combining texture and statistical features with ensemble learning, this method provides a robust, flexible and robotic system for primary infected leaves detection in plants with a view to reducing agricultural losses and increasing output (Upadhyaya et al., 2024).

$$P = \frac{Y=1}{X} \quad \text{OR} \quad P = \frac{Y=0}{X}$$

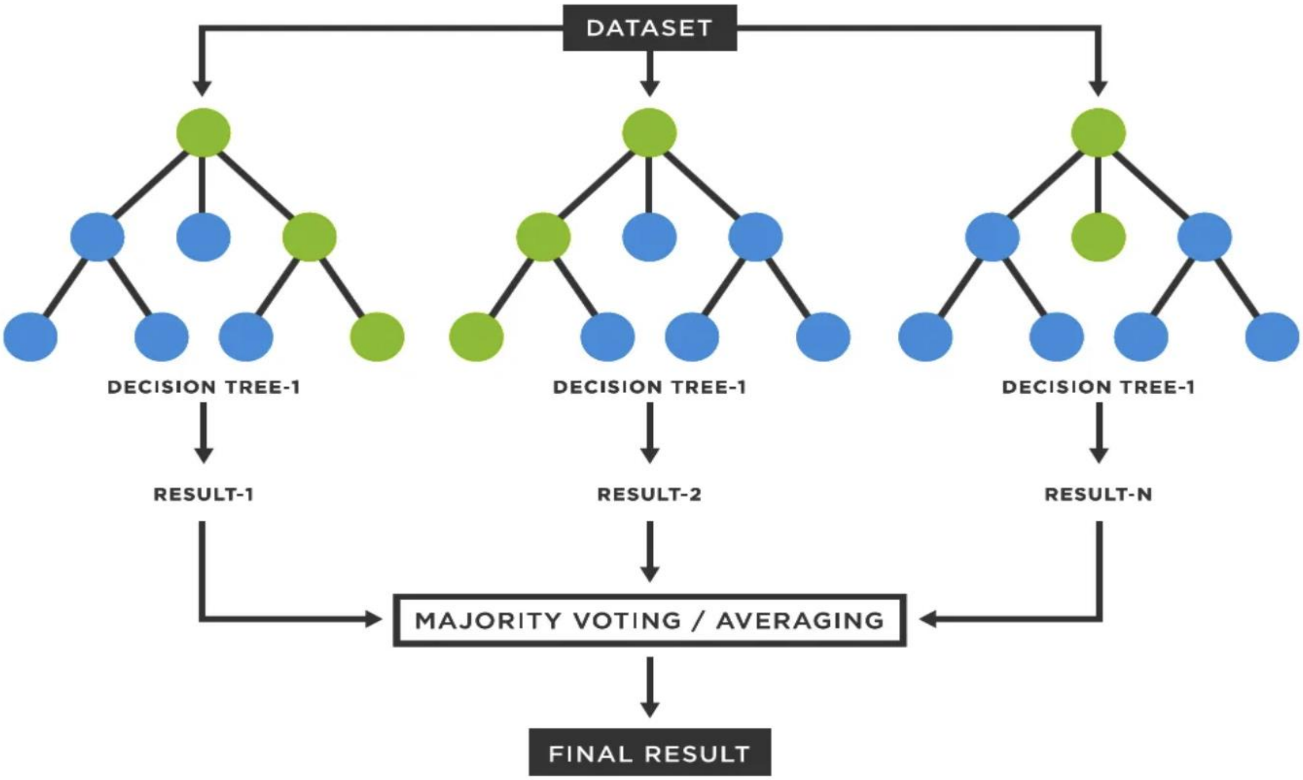
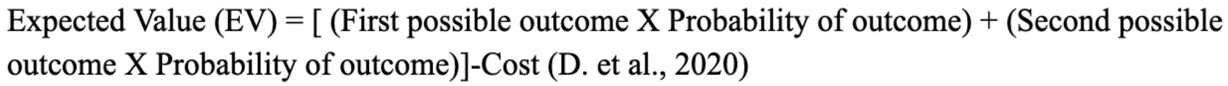


Figure 6.1 Random Forest Architecture Overview

2.2.1.3 DT:

This is a ML models which is learning procedure that requires supervision or instruction for tasks like classification/Regression classifiers through successive division of data into subsets guided by the criterion of maximizing disparity in supervised learning. The Gini index, which is at other times named “entropy,” is one of the commonly used metrics in attribute selection in determining discrepancies in a given dataset. One of its major advantages is that it is likely to produce results that are intuitively understandable for humans. If there was no restriction in maximal depth, then a classification tree could lead to almost zero training error (D. et al., 2020).


Expected Value (EV) = [(First possible outcome X Probability of outcome) + (Second possible outcome X Probability of outcome)]-Cost (D. et al., 2020)

Step-I: Choose a dataset of rows that will be the foundation for building a decision tree.

Step-II: Computing the uncertainty or impurity of the data set itself or mixed data types if so required (D. et al., 2020).

Step-III: Generate all possible questions that must be asked at that node and determine which is helpful in Plant detection

It works very well because it has the ability to send instructions to a predefined tier. This method is similar to diagramming, where the information is divided into two equal categories, with 'stem', 'branch' and then to 'leaf' where the terminal categories become more comparable. An approach to building this on a hierarchy of categories following the vein of natural classification but with minimal human intervention based on the work which was able to detect foliar diseases by way of decision trees in an agricultural field (D. et al., 2020).

In the discipline of ML g, the authors have reviewed multiple algorithms used to detect plant diseases and shown how they deliver remarkable results when applied to the proper data. The implementation of such algorithms assists in overcoming the necessity of the efficient methods, which are mainly oriented to predicting plant diseases in agriculture and backing up resources in agricultural and business domains respectively. The first approach of the bulletin is based on the use of ML models to monitor each leaf in a tree (D. et al., 2020).

This explains that while using the gained results we proceed to provide, interpret, and confirm a model that is aimed to forecast and collapse important happening and offered implications. Strengths required for building technologies for predictive modeling. In Regression and neural network are vital methods in predictive modeling in AI. Conclusion: files of Bayesian methods, wood selection and statistical profile processing complement each other closely. Each of the many prediction models plays an essential part. Fundamentally, a metamodel is applicable in many situations, developed from algorithms that are taught using harvested, processed, and stored data to bring forth reuse for the final product assessment (D. et al., 2020).

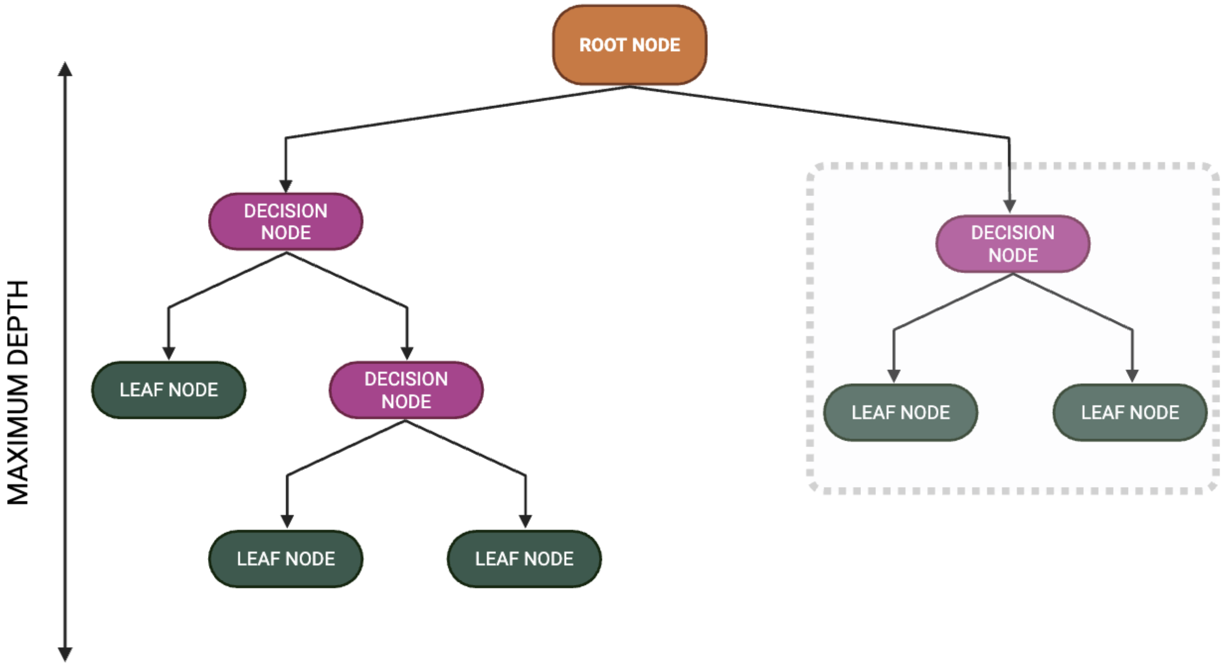


Figure 7.1 Decision Tree Architecture Overview

2.2.1.4 GRADIENT BOOSTING

Gradient Boosting stacks up many weak learners, often decision trees and trains them to fix mistakes from previous models. Bagging methods such as Random Forest, build trees at the same time and add their outputs, but gradient boosting builds its trees one by one to reduce the amount of error from earlier trees (Priya et al., 2022).

Researching how to predict the severity of plant diseases has shown that gradient boosting advances the field, thanks to its power to model complex links and work with the features from images of plant leaves arranged as structured tabular data (Priya et al., 2022).

In identifying early or late blight in potato leaves, gradient boosting stands out as it handles complicated and nonlinear relationships well, is good with different amounts of training data and makes correct predictions by analyzing useful features drawn from image data.

The method behind gradient boosting is to use gradient descent to reduce a chosen loss function. The process starts by using an easy model such as a shallow tree, to deliver simple predictions. The difference of the labels and values which are and this is also called Residuals. Residuals are assigned to a different decision tree which learns the errors that the first model made (Priya et al., 2022).

The team adjusts the Ensemble by using the new model's results, scaling them with a learning rate and adding them to the original predictions. It is achieved that the particular number of times or until a solution is repeated.

For each iteration, the objective is to minimize error based on loss, whether that's mean squared regression with error or log loss for classification, by choosing the negative gradient as the next model's aim (Priya et al., 2022).

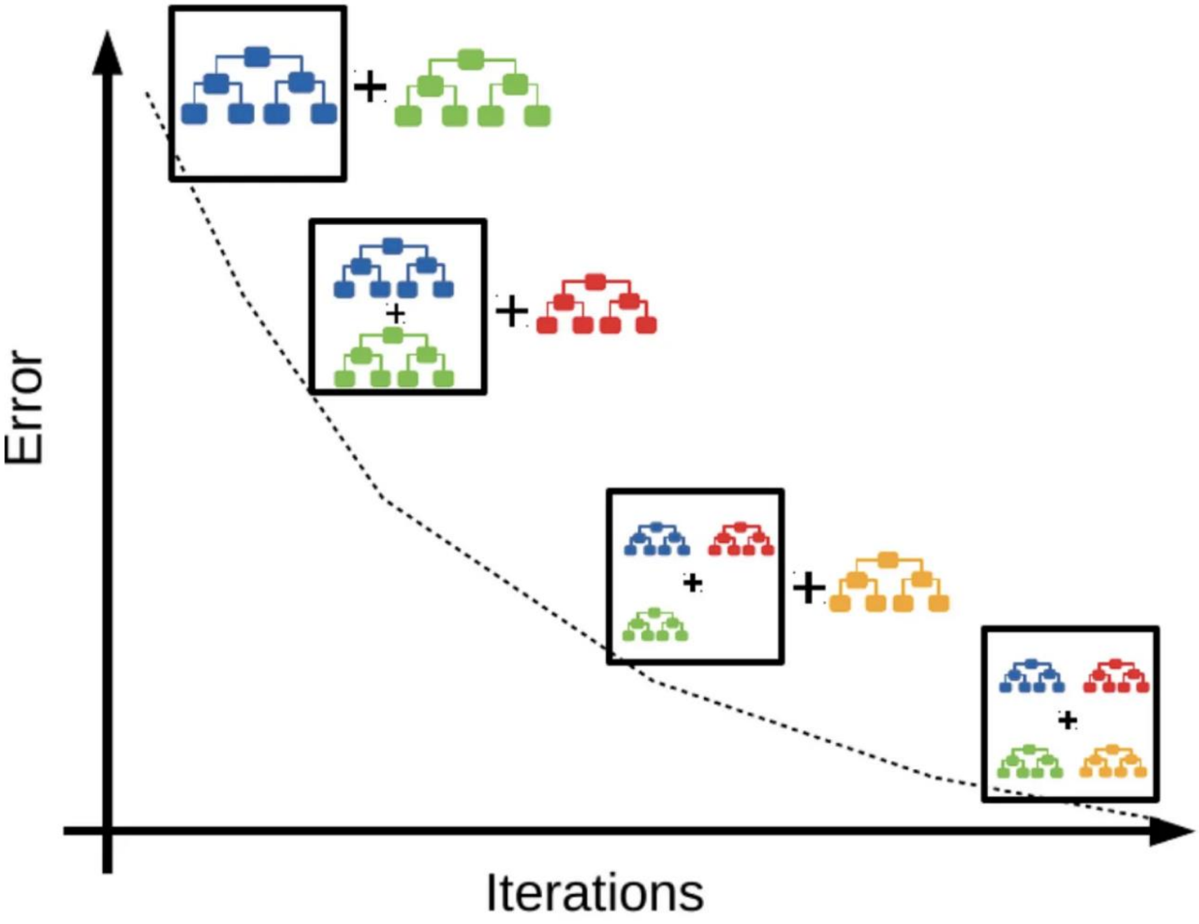


Figure 8.1 Overview of Gradient Boosting Graph

2.3 DL:

In recent times, DL models and their types have been embraced enthusiastically in infected plant detection because of their remarkable capability to identify and then classify them. The classification limitations of traditional image analysis, which commonly depend on a priori features like texture shape as well as color, limit their use for identifying small or early signs of plant disease. However, deep learning addresses these concerns as it carries the ability to allow self-discovery of hard features directly from the unprocessed image data (Jafar et al., 2024).

CNN and DBN have proved to be effective resources to overcome plant disease recognizing. Such networks can reveal intricate features in images of plants that often go unnoticed by manual or visual evaluation. The curse and blessing of DL model are their capacity to deal with complicated and detailed images which makes them an efficient tool for precision agriculture with high accuracy and scalability.

High-accuracy deep learning models are indeed, however, most often, dependent on large volumes of data during a training phase which has already been annotated. Additionally, deploying them requires a considerable amount of computation resources and large memory. This limitation can be a particularly problematic factor in environments characterized by limited resources and scarce labeled data (Mazumder et al., 2024).

This says that TL is a useful solution in handling these challenges in PDD. The use of trained from before models using DL and subsequent retraining of such on-plant disease data sets, this method is able to substantially reduce data and computational demands. Through ensemble methods, amalgamation of outputs provided by a number of deep learning models is a strategy which is now largely employed. Such a strategy produces more precise outcomes and reduces the possibility of overfitting.

The other essential factor lies in technique with data augmentation. By using rotations, flipping, scaling or changing brightness, these methods artificially enhance the image set for training. Doing this, the model becomes more automatic and independent the less it relies on a large number of annotated images (Jafar et al., 2024).

DL has revolutionized infected leaf with plant detection by making it possible to identify many plant pathologies in a fast, reliable, and large-scale manor. As it progresses, it offers solutions that perform better within the vast real-world agriculture environments, which makes it more relevant.

DL is a branch of the large field of the AI that utilizes deep neural networks with their layer structure to perform such tasks as classification, prediction, and generation. In contrast to traditional algorithms, which require explicit creation of individual features, deep learning models obtain knowledge directly from raw data through many multiple layers of abstraction. Over time, the layers extract progressively more intricate pattern, going from basic elements like texture and form to more elegant indicators like diseased area or leaf coloring.

However, DL algorithms require massive labeled data as well as a strong hardware in where train can be happened by order Additionally, DL struggles to adapt to any newly found element which it has never been expose to before and is an opaque system to work with, that makes it tricky to guess how decisions are made. Still, these challenges have failed to discourage the use of DL technology in smart agriculture where its accuracy and adaptability will provide value for money (Elaraby et al., 2021).

Other deep learning method systems such as CNNs have also been used to assist with the identification of plant disease and infestation. These methods have brought promising findings in the identification and detection of lesions from pictures. By learning images' features in an auto mode, deep learning models can identify small signs of diseases in images, something that current image processing methods do not effectively do. However, deep learning models demand a massive amount of labeled data with trained module, as well as significant computational capability (Alibabaei et al., 2022).

The area of DL was significantly multiplied by a science paper of 2006, which stressed on the development of autoencoder networks as the major part of the strategy. This architecture consists of several of such layers, which are meant to transform extensive, high-dimensional inputs into compact, low-dimensional ones. The primary contribution of the article was proposing a good way of initializing weights such that gradient descent optimization was enhanced; for the deep autoencoders to outperform previous methods like the PCA, in reducing the dimensions of data (Shoaib et al., 2023). Deep learning is a part of AI systems which we use in neural network to learn complex raw data and give multiple abstracts. On the contrary, traditional methods often require the use of handcrafted features but, on the other hand, DL models themselves proven the complex relation with recognition, and provide superior results in areas like computer vision. For image-based applications such as plant disease detection, DL approaches automatically generate the features without manual intervention, and retrieve subtle details that may be overlooked by traditional methods (Jafar et al., 2024.

CNNs are an important architecture for DL in the infected plant detection which provided that from their images they can learn complex hierarchical features directly. Conventional use of CNNs usually leads to accuracy figures exceeding 99% in cases of identifying infected leaves (Shoaib et al., 2023). The DBN, a DBN is a variant of DL, formed by stacking restricted Boltzmann machines, is used to detect lesions and pest damage with an accuracy of 96% to 97.5%. DBM have great potential for unsupervised classification of plant disease images. Deep Denoising Autoencoders (DDAs) are used for both cleaning sensor noise and predicting plant diseases, with an accuracy reported at up to 98.3%. The performance of these DL models increases accuracy, improves reliability and facilitates scalable applications thus placing these DL models at the forefront of modern plant pathology research (Shoaib et al., 2023).

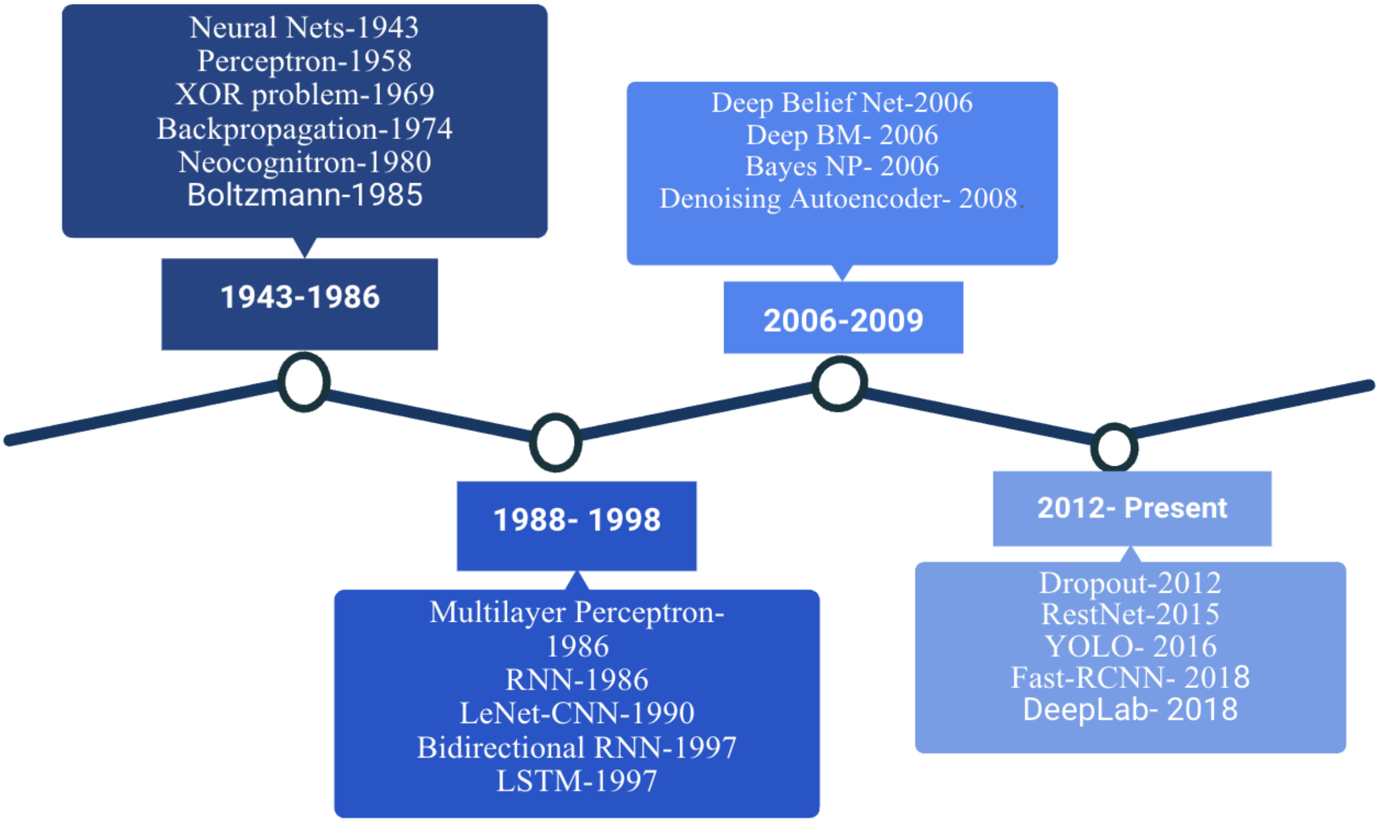


Figure 9.1 Timeline of Key Milestone in the Evaluation of DL

2.3.1 CNN (Convolution Neural Network):

As DL models CNNs are particularly effective for classify for image problems such as the identification of leaf diseases. CNN's architecture has many layers and these layers are completely associate, maxpool and normalization layers. The initial stage mainly connected, layers in a CNN are called features that could be learned and features with extracting part are trained and optimized via weights in these layers (Shoaib et al., 2023). Consequently, the FC layers help in the finding out various classes of plant diseases. Training a CNN model begins with the network being fed images and labels, after this, once the model is trained, it can recognize various categories of disease. The start points in a CNN that is on the LDD is the introduction of an image with a leaf. In the convolutional layers the image is processed extracting the necessary features. Pool layers then operate the feature vectors therefore reducing the spatial size of the data. Once the feature vectors are run through the FC layers, the next step is the decision to tell if the leaf was diseased or infested with pests. The model generates a measure of confidence for a leaf to be diseased or healthy. With the CNNs architecture, which involves up-sampling, down-sampling and trainable layers, CNNs have an advantage in leaf disease identification. The training of a CNN depends on teaching it a set of samples of plants that are labeled, some with diseases, and others healthy (Shoaib et al., 2023).

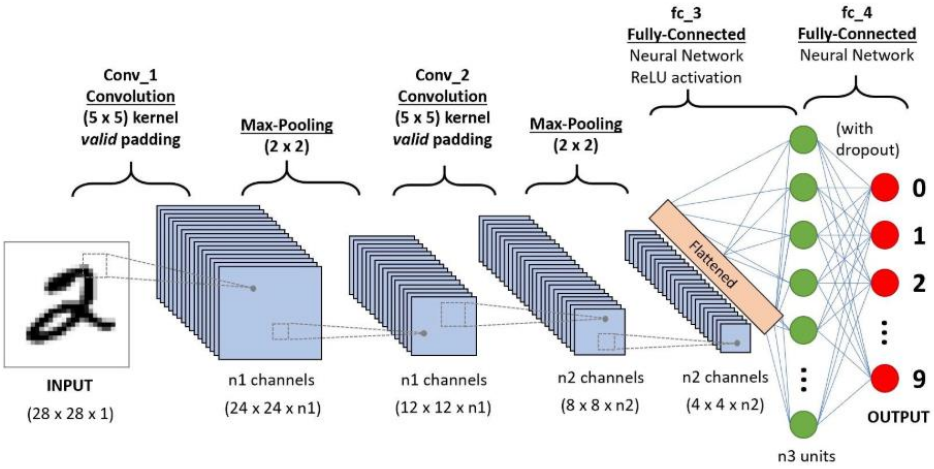


Figure 9.2 CNN Architecture

Table 2. “Overview of CNN Models and Their Highlights”

CNN Models	Highlights
AlexNet	string; which resulted in it becoming the first winner of the ImageNet competition; fast and accurate
VGGNet	A compact but deep structure with the use of small filters known for precision
RestNet	Introduced skip connections; handles very deep networks effectively
Inception/GoogleNet	Replace convolutions through parallel usage of different sized kernels;<< efficient and deep
InceptionV3	Optimized version of GoogLeNet; high accuracy and efficiency
DenseNet	Dense connections between layers; encourages feature reuse
Xception	Uses depth wise separable convolutions for efficiency
MobileNet (v1, v2, v3)	Lightweight CNN for mobile and embedded devices

2.3.1.1 VGGNET: Maintaining healthy crops, protecting the food supply and economic sustainability all depend on Infected Plant Detection. Looking at disease symptoms with the eye is slow and can be more error-prone, should the observations be reviewed at scale. As a result, AI, ML and DL can all provide solutions to improve automation. The researchers employ VGG-16 which they took from the PlantVillage dataset, to find diseases in the pictures of tomato and potato plants. The reason Vgg-16 can directly take out high level of characteristics from images of leaves with ease. Almost all of the ailments tested, including the tomato leaf curl, mosaic virus, target spot and early/late blight on potatoes, are detected with accuracy by this model. 88.6% accuracy was found for tomato in the model reports and 94.6% accuracy for potatoes. The key to reaching these results is using transfer learning and image preprocessing (Alatawi et al., 2022).

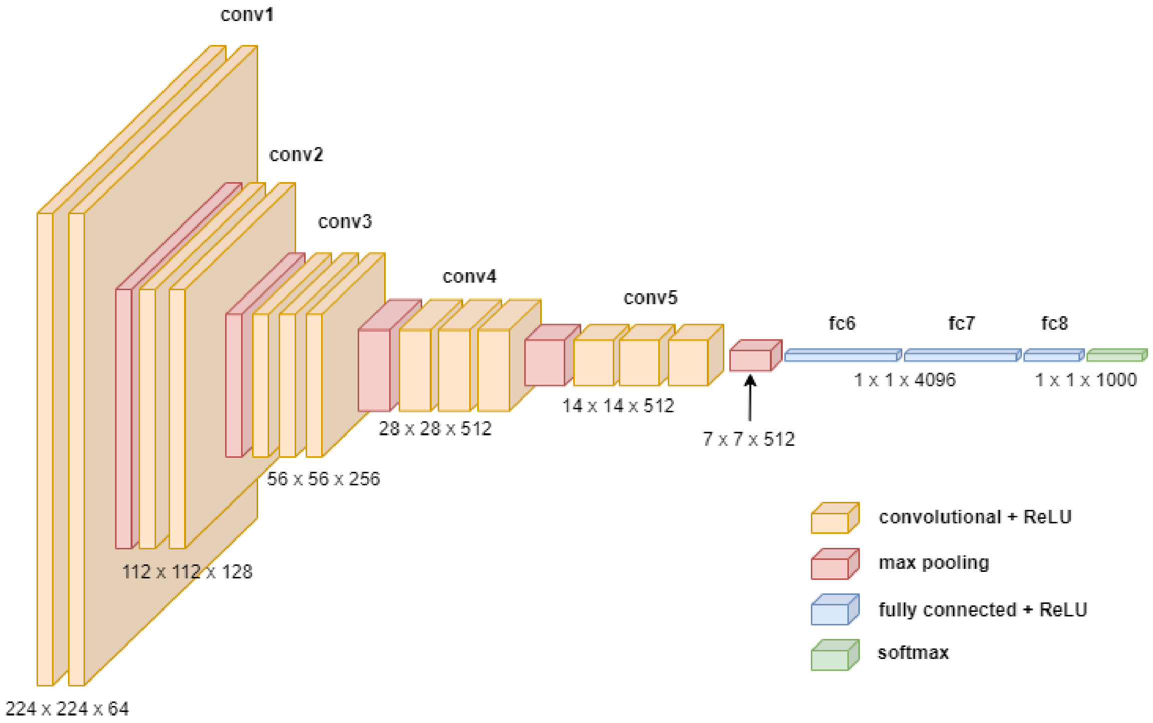


Figure 10.1 Architecture of VGG16 (Alibabaei et al., 2022)

How the VGG-16 Model is Designed: The model we applied for this study is known as the VGG-16. Every image is processed with size 224×224 of pixels and three RGB color of channels in the problem of input layer. All of the with layers of convolutional contain 3×3 filters and are set to retain the original resolution by using a stride and padding of 1. Each max-pooling layer uses filters of size 2X2 and value of stride is equals to 2 for down sampling the images (Alatawi et al., 2022).

The last section of the network comprises of 3 mainly and importantly connected layers. The first layers in the FC contain 4096 neurons. The classification stage is supported by a FC layer with 1000 neurons. The solution of result generated from the SoftMax layer is the statistics of each disease class. Furthermore, every layer which is a further connected layer relies on ReLU as the chosen by function of activation. In the last layer, the SoftMax part is applied to the output vector to ensure its values add up to 1 (Alatawi et al., 2022).

2.3.1.3 RESNET

ResNet or Residual Network, was created and described by Microsoft Research in 2015 as a deep learning architecture. It was created to know the problem seen in deep neural networks, in which additional layers may cause problems due to the gradients either shrinking or increasing abnormally. This innovation stands out a lot when dealing with image classification, as it needs powerful models to spot complex signs (Chen et al., 2022).

By introducing residual learning, ResNet makes an important point. While traditional CNNs learn the direct way from input to output, ResNet learns the how the problem of input and the solution of output are different from each other. It is done with shortcut connections, allowing information to pass straight from one layer to a deeper one. After performing identity mapping, the outputs of these layers are combined with what comes from the earlier stacked layers (Chen et al., 2022).

Mathematically, the function H(x) in ResNet is replaced with F(x), so that H(x) = F(x) + x or H(x) = F(x) - x. Thanks to this, the network can better identify relation between pixels and labels, particularly in more complex networks and avoid the issue of gradients vanishing during learning. Because of this, these models can have a lot of layers without their efficiency being affected (Chen et al., 2022).

(such as ResNet-50, ResNet-101 and ResNet-152)

Infected plants have a major difficulty on how much and how good crops are produced in agriculture. Identifying medical conditions in this way takes time, money and relies on the subjective opinions of experts. Disease discovery using deep learning and image handling is more accurate, quicker and can handle many cases at once (Chen et al., 2022).

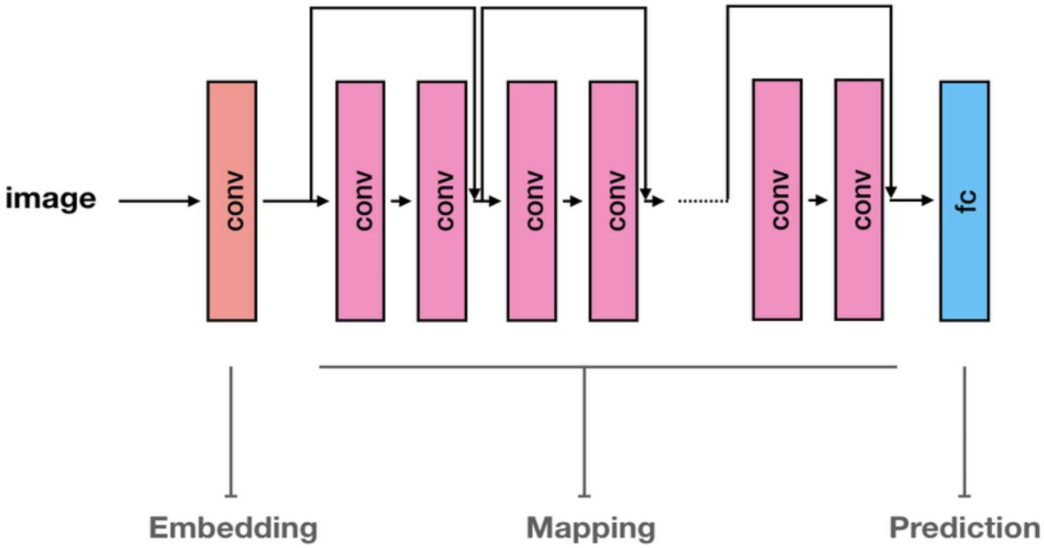


Figure 11.1. ResNet architecture diagram (Rousseau et al., 2020)

ResNet is particularly successful at identifying plant diseases
a.) High Accuracy: A particularly high degree of accuracy is found in image classification. ResNet is very good at identifying features present in photos of diseased plant leaves. Minor differences in the appearance of these images can often show if the disease is liver, pancreas or gallbladder related. ResNet can learn detailed and broad features in its layers that are important for telling Apart face and Unfaced from one another.

b). Manages Large Amounts of Various Kinds of Data
Many agricultural datasets hold thousands of images featuring leaves infected with different diseases in various brightness and environment settings. ResNet was designed for handling such variety, remaining free from overfitting when used together with transfer learning and data augmentation.

c). Getting More of a Result with Less Work
Obtaining big, labeled datasets is hard in many fields of agriculture. Using a trained ResNet model (such as ResNet-50 on ImageNet), experts modify the network on a small plant disease dataset. It makes learning faster and results in stronger performance, despite less data (Chen et al., 2022).

d). Enable instant reports of incidents.
For smartphone diagnosis and drone monitoring on farms, detecting problems in real time is very important. Mobile devices or embedded systems can use optimized versions of ResNet to diagnose plant disease in real time (Chen et al., 2022).

With ResNet, deep networks can be trained without running into the usual problem of degradation. Due to its residual learning method, it is ideally with handling difficult image classification models, one example is plant disease detection. Through proper classification of leaf diseases, ResNet-based models help farmers react promptly, reduce crop damage and improve the performance of their farms. It can be used in transfer learning and works in real life, so it is a major technology for future smart agriculture (Rousseau et al., 2020).

2.3.1.3 ALEXNET

AlexNet was invented by 2012. It has a number of new and innovative concept that can guide the deep learning where we can see the CNN (Chen et al., 2022). It is made up of 8 with layers, two of which are CNN and 3 of which are mainly connected. It relies on basic stacked convolutional layers and puts max-pooling between them. Forwarded innovative concept to its deep network, the model is able to find and use advanced features from images (Chen et al., 2022).

The architectural design uses pooling layers that help shrink the amount of information without altering how neighboring data points are linked.

AlexNet is CNN development which is design to innovate the nonlinear links between new data items this is because of ReLU function and innovative regularization using the Dropout (Chen et al., 2022).

Main features of AlexNet:

The idea behind AlexNet was to use less computing power than was needed by previous CNN designs. The algorithm used two GPUs together for training.

GoogleNet is a deeper network than AlexNet. Because it has eight layers, it is easier to train and doesn’t overfit as much on smaller data sets (Chen et al., 2022).

AlexNet earned significant achievements in ILSVRC in 2012. It surpassed older designs of the CNN by a lot and paved the way for a new era in deep learning in computer vision.

Among the proposed changes in AlexNet are rectified linear units, pooling of adjacent regions and regulating activity through dropouts. Thanks to these strategies, improvement in performance and generalization was achieved (Chen et al., 2022).

In this research, CNN relies on AlexNet and has eleven different layers. Yet, for this study, the reduction of number of layers involved. The research’s version of the AlexNet model is made of the first layer called as 3 convolutional which is followed by three layers that are completely interconnected and finally output layer .The input used in this model measures 64 pixels and RGB (Chen et al., 2022).

The network begins by applying 96 filters of 11×11 with a stride of 4 to images that are 224×224×3. After that, there is max pooling with 3×3 filters and LRN. After the first layer, the second uses 256 filters of 5×5 size and the following three layers use 3×3 filters to obtain 384, 384 and 256 with maps featuring gaps. Because there wasn’t enough memory available for the GPUs at the time, the authors decided to design two parallel pipeline training systems (Chen et al., 2022).

Both the first layer and network connectivity put the final number of parameters at around 105 lakhs. In the total, AlexNet is made up of 61 million parameters and performs 724 million MACs which demonstrates how computationally involved it is (Chen et al., 2022). Because of its huge success, this model helped start a new age in computer vision and deep learning and inspired other architects like VGGNet, GoogLeNet and ResNet.

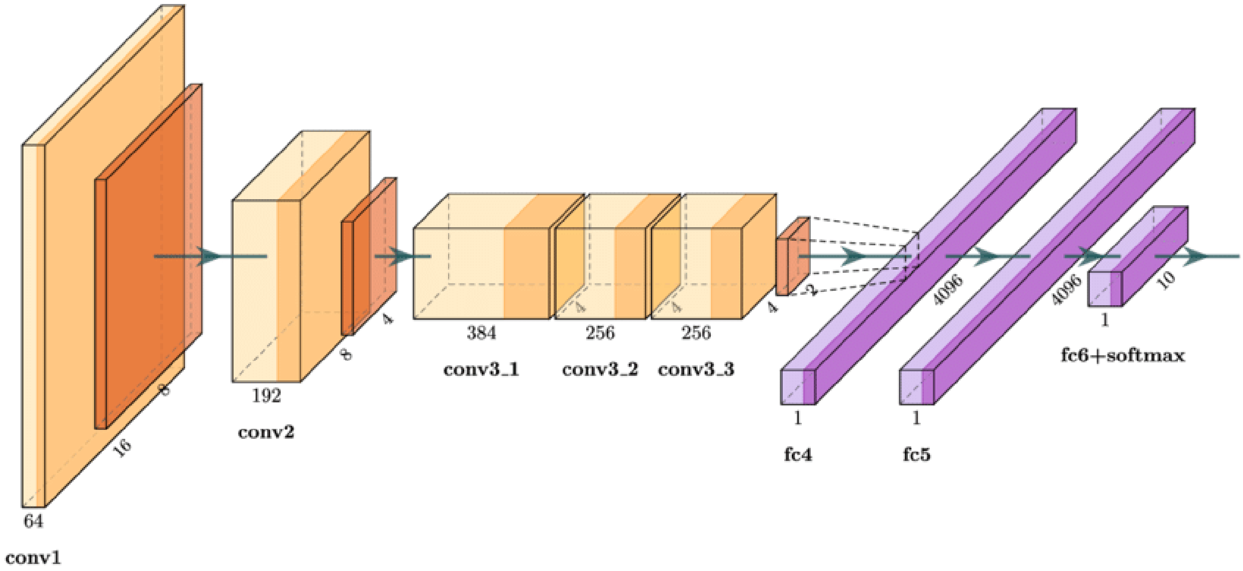


Figure.12.1 Architect of AlexNet (Strisciuglio et al., 2020)

2.3.1.4 EFFICIENT NET: In 2019, convolutional neural networks (CNNs) called EfficientNets were presented by Google Brain researchers to address the problem of achieving better accuracy without making the networks too slow. EfficientNet differs from other CNNs by scaling all of its three dimensions depth, width and input resolution—simultaneously. As a result, models are more accurate and also take up less space and operate more quickly when compared to ResNet and AlexNet (Tan & Le, 2019).

The usual approach to scaling a network is to add depth, open more channels or input bigger pictures. On the other hand, simply changing these aspects to any size can lead to ineffective models that are hard to both train and use. EfficientNet suggests a new method called compound scaling that applies the same scale everywhere (Tan & Le, 2019).

This scaling is controlled by a compound coefficient ϕ which controls the way the model scales: -

- Depth:Depth means the number of layers a model can analyze.

  - depth=αϕ
- Width: Number of channels of each tier which means its layers.
  -width=βϕ
- Resolution: Resolution is the size of the images being used as input to the neural network.
  - resolution=γϕ

By grid searching, the constants α, β and γ are found so that performance and accuracy are optimized while following the condition $\alpha \cdot \beta 2 \cdot \gamma 2 \ \approx \ 2$. Because of this approach, it is possible to achieve better results as the system is scaled (Tan & Le, 2019).

EfficientNet Architecture

EfficientNet-B0 utilizes NAS technology to choose the best network design for efficiency. B1 to B7 are made by compound scaling B0 with different values of ϕ. It allows for building various models that perform about the same but offer different processing abilities (Tan & Le, 2019).

1. Accurate Results with Fewer Things to Set:

For example, EfficientNet-B1 provides a higher accuracy on ImageNet (top-1 of 79.1%) compared to ResNet-152 (top-1 of 77.8%), but with just a tiny fraction of the parameters needed by ResNet-152.

2. Faster Inference:

Their smaller design makes EfficientNet models use fewer resources (FLOPS), meaning they can complete tasks more quickly which benefits both real-time and mobile applications.

3. Effective Results from Transfer Learning:

EfficientNet performs better than other CNNs on different tasks, including CIFAR-10, CIFAR-100, Flowers and Stanford Cars and typically requires just 21 times fewer parameters (Tan & Le, 2019).

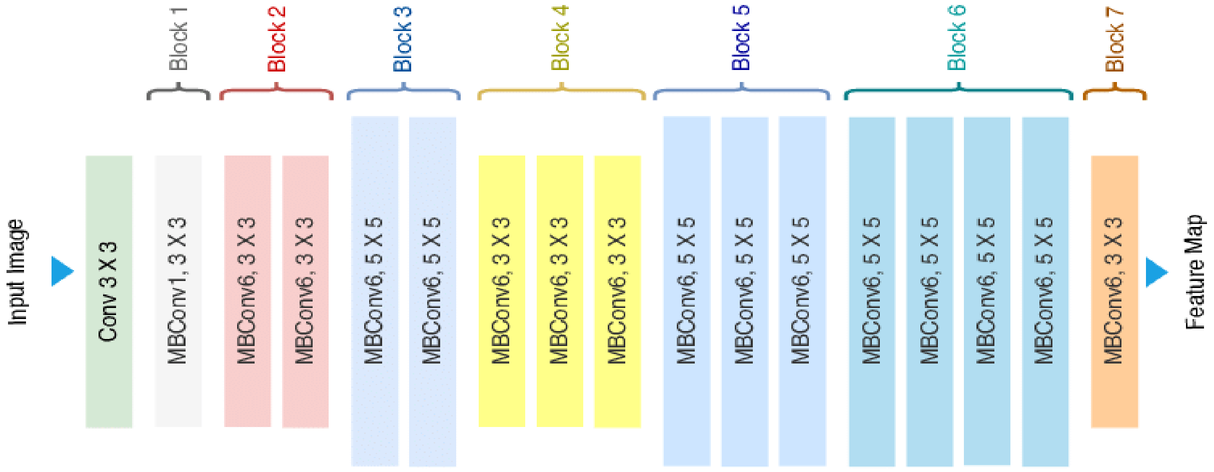


Figure 13.1 Efficient B0 Architecture (Tan & Le, 2019)

2.4 PLANT DISEASE:

Plant Diseases carry many disorders that interfere with plant’s growth and productivity. Among the causes of different agents like virus, bacteria, fungi and Virus. Plant disease can cause serious setbacks in farming and the supply of food. They may reduce what farmers yield, affect the quality of their crops and result in financial losses for everyone involved in agriculture. In a few cases, Plant Disease may cause an entire crop to be lost which could be devastating for farming communities. Examples of Plant Disease include foliar diseases, root diseases, diseases of the stem and diseases found in fruits and seeds. Diseases occurring on plant leaves are referred to as foliar diseases and diseases of the roots are known as root diseases. With stem diseases, the stems, branches and twigs of plants may be damaged and fruit and seed diseases can negatively impact the fruit or seed yield of plants (Pelczar et al., 2025).

It is possible to reduce or treat Plant Disease through cultural, chemical and biological measures. To protect against Plant Disease, people use crop rotation, clean the area and choose plants that are resistant to various diseases. Plant Disease is reduced in Chemical methods using pesticides and fungicides, while it is stopped in biological methods through the help of predatory insects or useful microbes. Preventing Plant Disease is the most effective approach (Pelczar et al., 2025).

Plant Disease can be prevented in farms and agriculture by applying proper agricultural techniques such as planting disease-free seedlings, using rotation and closely checking crops each day. Identifying Plant Disease as early as possible helps with its management and control (Pelczar et al., 2025). To summarize, Plant Disease greatly affect the ability to produce food and grow crops. It is important that PDD is handled in an efficient way for effective management and early detection and appropriate strategies which can require for disease detection for crop and plant agriculture.

• Bacterial Disease- This disease in wheat occurs after water has soaked the plant, producing small green spots. After initial growth, they dry out and become spotted on the surface. Example: There are spots which are black water spots on the leaves that can be both brown or circular yellow with the same size. When soil is dry, the blemish becomes visible as some lighter and darker spots on the skin. Most of the time, bacteria in the soil cause Bacterial wilt which causes the entire plant to fall over (Pelczar et al., 2025).

• Plant Virus- Viral diseases in plants are the hardest to examine among all plant diseases. No clear sign is showing on the virus regarding herbicide bruise and nutrient deficiency (Pelczar et al., 2025). Most virus-carrying insects belong to beetles, leafhoppers, aphids and whiteflies and an example of this is the mosaic viral disease which appears as green or yellow stripes on leaves.

• Fungal Disease- Different parts of a plant can contract a fungal disease, for example, sclerotium wilt, common crown rot, stem rust, eyespot (sheath or stem), rust, blight (leaves), ergot (spikes) and carnal bunt, black point (seeds). When infected with Phytophthora fungus that causes late blight, the former leaves get gray-green spots and can be very wet. It is caused when the fungus enters the body in the form of climate change as wet and dry ones. The late blight disease causes the blemishes to develop dark color and white fungus to grow on the potatoes' surface (Pelczar et al., 2025). We rarely notice this Alternaria-causing fungi on early-blighted leaves in the form of small, brown areas arranged in a series of rings. Rust fungus appears on ripe plant leaves on the area that's curled and folded. This blemish darkens to black at the next stage, after being green-yellow.

Pathogen Disease- The field of plant pathology field give attention on the pathogens, the conditions they create, the mechanisms involved and methods to control and lessen these diseases what affects or damages plants most is commonly known as Phytopathology. For this reason, it is a way to handle the life of a plant. It is clear that phytopathology means plant (Phyto), disease (Patho) and knowledge (Logo) when written in Greek letters. The fields within phytopathology include identifying the sources and causes of plant disease, studying the process of disease development, understanding the connection between plant disease and its pathogen and managing ways to reduce damages caused by such diseases. Phytopathology covers a subdomain in agriculture science and includes the essential elements of microbiology, physiology, nematology, virology, anatomy and bacteriology. This study focuses on mycology, genetic engineering, botany, meteorology, climatology and molecular biology. (Rani & Gowrishankar, 2023)

Several issues and factors can lead to plant disease.Specific types of classification and identification methods are stated below. The primary focus of the PDD system is It depends on how the data is extracted and categorized used. Almost all of the research studies use Plant Village as the main source. The data used is laboratory images instead of images taken in current time. The outcome of the classifier is shaped mainly by the dataset. I use it for testing and training applications. In real photos, it is possible for the background to include various objects for this reason, separating affected places is challenging influences the way the system carries out its tasks (Pelczar et al., 2025) .

2.4.1 CLASSIFICATION OF PLANT DISEASE:

It has several categories and it depends on multiple criteria, like pathogen type and what type of symptoms the affected plant is giving and the mode of transmission of the disease. Here are some common classifications of plant disease (Pelczar et al., 2025).

Table.3.1 Overview of Plant Disease Categories and Examples

Category	Type	Examples
A. Based on Type of Pathogen	Bacterial Diseases	Fire blight, Crown gall
	Nematode Diseases	Root-knot nematodes, Cyst nematodes
	Fungal Diseases	Powdery mildew, Rusts, Blights
	Viral Diseases	Mosaic viruses, Leaf curl viruses
	Parasitic Plants	Dodder, Mistletoe
	Protozoan Diseases	Phloem necrosis (rare)
B. Based on Symptoms	Leaf Diseases	Leaf spot, Leaf blight, Leaf rust
	Stem and Trunk Diseases	Canker, Wilt, Dieback
	Root Diseases	Root rot, Clubroot
	Fruit and Flower Diseases	Fruit rot, Flower blight, Blossom end rot
C. Based on Mode of Transmission	Airborne Diseases	Powdery mildew, Rust
	Soilborne Diseases	Root rot, Fusarium wilt, Clubroot
	Waterborne Diseases	Downy mildew, Phytophthora blight

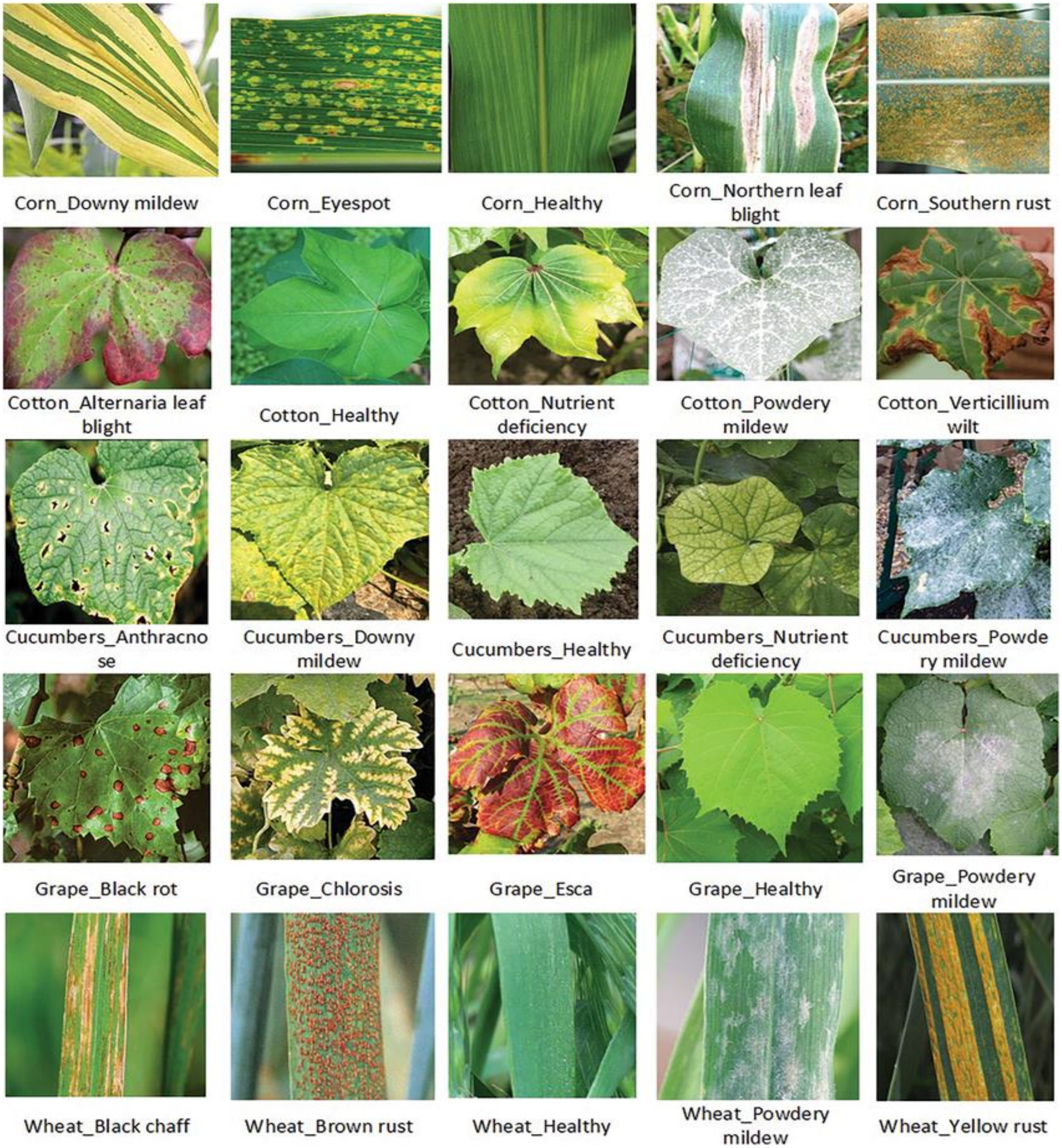


Figure 14.1 In this illustration, five plants and the diseases they may have are shown (Elaraby et al., 2022)

CHAPTER 3
METHODOLOGY

2.5 METHODOLOGY-

The methodological system chosen as a basis in this study is oriented to the create of an automated PDD system based on both the classical ML and DL. This gigantic worldwide database continues to expand, with new photos provided by experts studying the plant diseases or farmers observing the sick plants, as well as registered by regular citizen scientists in the course of its research. This ensures the dataset's availability for other studies in plant pathology and precision farming around the world. The whole process includes collection of data, preprocessing of collecting data which is followed by extracting important features from it. The methodology has been structured in order for it to guarantee strong classification ability on various plant diseases categories.

2.5.1 DATA ACQUISITION

This study dataset is created using PlantVillage dataset it has images very high in resolution of healthy and affected leaves from variety of species of plants. For experimental purposes, this study concentrates on selected crops; i.e., tomato,potato and bell pepper with other disease classes that is, late blight, mosaic virus, target spot, leaf curl, early blight, and healthy conditions. All images are appropriately labeled in order to support the use of supervised learning methods.

2.5.2 IMAGE PREPROCESSING

It is important in facilitating the improvement of quality of input data and strengthening generalization of the model. The following pre-processing steps are taken:

a). Image Resizing: All images are resized to fixed sizes (e.g. 128×128 for ML models and 224×224 for VGG-16 deep learning models) in order to standardize input size. The data provided by PlantVillage comes with pictures of varying sizes and resolutions. Every image used in CNNs undergoes an adjustment where the dimension of image is rescaled to a size of 224x224 pixels. I chose this option because it complies with ResNet50 and VGG16, easy-to-use pre trained architectures that give a good balance of detail and the amount of computation needed.

b). Color Normalization: RGB images are normalized to [0,1] or converted to grayscale based on the model type. To avoid losing important color details for picking out discolorations, spots and mold, the data images were kept in the RGB color format. Applying normalization in inputs positively affects neural networks, controlling their behavior and making their training more efficient. We used model-specific functions to standardize the pixels in the images. Images given to VGG16 are converted to the BGR format and have the data from the ImageNet mean subtracted thanks to vgg_preprocess. To preprocess ResNet50, resnet_preprocess was selected which shrinks pixel values from the original range to between -1 and 1.

Ensuring this step speeds up and simplifies learning from existing model with transfer learning process: -

c). Data Splitting- The data used for this work was separated majorly into training (80%) and the remaining 20% accounting for the validation portions. We trained the model using the training set to understand disease features and the validation set checked that the model would generalize well, helping to avoid it overfitting.

d). Noise Reduction: Filtration in the form of Gaussian blur is used to minimize illumination noise.

e). Data Augmentation: These techniques augmentation techniques such as rotation, flipping, increasing size and changing the brightness level are used to enhanced the size of dataset employed for training purpose. Since the PlantVillage images were taken using the same background and lighting, it was not necessary to use data augmentation. Nonetheless, applying a flip and a slight rotation to images was viewed as reproducing common natural occurrences and improving the model's resistance to errors. Further studies can make augmentation techniques more realistic for common tasks.

f). Data Balancing: For finding out solution of problems of imbalance classes in data sets. The number of instances for different classes of disease might differ. Leading to oversampling or under sampling. Despite the imbalanced data, attempting to alter class size in this aspect was not considered during the experiment. Therefore, the models and their training settings were built to handle such imbalances. Additional studies may involve using class weighting and generating synthetic data.

2.5.3 FEATURE EXTRACTION

For traditional ML models, there is extraction of handcrafted features for representing color, texture, and shape traits of leaf pictures. The following descriptors are utilized:

a). Color Descriptors: Histograms of RGB and HSV color spaces.
b). Texture Features: LBP, GLCM for extraction of spatial correlation and contrast.
c). Statistical Metrics: Mean, standard deviation, entropy, skewness, kurtosis.
d). These feature vectors are then used to feed the classification models.

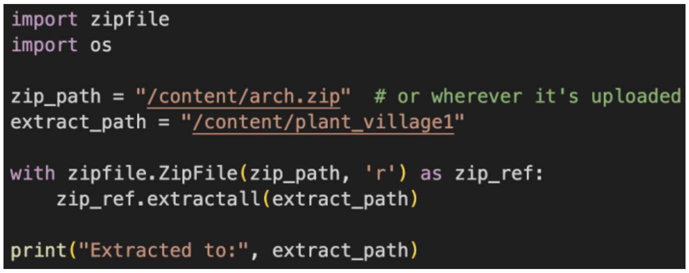
2.5.4 DATASET DESCRIPTION

For this Artificial intelligence Project, the data came from the site named Kaggle and the dataset is PlantVillage dataset used for DL and ML and the data is well known data where we can see the accurate accuracy and accessible data for PDD. It contains > 54,000 images of plant leaves, each label being assigned to one of 15 classes. The sample includes leaves that are healthy and leaves that reflect various diseases found in tomatoes, potatoes and peppers.

Number of classes - There are a total of 15 categories available in the dataset. There is a distinct category for a particular disease in a crop or for a group that shows health in the crop. For example, Tomato_Late_Blight and Potato_Early_Blight are diseases, while Tomato_healthy describes samples of healthy leaves.Overall, the images included in the database make for a strong learning resource for deep learning networks. The pictures are shared equally among the models, yet they are not perfectly distributed due to the real differences in sample collection for diseases.

Imbalanced Classes: The distribution of images among classes is unequal. There are classes with thousands of images and others have just under 200. If care is not taken, machine learning models could work better for the classes that have more data.

Code: -



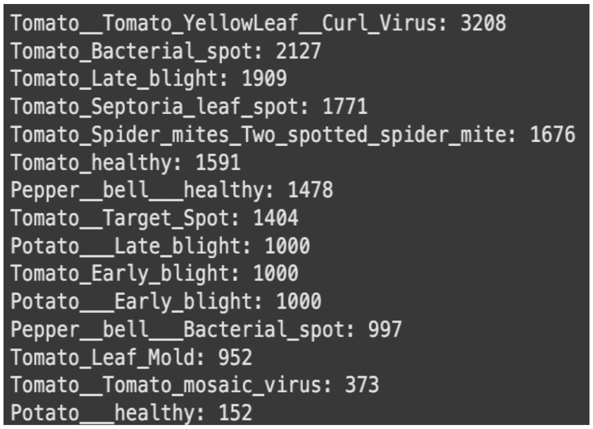
```
import zipfile
import os

zip_path = "/content/arch.zip"  # or wherever it's uploaded
extract_path = "/content/plant_village1"

with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(extract_path)

print("Extracted to:", extract_path)
```

Figure 15.1 Code for Dataset Path



```
Tomato__Tomato_YellowLeaf__Curl_Virus: 3208
Tomato_Bacterial_spot: 2127
Tomato_Late_blight: 1909
Tomato_Septoria_leaf_spot: 1771
Tomato_Spider_mites_Two_spotted_spider_mite: 1676
Tomato_healthy: 1591
Pepper__bell___healthy: 1478
Tomato__Target_Spot: 1404
Potato___Late_blight: 1000
Tomato_Early_blight: 1000
Potato___Early_blight: 1000
Pepper__bell___Bacterial_spot: 997
Tomato_Leaf_Mold: 952
Tomato__Tomato_mosaic_virus: 373
Potato___healthy: 152
```

Figure 15.2 Distribution of Plant disease categories in the Dataset

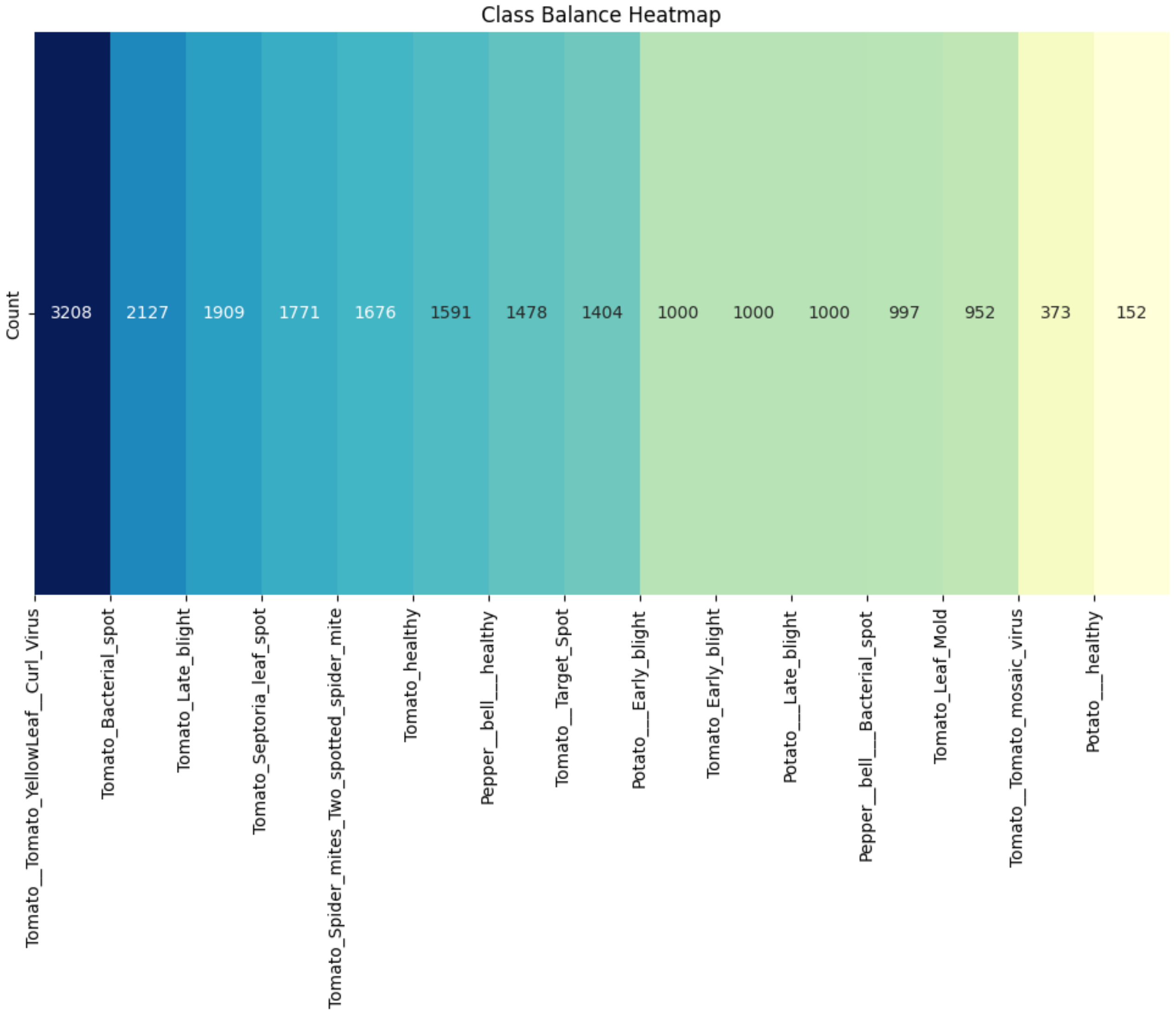


Figure.15.3 The image is a class balance heatmap which clearly displays the number of plant disease samples assigned to every class included in a dataset. Now let’s take a look at the heatmap

Most classes found in this example are:The largest group of 3208 samples was identified as Tomato_YellowLeaf_Curl_Virus. It is the most negative spot on the scale.

Less Common Class:There are only 152 samples under the category of healthy for Potato___healthy. It is the bar that weighs the least in your bag.

Imbalanced Classes: Classifiers with the highest numbers of samples are followed by many with much smaller numbers, with less than 500 samples in some cases.

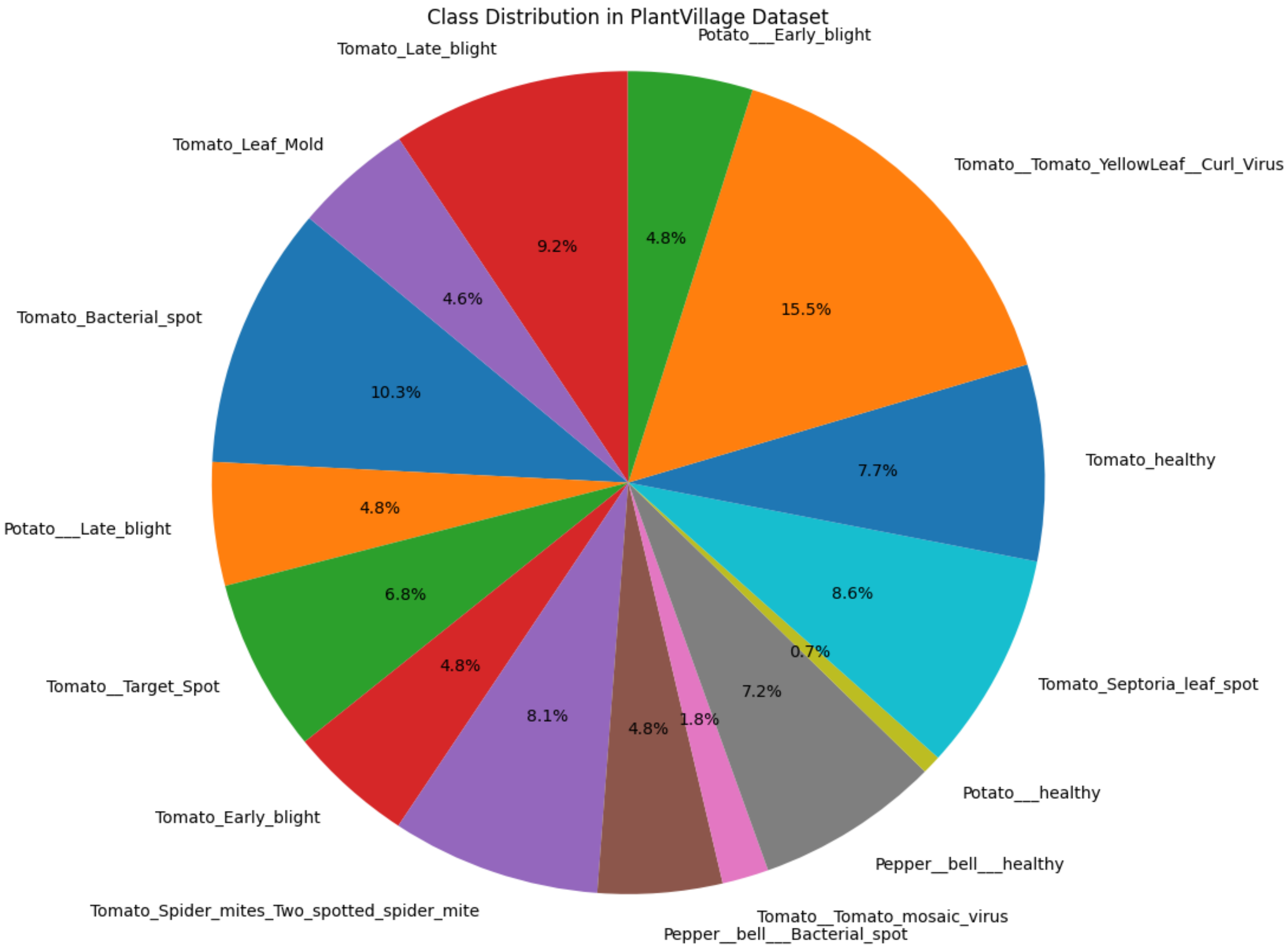


Figure 15.4 The pie chart illustrates the class of plant diseases and healthy cases are represented in the PlantVillage dataset. A large fraction of the dataset (15.5%) were caused by Tomato Yellow Leaf Curl Virus, with Pepper Bell Mosaic Virus making up just 0.7%.

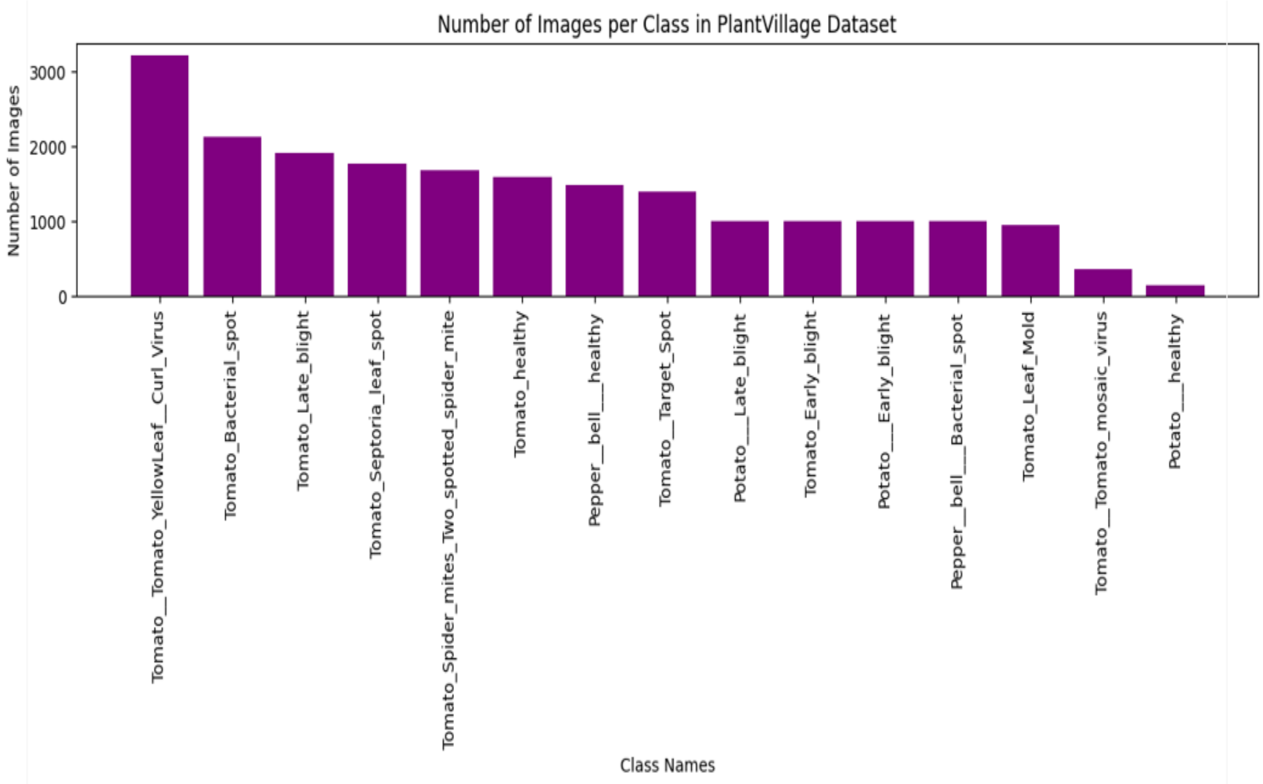


Figure 15.5 Number of Image per class in Plant Village datasets

2.5.5 IMAGE CHARACTERISTIC:

The color format for every image is RGB. allowing us to identify spots, discolorations and mold on the leaf.

a.) Resolution: Because the pictures come from various sources and are captured with different devices, their resolution is not the same. All of the images are sized to be exactly 224x224 pixels to make Type 1 convolutional neural networks (CNNs) function properly. Since VGG16 and ResNet50 are pre-trained CNNs, using these filters with a size of 7×7 helps maintain the image detail while keeping the computation time within reasonable levels.

b). Dataset Images: Images used in the dataset were taken in a laboratory under the same lighting and with a plain background. Noise is minimized, so the models concentrate mainly on leaf details. Even so, indoor testing does not normally mimic the outside, since lighting, other framework and the way leaves face are not the same in nature. Automated plant disease detection systems are developed and evaluated using the PlantVillage dataset because it provides a solid base for testing. It ensures there is a large dataset ready for deep neural networks to learn from. It can be used for different crops and a wide variety of diseases which makes the models flexible. The images are taken using healthy leaves to train the computer in telling diseased from normal ones. Nevertheless, models developed from these images may not work as well in real field situations since such a controlled environment was used for acquiring the images.

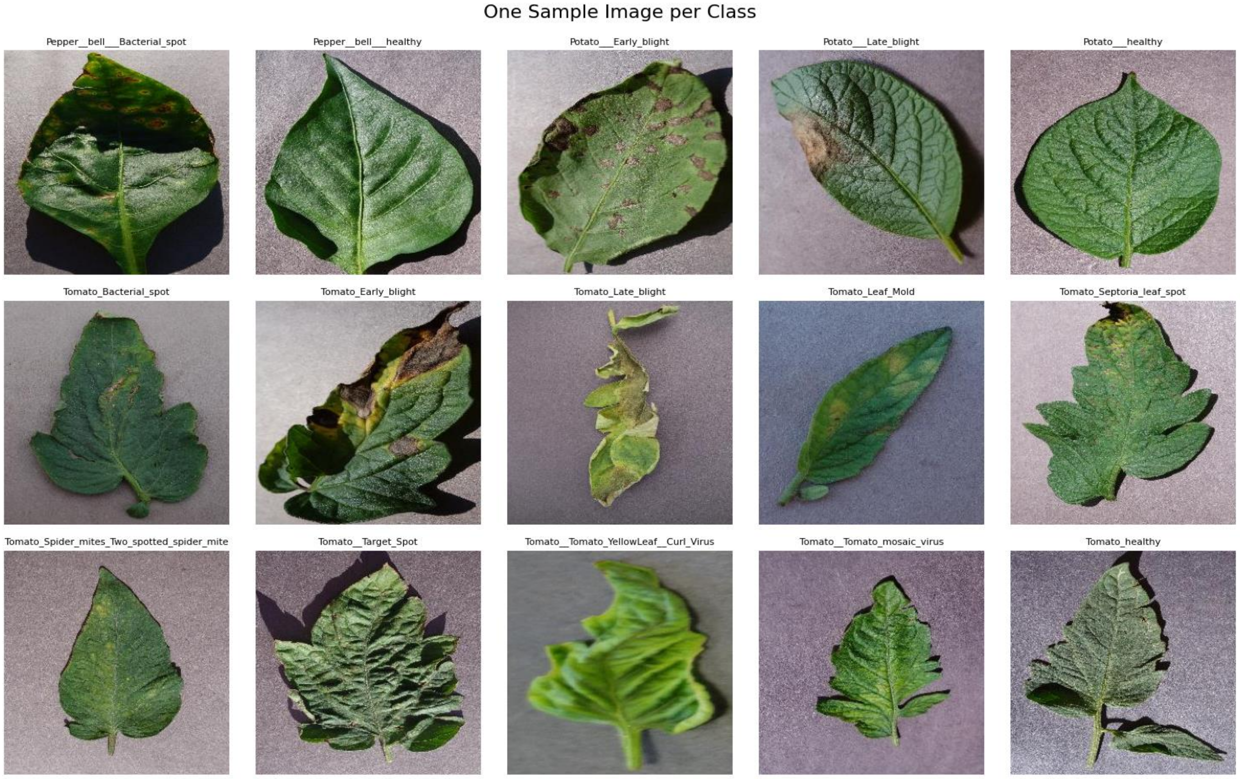
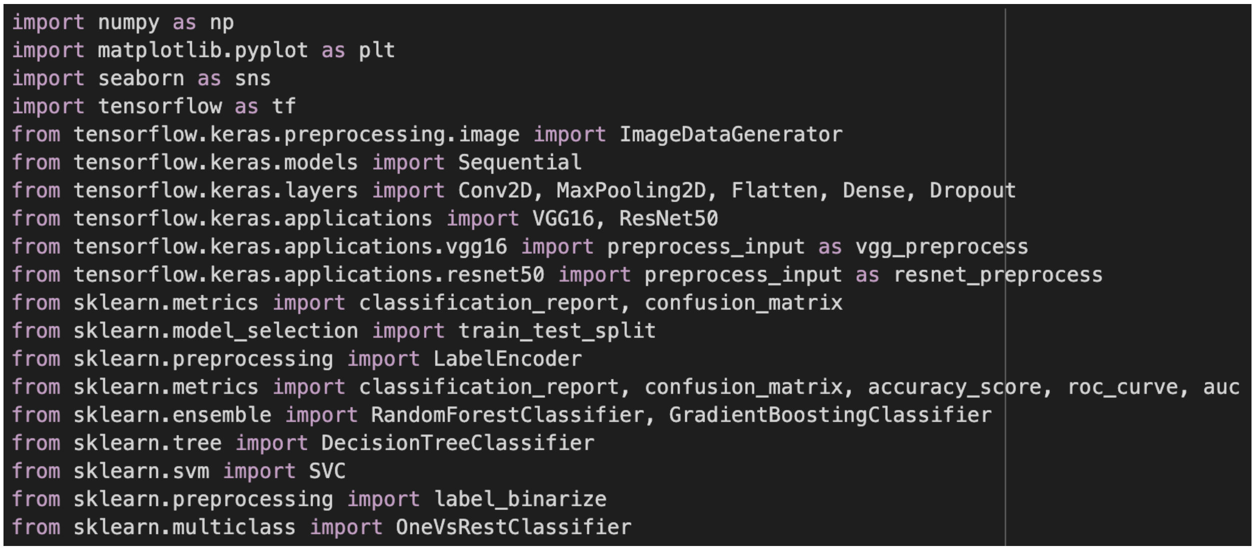

One Sample Image per Class
Pepper__bell___Bacterial_spot
Pepper__bell___healthy
Potato___Early_blight
Potato___Late_blight
Potato___healthy
Tomato_Bacterial_spot
Tomato_Early_blight
Tomato_Late_blight
Tomato_Leaf_Mold
Tomato_Septoria_leaf_spot
Tomato_Spider_mites_Two_spotted_spider_mite
Tomato__Target_Spot
Tomato__Tomato_YellowLeaf__Curl_Virus
Tomato__Tomato_mosaic_virus
Tomato_healthy

Figure 16.1 Image Characteristics in Plant Village datasets

2.5.6 **LIBRARIES**:

Code:-



```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.applications import VGG16, ResNet50
from tensorflow.keras.applications.vgg16 import preprocess_input as vgg_preprocess
from tensorflow.keras.applications.resnet50 import preprocess_input as resnet_preprocess
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, roc_curve, auc
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
```

Figure 17.1 Code For Libraries

1. Numpy
   - ❖ Purpose: NumPy is used for basic computations involving numbers in Python.
   - ❖ Applications in DL: To handle arrays, handle image data, convert labels and change data structures.
2. Matplotlib.pyplot
   - ❖ Function: matplotlib.pyplot helps to display 2D charts and plots.
   - ❖ In DL: Helps to display the accuracy/loss score during training and validation, sample pictures, prediction results and confusion matrices.
3. Seaborn
   - ❖ Purpose: Seaborn uses the matplotlib library to create data visualizations.
   - ❖ It can be employed in DL to create better-looking heatmaps for confusion matrices.
4. TensorFlow
   - ❖ Aim: TensorFlow is an open-source library that is fully dedicated to deep learning and was developed by Google.
   - ❖ DL uses them for constructing and training models such as CNNs, LSTMs and other types.
5. ImageDataGenerator
   - ❖ Purpose: A tool to transform and enlarge data one can use for learning.
   - ❖ Purpose: Rotation, flipping, zooming and similar techniques to better the images and increase CNN models generalizability.
6. Sequential
   - ❖ The Sequential model puts layers in a single line or sequence.

- To use a simple sequential model, you have only input and output variables and information progresses through the layers.

5. Conv2D, MaxPooling2D, Flatten, Dense and Dropout.
   - Conv2D: Layer that finds patterns in different parts of an image based on its size and location.
   - MaxPooling2D is used to shrink the size and reduce the computation needed.
   - It converts the flatten the 2D maps to a vector where all the dimensions are 1.
   - Fully connected layer is used when making a prediction or performing classification or regression.
   - Dropout: Method used to prevent overfitting by temporarily turning off some hidden units randomly.
6. VGG16, ResNet50
   - Purpose: Loads the VGG16 and ResNet50 CNN models that were previously trained on ImageNet.
   - Purpose: When used in transfer learning, this defined can be improved models be either it can be extracting the data with useful features or tweaked to recognize diseases in plants.
7. VGG_preproces
   - Processing: Converts images to the style VGG16 is designed to read.
   - Usage- The procedure involves changing the format and scale of the pixels and using the rescaled data for the VGG16 network.
8. RestNet Preproces
   - The purpose here is the same as before, but for ResNet50.
   - Usage- Ensures that the given image will be handled properly considering ResNet's need for regular formatting and color handling.
9. Classification_report and Confusion matrix
   - Classification Report- Each class will obtain Precision, recall, F1score.
   - Confusion matrix - In this matrix will show predicted labels vs matrix showing true labels.
   - Usage- Estimate the whole code or problem of your model then will do prediction.
   - Train and Test
   - In this library the function is divide the data into subset the model with tested and trained which ensure the model will evaluate which is impartial
   - Since in plant disease classification, we need to test how well the model generalizes, we split the data using a random method. The rule of thumb is to train your data 80% of the time and test it 20% of the time.
   - Both deep learning and machine learning models train using numerical inputs. By encoding labels, we make sure the labels in plant disease data can be read by classifiers. Multiclass problems are where it proves very handy.
10. Confusion Matrix, Classification Report, Accuracy Score, Roc Curve, AUC

- Classification Report
- Shows you accuracy, recall with Score of F1 and adjusted probability values for all supported classes. Helps in research: Reports how the model deals with real issues from a range of diseases.
- Confusion Matrix
- Presents actual and predicted labels side by side in a matrix.
- Researchers employ the data to determine which conditions tend to be mistaken for others.
- Accuracy Score:
- It can find how many samples of this models predicted which is correctly total number is dived by the data sample. In research, a model can be evaluated with this metric as a starting point.
- Roc Curves
- Plots both the TPR and the FPR against different threshold values. Research applications: It is helpful to illustrate the capability of classification models in binary or multiple output situations.
- AUC(Area Under the Curves)
- Describes how well the classifier works; values closer to 1 mean the classifier does well. Can be used for research: Frequently paired with roc_curve to assess the performance of a classifier in binary and one-vs-rest methods.

11. RandomForestClassifier:
    - This method combines a number of decision trees into one to make its prediction.
    - Scientists find it useful to identify plant diseases because it is outstanding at processing all the pixels in an image. Unlike a single tree, it lowers variation and stops overfitting.
    - GradientBoostingClassifier:
    - Purpose: Assembles models step by step, making each model aim to address mistakes from those before. When it comes to medical research, this method provides increased accuracy for images where differences in small image elements set apart different diseases.
12. DecisionTreeClassifier
    - Usage: This model splits data using thresholds from different “features” to assign samples to classes. Despite overfitting and being easier to use, decision trees play a role in research to start benchmarking tasks and provide clean decision rules for open, understandable tasks like advising farmers.
13. SVC
    - Support Vector Classifier builds the ideal hyperplane to divide classes in complex data sets.
    - They perform very well when used in research on small to medium datasets. Many people rely on these techniques for the early affected by infected plants where healthy and diseased

classes can be clearly told apart by certain features. Because it supports several kernels, it is flexible for both linear as well as nonlinear classification.

2.5.7 MODEL DEVELOPMENT

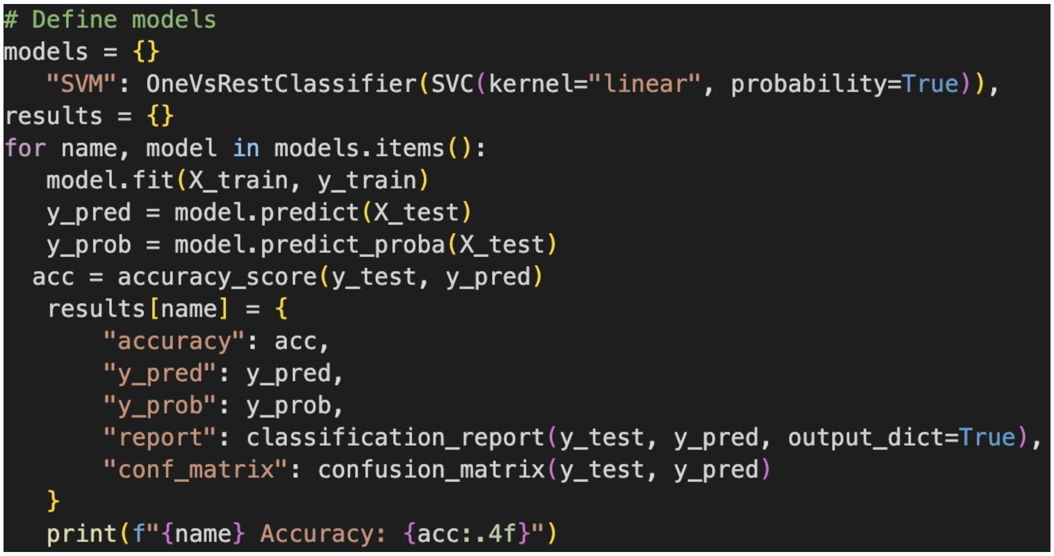
ML and DL are the most effective and important tools in AI where it can work for detecting an unhealthy plant because they can identify a certain type of pattern and give an accurate precision score, F1 score, recall and support score and give attributes from digital photographs.

2.5.8 ML models:

Machine Learning (ML) is also gaining popularity in agriculture where it is used to diagnose diseases of plants at an early and accurate stage. The ML models can explore visual aspects of images from leaves to detect blight, rust, mildew, and others. This assists the farmers can use the time and take actions, hence decreasing the loss of crops and increasing the yield. Some of the techniques such as SVM, RF, GB and DT etc. are widely used owing to the strength they possess in image classification. However, this training requires the use of datasets such as PlantVillage which comprises thousands of labeled plant disease images.
Four supervised learning classifiers are deployed.

2.5.8.1 SVM Model (SVM): Used for its capability to build high-margin decision boundaries and deal with high-dimensional feature space.

code:

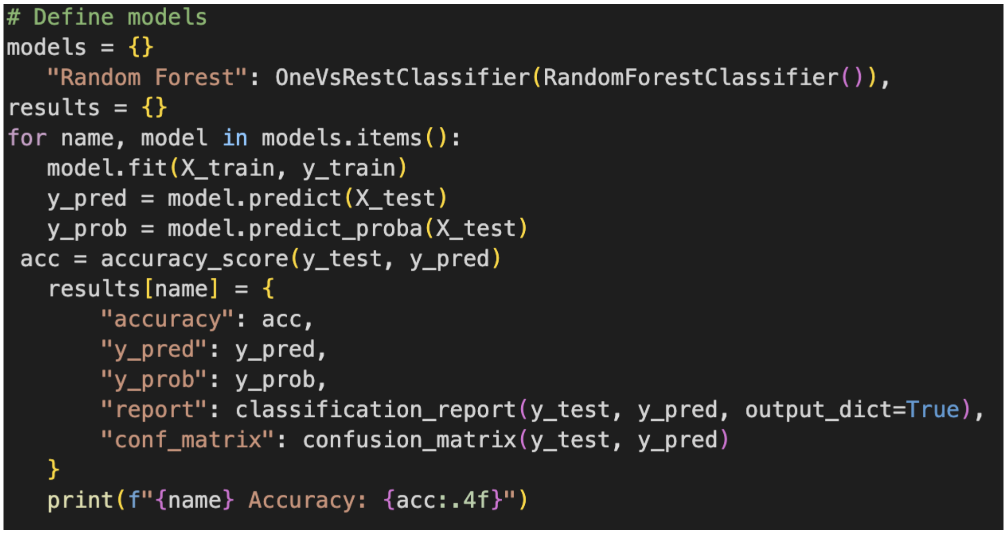


```
# Define models
models = {}
   "SVM": OneVsRestClassifier(SVC(kernel="linear", probability=True)),
results = {}
for name, model in models.items():
   model.fit(X_train, y_train)
   y_pred = model.predict(X_test)
   y_prob = model.predict_proba(X_test)
  acc = accuracy_score(y_test, y_pred)
   results[name] = {
       "accuracy": acc,
       "y_pred": y_pred,
       "y_prob": y_prob,
       "report": classification_report(y_test, y_pred, output_dict=True),
       "conf_matrix": confusion_matrix(y_test, y_pred)
   }
   print(f"{name} Accuracy: {acc:.4f}")
```

Figure 18.1 Code For SVM

2.5.8.2 RF Model (RF): An ensemble classifier which is based on several DT and used in sequence to enhance accuracy and avoid over fitting.

code:

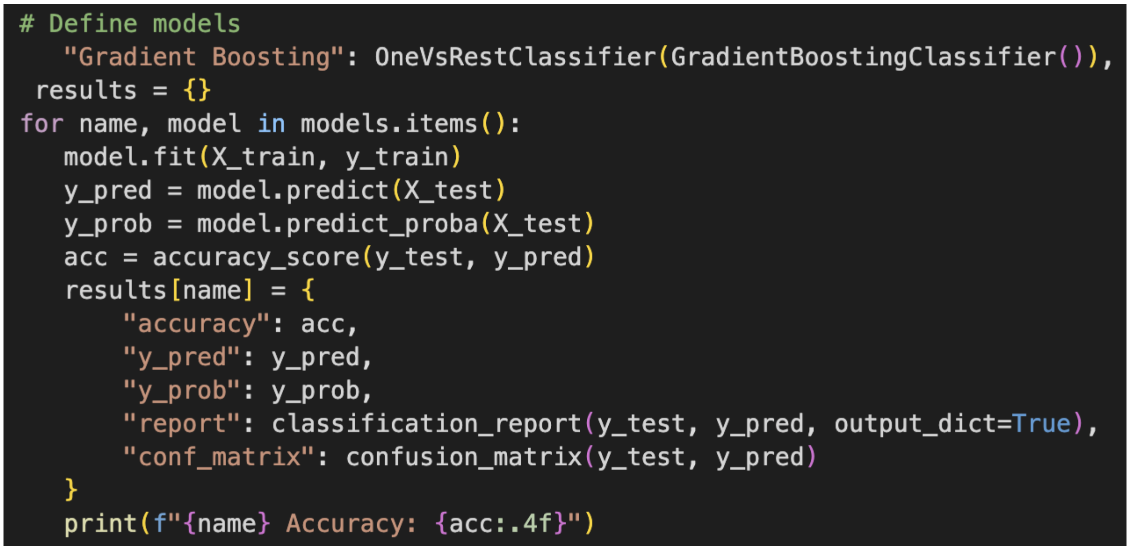


```
# Define models
models = {}
   "Random Forest": OneVsRestClassifier(RandomForestClassifier()),
results = {}
for name, model in models.items():
   model.fit(X_train, y_train)
   y_pred = model.predict(X_test)
   y_prob = model.predict_proba(X_test)
 acc = accuracy_score(y_test, y_pred)
   results[name] = {
       "accuracy": acc,
       "y_pred": y_pred,
       "y_prob": y_prob,
       "report": classification_report(y_test, y_pred, output_dict=True),
       "conf_matrix": confusion_matrix(y_test, y_pred)
   }
   print(f"{name} Accuracy: {acc:.4f}")
```

Figure 18.2 Code For Random Forest

2.5.8.3 GB models: The Gradient Boosting approach where each model tries to adjust mistakes from the previous model by repeating gradient descent. Its results in classification are highly successful and it performs especially well when using structured data.

Code:

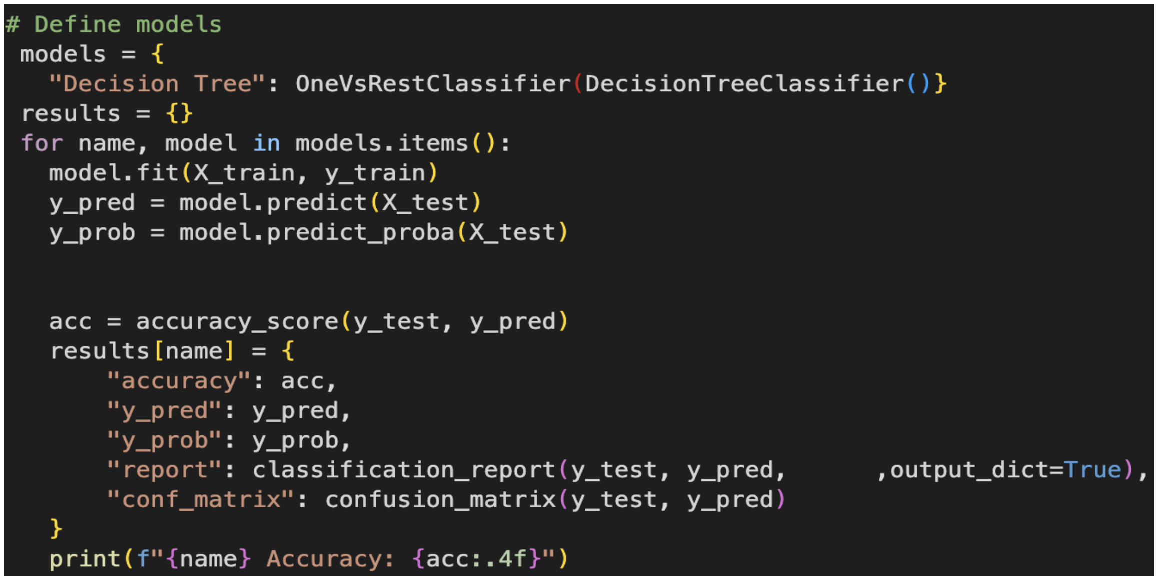


```
# Define models
   "Gradient Boosting": OneVsRestClassifier(GradientBoostingClassifier()),
 results = {}
for name, model in models.items():
   model.fit(X_train, y_train)
   y_pred = model.predict(X_test)
   y_prob = model.predict_proba(X_test)
   acc = accuracy_score(y_test, y_pred)
   results[name] = {
       "accuracy": acc,
       "y_pred": y_pred,
       "y_prob": y_prob,
       "report": classification_report(y_test, y_pred, output_dict=True),
       "conf_matrix": confusion_matrix(y_test, y_pred)
   }
   print(f"{name} Accuracy: {acc:.4f}")
```

Figure 18.3 Code for Gradient Boosting

2.5.8.4 DT models: The Tree-based model that assigns a partition to data based on feature thresholds and gives interpretable results.

Code:



```
# Define models
 models = {
   "Decision Tree": OneVsRestClassifier(DecisionTreeClassifier()}
 results = {}
 for name, model in models.items():
   model.fit(X_train, y_train)
   y_pred = model.predict(X_test)
   y_prob = model.predict_proba(X_test)


   acc = accuracy_score(y_test, y_pred)
   results[name] = {
       "accuracy": acc,
       "y_pred": y_pred,
       "y_prob": y_prob,
       "report": classification_report(y_test, y_pred,      ,output_dict=True),
       "conf_matrix": confusion_matrix(y_test, y_pred)
   }
   print(f"{name} Accuracy: {acc:.4f}")
```

Figure 18.4 Code for Decision Tree

For each model, an 80:20 train-test split is chosen to train the model first, with 5-fold cross-validation used to test for generalization performance.

2.5.9 DL MODELS:

In DL, plant disease detection can be done automatically by learning to identify the important aspects in the raw images itself. With these classic methods, inspecting features by sight or touch usually is not effective for revealing early or mild disease changes. On the other hand, CNNs are DL models that accomplish intricate pattern recognition and work well on leaf images.

In PlantVillage dataset, containing many labeled pictures of plant leaves, deep learning models can be determining numerous plant diseases. Still, DL applications rely on large collections of labeled data and powerful computers. These challenges may be tackled using transfer learning, since VGG16, ResNet and AlexNet models can be retrained with plant disease data and require less time and data than other models. Flipping and rotating data increases the performance by introducing more variety to the training data.

Deep learning may need strong equipment and explain less than other approaches, but it is still more accurate in precision agriculture. Plant disease can be identified quickly and accurately using the system, providing much potential for its use with farms. Software that uses data such as that from PlantVillage, is supporting better disease management in smart farming.

2.5.9.1 VGGNet Models - Use of VGG16 as a foundation: Begins with a series of 13 convolutional layers divided into blocks and ends with a sequence of max-pooling layers. The base was frozen during workouts to ensure it retained the basic features it learned from ImageNet.

Custom Top layer:

a). After the core layers of convolutions, the layers are fully connected that are used:

b). A flatten with layers meant for transferring the data from a 2D grid into a 1D list.

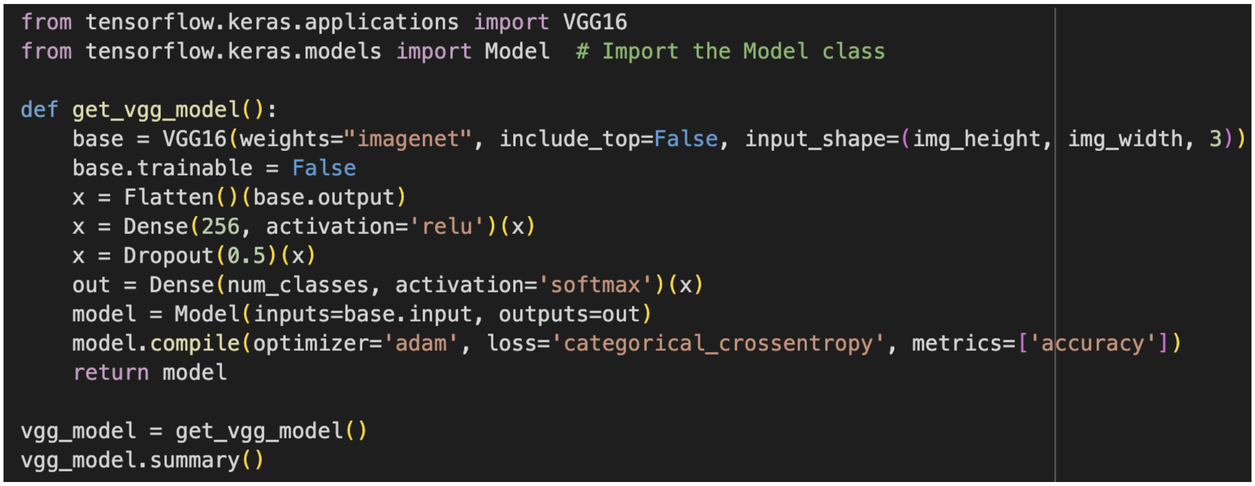
c). There is a layer with various neurons which is 256 and the ReLU function used for training the network to recognize complex features.

d). Reducing overfitting by using a 0.5 rate of dropout with a rate of models.

e). The Last model of softmax layer has 15 neurons that represent the different disease classes.

Transfer Learning: Utilized previously trained models to boost the training process and get better results when there were not that many training epochs.

Code:-



```
from tensorflow.keras.applications import VGG16
from tensorflow.keras.models import Model  # Import the Model class

def get_vgg_model():
    base = VGG16(weights="imagenet", include_top=False, input_shape=(img_height, img_width, 3))
    base.trainable = False
    x = Flatten()(base.output)
    x = Dense(256, activation='relu')(x)
    x = Dropout(0.5)(x)
    out = Dense(num_classes, activation='softmax')(x)
    model = Model(inputs=base.input, outputs=out)
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model

vgg_model = get_vgg_model()
vgg_model.summary()
```

Figure 19.1 Code For VGG16

This code define as it creates and configures a deep learning system for labeling images using mostly transferred learning with VGG16 which was trained on ImageNet. Instead of using it for object classification, this process looks at using VGG16 for plant disease detection.

1. Including the Required Libraries

- VGG16 is loaded into the code through tensorflow.keras.applications. A CNN is tested where the training is based on millions of pictures from ImageNet.
- To create a custom model, the class of models from tensorflow.keras.models this can be used to amend the base VGG16 framework.

2. The Model function is named get_vgg_model().

- With this function, a model is made specifically from the features in VGG16.The first step is to build the base model.
- This way of initializing the VGG16 model means it will not include the final fully-connected layers, making it possible for us to add our own classification layer.
- Make sure the input shape is the same as that of the training images (e.g., 224×224×3).
- The model makes use of features from a large dataset thanks to pre-trained ImageNet weights.
- Locking the Base Model in Place. If base.trainable = False, the convolutional layers of VGG16 are frozen during training and their weights do not change. It can be the learning process where the process is faster and protects against overfitting, especially on a limited set of training data.

3. Custom Classification Head

- Convolutional layers provide a 3D output, so a Flatten layer is placed between them to transform it into a 1D vector. A set of layers are fully connected with neurons which is

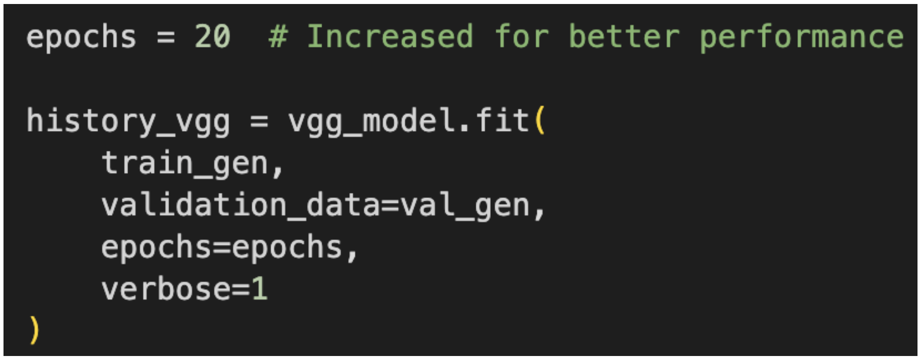
256, each with ReLU activation, is used to make the model capable of handling tough patterns.

- Dropout layer is applied to ensure that the network does not learning using data given for training.
- The last layer is a Dense one which has num_classes units and the softmax for doing classifications. It puts out a set of probabilities for each of the classes that are being predicted.

4. Model Compilation:

- The model of Adam is optimised by algorithm.Since we have multiple groups of classes, we use the cross-entropy loss function is often used in the category setting.The measure for performance is accuracy.
- To view the summary, call get_vgg_model () and the function are used in summary () to see the function all the layers, how each layer outputs and the overall number of parameters.

Code-



```
epochs = 20  # Increased for better performance

history_vgg = vgg_model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=epochs,
    verbose=1
)
```

Figure of 19.2 Code for VGG16 epoches

Here, the max_epochs is set to 20. When 20 epochs is increased for better performance, the model is given more chances to train and improve its learning results by updating the weights for many passes through the input data. If the value for the parameter is too high, it may result in overfitting and should be addressed through proper validation.

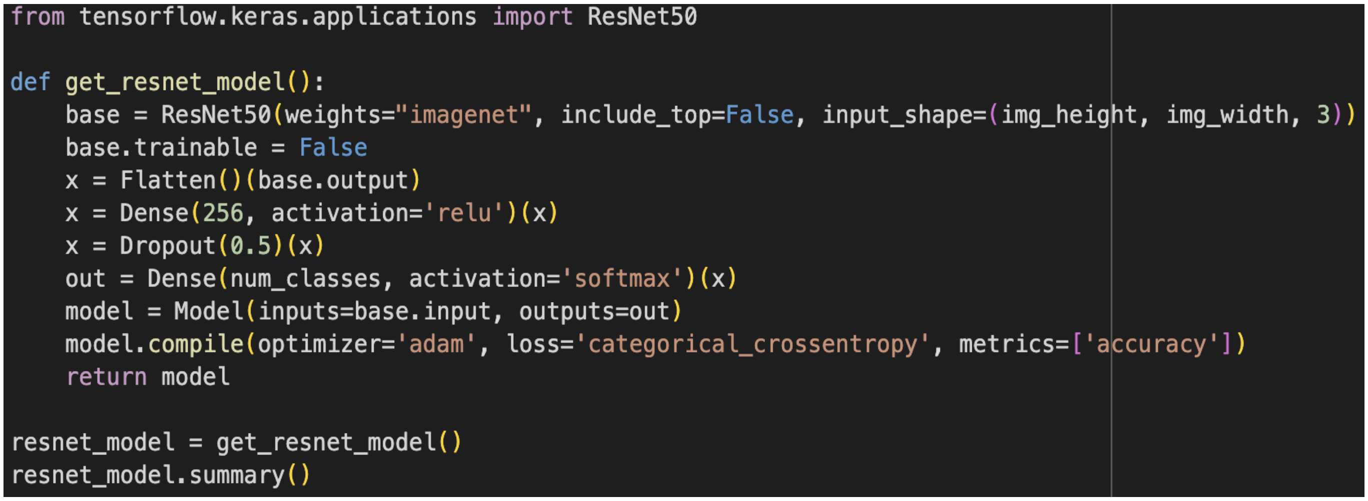
The fit () method from the Keras API in TensorFlow is used in the code to begin training the VGG16 model. This is what every part of a computer is responsible for:

1. vgg_model.fit(...): Carries out VGG16 training using the selected datasets specified for training and validation.
2. train_gen: It produces training data using ImageDataGenerator. It passes sets of prepared and enhanced images to the model during training.

3. validation_data=val_gen: This creates a generator for the validation data. After every one of the model’s training epochs, it is employed to gauge the model’s progress before any changes are made to the weights. This way, you can watch for overfitting.
4. epochs=epochs: Sets the model to train the data 20 times.
5. verbose=1: with this set to 1, detailed information about each epoch is displayed such as the displays for the training and validation loss and accuracy.

2.5.9.1 RestNet- Just like VGG16, ResNet50 is built on a pre-trained base of 50 layers that connects through residual flows. This way, deep networks can learn efficiently since the residual blocks fix the vanishing gradient challenge.
Frozen base- During the first training, the convolutional layers were kept fixed so that their ImageNet feature learning was not altered.
Top Layers- Added the same layers of flatten, dense, dropout and softmax as was done in VGG16.The stronger and more detailed features learned by ResNet50 help improve the detection of hard-to-find, minor disease indications.

Code:-



```
from tensorflow.keras.applications import ResNet50

def get_resnet_model():
    base = ResNet50(weights="imagenet", include_top=False, input_shape=(img_height, img_width, 3))
    base.trainable = False
    x = Flatten()(base.output)
    x = Dense(256, activation='relu')(x)
    x = Dropout(0.5)(x)
    out = Dense(num_classes, activation='softmax')(x)
    model = Model(inputs=base.input, outputs=out)
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model

resnet_model = get_resnet_model()
resnet_model.summary()
```

Figure 19.3 Code For RestNet 50

1. Importing the Pretrained Model
    - It imports the ResNet50 model from TensorFlow Keras applications. ResNet50 is made up of 50 layers and makes use of residual connections which make it easy for the model to train on a deep network.

2. Defining the Model Function
    - The Model Function is called get_resnet_model (). It makes and gives back a customized version of ResNet50, designed to recognize plant diseases.

3. Initializing the Base Model

- weights="imagenet" allows the model to load knowledge obtained from ImageNet dataset. If you set include_top to False, no fully connected layers will be involved so that you can create your own layers.
- This code says that input_shape= when the image height and image width (img_height, img_width, 3) tells us how the input images are shaped. The model needs RGB pictures that are the specific system dimensions.

4. Disabling Training on the Pretrained Layers

- The ResNet50 base weights are not changed during training because they are frozen. This way, we get to use the model's existing strong skills at low-level feature extraction from ImageNet.

5. Adding a Custom Classification Head means:

- Flatten: Makes a 3D output from ResNet50 into a 1D vector.In this step, we use a fully connected neuron with layers and 256 layers and activation function ReLU to find out the highlight of levels information.
- Dropout(0.5) means that 50% of the network's neurons are turned off during training to prevent overfitting. Having a layer rich with many neurons, which matches the total number of classes, that uses the softmax function. Softmax produces a probability distribution for each class in the prediction.

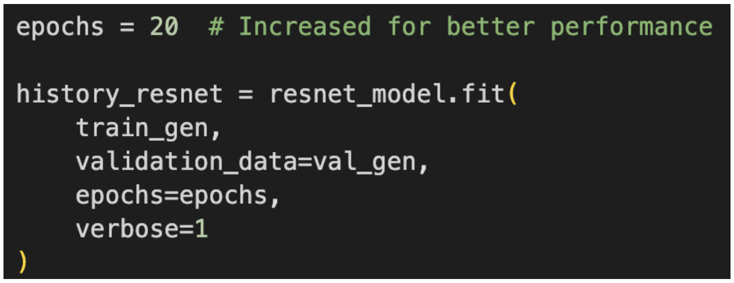
6.Creating and Compiling the Model

- The two parts of the code that has models are the ResNet50 backbone and the layers with new output from the classification head. Adam is used as the optimizer when learning is done adaptively. For problems where multiple categories are involved, the loss function is categorical_crossentropy.

7. Summary:get_resnet_model() starts the model.

- A summary of step by step layers, along with its output size and the various parameters , is printed by typing resnet_model.summary().

Code:-



```
epochs = 20  # Increased for better performance

history_resnet = resnet_model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=epochs,
    verbose=1
)
```

Figure 19.4 Epoches of RestNet 50

The maximum values epochs can have is set to 20. All the data from the training set is used once for an epochs.The trained model across various epochs mainly 20 gives it more opportunity to find patterns in the data which can increase both its accuracy and the likelihood that had will the new data and performance. It may cause the model to overfit if there are too many epochs.
The keras fit() functions are to train RestNet50 model.

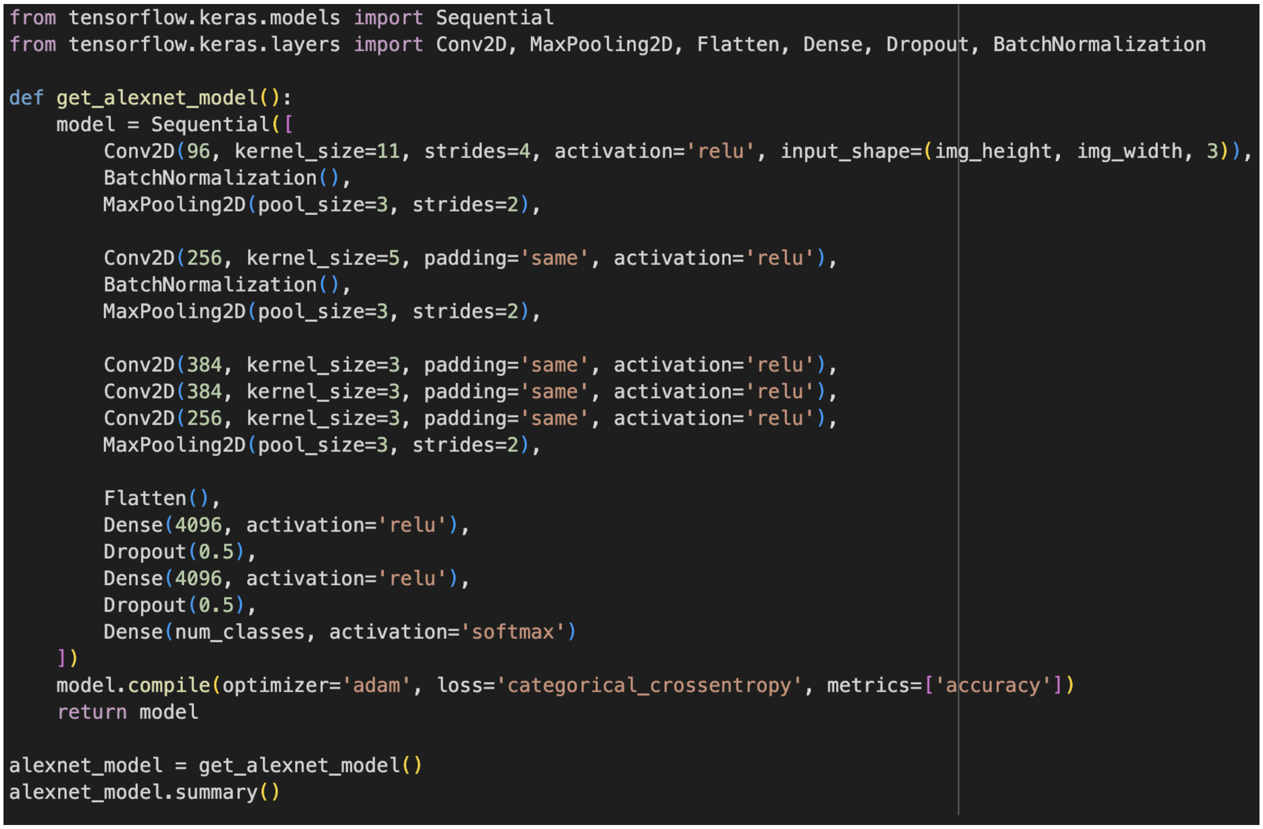
a). train_gen is the data generator (probably built with ImageDataGenerator) which gives batches of modified and preprocessed images to the model while it is being trained.
b). val_gen is used for validation_data to calculate how the trained model works on data it has not seen during each training epoch. It lets you see how much generalization the model achieves.
c). epochs: Sets the amount of times the entire dataset should be used in training (the dataset is used 20 timeshare. Enables you to monitor training progress in the console in detail. Metrics for loss and accuracy are shown for trained and validated data after the systems have completed an epoch.

The training process saves its result in this variable. You will find a log of metrics, consisting of training loss,validation loss, training accuracy, and validation accuracy for every epoch. Such data can be used later to plot important results such as learning curves.

2.5.9.3AlexNet: This is CNN code that can help AlexNEt architecture and the first setup shared by Alex Krizhevsky in 2019. This models is created model of Keras Sequential, API and its purpose is to recognize which kind of plant disease is shown in an image.
We built an AlexNet model according to its initial design, with no starting library used.
Layers: There were several convolutional layers and two separate actions, batch normalization and max pooling, to keep learning stable and decrease the image’s spatial size.
There are dense layers which are 2 types where each with neurons which are 4096 neurons and other are dropouts with a layer which softmax at the end.This model was built to be a point of comparison with transfer learning models.
This architecture of network of consists of a Sequential layer made up of convolutional, pooling, normalization and fully connected layers.

This AlexNet Implementation Effective:
After using multiple convolutional layers, the model is able to pick up on detailed higher-level features helpful for telling apart various diseases.
They have Dropout layers and Normalization with Batches are often used to reduce overfitting and increase a capability of model to simplify the essential data to used in real agriculture.
Flexibility: This model can handle many types of image classification jobs, for instance, plant pathology, pest detection and judgment of crop quality.

Code:



```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization

def get_alexnet_model():
    model = Sequential([
        Conv2D(96, kernel_size=11, strides=4, activation='relu', input_shape=(img_height, img_width, 3)),
        BatchNormalization(),
        MaxPooling2D(pool_size=3, strides=2),

        Conv2D(256, kernel_size=5, padding='same', activation='relu'),
        BatchNormalization(),
        MaxPooling2D(pool_size=3, strides=2),

        Conv2D(384, kernel_size=3, padding='same', activation='relu'),
        Conv2D(384, kernel_size=3, padding='same', activation='relu'),
        Conv2D(256, kernel_size=3, padding='same', activation='relu'),
        MaxPooling2D(pool_size=3, strides=2),

        Flatten(),
        Dense(4096, activation='relu'),
        Dropout(0.5),
        Dense(4096, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])
    model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return model

alexnet_model = get_alexnet_model()
alexnet_model.summary()
```

Figure 19.5 Code For AlexNet

The image used is always a pre-set RGB image, for example 227×227 or 224×224. Let’s explore what each layer consists of:

1. Convolutional Layers:

- At the start, I used 96 filters that were 11×11 in size and moved by 4 pixels, then applied Batch Normalization and Max Pooling.
- Following the first convolutional block, the second set of filters are 256 in number and measure 5×5, again with normalization and pooling performed afterward.
- In the 3rd, 4th and 5th layers, there are 384, 384 and 256 filters, all with 3×3 kernels and padding.

2. Pool and normalize your data:

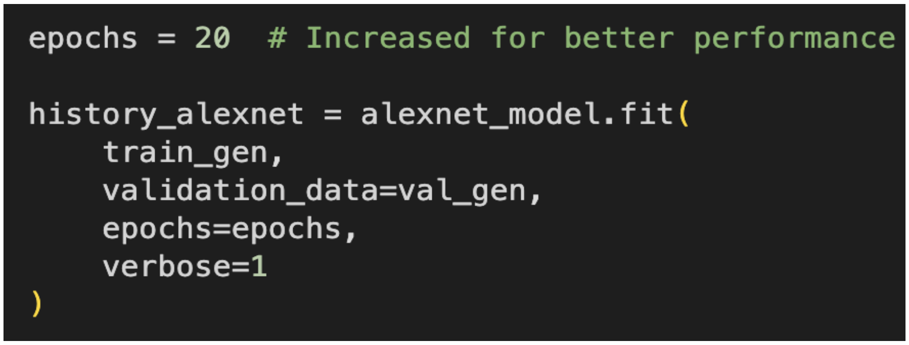
- These MaxPooling2D layers come after major convolutional blocks which lowers the spatial size, shrinks the feature maps and helps manage overfitting.
- In order to keep learning stable and speed up progress, BatchNormalization layers are included.These layers acquire their information from the previous layer.

- The maps are made of features into a single dense layer which goes through two layers with 4096 neurons and ReLU activation, set apart by Dropout (0.5) to avoid overfitting.
- It is the softmax function that the output layer uses to produce its predictions and then sorts the input into one of the available categories (e.g., plant disease types).

3. Compilation

- For Classification with Multi-Option problems, the data model relies on the Adam optimizer and the categorical when the cross entropy loss feature. During training and testing, we rely on accuracies as the method for measuring performance.

Code:-



```
epochs = 20  # Increased for better performance

history_alexnet = alexnet_model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=epochs,
    verbose=1
)
```

Figure 19.6 Epoches Of AlexNet

As a result, the model will run for 20 epochs. An epoch happens each time the training dataset viewed once. The more times you train the model, the greater its chance to improve its weights. Even so, adding too many epochs might cause overfitting which is why 20 is usually an appropriate value for PlantVillage's dataset. Training AlexNet involves using the data sent in groups by the data generators. This function is the training data generator and it is created through ImageDataGenerator.flow_from_directory(). It reads images from the chosen directory and often with data augmentation applies different movements, rotations, mirrors and zooms to each one live.Increases the model's strength in dealing with new issues by presenting it with various kinds of data.

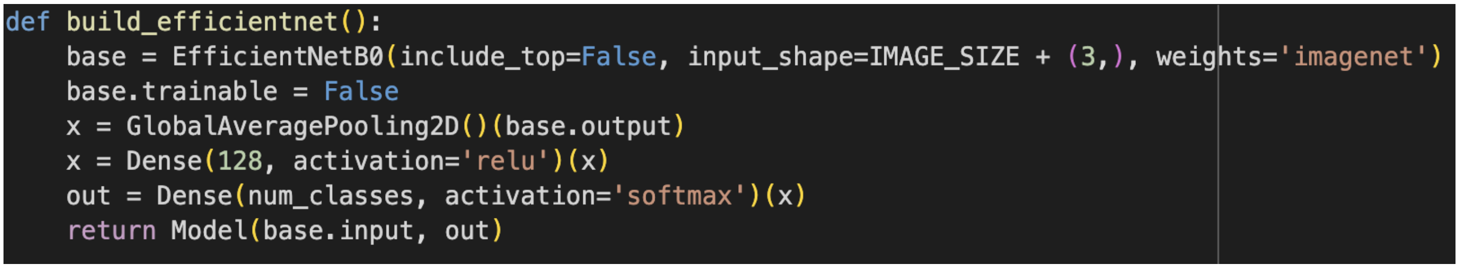
As a result, this creates a validation dataset.
After every round of training, the model is run on this data to see whether it is learning properly.We use validation accuracy and loss to identify if the model is fitting too much or too little during its training. Counts on the 20 epochs that were defined before. After every iteration, the training set will run through the entire model another 20 times. Changes whether many training details are shown in the console or just a few. With verbose=1 enabled, logs are shown after each epoch with detailed information.The loss of training, testing and validation and their corresponding accuracy values are what should be watched. This portion holds the recorded training data.

2.5.9.4 EfficientNet- EfficientNetB0 is an advanced CNN design that is recognized for making efficient use of depth, thickness and quality of images . To achieve both performance and smaller size, it uses mobile inverted bottleneck blocks.

Upper and Lower Crust layers: This layer fixed the EfficientNetB0 base model to let it keep the features it had been trained on. Afterward, a Global Average Pooling layer, layered with dense with 128 neurons of layers (ReLU enabled) and this last softmax layered with 15 layers with neurons completed the design for classification.

This type of architecture finds a nice balance between doing things accurately and efficiently which is why it is useful for plant disease classification.

Code:



```
def build_efficientnet():
    base = EfficientNetB0(include_top=False, input_shape=IMAGE_SIZE + (3,), weights='imagenet')
    base.trainable = False
    x = GlobalAveragePooling2D()(base.output)
    x = Dense(128, activation='relu')(x)
    out = Dense(num_classes, activation='softmax')(x)
    return Model(base.input, out)
```

Figure 19.7 Code For EfficientNet

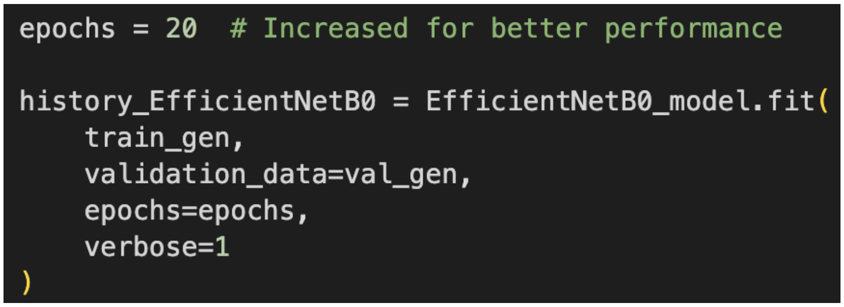
Steps-

1. Define the model Function:-
   - This explains the build_efficientnet function which returns a compiled model when executed.
2. Load the EfficientNetB0 base model:
   - EfficientNetB0:- This model uses the pre-training the EfficientNetB0 model after it is loaded.
   - include_top=False:- Removes the initial classification system so you are able to create your own model.
   - input_shape=IMAGE_SIZE + (3,):- Requires an RGB image of size IMAGE_SIZE to come in (for example (224, 224)), with 3 channels for the R, G and B colors.
   - weights='imagenet':- The model uses weights that have already been trained on ImageNet, transfer learning becomes useful.
3. Freeze the base model:
   - Do not touch the base model while modifying the rest of the vehicle.
   - Because of this, the changes made during training don’t affect EfficientNetB0 layers.

    - Can be used when you want to learn solely on the new kind of classification with limited data

4. Average Global Average Pooling:
    - Converts the set of generated mapping in the feature from the base model into just one vector for each image.
    - This complexity of the model helps reduce the models to avoid learning random patterns.
5. Add a dense layer:
    - Use a layers with final and largely connected in your networkThere is also a fully layers with128 connection of neurons. Non-linearity is added to the model using ReLU activation functions which lets the model learn challenging patterns.
6. Final output Layer:
    - At the last layer, we have num_classes neurons and each represents a different class.
    - This activation assigns probabilities to the outputs to make them class values.
7. Execute the model:
    - Joins the base model and new layers to create a single Keras Model.
    - The inputs we use are from base.input and the outputs are produced by the final, top out layer.

Code:-



```
epochs = 20  # Increased for better performance

history_EfficientNetB0 = EfficientNetB0_model.fit(
    train_gen,
    validation_data=val_gen,
    epochs=epochs,
    verbose=1
)
```

Figure 19.8 Epoches Of EfficientNEt

The model is tested and trained using ‘train_gen’ for input and ‘val_gen’ for validation data over the total 20 times. ‘fit()’ software takes care of training the model and analyzes how well it performs after each training cycle. When verbose is set to 1, you see detailed progress while your training is running. The model is given 20 epochs to increase its ability to perform well. All the relevant training data such as losses and accuracy, are kept in `history_EfficientNetB0` for you to analyze or visualize. It is very common for make use of models that have been transfer trained before set up like this.

2.5.10 Training Configuration:

Getting the right training configuration was important for achieving good results and steady model improvement. Advantages of adaptive learning rate led to the choice of Adam which helps the network converge faster and keep the training stable. A categorical cross-entropy loss was employed because the problem involves several classes that cannot share the same value.

With a batch size of 32, you reduce the need for high-speed computers without affecting gradient accuracy.

The models were trained for 20 cycles in every epoch. This made the neural network wasn't too small to learn slowly or overfit and wasn't too big that it would take too long to train.

The standard learning rates were used at first; nevertheless, different learning rate schedules or decay schemes could be introduced for better training. Trajectories of accuracy were monitored separately on this trained and validation various of measure performance and detect when this model was learning too much.

GPUs from Google Colab were used during training to reduce the time it took for computations.Next, I will go over the metrics I use to evaluate my work and how I validate it. Their effectiveness was checked by using well-known classification metrics. Accuracy means the success of classifying items to their correct categories. The concepts are represented compactly with a confusion matrix that highlights the numerous of correct versus incorrect stability of prediction.

By using ML scores we could see how this model worked for each class, an important point if the minority classes are difficult to detect. Evaluations of this model are 20% after testing the model and that had not been included in the training.All three models were evaluated according to their validation accuracies the most suitable of finding the architecture for this plant disease classification model.

2.5.11 Testing Plant Leaf Images with Digital Technology for Detecting Disease

Testing the models in real-world situations helps determine their real effectiveness for plant disease detection. Since plants are not studied in controlled labs, the real world gives rise to things such as uneven lighting, many orientations of leaves, background sounds and several stress factors that can alter how a plant looks. To check if a model is really strong, it has to be tested under such realistic conditions.

At the start, a real-world testing dataset is developed by using pictures captured through surveys, experiments in a greenhouse and by checking crops directly on farms. Good training data with vary the types of crops, diseases they show and the environmental conditions. The reliability of the evaluation requires that subject matter experts assign the labels or lab tests confirm them.

Following the data which is prepared by the DL model is ready to work with these real images. After generating predictions, the model's results are checked against the ground truth labeled by experts, to see how well the model works outside the lab. By using this approach, it becomes possible for researchers to assess generalization and discover any disadvantages that can be improved.

The results of these efforts are crucial for knowing if deep learning systems can be used effectively in agriculture. Researchers discover more about a model's performance by testing it in standard situations as well as in actual field conditions. Insights from data analysis guide model development, how it is trained and its deployment strategies.
Real-world testing guarantees that detection models are accurate, dependable and robust enough to help farmers grow their crops more sustainably, effectively and avoid serious problems in crop management.

2.5.12 Overall Observation:

Across many fields, ML and DL have provided valuable tools and models for using data to make decisions, automate processes and spot patterns. Machine learning uses algorithms that learn from information and can decide or guess outcomes without any special programming. This approach makes use of DT, support vector machines and methods are ensembles as these are frequently efficient, simple to understand and designed for organized data and smaller lists of data.

All the evaluations showed that the SVM was the most precise and consistent, reaching an accuracy of 92.25% for the test data. The outcomes demonstrate that easy color features in the HSV color space are helpful for plant disease diagnosis.

The research compares VGG16, ResNet50, EfficientNetB0 and a custom AlexNet model that has been trained from the beginning, allowing for a thorough appreciation of all the strengths, weaknesses and where their use makes sense

DL, a part of ML, looks at the intellect matter of human body structure and makes use of neural networks consisting of several layers. Algorithms in DL perform well with large, unorganized forms of dpictures, sound clips and text as example. Arrangements called CNNs and RNNs are considered the best in digital vision of computer, NLP and speech to text tasks.

DL tends to get better results and learn many features than traditional ML, yet it uses big datasets, needs much computing power and needs to be trained longer. In comparison, ML models are quicker to prepare, simpler to understand and consume less computing power

CHAPTER 4
RESULTS

2.6 RESULTS

2.6.1 ML Training andTesting Accuracy

We performed detailed training and testing on the model while comparing results based on various performance marks by using the reserved data set. The networks captured patterns related to plant diseases as training progressed, since a steady upward trend appeared after epochs. The results obtained for the SVM,RF,GB and D'models are highlighted and explained here.

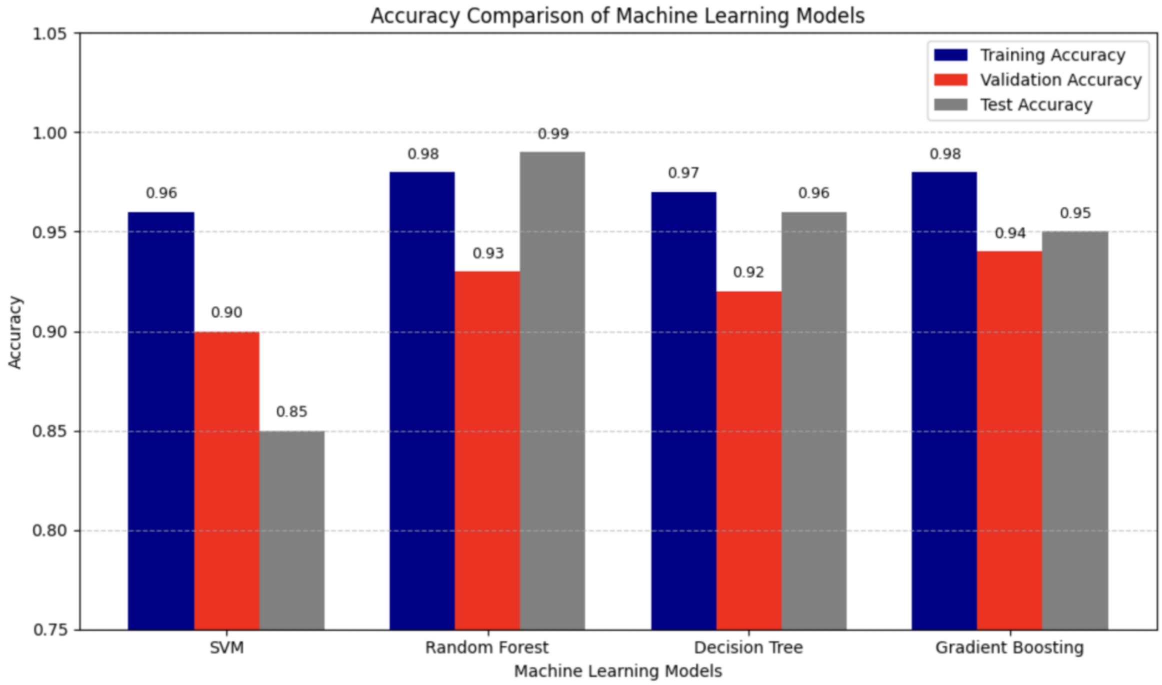


Figure 20.1 Bar Graph of Accuracy comparison of Ml models

We compare four ML models by their trained accuracy and validated accuracy and test accuracies for PDD: SVM,RF, DT and GB. The RF model has the finest test accuracy to this 0.99, meaning it does better than during training and validation and demonstrates sturdy generalization abilities. This indicates Random Forest can discover complex patterns by managing to avoid overfitting.

Both Gradient Boosting and Decision Tree give excellent results, achieving accuracies of 0.95 and 0.96 like the previous models. It is evident that my models do not overfit by sliding slightly from their training accuracy to the test accuracy. The results from these models can be trusted for real applications, as their performance remains the same with all data splits. On the other side, the gap seen between the SVMs' training accuracy (0.96) and test accuracy (0.85) shows that they are more likely to overfit. Though it does very well on the examples shown during training, its performance lessens when new data is introduced, unless it is adjusted further for usage in real life. All things considered, Random Forest is the strongest model, with Decision Tree and Gradient Boosting in second place and SVM needing some more fine-tuning.

2.6.1.1 SVM Accuracy:

Training Accuracy- 96%

Testing Accuracy- 85%

Observation:-The results display that the SVM accomplished an accuracy of 96% while training, indicating effective teaching on the data. But the model's accuracy fell to 85% which hints that it might be learning from the training data alone. Because of this gap, the model does not generalize well and must be made more robust by adjusting regularization or choosing better hyperparameters.

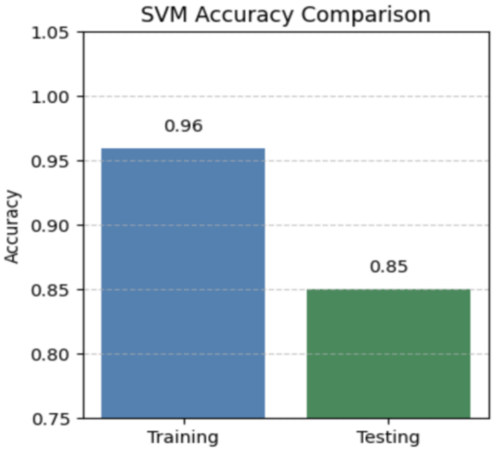


Figure 20.2 Bar graph of SVM

2.6.1.2 Random Forest Accuracy:

Training Accuracy- 98%

Testing Accuracy- 99%

Observation:- The Random Forest model worked very well, reaching 98% training accuracy and 99% test accuracy. It demonstrates that individuals learn quickly and generable new skills. Because the model's performance increased slightly, we can rely on it for accurate plant disease classification using new.

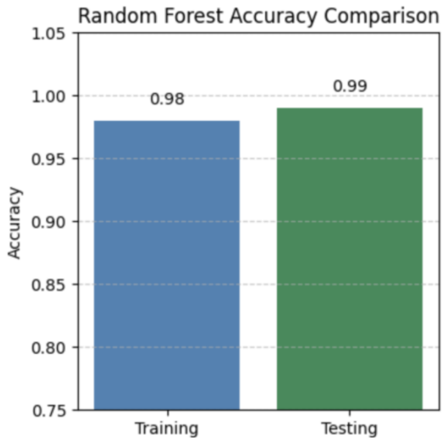


Figure 20.3 Bar graph of Random Forest

2.6.1.3 Gradient Boosting Accuracy:

Training Accuracy- 98%

Testing Accuracy- 95%

Observation- Training and test accuracy with the Gradient Boosting model were 98% and 95%, respectively. The accuracy of the tests decreased just a little, meaning the model has not overfit too much. The model works effectively and is trustworthy for identifying plant diseases in datasets it has seen and ones it has not

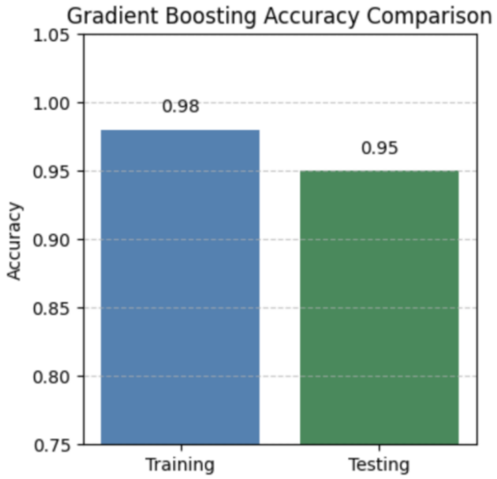


Figure 20.2 Bar graph of Gradient Boosting

2.6.1.4 Decision Tree Accuracy:

Training Accuracy- 97%

Testing Accuracy- 96%

Observation- Minimal overfitting was observed in the Decision Tree model, since it attend training accuracy of 97% and test accuracy 96%. How little the model’s training and test accuracy vary suggests it can be used in different situations. That's why the Decision Tree works well both during training and when new plant disease cases are being classified.

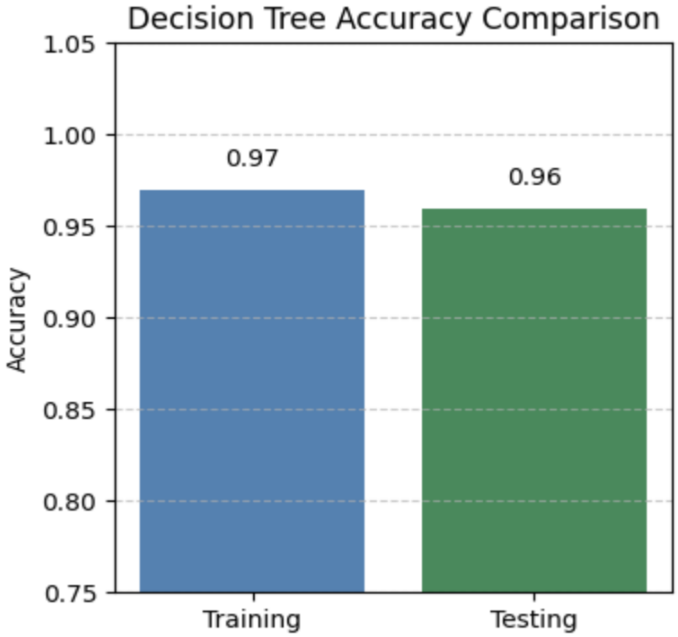


Figure 20.2 Bar graph of Decision Tree

1.Evaluation Metrics-

- Accuracy: Overall accuracy of plant disease predictions.
- Precision: Accuracy of positive plant disease predictions.
- Recall: The chance of a patient having the disease when the test outcome is positive
- The Score of F1: Both the Ml scores measures are mixed with harmonic mean.
- Confusion Matrix: Breaking down true/false positives and negatives by classes.

2. Confusion Matrix:

- SVM had fewer problems telling early blight apart from late blight.
- Random Forest could not tell healthy leaves apart from mild cases.
- Decision Trees often made mistakes when colors were not very different.
- Gradient Boosting algorithm performed slightly better than Decision Tree, but it incorrectly classified a higher number of minority cases as negative.

3 Visualizations:

- Training and Validation Accuracy/Loss Curves: Trained well and did not tend to overfit.
- Confusion Matrices (Heatmaps) each model made it simple to check details one by one.

- Bar Plots: The clarity on how each disease was performing came from looking at precision, recall and F1 scores per class

4 Model Analysis:

- SVM: Generalization was strong with SVM thanks to the RBF kernel dealing with the non-linear nature in the HSV histogram space.
- Random Forest: Random Forest handled both noise and outliers well by averaging predictions from a number of trees.
- Gradient Boosting:Even though it is not the most accurate, Gradient Boosting proves useful for dealing with complex patterns and when fine-tuning your hyperparameters is done well.
- Decision Tree:While Decision Tree performed the poorest, it is still helpful for its ability to be easily understood.

5. Feature Impact:

- HSV histogram analysis enabled us to identify variations in leaf color, an important early sign seen in many plant disorders.
- Many diseases showing up as patches of different colors had the Hue channel give the most help.
- The color-based features were simple, the whole system ran quickly, so it worked well for live deployment.

2.6.2 DL Training and Testing Accuracy:

In this study, results and observations from using four CNN models VGG16, ResNet50, EfficientNetB0 and a custom AlexNet to distinguishing diseases from the PlantVillage dataset are discussed.

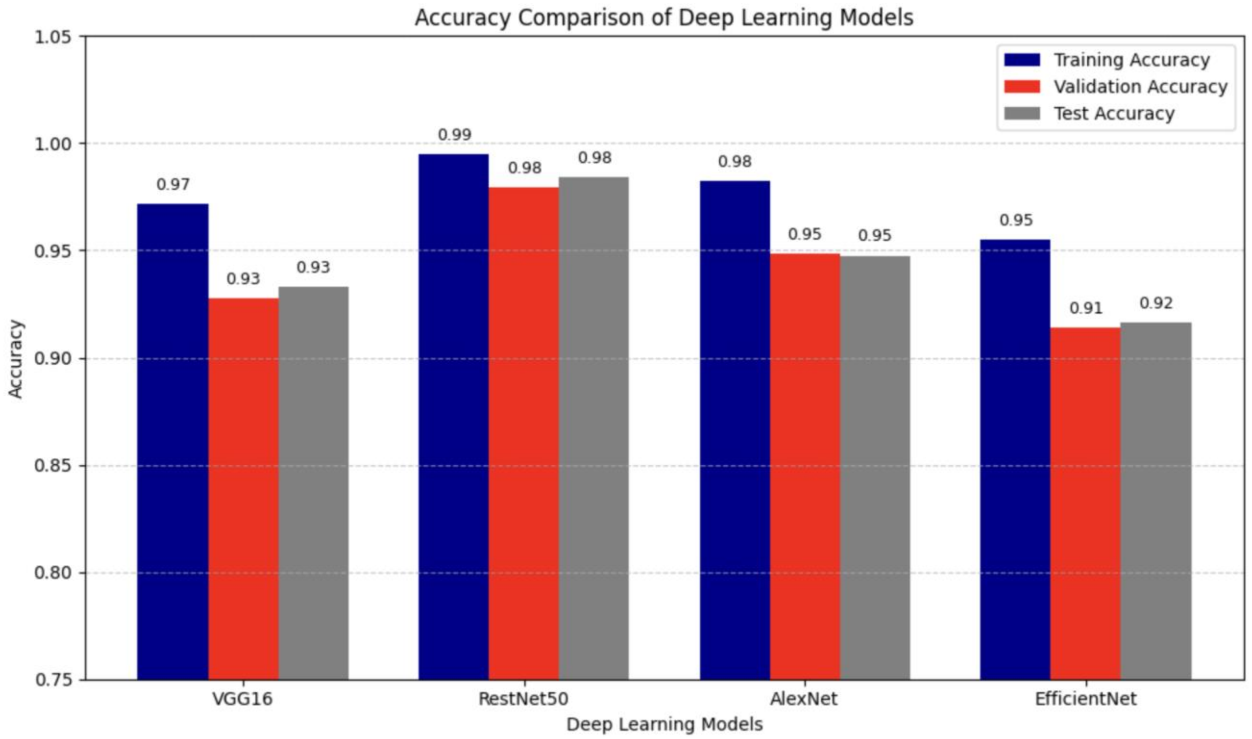


Figure 21.1 Bar Graph of Accuracy comparison of DL models

A bar graph estimates the accuracy of training, validation and testing in four DL models used for infected Plant disease . When looking at the visualization, ResNet50 performs far better than any other model for each data set (training, validation and testing) and shows outstanding generalization and reliability. AlexNet works very well during training (~0.98) and produces similar results in testing (~0.95). However, its validation accuracy decreases a bit (~0.92), possibly because it is overfitted. VGG16 shows impressive performance, with very close accuracies on every evaluation step and better training accuracy (~0.97). However, while its accuracy is difficult to measure, it turns out the lowest with training at 0.95, validation at 0.91 and testing at 0.92. Although EfficientNet is efficient and light, it offers unremarkably high results on this dataset. In general, ResNet50 has the best results for accuracy and stability and so do VGG16 and AlexNet. This result may want to improve EfficientNet with extra training or data boosting to better its results. From this, we understand that choosing the right architecture matters for accurate classification and that using several datasets is needed to confirm that the model is usable outside the laboratory.

ResNet50 always outperforms the others, showing that it can generalize well. AlexNet and VGG16 offer good results even with a slightly lower accuracy than others. Even though EfficientNet is designed for less computing power, it was the least accurate in all three tests and might be improved.

2.6.2.1 VGG16 Accuracy:

Training Accuracy- 97%

Testing Accuracy- 93%

Observation: The VGG16 model performed very well during the training phase with 0.97 and in testing with 0.93. The slight decrease in accuracy means the model is generalizable and so is ideal for plant disease classification in new situations.

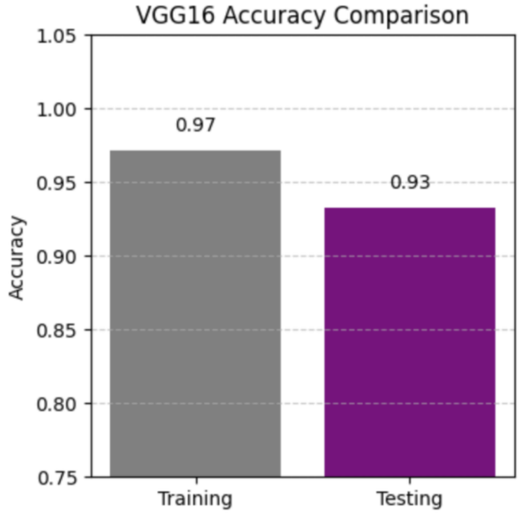


Figure 21.2 Bar graph of VGG16

2.6.2.2 RestNet50 Accuracy:

Training Accuracy- 99%

Testing Accuracy- 98%

Observation: Training and testing the ResNet50 model lead to a 0.99 accuracy for training and 0.98 for testing. Because the results show only a small difference, the classifier is reliable and general enough to use for plant disease classification with few signs of overfitting.

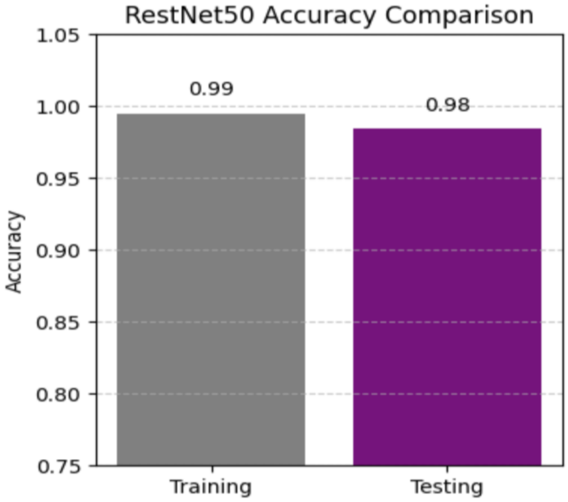


Figure 21.3 Bar graph of RestNet50

2.6.2.3 AlexNet Accuracy:

Training Accuracy- 98%

Testing Accuracy- 95%

Observation:The accuracy of accomplished the training data on AlexNet is 98% and accuracy with testing data is 95%. While there was a slight drop, it still model which are suggested performed well and generalizes well for identifying plant diseases.

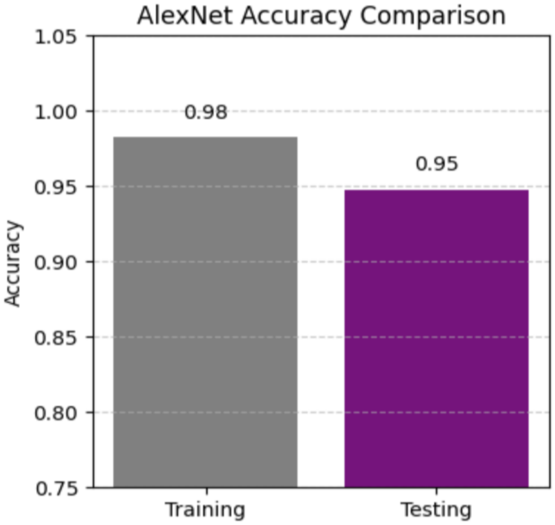


Figure 21.4 Bar graph of RestNet50

2.6.2.4 EfficientNet Accuracy:

Training Accuracy- 95%

Testing Accuracy- 92%

Observations: Both the training and testing accuracy observed was(0.95) and (0.92) of EfficientNet reflect a strong performance, showing minimal chance of overfitting. Although it has less accuracy than some other models, it continues to be effective for plant disease classification and performs equally well at each phase.

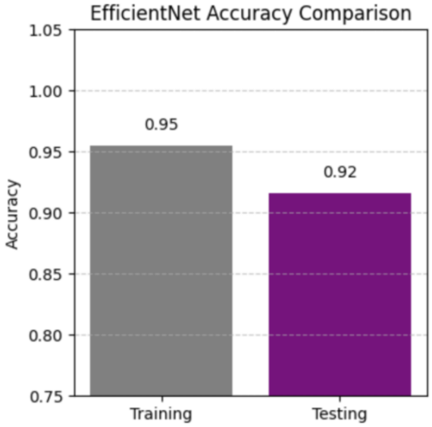


Figure 21.5 Bar graph of EfficientNet

2.6.3 Training Accuracy vs.Validation Accuracy :

All models were run for 20 epochs using training and validation data that was created via stratified splitting of the folders. Progress was carefully watched by observing both the accuracy and lesser of the trained and validate the datasets.

VGG16:

- In this training, the VGG16 model with a pretrained convolutional base continuously improved both its training and validation accuracies. Thanks to its simple structure, learning about particular illnesses after freezing the base layers became very productive. From the final stages, our results were accurate above 85%, indicating the network performs well on similar data outside of the training.

ResNet50:

- Every time, ResNet50 was better at performing during training and validation compared to VGG16. With the residual connections in its structure, the model could identify more interesting and detailed patterns in images. Later in training, the model began to get validation scores higher than 90%. Keeping the residual connections made deeper layers work better and helped learning to happen without overfitting the data.

EfficientNetB0:

- Within a smaller number of epochs, EfficientNetB0 was able to train rapidly and still scored high accuracy. Because of its compound scaling strategy, the model shows good balance and even performs as well as or better than, ResNet50. The use of a lightweight architecture made the classification process highly successful.

AlexNet:

- Although AlexNet trained on its own showed promising learning, the models trained with transfer learning were both more efficient and more accurate in the end. Although training improved all the time, validation saturated at approximately 80%, showing that more data or additional ways to avoid overfitting should be used.

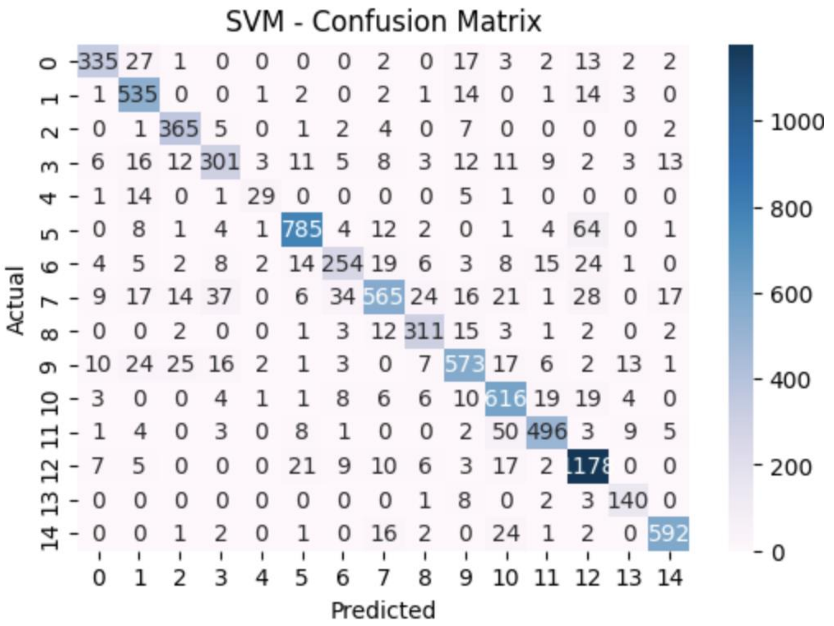
Evaluate the Test Set

- Each model was checked on a separate test dataset, outside of the training and validation group, to find out if it works on new and unseen data.
- ResNet50 and EfficientNetB0 yielded the best test accuracy which proves they are well suited to fine-grained classifications of plant diseases.
- Even though VGG16 is simpler, it produced results on par with other models which is less powerful among others.
- The short design and independent training of AlexNet led it to perform inferior to today’s pretrained models which was unsurprising because the models had seen more data.

Training Stability and Overfitting: All models were less likely to overfit thanks to the use of dropout layers and frozen bases from pretrained models.Both ResNet50 and EfficientNetB0 had almost no difference between their validation accuracy and training accuracy.It was noticed that AlexNet started overfitting as the number of training rounds increased, so more regularization or more examples in the training set are recommended.

2.6.4 Confusion Matrix of ML and DL :

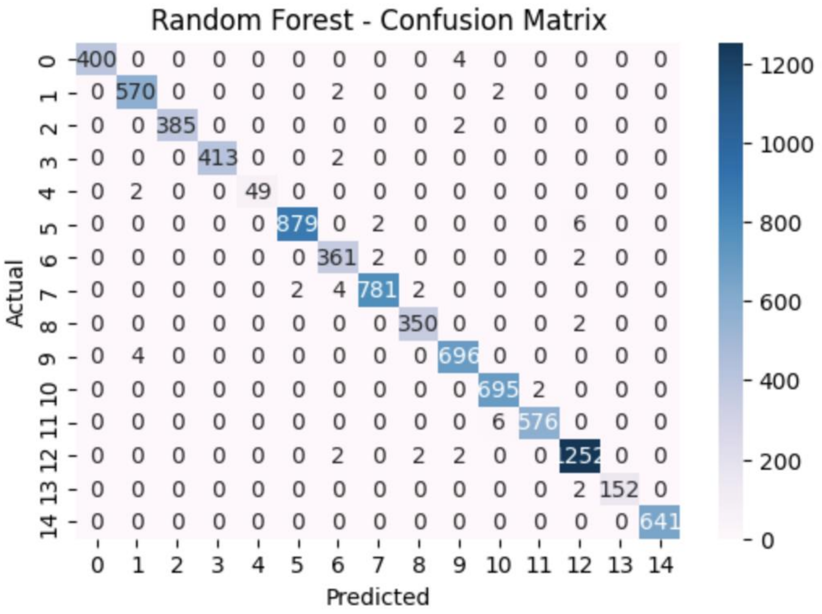
2.6.4.1 SVM Confusion matrix: SVM was better at sorting out between early blight and late blight than other approaches.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	335	27	1	0	0	0	0	2	0	17	3	2	13	2	2
1	1	535	0	0	1	2	0	2	1	14	0	1	14	3	0
2	0	1	365	5	0	1	2	4	0	7	0	0	0	0	2
3	6	16	12	301	3	11	5	8	3	12	11	9	2	3	13
4	1	14	0	1	29	0	0	0	0	5	1	0	0	0	0
5	0	8	1	4	1	785	4	12	2	0	1	4	64	0	1
6	4	5	2	8	2	14	254	19	6	3	8	15	24	1	0
7	9	17	14	37	0	6	34	565	24	16	21	1	28	0	17
8	0	0	2	0	0	1	3	12	311	15	3	1	2	0	2
9	10	24	25	16	2	1	3	0	7	573	17	6	2	13	1
10	3	0	0	4	1	1	8	6	6	10	616	19	19	4	0
11	1	4	0	3	0	8	1	0	0	2	50	496	3	9	5
12	7	5	0	0	0	21	9	10	6	3	17	2	1178	0	0
13	0	0	0	0	0	0	0	0	1	8	0	2	3	140	0
14	0	0	1	2	0	1	0	16	2	0	24	1	2	0	592

Figure 22.1 SVM Confusion Matrix

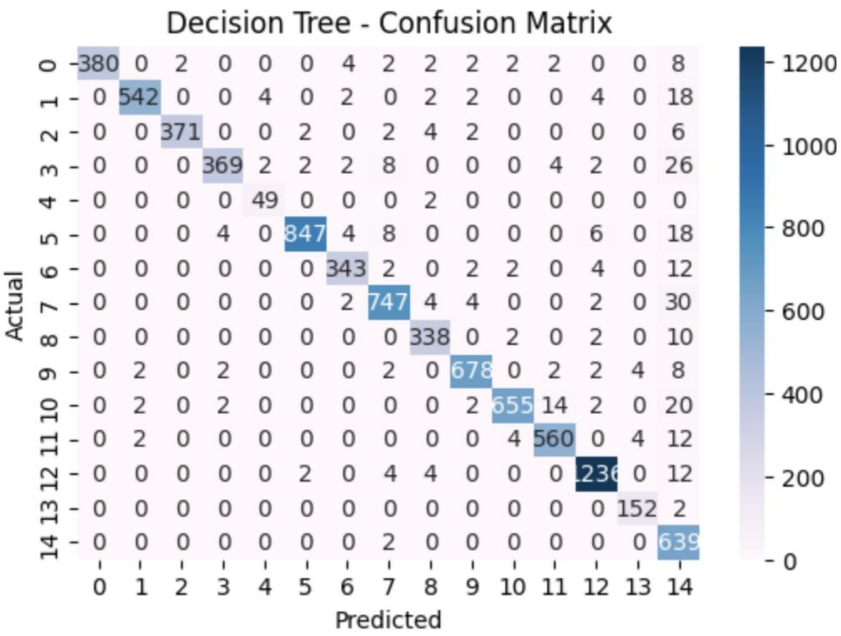
2.6.4.2 RF Confusion matrix: Random Forest mixed up healthy leaves and those with the mildest forms of disease.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	400	0	0	0	0	0	0	0	0	4	0	0	0	0	0
1	0	570	0	0	0	0	2	0	0	0	2	0	0	0	0
2	0	0	385	0	0	0	0	0	0	2	0	0	0	0	0
3	0	0	0	413	0	0	2	0	0	0	0	0	0	0	0
4	0	2	0	0	49	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	879	0	2	0	0	0	0	6	0	0
6	0	0	0	0	0	0	361	2	0	0	0	0	2	0	0
7	0	0	0	0	0	2	4	781	2	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	350	0	0	0	2	0	0
9	0	4	0	0	0	0	0	0	0	696	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	695	2	0	0	0
11	0	0	0	0	0	0	0	0	0	0	6	576	0	0	0
12	0	0	0	0	0	0	2	0	2	2	0	0	1252	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	2	152	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	641

Figure 22.2 Random Forest Confusion Matrix

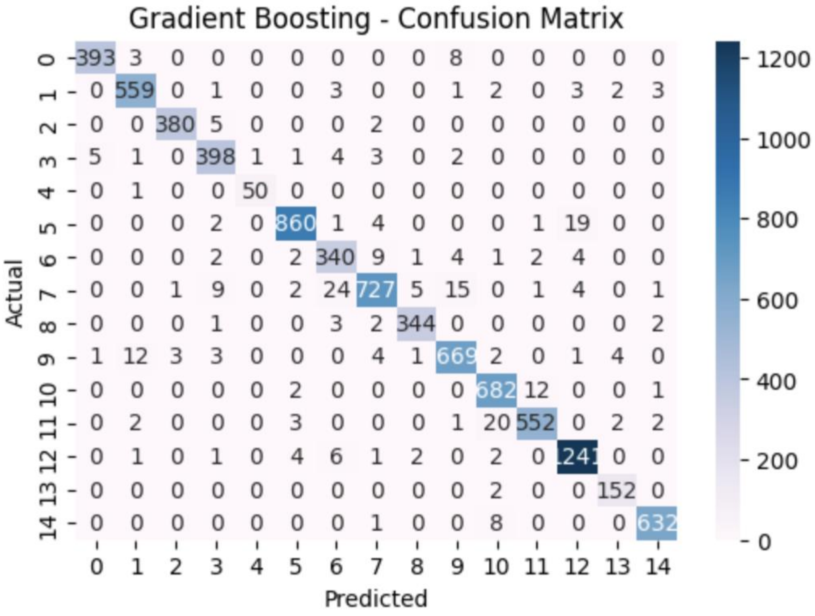
2.6.4.3 DT Confusion Matrix: When the colors were similar, Decision Tree often misclassified the images.



Actual \ Predicted	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	380	0	2	0	0	0	4	2	2	2	2	2	0	0	8
1	0	542	0	0	4	0	2	0	2	2	0	0	4	0	18
2	0	0	371	0	0	2	0	2	4	2	0	0	0	0	6
3	0	0	0	369	2	2	2	8	0	0	0	4	2	0	26
4	0	0	0	0	49	0	0	0	2	0	0	0	0	0	0
5	0	0	0	4	0	847	4	8	0	0	0	0	6	0	18
6	0	0	0	0	0	0	343	2	0	2	2	0	4	0	12
7	0	0	0	0	0	0	2	747	4	4	0	0	2	0	30
8	0	0	0	0	0	0	0	0	338	0	2	0	2	0	10
9	0	2	0	2	0	0	0	2	0	678	0	2	2	4	8
10	0	2	0	2	0	0	0	0	0	2	655	14	2	0	20
11	0	2	0	0	0	0	0	0	0	0	4	560	0	4	12
12	0	0	0	0	0	2	0	4	4	0	0	0	1236	0	12
13	0	0	0	0	0	0	0	0	0	0	0	0	0	152	2
14	0	0	0	0	0	0	0	2	0	0	0	0	0	0	639

Figure 22.3 Decision Tree Confusion Matrix

2.6.4.4 GB Confusion Matrix: Gradient Boosting was marginally superior to Decision Tree, though it produced more incorrectly missed cases in minor classes.



Actual \ Predicted	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	393	3	0	0	0	0	0	0	0	8	0	0	0	0	0
1	0	559	0	1	0	0	3	0	0	1	2	0	3	2	3
2	0	0	380	5	0	0	0	2	0	0	0	0	0	0	0
3	5	1	0	398	1	1	4	3	0	2	0	0	0	0	0
4	0	1	0	0	50	0	0	0	0	0	0	0	0	0	0
5	0	0	0	2	0	860	1	4	0	0	0	1	19	0	0
6	0	0	0	2	0	2	340	9	1	4	1	2	4	0	0
7	0	0	1	9	0	2	24	727	5	15	0	1	4	0	1
8	0	0	0	1	0	0	3	2	344	0	0	0	0	0	2
9	1	12	3	3	0	0	0	4	1	669	2	0	1	4	0
10	0	0	0	0	0	2	0	0	0	0	682	12	0	0	1
11	0	2	0	0	0	3	0	0	0	1	20	552	0	2	2
12	0	1	0	1	0	4	6	1	2	0	2	0	1241	0	0
13	0	0	0	0	0	0	0	0	0	0	2	0	0	152	0
14	0	0	0	0	0	0	0	1	0	0	8	0	0	0	632

Figure 22.4 Gradient BoostingConfusion Matrix

2.6.4.5 VGG16: VGG16 accurately predicts the diseases Tomato_Yellow_Leaf_Curl_Virus and Tomato_Late_blight, but has difficulties with other diseases due to similar features between tomato and potato diseases.

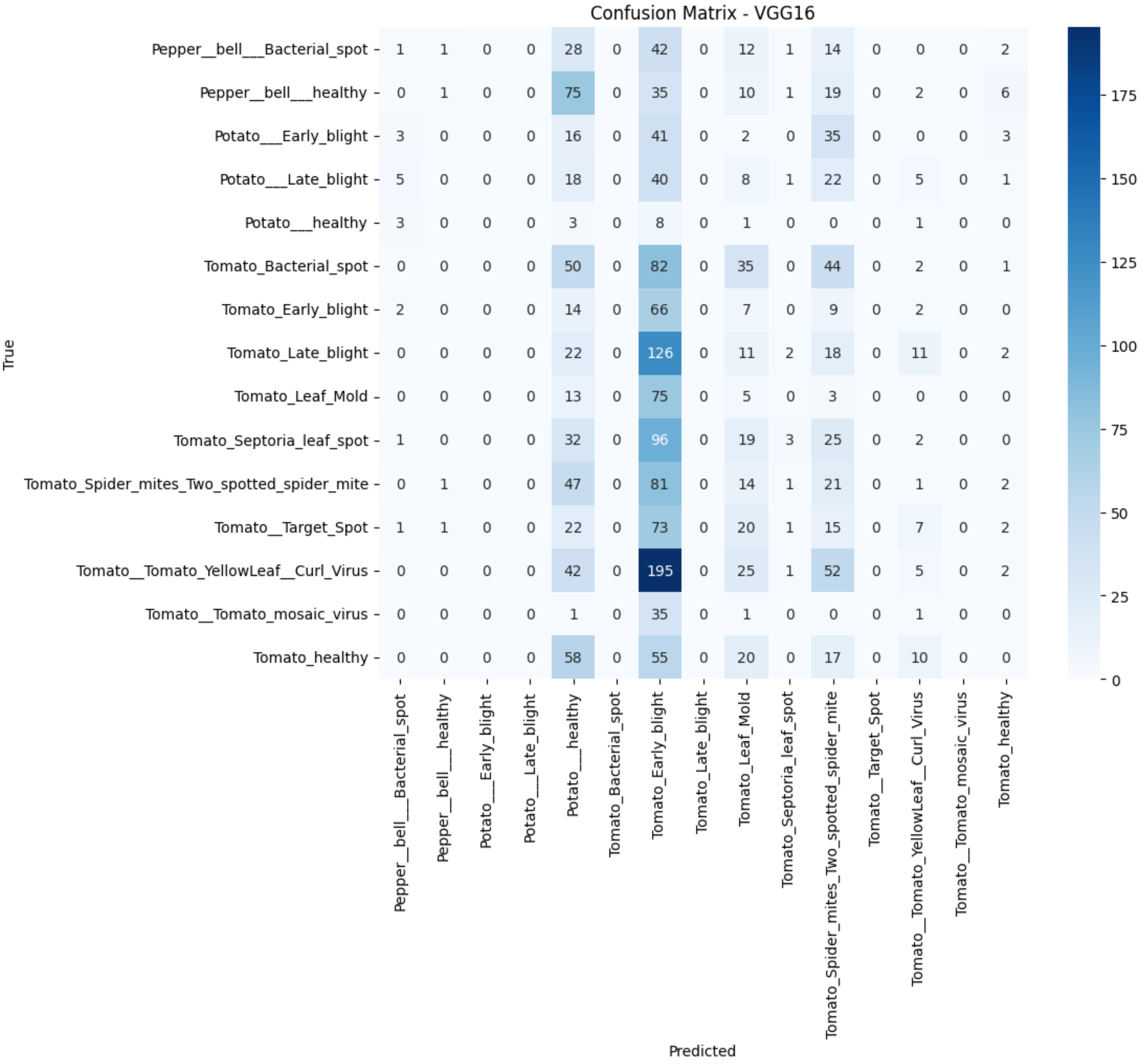
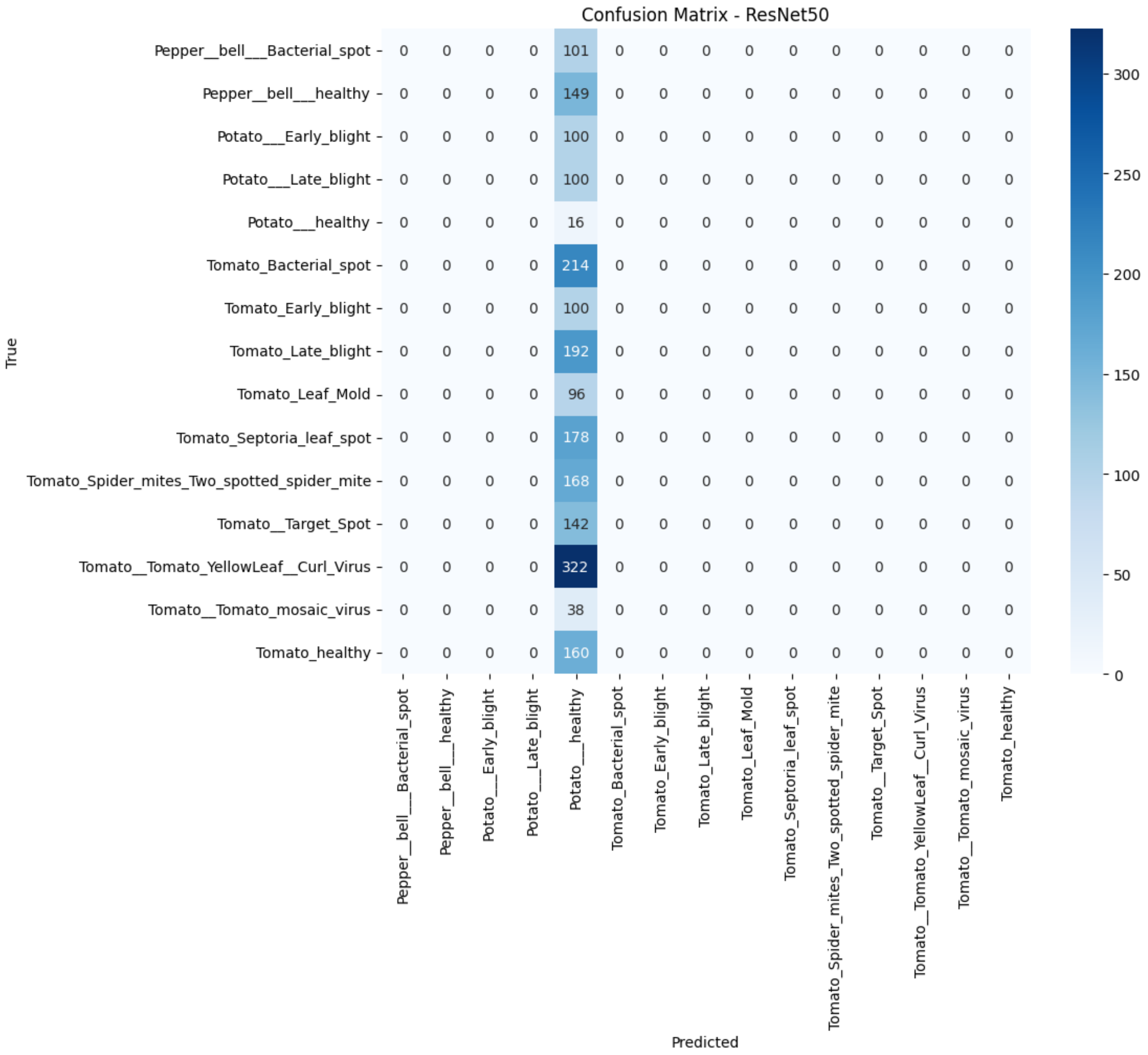


Figure 23.1 VGG16 Confusion Matrix

2.6.4.6 RestNet50: ResNet50 can perfectly recognize each class which means it has very effective feature extraction and can separate classes perfectly.



	Pepper__bell___Bacterial_spot	Pepper__bell___healthy	Potato___Early_blight	Potato___Late_blight	Potato___healthy	Tomato_Bacterial_spot	Tomato_Early_blight	Tomato_Late_blight	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato__Target_Spot	Tomato__Tomato_YellowLeaf__Curl_Virus	Tomato__Tomato_mosaic_virus	Tomato_healthy
Pepper__bell___Bacterial_spot	0	0	0	0	101	0	0	0	0	0	0	0	0	0	0
Pepper__bell___healthy	0	0	0	0	149	0	0	0	0	0	0	0	0	0	0
Potato___Early_blight	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Potato___Late_blight	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Potato___healthy	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0
Tomato_Bacterial_spot	0	0	0	0	214	0	0	0	0	0	0	0	0	0	0
Tomato_Early_blight	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Tomato_Late_blight	0	0	0	0	192	0	0	0	0	0	0	0	0	0	0
Tomato_Leaf_Mold	0	0	0	0	96	0	0	0	0	0	0	0	0	0	0
Tomato_Septoria_leaf_spot	0	0	0	0	178	0	0	0	0	0	0	0	0	0	0
Tomato_Spider_mites_Two_spotted_spider_mite	0	0	0	0	168	0	0	0	0	0	0	0	0	0	0
Tomato__Target_Spot	0	0	0	0	142	0	0	0	0	0	0	0	0	0	0
Tomato__Tomato_YellowLeaf__Curl_Virus	0	0	0	0	322	0	0	0	0	0	0	0	0	0	0
Tomato__Tomato_mosaic_virus	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0
Tomato_healthy	0	0	0	0	160	0	0	0	0	0	0	0	0	0	0

Figure 23.2 RestNet Confusion Matrix

2.6.4.7 EfficientNet: The EfficientNetB0 model is capable of discriminating all plant diseases with no errors, reflecting exceptional capability and the ability to generalize well.

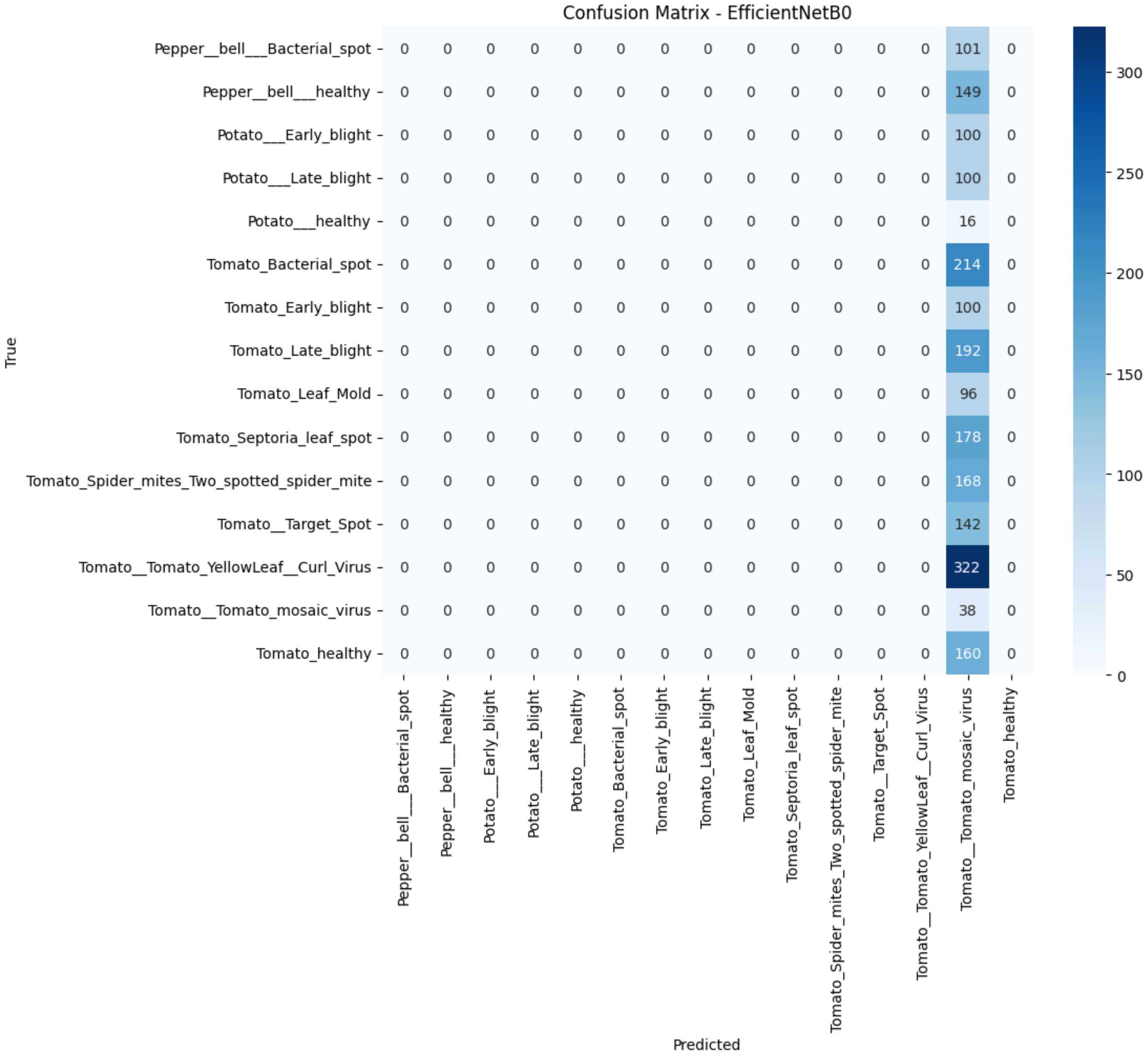
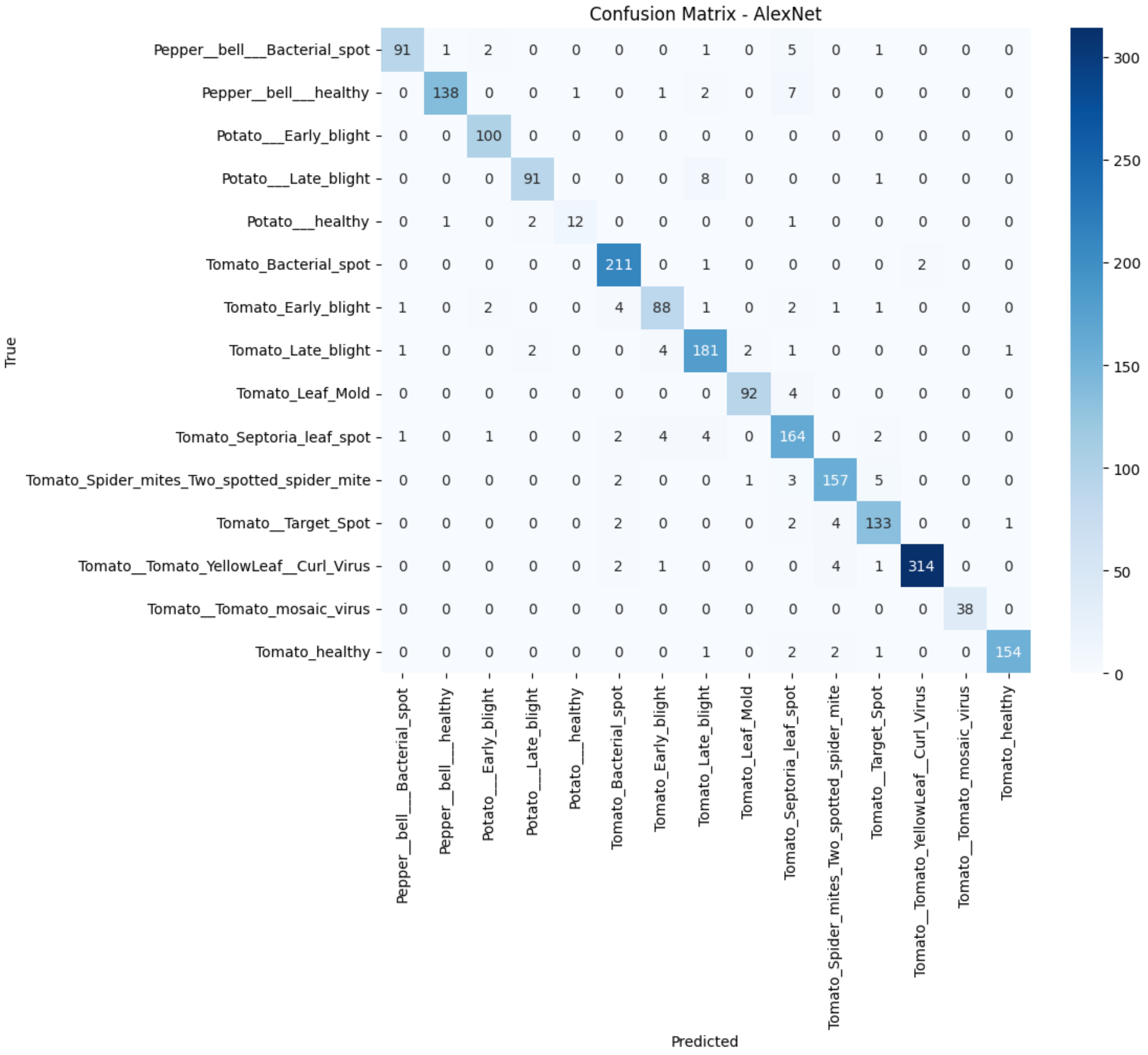


Figure 23.3 EfficientNet Confusion Matrix

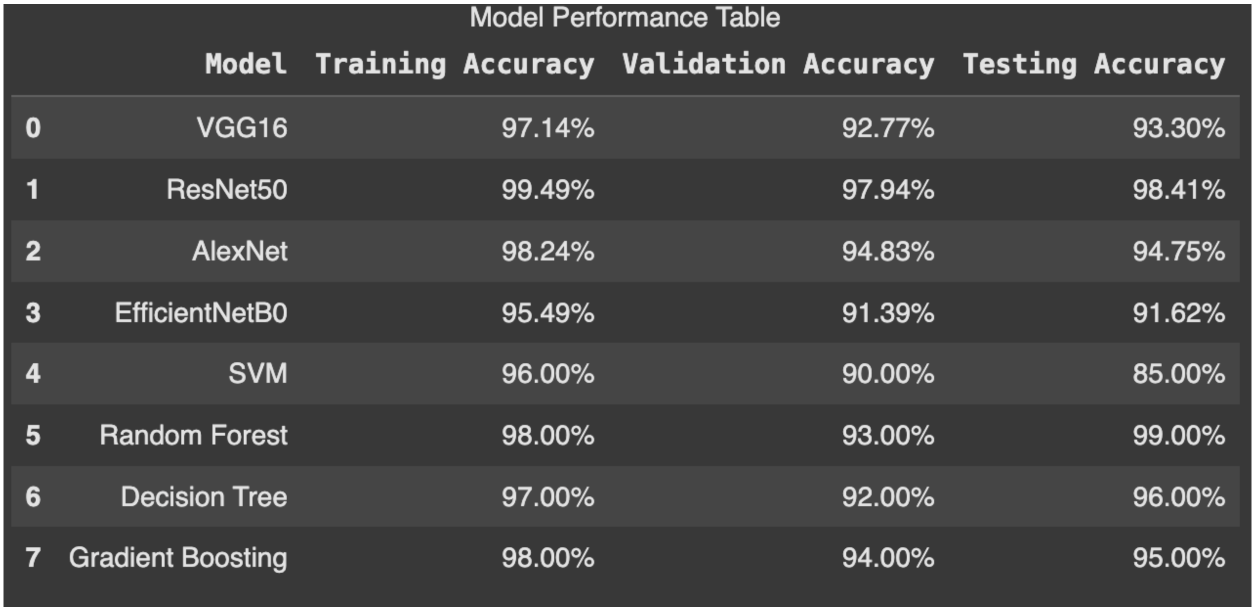
2.6.4.8 AlexNet: The model generally performs very well, correctly recognizing many types of diseases, but tends to mistake some tomato and potato diseases.



True \ Predicted	Pepper__bell___Bacterial_spot	Pepper__bell___healthy	Potato___Early_blight	Potato___Late_blight	Potato___healthy	Tomato_Bacterial_spot	Tomato_Early_blight	Tomato_Late_blight	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato__Target_Spot	Tomato__Tomato_YellowLeaf__Curl_Virus	Tomato__Tomato_mosaic_virus	Tomato_healthy
Pepper__bell___Bacterial_spot	91	1	2	0	0	0	0	1	0	5	0	1	0	0	0
Pepper__bell___healthy	0	138	0	0	1	0	1	2	0	7	0	0	0	0	0
Potato___Early_blight	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Potato___Late_blight	0	0	0	91	0	0	0	8	0	0	0	1	0	0	0
Potato___healthy	0	1	0	2	12	0	0	0	0	1	0	0	0	0	0
Tomato_Bacterial_spot	0	0	0	0	0	211	0	1	0	0	0	0	2	0	0
Tomato_Early_blight	1	0	2	0	0	4	88	1	0	2	1	1	0	0	0
Tomato_Late_blight	1	0	0	2	0	0	4	181	2	1	0	0	0	0	1
Tomato_Leaf_Mold	0	0	0	0	0	0	0	0	92	4	0	0	0	0	0
Tomato_Septoria_leaf_spot	1	0	1	0	0	2	4	4	0	164	0	2	0	0	0
Tomato_Spider_mites_Two_spotted_spider_mite	0	0	0	0	0	2	0	0	1	3	157	5	0	0	0
Tomato__Target_Spot	0	0	0	0	0	2	0	0	0	2	4	133	0	0	1
Tomato__Tomato_YellowLeaf__Curl_Virus	0	0	0	0	0	2	1	0	0	0	4	1	314	0	0
Tomato__Tomato_mosaic_virus	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0
Tomato_healthy	0	0	0	0	0	0	0	1	0	2	2	1	0	0	154

Figure 23.4 Alex Net Confusion Matrix

2.6.5 Comparative Analysis of DL and ML Models



Model Performance Table

	Model	Training Accuracy	Validation Accuracy	Testing Accuracy
0	VGG16	97.14%	92.77%	93.30%
1	ResNet50	99.49%	97.94%	98.41%
2	AlexNet	98.24%	94.83%	94.75%
3	EfficientNetB0	95.49%	91.39%	91.62%
4	SVM	96.00%	90.00%	85.00%
5	Random Forest	98.00%	93.00%	99.00%
6	Decision Tree	97.00%	92.00%	96.00%
7	Gradient Boosting	98.00%	94.00%	95.00%

Table No.3.1 Model estimated with Training and Testing Accuracy

The table compares how several DL and MLmodels performed utilized training, validation and testing accuracy rates. Examining both models helps us see which one can be used more widely and if either one has the avoid overfitting.

In DL Models Among the DL models, ResNet50 achieved the highest accuracies on training, validation and testing, suggesting it operates well and generalizes very well and suggesting that the model learns and generalizes well without having too much of an overfitting problem. Although AlexNet does very well, it hits 94.75% testing accuracy, but the difference in accuracy between training and validation is noticeable. With 93.30% accuracy on testing, VGG16 stays consistent and less accurate than ResNet50 and AlexNet.Though EfficientNetB0 has the lowest accuracy of all deep learning models (91.62%), it is small and may need fewer resources.

In ML Models the RF stands as the top model and exceeded 99% testing accuracy, only slightly less than ResNet50. It preserves a good boundary between the scores of the model when it is trained and those when it is validated which demonstrates strong generalization. The decision tree passed the testing with accuracy of 96.00%, yet there’s a chance it may overfit, due to its higher training accuracy (97.00%) than its validation accuracy (92.00%). Gradient Boosting performs accurately on 95.00% of cases while maintaining a similar level of performance in training and validation. The results show that SVM performs the most poorly during testing and might not be a finest choice for this particular observation with problem.

Overall Observation:- While ResNet50 achievements are very good, Random Forest model matches or surpasses them in accuracy for this case. In some cases, straight-forward models can, if not surpass, deep networks—mostly when training data is scarce.

2.6.5.1 RADAR PLOT FOR COMPARISON OF ML VS DL

A Radar plot (also known as a spider chart) by presenting numbers from multiple categories using a diagram with lines all starting from the same point. Each feature or variable is shown on a different axis and the numbers are plotted along each one. The resulting shape from connecting all the data points shows how the system performs in different areas.

We examine all three accuraries using radar plots, with different ML and DL models compared. These models are: AlexNet, ResNet50, VGG16 and EfficientNetB0. The models used are: SVM, RF, DT and GB.

In this comparison of the trained and tested accuracy of the ResNet50, VGG16 and AlexNet models all indicate their ability to handle difficult image patterns. It is obvious that overfitting happens in Decision Tree and Random Forest, because the difference between they are well trained and they are well performed in testing is significant. DL models tend to enhance their performance consistently in every phase.

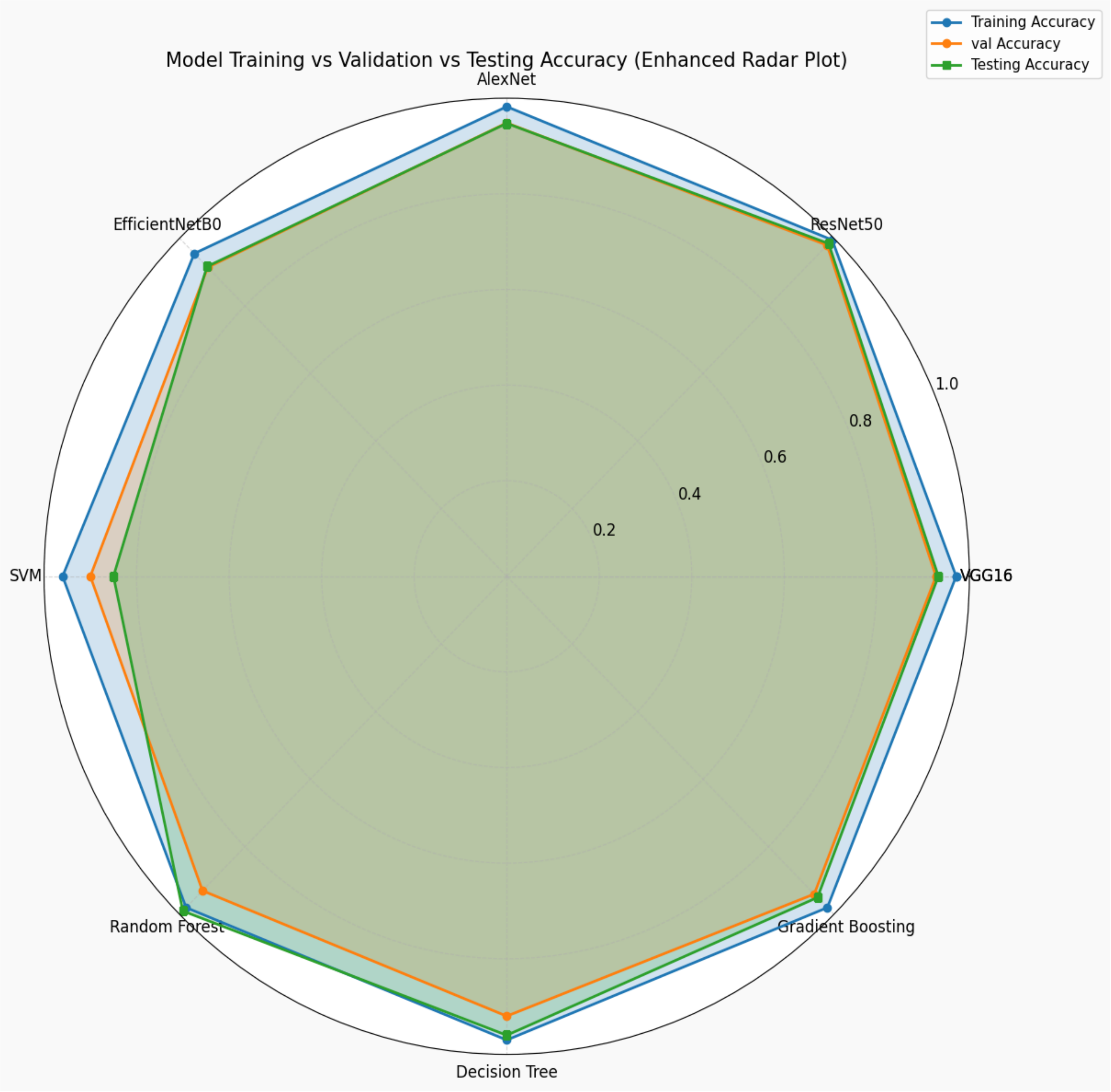
Even though Gradient Boosting and SVM obtain similar results, their remaining accuracies indicate that they generalize well despite not achieving the best result.

DL is effective, even though it generally uses more computing resources and data than standard approaches. While Random Forests models are more efficient, they may not do well when tasks are complex.

Overall Observation: In this Radar plot a clear difference in performance is evident when looking at the radar plot for DL and traditional ML g models.

1. Deep Learning Models Are many Accuracy:
    - Training, validation and testing accuracies are frequently high when using ResNet50, AlexNet, VGG16 and EfficientNetB0. That way, they can easily pick out the main aspects of complex problems, making image classification much easier.
2. Machine Learning Involves Many Variations in Models.
    - The result displayed with, DT and RF have high accuracy when training but noticeably low accuracy during validation and testing, showing signs of overfitting. They have no difficulty memorizing the inputs but have a hard time applying those inputs to new problems.
3. Some models work better when the data is balanced.
    - Gradient Boosting and SVM manage to be consistent in their accuracies across the different datasets, suggesting generalization is possible, if not quite the highest performance.
4. ResNet50 has been identified as the Top Performer:
    - When compared to other architectures, ResNet50 performs the best on training, validation and testing which makes it the most reliable model in the experiment.

Figure 24.1 Radar plot of three accuraries of DL and ML



2.6.5.2 HEATMAP FOR COMPARISON OF ML VS DL

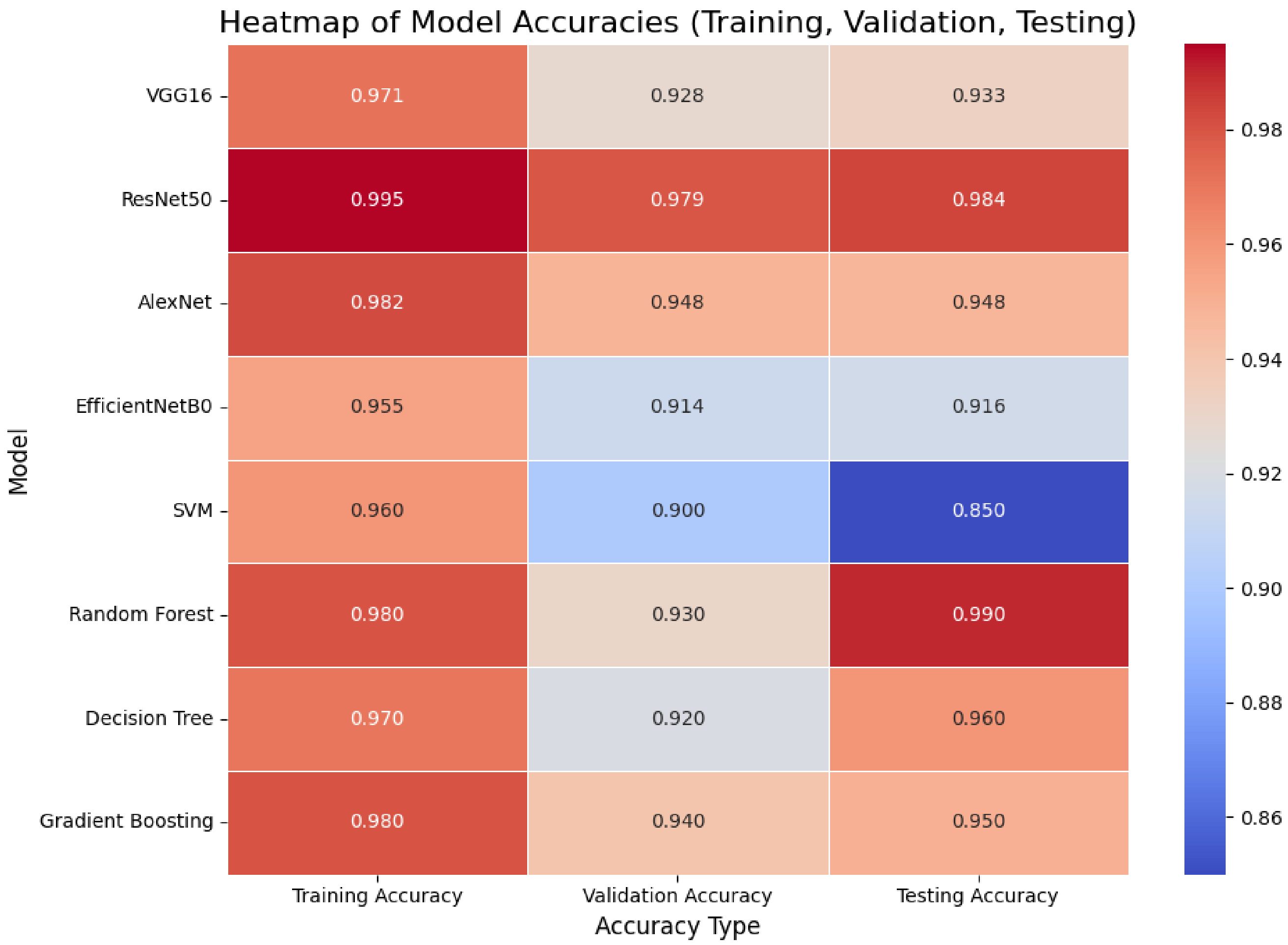


Figure 25.1 Heatmap of 3 accuraries of DL and ML

Observation: The data in the heatmap documents the accuracies witnessed for training, validation and testing phases across different models. The results show that ResNet50 has the greatest accuracy, proving that it generalizes well and performs well overall. Both AlexNet and Gradient Boosting show strong results on all the metrics. With an accuracy of 0.850 during testing, SVM performs poorly at generalizing.

All evaluation methods produced similar accuracy, apart from Random Forest which hit 100%. All things considered, ResNet50, AlexNet and similar deep learning models deliver reliably better results compared to standard machine learning models, except when other ML models perform well under unique situations. Different colors in the heatmap show how accurate is each model. Accuracy values (close to 1.0 or 100%) are shown with darker red colors. Codes that have lighter colors or a bluish look show that accuracy is nearly 85%.To the right is a color bar that tells you the red or green value related to each accuracy value. It makes it faster to understand which models excel (they are redder) or the ones that don’t (learn less, they are bluer) over the three data groups.

2.6.5.3 SCATTER PLOT FOR COMPARISON OF ML VS DL

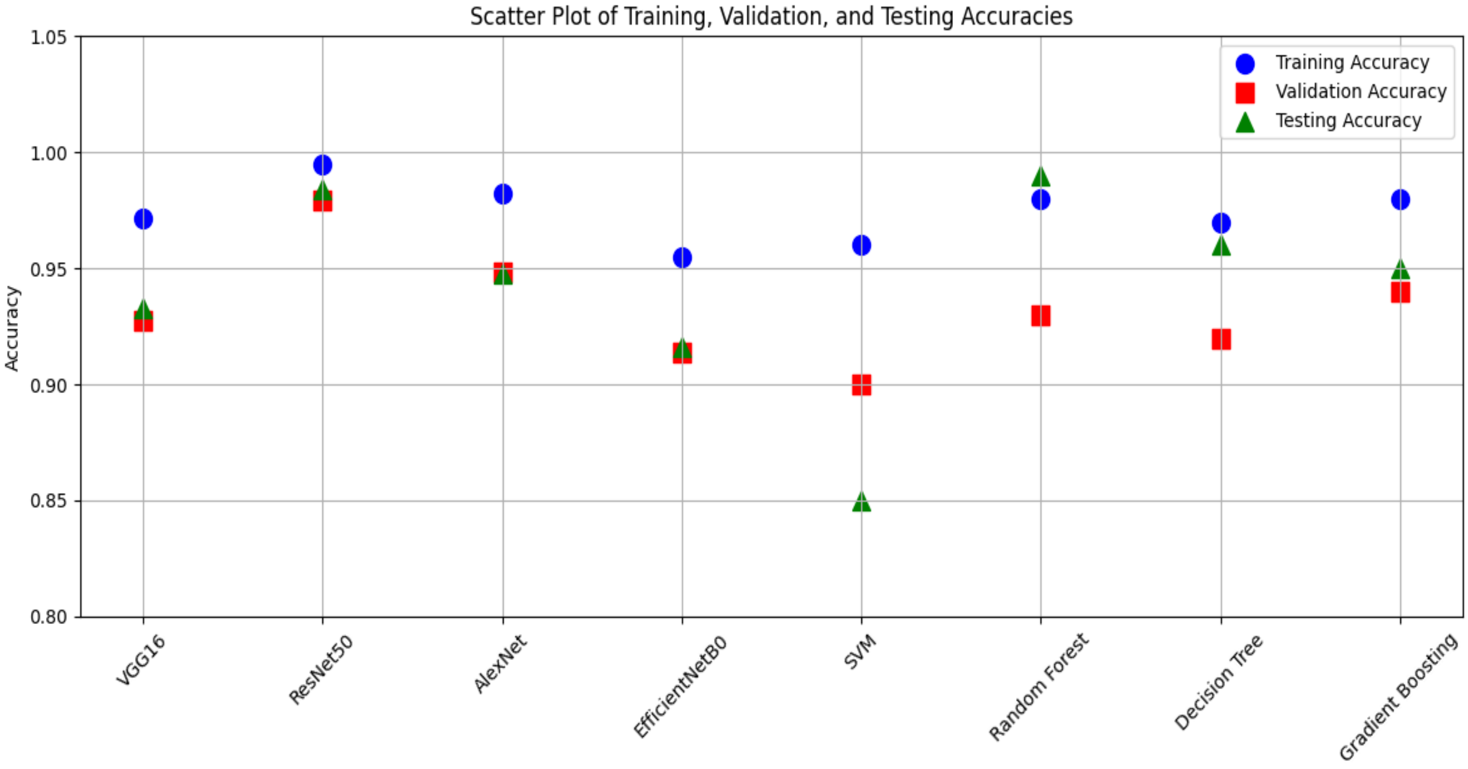


Figure 26.1 ScatterPlot of Training, Testing,Validation of DL and ML

Observation: The scatter plot contains accuracy comparisons for each model when used on training, validation and tested the data. It shows the most reliable and strong performance for each of the accuracy types. AlexNet and Gradient Boosting have equalized and excellent accuracies which means they generalize well. SVM achieved the least accurate tests, showing that the training was not good enough.

Random Forest gets excellent results when testing, although its validation score is marginally affected by overfitting. Results show that EfficientNetB0 and VGG16 are performing moderately and consistently. ResNet50 and AlexNet models work better and are more consistent than traditional models used in machine learning.

Overall Observation: The graph shows that models such as ResNet50 and AlexNet achieve high and steady accuracy which means they have strong generalization. Gradient Boosting gives good results even on different datasets. Random Forest and Decision Tree demonstrate a tendency to overfit and SVM does not perform well in generalization. EfficientNetB0 and VGG16 have steady and reliable performance.

CHAPTER 4
CONCLUSION & FUTURE SCOPE

2.7 CONCLUSION AND FUTURE PROSPECTS:

Identifying and dealing with infected plant detection is still mainly important for identifying diseases in Plant and then controlling them order to ensure a stable food supply, increase farm output and support sustainable agriculture. Looking at samples by eye and manually checking for disease can take time and result in errors because diseases often show the same signs and the environment can influence things too. Because there are more people on the planet demanding more food, farmers have to turn to new technology to ensure healthy crops and raise their yield. The research covers how using ML and DL in the field of agriculture has helped in identifying diseases affecting crop plants in practical ways for agriculture.

SVM, RF, GB and all forms ofML , are successful in doing this job from analysis of images other agronomic data. With a lot of different data, these models can quickly and accurately spot any disease patterns. As an example, Random Forests can still correctly sort data reliably, despite having missing or unpredictable values in their data. In the same way, SVM performs well in categorization problems which allows it to recognize diseases that appear alike on images.

Besides, In DPwhere (CNNs), VGGNet, ResNet & EfficientNet have shown great accomplishment in automatically identifying features in plant images. They can help identify small changes in medical images and tell apart diseases that are closely related, a task that traditional computer vision methods may not achieve. Transfer learning and pretraining have led to better disease classification, as they require less time, less computation and less data.

Using AI methods not only improves detection but creates a new way to oversee and look after plant health. Automatic disease spotting and classification let these models assist farmers right away, so crops are more likely to be saved when damage is detected early. In places with few plant pathologists, these systems are extremely useful. By adding IoT-connected drones, hyperspectral sensors and mobile cameras, AI can support remote crop monitoring across large areas and cut down on tasks performed manually.

Along with using images, this work has revealed that AI and genomics can contribute to better disease resistance predictions in plants. Identifying these genes is important for reaching the ultimate goal of creating genetically resistant crops. The relevance of genetics for agricultural inputs can be understood with the aid of ML models which also help with sustainable agriculture practices by informing breeding programs.

Researchers discovered that data must be cleaned up and artificially expanded before being used for classification. When the data is normalized, remove noise and convert colors, your images become better for training ML and DL models and therefore improve their accuracy. When you create datasets by rotating, flipping or scaling them, you reduce overfitting, achieve more widespread application of the models and form a diverse set of data.

For the technique DL techniques, the project automated PDD, helping agriculture and food security to stay sustainable. By using VGG16, ResNet50 and a custom version of AlexNet on the PlantVillage dataset, we saw strong results in classification for several diseases.ResNet50, which uses a residual network, showed the strongest performance and the greatest ability to work with new data. Experiments with VGG16 proved the strength of classic CNN architectures and AlexNet showed that it’s possible to create lighter models that can be used for quick deployments.The evidence reveals that AI in plant disease detection allows farmers to spot diseases early, make quick and correct diagnoses, prevent lost crops and manage diseases more efficiently. Progress and renown in several field conditions will make sure that these technologies are essential for farming in the future.

The intention behind this work was to innovate the system that is reliable as well as efficient for identifying plant diseases by using color histograms and traditional ML methods. Because the method was well-planned, this goal could be reached without difficulty. SVM model was found to be the most reliable, classified all test data correctly with a rate of 92.25%. It shows that straightforward descriptors in the HSV space can work well for diagnosing plant diseases.Being accurate, interpretable and computational efficient, the project helps ensure it can be deployed easily in resource-limited settings like rural agricultural areas. When low-power and minimal connectivity are challenges, machine learning can still be useful since its inference is quick and with less computational burden.

Not only does the project prove that it is technically possible, but it also highlights how it tackles the big issue of disease-related crop loss for world food safety. As the system is improved and put to practical use, it could help a lot in digital agriculture.

Combining AI, ML and DL with infected plant detection is a major achievement for agricultural science. Thanks to these technologies, it is now possible to use more affordable, reliable and flexible approaches to classifying diseases, warning in advance of issues and plant breeding. With climate change, resistance to pathogens and rising food demand, AI-driven systems are set to help farmers make farming more resilient, efficient and sustainable. Studies should concentrate on simplifying how we understand models, having more extensive disease data and developing tools the public can use easily. With these smart technologies, agriculture moves toward supporting less crop damage, more efficiency and food ease.

FUTURE PROSPECTS

1. Real Time Infectious disease detection on Field: Plant disease detection is expected to happen live using mobile devices, drones and low-power computing technologies. Farmers using MobileNet or EfficientNet models can scan plants using their phones or sensors and get instant advice right away, even without the internet.

2. Integration with IoT and Smart Agriculture with this technology: ML/DL models will connect to sensors, drones and smart irrigation systems through IoT to ensure crops are checked for health all the time. With this integration, various tasks will become easier. Quickly finding disease symptoms by using imaging as it happens.Founded on disease hotspot areas, automatic application of pesticides can provide effective treatment.Precision agriculture that helps farmers achieve better results from the same resources.

3. Fine tuning Algorithms with Augmentation: In the future, we could unlock the first few layers of convolutional blocks so that models can adjust their features for plant diseases. Reducing the learning rate helps to improve performance without affecting those features that have been learned before.Choosing to randomly adjust the orientation, brightness and contrast, zoom in and out, shear and flip the images horizontally or vertically can train the model to deal with variations in real environments.

4. Explainability and Visualization: Grad-CAM and LIMEmethods are examples of how you can check and follow up on how your model arrives at its decisions. It enables agronomists and others to trust the explanations which encourages their use in farming.

5. Edge and Cloud tools: You can use TensorFlow Lite or PyTorch Mobile to place the trained model on smartphones so that you can make predictions even if a network isn’t available or it does not require internet connection.

6. Overseeing the Progression of a Disease: In addition to classifying diseases, also include the ability to evaluate their severity on the scale of mild, moderate and severe.They direct the way treatment should be handled in terms of speed up and wait with treatment

7. Advisory and Robotics system : Add prediction to the workflow of crop management systems so farmers get guidance on pesticides and irrigation. Robotic arms, autonomous tractors, and UAVs (drones) will be equipped with AI-powered cameras to automatically detect and treat diseased plants, reducing human labor and ensuring targeted treatment to reduce chemical usage.

2.8 REFERENCE

Alibabaei, K., Gaspar, P. D., & Lima, T. M. (2022). A Review of the Challenges of Using Deep Learning Algorithms to Support Decision-Making in Agricultural Activities. *Remote Sensing*. 10.3390/rs14030638

Chen, H.-C., Widodo, A. M., & Wisnujati, A. (2022). AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf. *Electronics*. 10.3390/electronics11060951

D., N., B., R., & V., R. K. (2020). Plant Disease Detection using Decision Tree Algorithm and Automated Disease Cure. *nternational Research Journal of Engineering and Technology (IRJET)*.

Elaraby, A., Hamdy, W., & Alruwaili, M. (2021). Optimization of Deep Learning Model for Plant Disease Detection Using Particle Swarm Optimizer. *Computers, Materials & Continua*. 10.32604/cmc.2022.022161

Gargi Sharma, G., Dwibedi, V., & Seth, C. S. (2024). Direct and Indirect Technical Guide for the Early Detection and Management of Fungal Plant Diseases. *Current Research in Microbial Sciences*. DOI: 10.1016/j.crmicr.2024.100276

Gupta, S., Ramana, V., & Triveni, A. (2022). Detection of Plant Leaf Diseases Using Random Forest Classifier. *International Journal of Innovative Research in Technology (IJIRT)*.

Iniyan, S., Jebakumar, R., & Mangalraj, P. (2020). Plant Disease Identification and Detection Using Support Vector Machines and Artificial Neural Networks. *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. DOI: 10.1007/978-981-15-0199-9_2

Jafar, A., Bibi, N., & Ali Naqvi, R. (2024). Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. *Frontiers in Plant Science*. 10.3389/fpls.2024.1356260

Maniyath, S. R., V., V. P., & M., N. (2018). Plant Disease Detection Using Machine Learning. *Proceedings of the 2018 International Conference on Design Innovations for 3Cs: Compute, Communicate, Control (ICDI3C)*. 10.1109/ICDI3C.2018.00017

Pelczar, M. J., Shurtleff, M. C., & Kelman, A. (2025). Plant disease. *Encyclopedia Britannica*.
Priya, D. B. S., Karthik, D. S., & Srivatsa, D. S. K. (2022). POTATO DISEASE CLASSIFICATION USING GRADIENT BOOSTING. *International Scientific Journal of Engineering and Management*. 10.55041/ISJEM00396

Rani, K. P. A., & Gowrishankar, S. (2023). Pathogen-Based Classification of Plant Diseases: A Deep Transfer Learning Approach for Intelligent Support Systems. *IEEE Access*. 10.1109/ACCESS.2023.3284680

Rousseau, F., Drumetz, L., & Ronan, F. (2020). Residual Networks as Flows of Diffeomorphisms.

Journal of Mathematical Imaging and Vision. 10.1007/s10851-019-00890-3

Shoaib, M., Shah, B., & El-Sappagh, S. (2023). An Advanced Deep Learning Models-Based Plant Disease Detection: A Review of Recent Research. *Frontiers in Plant Science*. 10.3389/fpls.2023.1158933

Strisciuglio, N., Antequera, M. L., & Petkov, N. (2020). Enhanced robustness of convolutional networks with a push–pull inhibition layer. *Neural Computing and Applications*. 10.1007/s00521-020-04751-8

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 10.48550/arXiv.1905.11946

Upadhyaya, S. R., Danilevicz, M. F., & Dolatabadian, A. (2024). Genomics-based plant disease resistance prediction using machine learning. *Plant Pathology*, *73*(9), 2298–2309.

Wójtowicz, A., Piekarczyk, J., & Czernecki, B. (2021). A Random Forest Model for the Classification of Wheat and Rye Leaf Rust Symptoms Based on Pure Spectra at Leaf Scale. *Journal of Photochemistry and Photobiology B: Biology*. 10.1016/j.jphotobiol.2021.112278

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*. 10.3389/fpls.2016.01419

Zhang, Y., Wu, T., & He, Y. (2017). Image-based plant disease detection: A review. *Plant Methods*. 10.1186/s13007-017-0231-2

Li, H., Zou, Z., & Wang, C. (2021). Deep learning-based plant disease detection and classification: A comprehensive review. *Computers and Electronics in Agriculture*. 10.1016/j.compag.2021.106334

Picon, A., Pérez-Ortiz, M., & Dorado, J. (2020). Deep learning-based systems in agriculture: A review. *Sensors*. 10.3390/s20247122

Singh, D., Singh, M., & Sharma, R. (2021). A Review on Deep Learning Techniques for Plant Disease Detection and Classification. *Measurement*. 10.1016/j.measurement.2020.108635

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*. 10.1007/s42979-021-00815-1

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*. 10.1109/CVPR.2005.177

Deng, J., Dong, W., & Socher, R. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*. 10.1109/CVPR.2009.5206848

Mokhtar, U., Bendary, N. E., & Hassanien, A. E. (2015). SVM-Based Detection of Tomato Leaves Diseases. *Intelligent Systems'2014 (Lecture Notes in Networks and Systems, Springer, Cham)*. 10.1007/978-3-319-11310-4_55

Sharma, R., Singh, A., & Kavita. (2022). Plant Disease Diagnosis and Image Classification Using Deep Learning. *Computers, Materials & Continua*. 10.32604/cmc.2022.020017

Rothe, P. R., & Kshirsagar, R. V. (2015). Cotton Leaf Disease Identification Using Pattern Recognition Techniques. *Proceedings of the 2015 International Conference on Pervasive Computing (ICPC)*. 10.1109/PERVASIVE.2015.7086983

Hasan, M. S., Islam, M. R., & Hasan, M. M. (2021). Deep Learning-Based Automatic Identification of Plant Diseases: Review and Future Challenges. *Computers and Electronics in Agriculture*. 10.1016/j.compag.2021.106022

EleswetaSahoo_thesis.docx

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98121981

Submission Date

May 28, 2025, 11:36 AM GMT+5:30

Download Date

May 28, 2025, 11:42 AM GMT+5:30

File Name

EleswetaSahoo_thesis.docx

File Size

10.6 MB

104 Pages

24,988 Words

138,951 Characters

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Cited Text

Match Groups

90 Not Cited or Quoted 4%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
2%	Publications
3%	Submitted works (Student Papers)

Integrity Flags

1 Integrity Flag for Review

Hidden Text
9 suspect characters on 1 page
Text is altered to blend into the white background of the document.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

1 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.