



ROME BUSINESS SCHOOL

MASTER IN DATA SCIENCE MANAGEMENT

Final Project by

**ADEDOTUN ADEDAMOLA AYODELE
(REG. NO: 23DS2606AA90)**

**IMPLEMENTATION OF PLANT DISEASE CLASSIFICATION
USING IMAGE PROCESSING:
A CASE STUDY USING TOMATO LEAVES**

Academic Year: 2023/24

ROME, OCTOBER 2024

CERTIFICATION



ROME BUSINESS SCHOOL

This project work titled, **IMPLEMENTATION OF PLANT DISEASE CLASSIFICATION USING IMAGE PROCESSING: A CASE STUDY USING TOMATO LEAVES** prepared and submitted by **SEGUN SAMUEL OTITOLAYE, BABASOJI MORAYO ARIJE, ANTONIA CHINAZOR EZULIKE, NNENNA AGBEZE OKO AND ADEDOTUN ADEDAMOLE AYODELE** in partial fulfilment

of the award of the degree of **MASTER IN DATA SCIENCE MANAGEMENT**

is hereby accepted.

Date Defended / Approved: September 14th, 2024

[Solomon Adebayo]
Program Director

Samuel Igwe
Head of Academic



EXECUTIVE SUMMARY

Introduction:

Plant diseases are a major hindrance in Agriculture, causing significant crop losses as well as negatively impacting human health. Early disease detection is crucial for minimizing these losses and ensuring food security. Traditional methods of disease detection are time-consuming, prone to errors, and impractical for large-scale farms. This study will focus on implementing machine learning models to automate disease classification using image processing. Tomato leaves will be used as the case study. The significance of this research lies in its potential to enhance crop yields, farmer income, and food security by providing a reliable tool for early disease detection.

Methodology:

The study used an experimental design. Both classical machine learning and deep learning models were implemented and compared on a dataset of tomato leaf images from Kaggle. An algorithm for detecting tomato diseases was proposed, focusing on image segmentation and classification. Limitations in dataset quality and generalizability were acknowledged.

Results:

SVC Achieved 92% accuracy, excelling in distinguishing healthy leaves and certain diseases, but faced issues with diseases having similar symptoms. KNN Recorded 87% accuracy with high precision and recall, but struggled with diseases having overlapping symptoms. Logistic Regression Obtained 83% accuracy, showing strong precision and recall but lower accuracy compared to other models. Random Forest Achieved 90% accuracy, performing well overall but facing challenges with diseases having similar symptoms. Decision Tree showed 88% accuracy and high precision and recall, though the absence of an F1-score makes its full performance assessment difficult.

Conclusion:

KNN and Logistic Regression are recommended for balanced performance with high Precision and Recall, and the highest F1-scores. SVC is suitable when Accuracy and Specificity are prioritized. Random Forest is a robust choice for high Precision, Recall, and F1-Score, despite a slightly lower Accuracy. Decision Tree may be considered if interpretability is critical, but its lower Accuracy and missing F1-score should be noted.



TABLE OF CONTENTS

EXECUTIVE SUMMARY	2
TABLE OF CONTENTS.....	3
LIST OF FIGURES AND TABLES.....	5
LIST OF ABBREVIATIONS AND GLOSSARY	6
PREFACE.....	8
CHAPTER ONE	9
INTRODUCTION	9
1.1 Background.....	9
1.2 Statement of the Problem.....	10
1.3 Aims and Objective.....	10
1.4 Significance of the Study	11
1.5 Scope of the Study	11
CHAPTER TWO	12
LITERATURE REVIEW	12
2.1 Overview.....	12
2.1.1 First Phase of Deep Learning (1943-2006).....	13
2.1.2 The Rise of Deep Learning Applications (Second Phase Beyond 2006)	14
2.2 Plant disease detection by DL architecture	15
2.3 Convolutional Neural Network (CNN).....	16
2.4 Advantages and Disadvantages of Convolutional Neural Networks (CNNs):	19
2.4.1 Advantages of Convolutional Neural Networks (CNNs):	19
2.4.2 Disadvantages of Convolutional Neural Networks (CNNs):.....	19
CHAPTER THREE	20
RESEARCH AND METHODOLOGY	20
3.1 Research Design.....	20
3.2 Data Preparation - Collection and Preprocessing	20
3.3. Classical Machine Learning Techniques	23
3.4 Deep Learning Techniques	23
3.5. Performance Evaluation Metrics.....	26
3.6. Proposed Algorithm for Disease Detection	27



ROME BUSINESS SCHOOL

3.7 Ethical Considerations	27
3.8 Limitations of the Methodology	27
CHAPTER FOUR.....	28
RESULTS AND DISCUSSIONS.....	28
4.1 Result Analysis	29
4.2 Analysis of the Training and Validation Loss /Training and accuracy Plot	37
4.3 Comparative Analysis of the five different Models.....	40
CHAPTER FIVE	43
CONCLUSION AND RECOMMENDATION.....	43
5.1 Conclusion	43
5.2 Recommendation	43
BIBLIOGRAPHY.....	44



LIST OF FIGURES AND TABLES

LIST OF FIGURES

Figure 1: Timeline of Deep Neural Network Development	14
Figure 2: Plant Disease Detection and Classification Workflow	15
Figure 3: Simple CNN Architecture	16
Figure 4: Diagram of Object with Dimensions and Color Channels	17
Figure 5: Convolution Operation	18
Figure 6: Model Involving Detection and Recognition of Tomato Plant Disease	19
Figure 7: Examples of Tomato Leaf Images Showing Various Conditions	21
Figure 8: Classifier Model Used	22
Figure 9: CNN Model Architecture	24
Figure 10: Images of Tomato Leaves with Disease Labels and Texture Features	28
Figure 11: Confusion Matrix for SVC	30
Figure 12: Confusion Matrix for KNN	32
Figure 13: Confusion Matrix for Logistic Regression	33
Figure 14: Confusion Matrix for Random Forest Classifier	35
Figure 15: Confusion Matrix for Decision Tree Classifier	37
Figure 16: Training and Validation Loss Graph	38
Figure 17: Training and Validation Accuracy Graph	40
Figure 18: Comparative Analysis of the five different models	42

LIST OF TABLES

Table 1: Hyper-parameter of Deep Neural Network	25
---	----



LIST OF ABBREVIATIONS AND GLOSSARY

Table	Meaning
AN	Generative Adversarial Nets
CA	Classification Accuracy
CNN	Convolutional Neural network
DCNN	Deep Convolutional Neural Network
DCRN	Densely Connected Recurrent Convolutional Network
DL	Deep Learning
ELM	Extreme Learning Machine
FCN	Fully Convolutional Neural Network
FRKNN	Feature Ranking K-Nearest Neighbor
GRU	Gated Recurrent Unit
HSI	Hyperspectral Imaging
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
INAR-SSD	Single Shot Detector with Inception module and Rainbow concatenation
IoU	Intersection of Union
IRCNN	Inception Recurrent Convolutional Neural Network
IRRCNN	Inception Recurrent Residual Convolutional Neural Network
KNN	K-Nearest Neighbor
LR	Logistic Regression



LRP	Layer-wise Relevance Propagation
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
NiN	Network in Network
PCA	Principal Component Analysis
PSPNet	Pyramid Scene Parsing Network
R-FCN	Region-based Fully Convolutional Networks
R2U-Net	Recurrent Residual Convolutional Neural Network based on U-Net model
RCN	Region based Convolutional Neural Network
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SRCNN	Super-Resolution Convolutional Neural Network
SSD	Single Shot Detector
SVM	Support Vector Machines
ToF	Time-of-Flight



PREFACE

I gladly dedicate this work to God, acknowledging His provision of means and strength that enabled me to complete this task. My deepest appreciation goes to my family, who have been a constant source of support and inspiration throughout my journey. As the youngest among siblings, I have had numerous role models to emulate, learning invaluable lessons in determination and work ethic from each of them.

I am sincerely grateful to the Data Science faculty at Rome Business School Nigeria for equipping me with a robust foundation in Data Science. Their unwavering support, both didactic and administrative, has been instrumental in my academic and professional growth. To the faculty and staff, I extend my heartfelt thanks.

Special appreciation goes to my project supervisor, Mr. Adebayo Solomon, whose expertise, dedication, and insightful guidance were instrumental in shaping this project. His unwavering commitment to excellence and his ability to challenge and inspire me to think critically pushed me beyond my comfort zone, leading to the successful completion of this work. Mr. Solomon's meticulous attention to detail, his patience in addressing my queries, and his willingness to provide constructive feedback at every stage of the project were invaluable. His mentorship not only enhanced the quality of this project but also significantly enriched my learning experience. I am truly grateful for the opportunity to have worked under his supervision and for the profound impact he has had on my academic and professional development.

I also extend my gratitude to all members of the Francesco Group for their collective effort, dedication, and collaboration, which played a crucial role in ensuring the success and impact of this work. This project entailed countless hours of research, rigorous editing, virtual meetings, tutorials, discussions, and meticulous planning. I deeply appreciate the sacrifices and contributions of everyone involved, without which this endeavor would not have been possible.

Adedotun Ayodele

Lagos.

August 2024



CHAPTER ONE

INTRODUCTION

1.1 Background

The Agricultural industry underpins global food security by providing sustenance for the rapidly growing population. However, numerous threats hinder crop production, with plant diseases being a major concern. These diseases not only cause significant crop losses, but also negatively impact human health and the livelihoods of farmers who rely on healthy harvests. Early and accurate disease detection is crucial for minimizing these losses and ensuring food security. Tomato (*Lycopersicon Esculentum*) is one of the most consumed fruits daily. As tomatoes are utilized in condiments such as ketchup, sauce, and puree, their global utilization rate is high. It constitutes approximately 15% of all vegetables and fruits, A person consumes approximately 42 kilograms of tomatoes annually (Laranjeira *et al.*, 2022). The high demand for tomatoes necessitates the development of early detection technologies for viruses, bacterial, and viral contaminations.

Traditionally, farmers rely on visual inspection during harvest to identify and remove diseased plants. This manual method is time-consuming, particularly for large farms, and prone to human error due to factors like fatigue and lack of expertise in specific diseases. Additionally, visual inspection often occurs at later stages, limiting the effectiveness of treatment options. Advancements in machine learning and image processing offer a powerful solution for early and accurate plant disease detection. Image processing techniques can extract relevant features from images of plant leaves, while machine learning models can leverage these features to automate disease classification. This automation significantly reduces the time and resources required for disease identification compared to traditional methods. This thesis focuses on implementing and evaluating machine learning models for the classification of diseases in tomato leaves. Tomato crops are particularly susceptible to various diseases, and early detection is crucial for ensuring high yields and farmer income

Image processing plays a vital role in this research by preparing the data for machine learning models. Several techniques will be employed to extract relevant features from tomato leaf images. These techniques may include image pre-processing steps like noise reduction, color normalization, and resizing for consistency. Feature extraction techniques such as color histograms, texture analysis, and edge detection will be utilized to capture crucial information about the disease symptoms visible on the leaves. This research will explore the implementation of various machine learning models for detecting and classifying tomato leaf diseases. Convolutional Neural Networks (CNNs) are a popular choice due to their ability to automatically learn features directly from images, potentially achieving high accuracy. Additionally, models like K-Nearest Neighbors (KNNs) may be explored for their simplicity and interpretability, which can be valuable for understanding the model's decision-making process.



The main objectives of this thesis are:

- To develop a system for automatic classification of tomato leaf diseases using image processing and machine learning techniques.
- To evaluate the performance of different machine learning models on a benchmark dataset of tomato leaf images.
- To investigate the explainability of the chosen machine learning models for improved user trust and understanding.

This research contributes to the development of a practical and user-friendly tool for farmers to identify tomato leaf diseases early and accurately. By automating the disease identification process, this system can potentially improve crop yields, farmer income, and overall food security. Additionally, the focus on explainability can build trust and encourage user adoption of this technology. By developing an automated image-based disease classification system, farmers and agricultural professionals can benefit from a faster, more reliable, and potentially more cost-effective tool for monitoring and managing tomato crop health.

1.2 Statement of the Problem

Traditional methods of disease identification often rely on visual inspection by experts/farmers. Visual inspection of crops for disease diagnosis is highly subjective, relying on the inspector's experience and expertise, which can lead to inconsistencies and misdiagnoses, especially for less common diseases or those with subtle symptoms. This process is time-consuming, requiring thorough examination of large fields, which delays treatment and potentially worsens the disease's impact. Many farmers lack the expertise to accurately identify every crop disease, and hiring specialists is expensive and often not feasible in all locations. Early detection is critical for effective treatment, but visual inspection frequently misses diseases in their preliminary stages when symptoms are subtle. Inspectors may experience fatigue during extended periods of visual inspection, resulting in decreased accuracy and missed diagnoses. Additionally, visual inspection is impractical for large-scale farms, limiting its effectiveness in managing widespread disease outbreaks. Environmental factors such as lighting conditions and weather further complicate visual inspection by affecting the appearance of disease symptoms, making accurate diagnosis even more challenging (Razzaq et al., 2019)

1.3 Aims and Objective

This project aims to propose a solution using image processing techniques. This approach aims to address the limitations of traditional methods by:



- **Automating disease detection:** This reduces reliance on manual inspection, enabling faster and more efficient disease identification as demonstrated in the work by Ngugi, Abelwahab, and Abo-Zahhad (2021).
- **Improving accuracy:** Utilize image processing algorithms to potentially achieve higher accuracy in disease classification compared to the human eye (Somowiyarjo, 2011). Early detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network reports promising results using deep learning techniques.
- **Facilitating early detection:** Enable earlier identification of diseases, allowing for timely intervention and improved crop management strategies as highlighted in (Somowiyarjo, 2011).
- **Standardizing disease identification:** Provide a consistent and objective method for disease classification, reducing subjectivity inherent in visual inspection.

1.4 Significance of the Study

This research explores the use of image processing and machine learning for automated disease classification in tomato leaves. This is significant because the algorithms used analyze images consistently, thus reducing subjectivity. By doing this, we save time and resources otherwise associated with manual inspection, it also means that we can apply these methods to large farms, which helps in efficient disease management. It can be applied to identifying diseases in early stages based on visual cues, this reduces dependency on experts to identify the diseases, thus saving costs.

1.5 Scope of the Study

This project is focused on using an artificial neural network algorithm to develop a predictive model that will enable autonomous systems to detect, recognize and classify the tomato plant disease, with a high level of accuracy.



CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

This chapter covers both the literature review and the technical survey of the techniques used in this project. Recent advancements in image processing and machine learning offer a powerful solution for automated plant disease classification. A literature review is carried out to cover research in the fields of techniques of identification and classification in Plant disease detection. A varying range of past and present techniques are studied and evaluated. The Deep Learning (DL) approach which is a subcategory of Machine Learning (ML) was introduced in 1943 when threshold logic was introduced to build a computer model closely resembling the biological pathways of humans (McCulloch and Pitts, 1943). The classification of tomato diseases using conventional machine learning (ML) techniques has been a significant area of research due to its potential impact on agricultural productivity and sustainability. Myriad studies have explored different ML algorithms, feature extraction methods, and datasets to improve the accuracy and efficiency of disease detection and classification.

Techniques for Disease Classification

Researchers have employed a range of techniques to classify tomato diseases effectively. For instance, Support Vector Machines (SVM) have been widely utilized. Amara et al. (2017) applied SVM to classify tomato leaf diseases, using image processing techniques to extract features like color, texture, and shape. Their results demonstrated that SVM could effectively classify diseases such as late blight, early blight, and leaf mold with high accuracy. Similarly, the k-Nearest Neighbors (k-NN) algorithm has been popular for its simplicity and effectiveness. Zhang et al. (2018) utilized k-NN for identifying tomato leaf diseases by extracting color and texture features from leaf images. Their study showed that k-NN could achieve satisfactory classification results, particularly when combined with effective feature selection techniques.

Decision-Making Models

Decision Trees and ensemble methods like Random Forests have also been employed for tomato disease classification. Dubey and Jalal (2016) used Decision Trees to classify tomato leaf diseases based on color and texture features, providing an interpretable model beneficial for understanding the decision-making process. Additionally, Bhange and Hingoliwala (2015) demonstrated that Random Forests, which combine multiple decision trees, could improve classification accuracy and robustness, making them suitable for complex classification tasks.

Probabilistic Approaches



Probabilistic models like the Naive Bayes classifier have been applied in the context of tomato disease classification. Phadikar and Sil (2008) used the Naive Bayes classifier to categorize tomato diseases, focusing on extracting relevant features from leaf images. They demonstrated that the Naive Bayes classifier could achieve reliable performance, especially where the assumption of feature independence was reasonable.

Image Preprocessing and Feature Extraction

Image preprocessing and feature extraction are crucial aspects of conventional ML approaches to tomato disease classification. Techniques such as image segmentation methods like thresholding and clustering have been used to isolate diseased areas of leaves. Feature extraction techniques, including color histograms, texture descriptors like Local Binary Patterns (LBP), and shape features, have been pivotal in improving classification performance. Haralick et al. (1973) introduced texture features that have been widely adopted for their effectiveness in capturing spatial patterns in images.

Challenges and Limitations

While conventional ML techniques have shown promise in tomato disease classification, several challenges and limitations exist. A major challenge is the dependency on handcrafted features, which requires domain expertise and may not generalize well across different datasets or disease types. Additionally, these models can be sensitive to variations in lighting conditions, leaf orientations, and image resolutions. The need for extensive preprocessing and feature extraction steps can also make these approaches time-consuming and computationally expensive.

Conventional machine learning techniques have been extensively studied for tomato disease classification, with various algorithms demonstrating effectiveness in different contexts. However, the reliance on feature extraction and the limitations in handling complex and diverse datasets highlight the need for continued research and the potential benefits of integrating more advanced techniques, such as deep learning, to further enhance classification accuracy and robustness. Future research should focus on developing more generalized models that can perform well across different environmental conditions and disease types, leveraging advancements in both conventional and deep learning methodologies.

2.1.1 First Phase of Deep Learning (1943-2006)

The historical development of deep learning concepts has evolved through several key phases, each marked by significant advancements and challenges. In the 1940s and 1950s, the field of cybernetics pioneered the idea of artificial learning inspired by biological brains, while Frank Rosenblatt developed the Perceptron for image recognition and Bernard Widrow introduced ADALINE, both limited in handling complex functions. The 1960s and 1970s saw a decline in research enthusiasm due to the limitations of linear models like Perceptron and ADALINE,



particularly their inability to handle XOR functions. During this period, connectionism emerged as an alternative approach inspired by cognitive science, challenging the dominance of symbolic reasoning. The 1980s marked the introduction of Artificial Neural Networks (ANNs) with hidden layers, mimicking the nervous system's structure, and the development of foundational deep learning concepts such as LSTMs, distributed representations, and backpropagation. Key developments during this phase included Dreyfus's work on backpropagation (1962), Kelley's chain rule (1960), Fukushima's Neocognitron (1980), LeCun et al.'s handwritten text recognition (1998), and the resolution of training problems by Hinton and Salakhutdinov (2006).

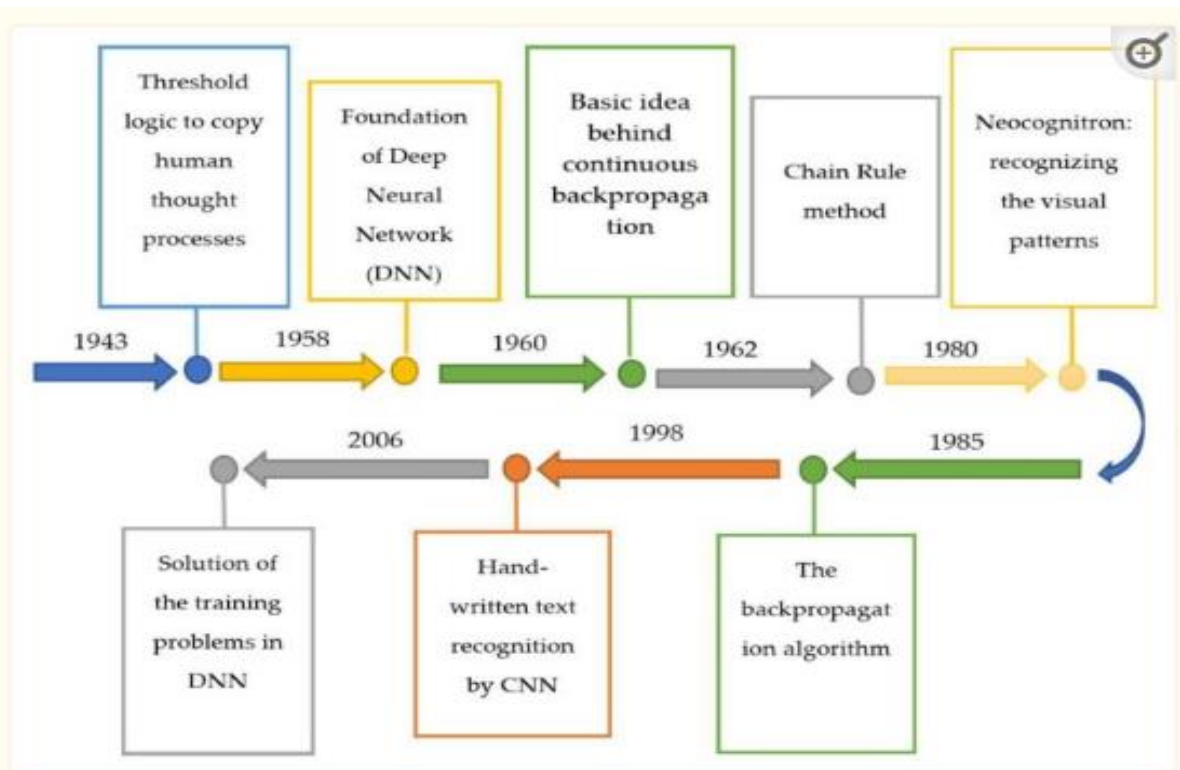


Figure 1: Timeline of Deep Neural Network Development (Goodfellow, Bengio and Courville, 2016).

2.1.2 The Rise of Deep Learning Applications (Second Phase Beyond 2006)

This phase witnessed the development of cutting-edge deep learning algorithms and architectures for applications ranging from self-driving cars (Bojarski et al., 2016) and healthcare (Esteva et al., 2017) to text recognition (LeCun et al., 1998), earthquake prediction (Mousavi et al., 2019), marketing (Lai et al., 2019), finance (Heaton et al., 2017), and image recognition (Krizhevsky et al., 2012). A notable achievement was AlexNet, which won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and inspired numerous new architectures to address earlier limitations (Krizhevsky et al., 2012). To evaluate deep learning performance, various metrics were employed, including Top-1%/Top-5% Error Rate (Krizhevsky et al., 2012), Precision and Recall (Szegedy et al., 2015), F1 Score (Szegedy et al., 2015), Training/Validation Accuracy and Loss



(Simonyan and Zisserman, 2014), and Classification Accuracy (CA) (He et al., 2016). Implementing deep learning models involves essential steps such as data collection, model training, and result visualization, as illustrated in Figure 2.

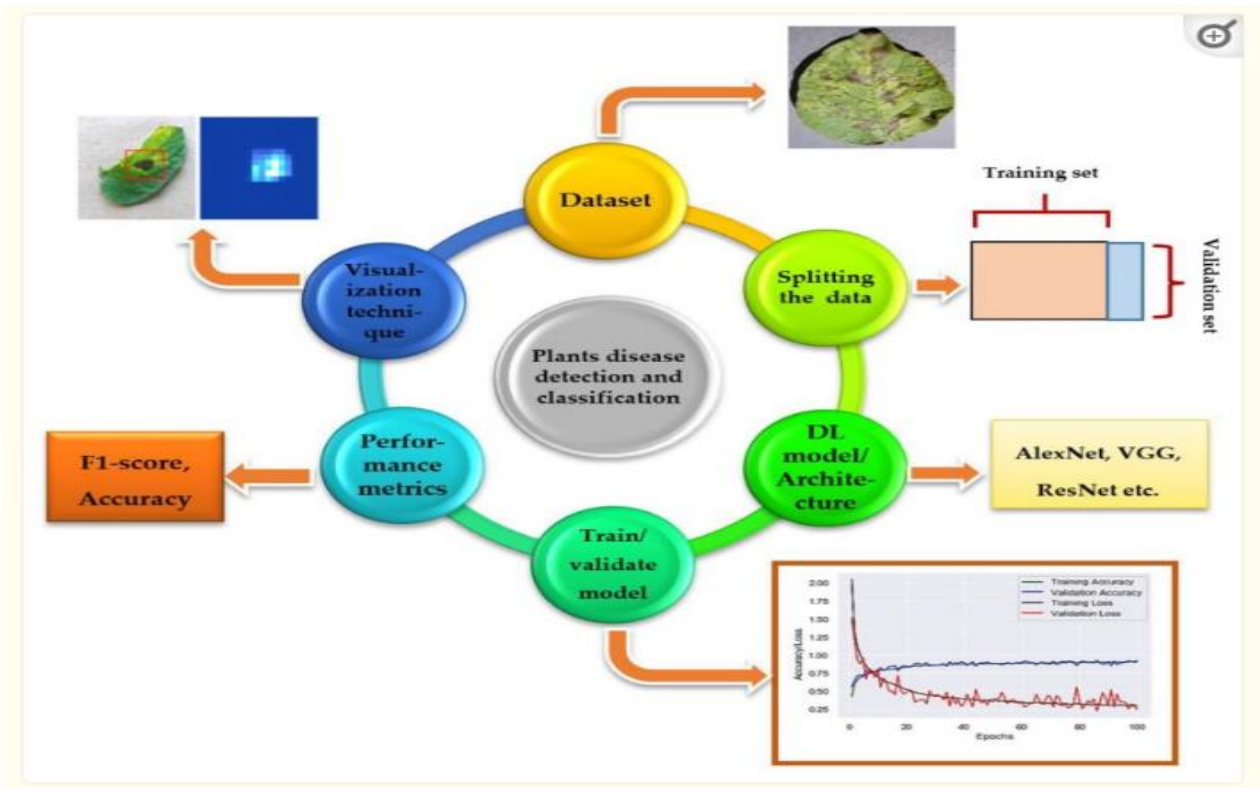


Figure 2: Plant Disease Detection and Classification Workflow Arsenovic *et al* (2019).

2.2 Plant disease detection by DL architecture

With advancements in deep learning (DL), researchers have successfully applied these architectures to image recognition and classification, including agricultural applications. For instance, Hall *et al* (2015) achieved 97.3% classification accuracy (CA) on 32 leaf species using a combined approach of a modified CNN and Random Forest (RF) classifier.

The introduction of AlexNet (Krizhevsky *et al.*, 2012) marked a turning point, leading to the development of numerous advanced DL models for image detection, segmentation, and classification. This section focuses on research utilizing prominent DL architectures for plant disease identification and classification. Several studies have explored novel visualization techniques and modified DL architectures to improve disease detection accuracy.



2.3 Convolutional Neural Network (CNN)

This deep learning neural network architecture is commonly used in computer vision, a field of artificial intelligence that enables computers to understand and interpret visual data. Artificial Neural Networks (ANNs) perform well in various datasets like images, audio, and text, with several types used for specific purposes. For instance, Recurrent Neural Networks, particularly LSTMs, are used for predicting word sequences, while Convolutional Neural Networks (CNNs) are employed for image classification. A regular neural network comprises three types of layers: Input Layers, where the number of neurons equals the total number of data features (e.g., pixels in an image); Hidden Layers, which process input through multiple neurons via matrix multiplication, learnable weights, biases, and activation functions; and Output Layers, where a logistic function like sigmoid or softmax converts the output into probability scores for each class. The process of feeding data into the model and obtaining output is called feedforward, followed by error calculation using functions like cross-entropy or square loss. Backpropagation is then performed to minimize loss by calculating derivatives. CNNs extend ANNs and are predominantly used to extract features from grid-like matrix datasets, such as images or videos, where data patterns are crucial.

2.3.1 CNN architecture

Convolutional Neural Network consists of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

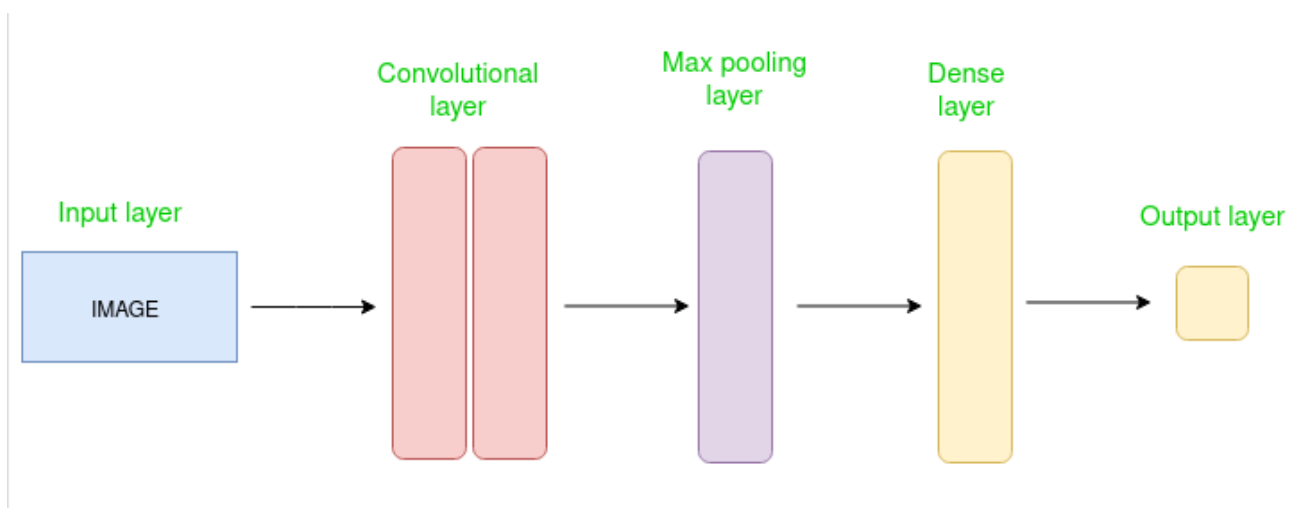


Figure 3: Simple CNN architecture.



The Convolutional layer applies filters to the input image to extract features, the Pooling layer down samples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through backpropagation and gradient descent.

2.3.2. How Convolutional Layers works

Convolution Neural Networks or convnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (i.e. the channel as images generally has red, green, and blue channels).

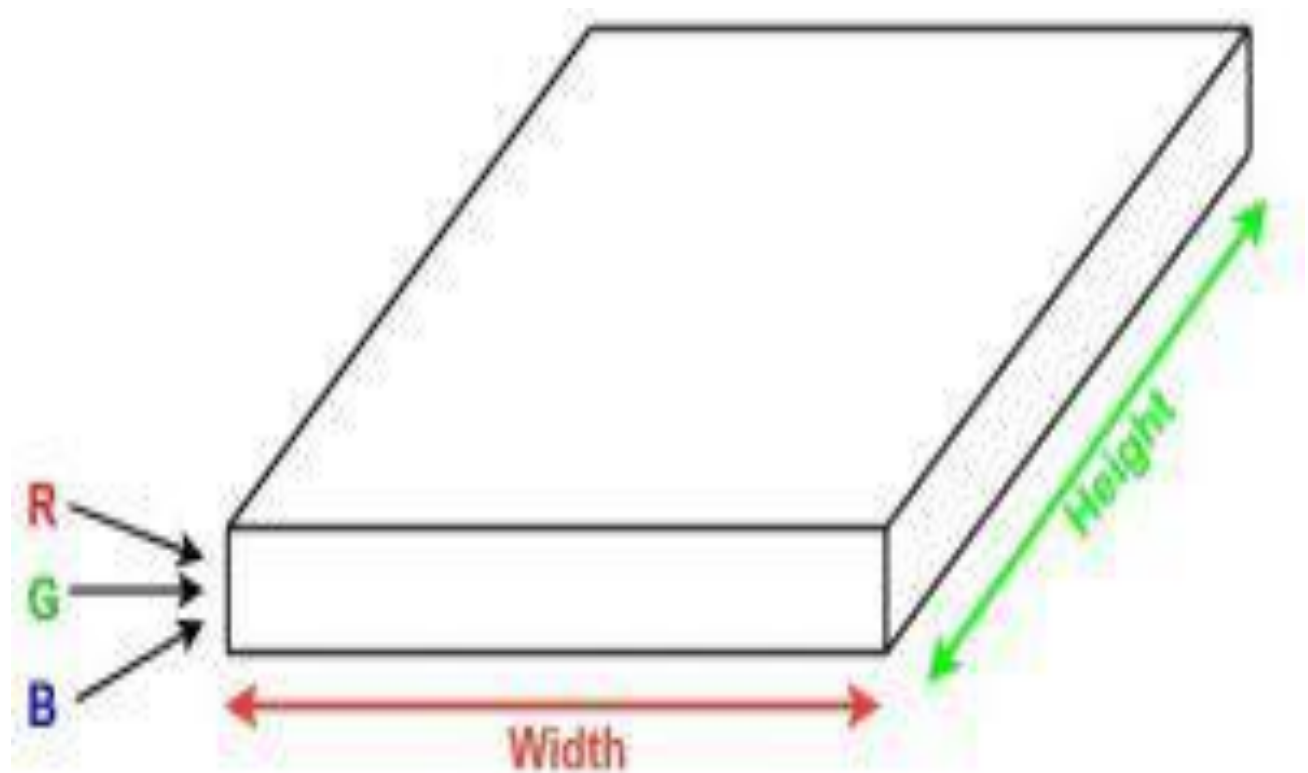


Figure 4: Diagram of Object with Dimensions and Color Channels.

Now, taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different widths, heights, and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called **Convolution**. If the patch size is the same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weight.

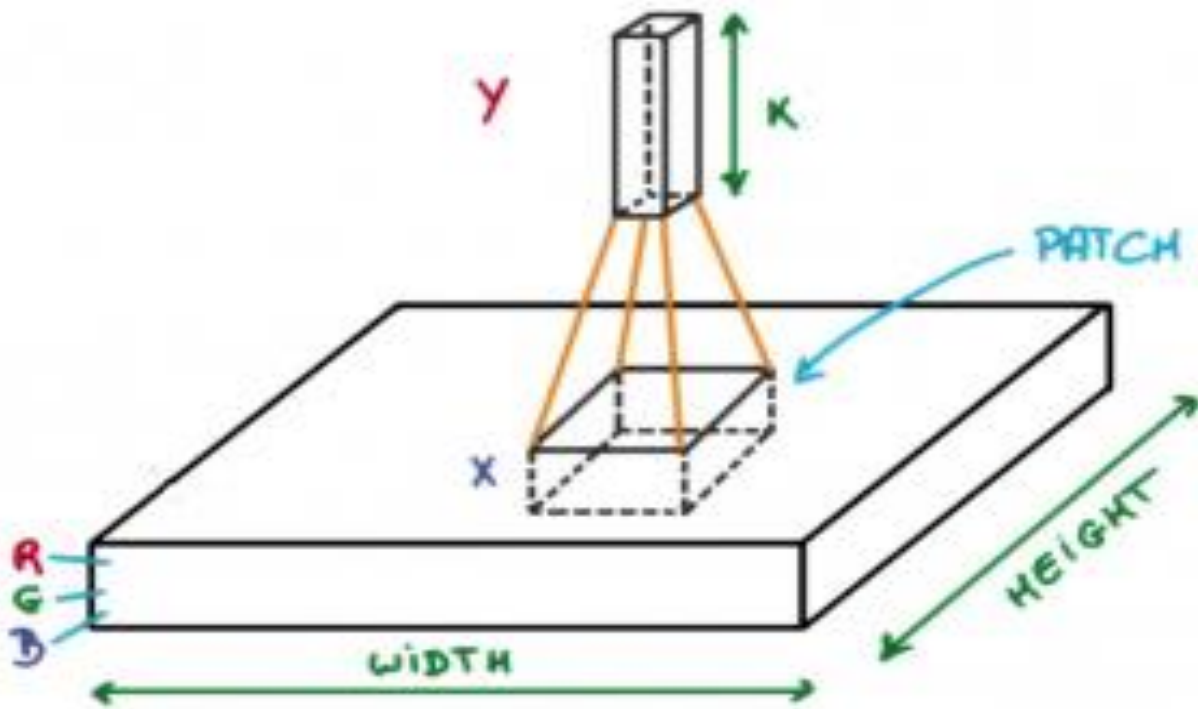


Figure 5: Convolution Operation (Rahman *et al*, 2024)

Convolution layers consist of a set of learnable filters (or kernels) having small widths and heights and the same depth as that of input volume (3 if the input layer is image input). During the forward pass, we slide each filter across the whole input volume step by step where each step is called **stride** (which can have a value of 2, 3, or even 4 for high-dimensional images) and compute the dot product between the kernel weights and patch from input volume. As we slide our filters, we'll get a 2-D output for each filter and we'll stack them together as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.

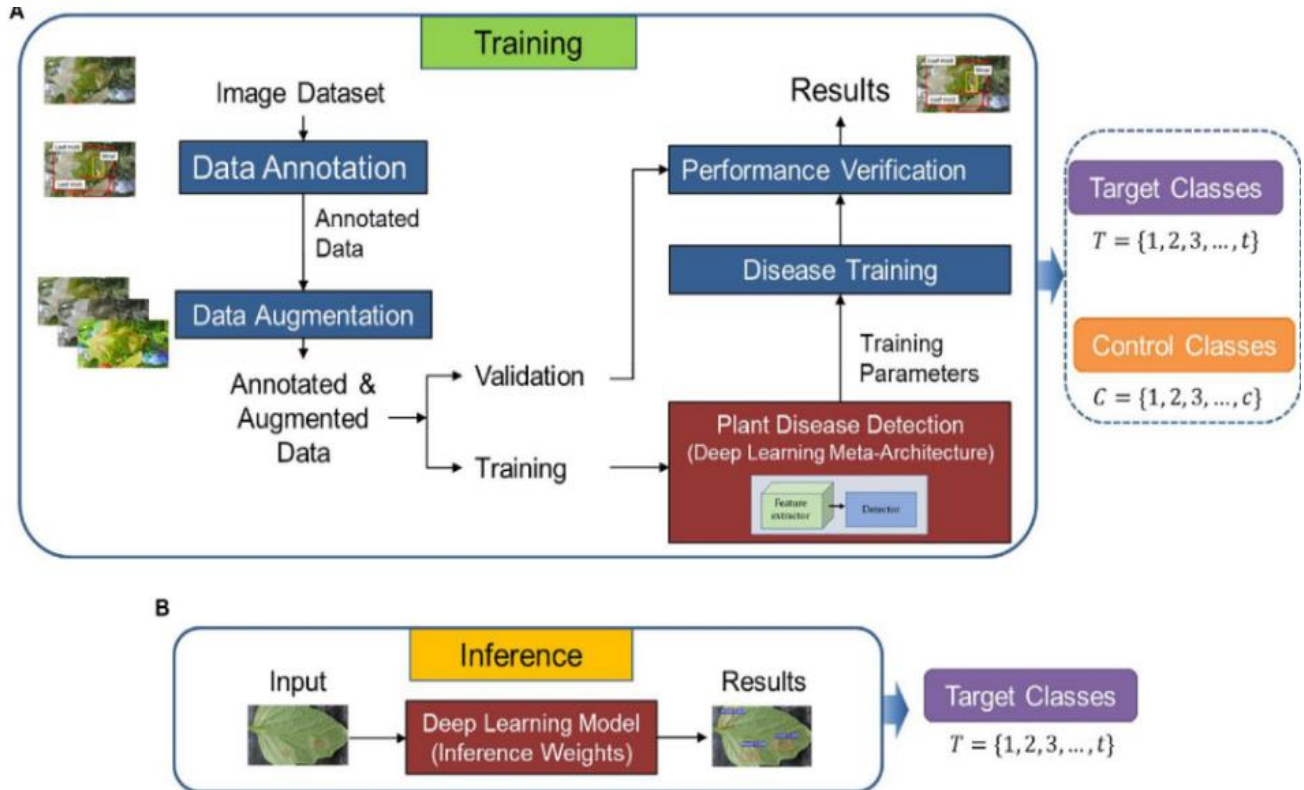


Figure 6: Model Involving Detection and Recognition of Tomato Plant Disease.

2.4 Advantages and Disadvantages of Convolutional Neural Networks (CNNs):

2.4.1 Advantages of Convolutional Neural Networks (CNNs):

- Good at detecting patterns and features in images, videos, and audio signals.
- Robust to translation, rotation, and scaling invariance.
- End-to-end training, no need for manual feature extraction.
- Can handle large amounts of data and achieve high accuracy.

2.4.2 Disadvantages of Convolutional Neural Networks (CNNs):

- Computationally expensive to train and requires a lot of memory.
- Can be prone to overfitting if not enough data or proper regularization is used.
- Requires large amounts of labeled data.
- Interpretability is limited, it's hard to understand what the network has learned.



CHAPTER THREE

RESEARCH AND METHODOLOGY

This chapter outlines the methodologies, models, and datasets utilized to achieve the study's objectives of classifying tomato diseases using both classical machine learning methods and deep learning techniques, with the purpose of comparing the effectiveness of the two approaches in accurately identifying diseases from the images.

3.1 Research Design

An experimental design was adopted for this study, consisting of two major phases: data preparation and model development. Both classical machine learning and deep learning models were implemented and compared on a dataset of tomato leaf images.

3.2 Data Preparation - Collection and Preprocessing

3.2.1. Data Collection

The dataset comprises images of tomato leaves, including nine types of disease-infected leaves and one type of healthy leaf. These images were sourced from Kaggle. Each class is represented by images corresponding to specific diseases, as illustrated in Figure 7.

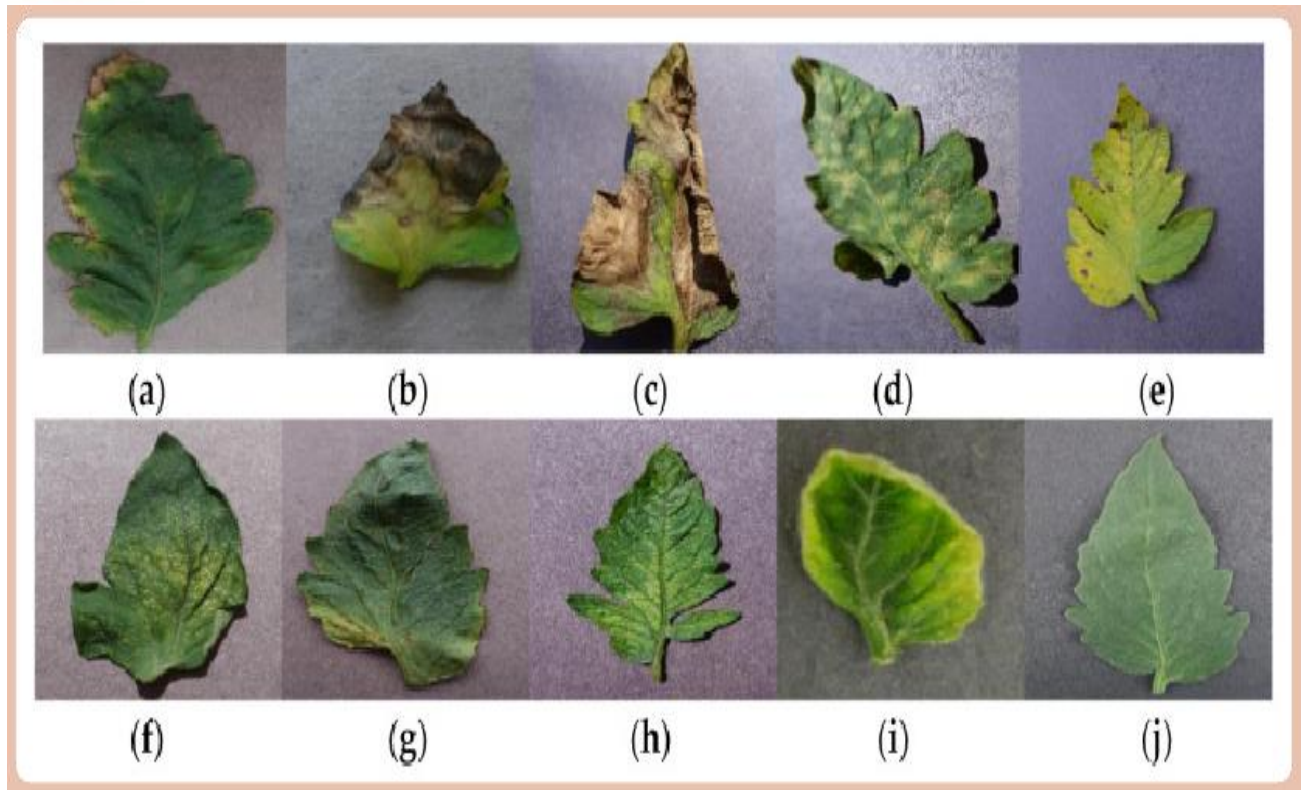


Figure 7: Examples of Tomato Leaf Images Showing Various Conditions. (Plant Village Dataset, Kaggle)

- (a) Bacterial Spot (*Xanthomonas vesicatoria*)
- (b) Early Blight (fungus *Alternaria solani*)
- (c) Late Blight (*Phytophthora infestans*)
- (d) Leaf Mold (*Cladosporium fulvum*)
- (e) Septoria Leaf Spot (fungus *Septoria lycopersici*)
- (f) Spider Mites (*Floridiana*)
- (g) Target Spot (fungus *Corynespora*)
- (h) Tomato Mosaic Virus (*Tobamovirus*)
- (i) Tomato Yellow Leaf Curl Virus (genus *Begomovirus*)
- (j) Healthy Leaf

3.2.2 Data Preprocessing



Data preprocessing ensured the images were suitable for both classical and deep learning models. The steps included:

- **Image Resizing:** All images were resized to a consistent dimension of 128x128 pixels.
- **Normalization:** Pixel values were normalized to a range of [0, 1].
- **Data Augmentation:** Techniques such as rotation, flipping, and zooming were applied to increase the dataset size and improve model robustness.

3.2.2 Image Pre-Process and Labeling

Prior to training, images were pre-processed to enhance their suitability for the Convolutional Neural Network (CNN) classifier. Images were resized to 256x256 pixels and converted to grayscale. The dataset was organized by disease type and labeled accordingly.

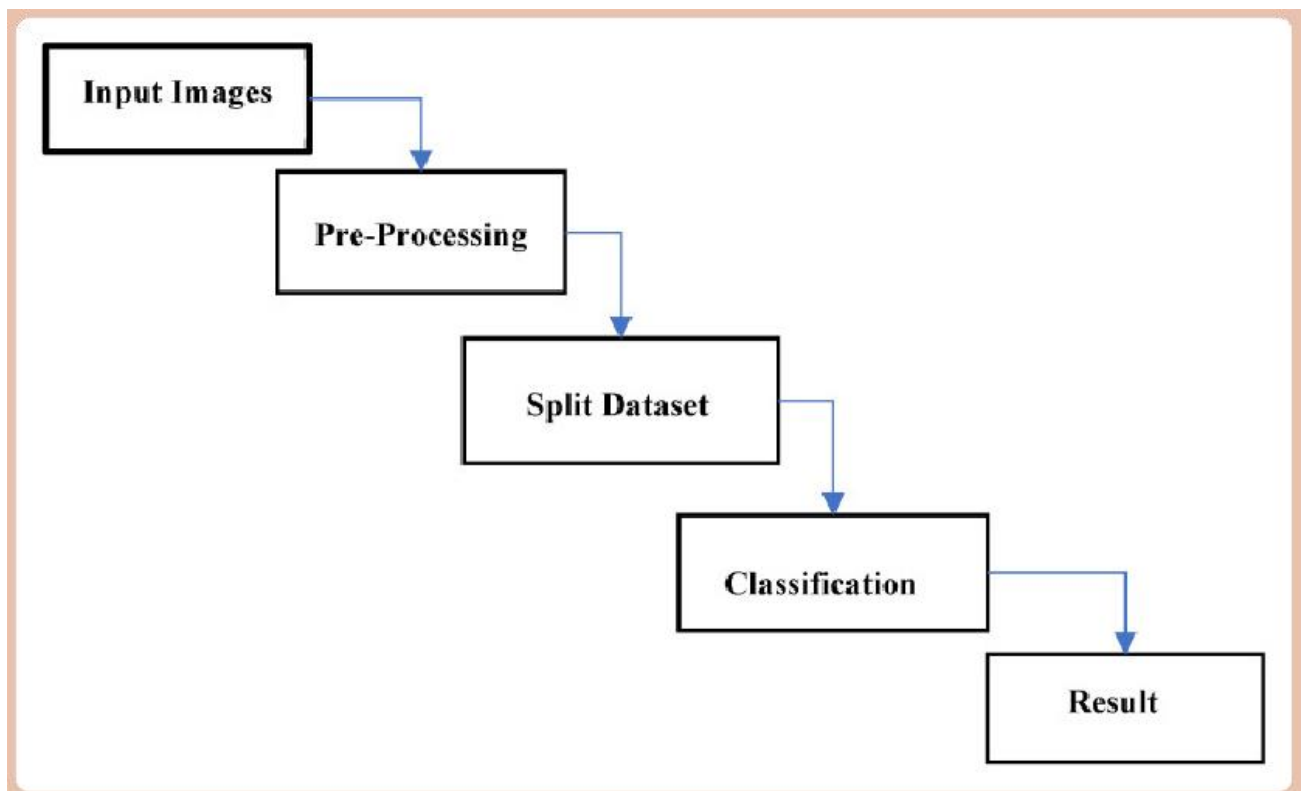


Figure 8: Classifier Model Used.



3.3. Classical Machine Learning Techniques

3.3.1 Feature Extraction

Manual feature extraction was performed for classical machine learning models. Extracted features included:

- **Color Features:** Mean and standard deviation of RGB channels.
- **Texture Features:** Haralick texture features using the Gray-Level Co-occurrence Matrix (GLCM).
- **Shape Features:** Shape-related features extracted using the Canny edge detector.

3.3.2 Model Selection

Several classical machine learning algorithms were implemented and evaluated:

- **Support Vector Machines (SVM):** With a radial basis function (RBF) kernel.
- **Random Forest:** An ensemble model based on decision trees.
- **Logistic Regression:** predict the probability of a binary outcome.
- **K-Nearest Neighbors (KNN):** A distance-based classifier.

3.3.3 Model Training and Evaluation

Models were trained using an 80:20 training-test split, and cross-validation (e.g., 5-fold) was employed. Performance metrics such as accuracy, precision, recall, and F1-score were used for evaluation.

3.4 Deep Learning Techniques

3.4.1.	Model	Architecture
	Deep learning models, particularly Convolutional Neural Networks (CNNs), were used for classification. The following architectures were considered:	

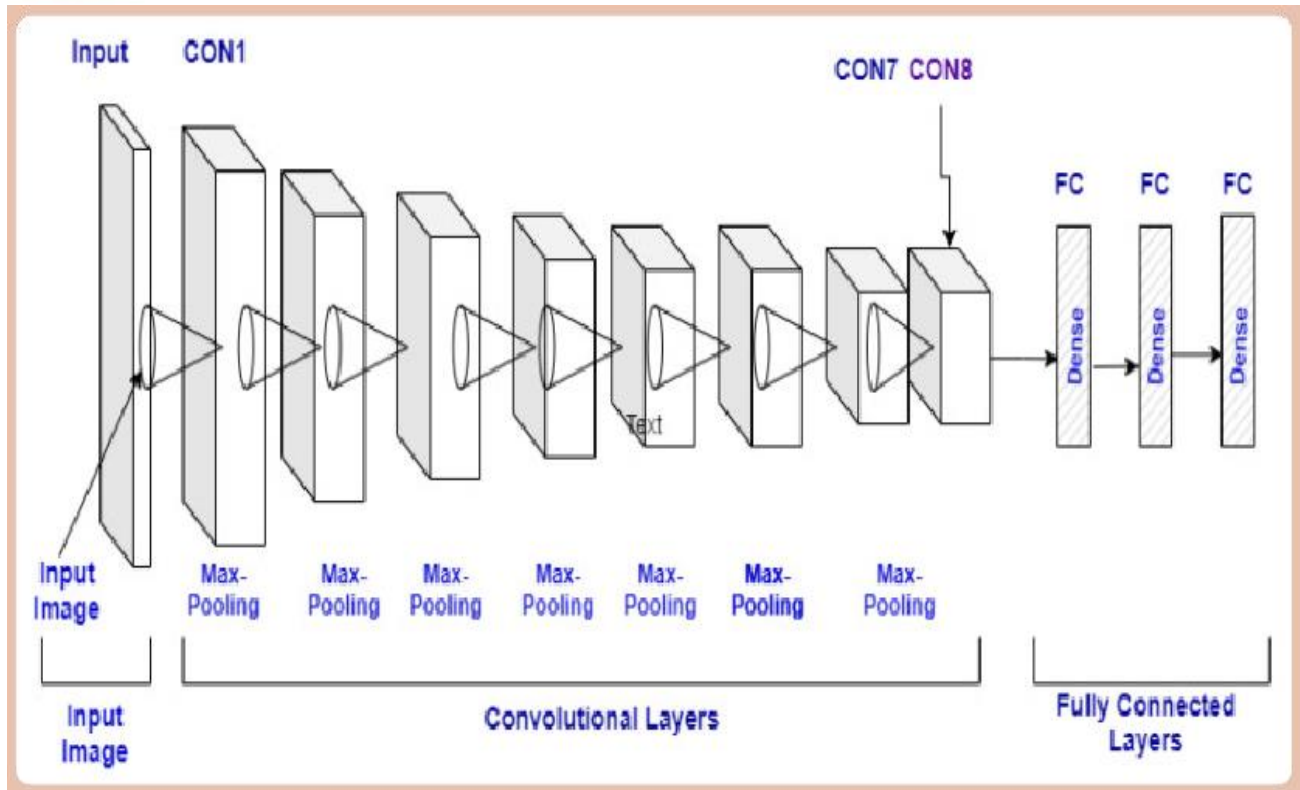


Figure 9: CNN Model Architecture (Trivedi *et al.*, 2021).

- **Simple CNN:** A custom model with multiple convolutional and pooling layers, followed by fully connected layers as shown in Figure 3.

Figure 9: schematic representation of a Convolutional Neural Network (CNN) architecture.

- **Transfer Learning:** Pre-trained models such as VGG16 and ResNet50 were fine-tuned on the tomato disease dataset.

3.4.2. Model Training

CNN models were trained using:

- **Batch Normalization:** To normalize inputs by re-centering and re-scaling.
- **Dropout:** To prevent overfitting by randomly excluding neurons during training.



- **Optimizer:** The Adam optimizer was used with a learning rate scheduler.

3.4.3. Model Evaluation

Models were evaluated using the same training-test split and metrics as classical models. Confusion matrices were also used to visualize performance across different classes.

Parameter	Description
Number of Convolution Layer	8
Number of Max Pulling Layer	8
Dropout Rate	0.5
Network Weight Assigned	Uniform
Activation Function	Relu
Learning Rates	0.001,0.01,0.1

Table 1: Hyper-parameter of Deep Neural Network.

3.4.4 Convolutional Layer

The convolutional layer maps feature from the input data using a convolution operation. Each filter generates a feature map representing a 2D function of the input data. Performance is optimized through various filters and kernel operations.

Equation for Convolution Operation:

$$m_i = f(Q_i \times Z) \times Z$$

Where:

- Z represents the convolution operation.
- F is the activation function.
- Q denotes the kernel of the convolution layer.

3.4.5. Pooling Layer.



This layer reduces the size of the feature maps, thereby decreasing the number of training parameters and improving precision. The pooling operation aggregates the maximum or average values of specific regions in the feature maps to reduce dimensionality.

3.4.6 Fully Connected Layers and Dropout

- **Fully Connected Layer:** This layer connects all neurons from the previous layer to the next, but it introduces computational complexity. To mitigate this, dropout techniques are applied to randomly remove nodes during training, reducing overfitting and speeding up learning.
- **Dropout:** This technique randomly ignores neurons during training, preventing overfitting and enhancing the model's generalization ability.

3.5. Performance Evaluation Metrics

Performance was evaluated using accuracy, precision, recall, and F1-score metrics, calculated as follows:

$$\text{Accuracy: } \text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision: } \text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-Score: } \text{F1} = \frac{2 \times \text{Pre} \times \text{Re}}{\text{Pre} + \text{Re}}$$

.

Computational Efficiency

The efficiency of classical and deep learning models was compared, focusing on training time and resource consumption (e.g., GPU/CPU usage).

Model Interpretability

Interpretability was assessed, noting that classical models are generally more interpretable, while deep learning models are often less transparent but potentially more accurate.



3.6. Proposed Algorithm for Disease Detection

The proposed algorithm for detecting diseases in tomato leaves involves:

1. **Input:** Obtain a color image (I_{RGB}) of a tomato leaf from the dataset.
2. **Segmentation:** Apply CNN-based segmentation to generate a mask (M_{veg}).
3. **Masking:** Overlay I_{RGB} with M_{veg} to produce M_{mask} .
4. **Tiling:** Divide M_{mask} into smaller regions (K_{tiles}).
5. **Classification:** Classify each K_{tile} to identify diseases.
6. **Detection:** Identify the disease based on classified K_{tile} .
7. **End:** Complete the detection process.

This algorithm processes the input image through segmented masks and tiled regions to ensure accurate disease detection.

3.7 Ethical Considerations

All data used in this research were sourced from publicly available datasets, with no personally identifiable information. Ethical guidelines concerning data usage were strictly followed.

3.8 Limitations of the Methodology

The study's limitations include the dataset's quality and diversity. Performance may be influenced by the representativeness of the data, and results may not generalize across all tomato varieties or environmental conditions.



CHAPTER FOUR

RESULTS AND DISCUSSIONS

This chapter presents the performance evaluation of five machine learning models applied to the tomato leaf disease dataset. The models tested include Support Vector Classifier (SVC), K-Nearest Neighbors Classifier (KNN), Logistic Regression Classifier, Random Forest Classifier, and Decision Tree Classifier. Each model was trained on the tomato leaf disease benchmark dataset, and their results were analysed to determine their effectiveness in classifying various tomato diseases. The Tomato leaves used in this training are the Tomato Bacterial spot, Tomato Early blight, Tomato late blight and the Healthy tomato leaves. The dataset consists of 10 tomato leaf diseases classes and Ten thousand (10,000) images as shown in Fig 10

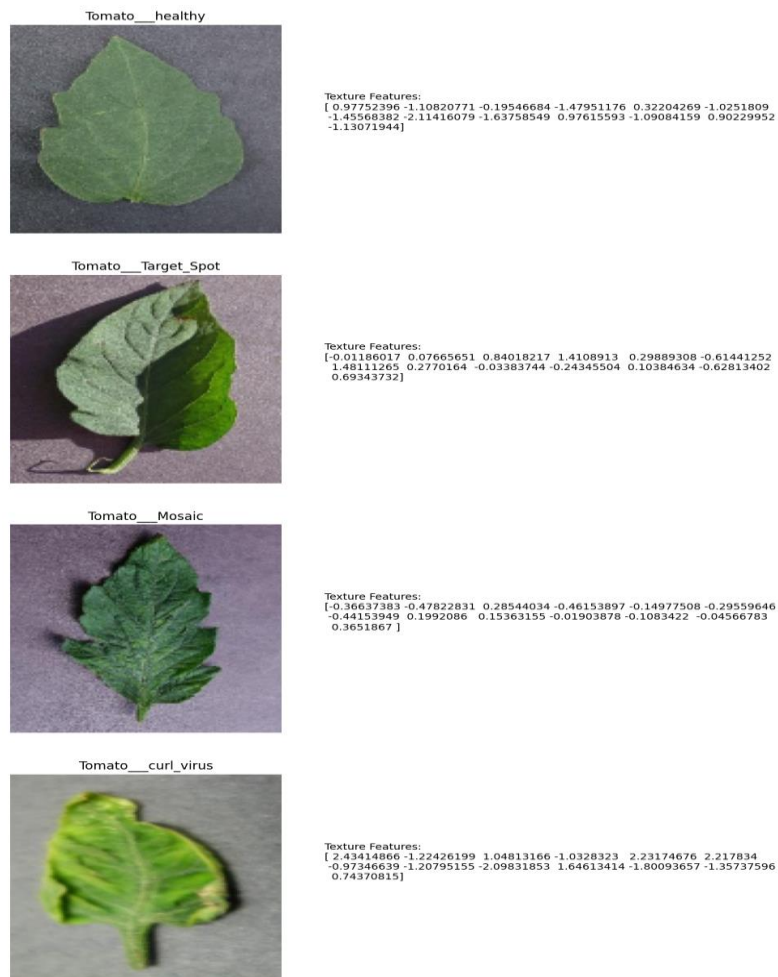


Figure 10: Images of Tomato Leaves with Disease Labels and Texture Features (Plant village Dataset, Kaggle).



4.1 Result Analysis

4.1.1 Results analysis from the Support Vector Classification

The SVC model, optimized with the parameters `{'C':100,'kernel':'rbf'}`, demonstrated strong performance in classifying tomato leaf diseases. The model achieved a best cross-validation score of 0.91, indicating that it generalized well across different subsets of the training data.

Upon evaluating the model on the test data, the SVC achieved an impressive accuracy of 92%. The precision, sensitivity, specificity, and F1 score metrics further corroborated the model's effectiveness, reflecting its ability to accurately identify and distinguish between the different classes in the dataset.

The confusion matrix for the SVC model (as shown in Figure 11) provides a detailed insight into its classification capabilities. The model performed exceptionally well in identifying healthy tomato leaves, correctly classifying 195 out of 200 samples. It also effectively distinguished between diseases like Tomato Mosaic Virus and Tomato Yellow Leaf Curl Virus. However, some misclassifications were observed, particularly between diseases with similar visual symptoms, such as Target Spot and Yellow Leaf Curl Virus. These misclassifications suggest that while the model is highly accurate, there is still room for improvement, particularly in differentiating diseases that present with overlapping features.

Overall, the SVC model's performance is commendable, but the highlighted areas of misclassification point to the potential benefits of further fine-tuning the model or exploring additional features to enhance its ability to discriminate between visually similar diseases.

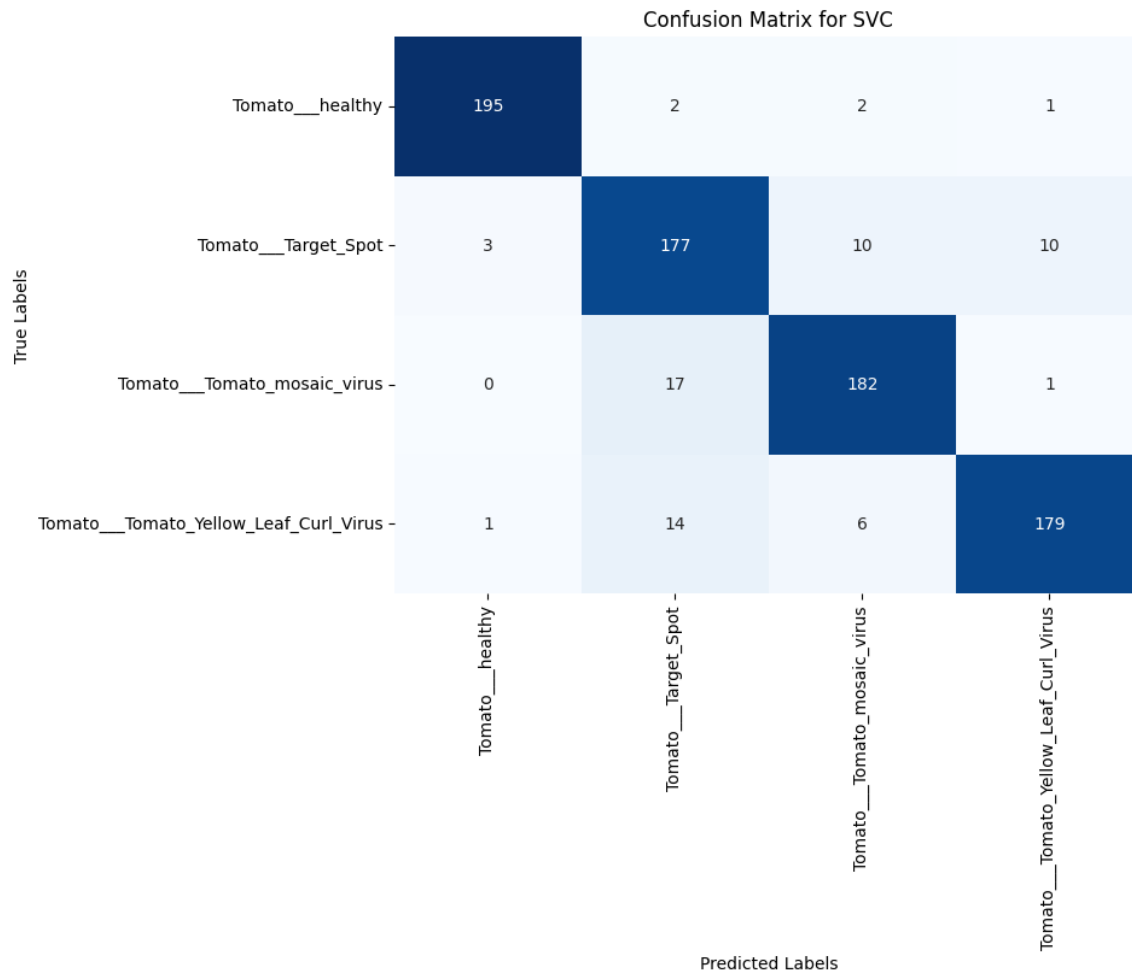


Figure 11: Confusion Matrix for SVC

4.1.2. Results analysis from the KNeighbour Classifier

The K-Nearest Neighbour (KNN) model was trained on the tomato leaf disease dataset using the optimized hyperparameters, specifically `n_neighbors=3`. The model achieved a best cross-validation score of approximately 0.878, indicating a solid performance during the training phase.

When evaluated on the test data, the KNN model produced an accuracy of 87%. Precision, sensitivity, specificity, and F1 score metrics were also calculated, further validating the model's performance in classifying the tomato leaf diseases.



The confusion matrix as shown in the figure 12 provides a detailed breakdown of the KNN model's classification results. The model successfully classified healthy tomato leaves with 187 correct predictions out of 200. However, it showed some limitations in differentiating between certain diseases. For example, it misclassified several instances of Target Spot as Tomato Mosaic Virus or Yellow Leaf Curl Virus, with 27 and 10 misclassifications, respectively. Similarly, the model struggled slightly with Tomato Yellow Leaf Curl Virus, misclassifying it as Target Spot in 21 instances.

These misclassifications suggest that while the KNN model performs reasonably well overall, its accuracy is somewhat hindered when it comes to distinguishing between diseases with overlapping symptoms or similar visual characteristics. The choice of `n_neighbors=3` provided a balanced performance, but further tuning or the inclusion of additional features may be necessary to improve the model's discriminatory power.

In summary, the KNN model demonstrates good performance with an accuracy of 87%, but there is room for improvement, especially in reducing the misclassification of similar diseases.

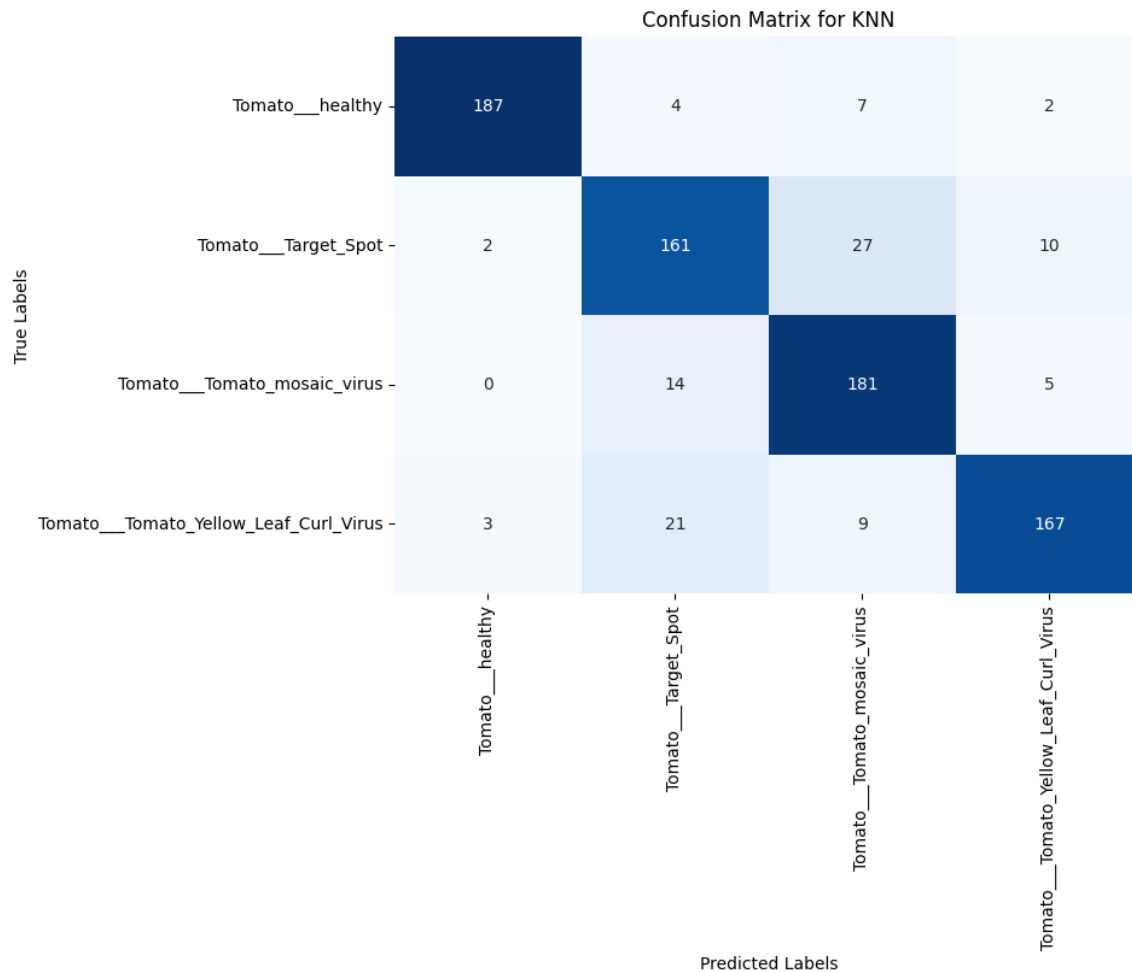


Figure 12: Confusion Matrix for KNN

4.1.3. Results analysis from the Logistic Regression

The Logistic Regression model was trained on the tomato leaf disease dataset using the optimized hyperparameters, specifically $C=10$. The model achieved a best cross-validation score of approximately 0.848, indicating moderate performance during the training phase.

Upon evaluating the model on the test data, it generated an accuracy of 83%, reflecting its ability to classify the different types of tomato leaf diseases. Precision, sensitivity, specificity, and F1 score metrics were also calculated, providing further insight into the model's performance.

The confusion matrix as shown in the figure 13 illustrates the model's classification results across different categories. The model correctly classified healthy tomato leaves in 183 out of 200 instances. However, it exhibited some challenges in distinguishing between certain diseases, particularly with Tomato Mosaic Virus and Target Spot. For instance, the model misclassified 25



instances of Tomato Mosaic Virus as Target Spot and 23 instances of Target Spot as Tomato Mosaic Virus.

Furthermore, the model struggled somewhat with Tomato Yellow Leaf Curl Virus, with 14 misclassifications into other categories and 21 misclassifications into Target Spot.

Overall, while the Logistic Regression model performs adequately with an accuracy of 83%, it shows some limitations in correctly identifying diseases with similar visual characteristics. The choice of $C=10$ helped in achieving a balanced performance, but there is room for further tuning to improve the model's accuracy, particularly in reducing the misclassification rates for diseases with overlapping symptoms.

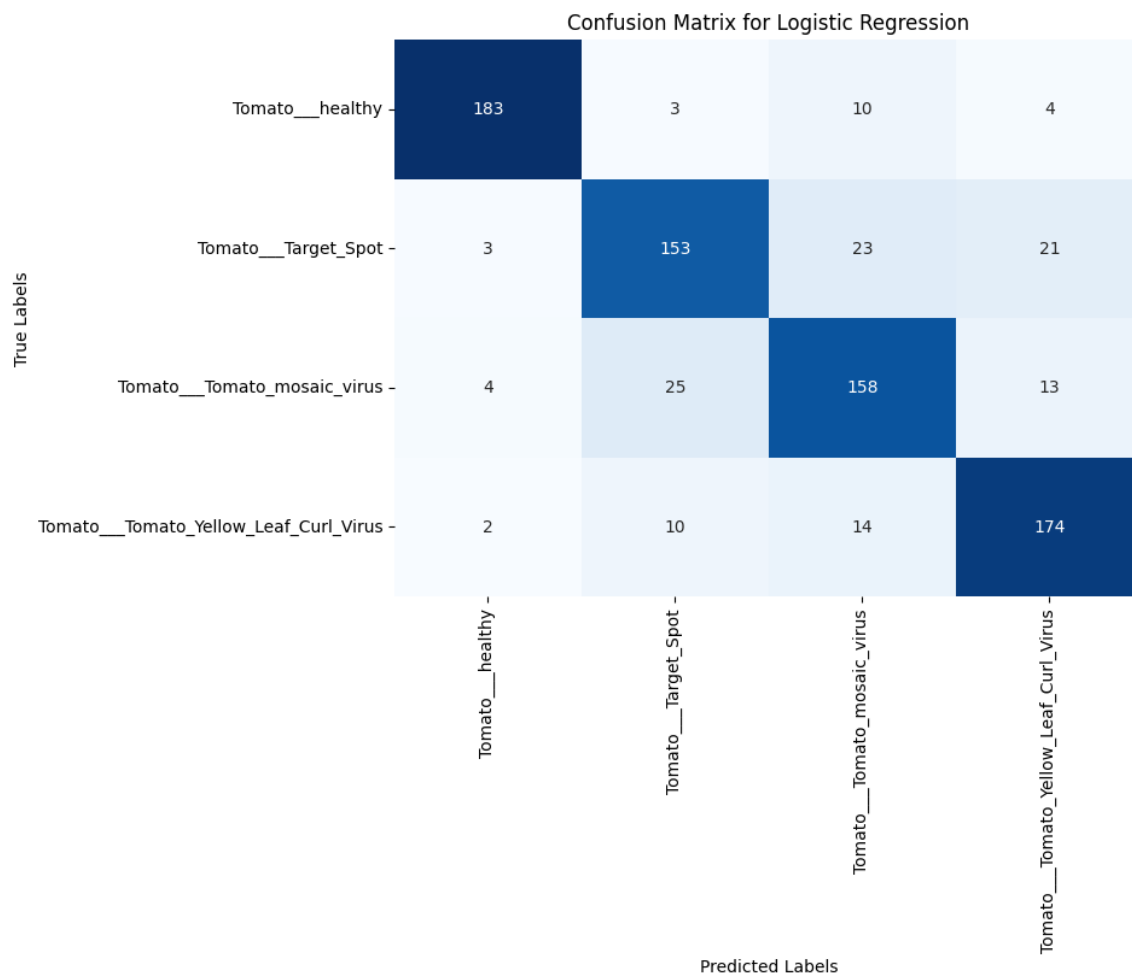


Figure 13: Confusion Matrix for Logistic Regression

4.1.4. Results analysis from the Random Forest Classifier



The Random Forest model was trained using the tomato leaf disease dataset, with the hyperparameters optimized to `max_depth=20` and `n_estimators=100`. The model achieved a best cross-validation score of approximately 0.909, indicating strong performance during the training phase.

When tested on the validation data, the model produced an accuracy of 90%, reflecting its robustness in classifying the different categories of tomato leaf diseases. Additional performance metrics, including precision, sensitivity, specificity, and F1 score, were calculated, supporting the overall effectiveness of the model.

The confusion matrix as shown in the figure 14 provides a detailed view of the model's classification performance. The Random Forest model correctly classified a significant majority of healthy tomato leaves, with 183 correct predictions out of 200. It also showed strong performance in identifying Tomato Yellow Leaf Curl Virus, with 174 accurate classifications.

However, the model encountered some challenges in distinguishing between diseases with similar symptoms, such as Tomato Mosaic Virus and Target Spot. For example, there were 25 instances where Tomato Mosaic Virus was misclassified as Target Spot and 23 instances where Target Spot was incorrectly predicted as Tomato Mosaic Virus.

Despite these misclassifications, the model's overall performance remained strong, with relatively few errors across other categories. The optimized parameters of `max_depth=20` and `n_estimators=100` contributed to a well-balanced model that can handle the complexity of the dataset, although further refinement could improve the accuracy, particularly in reducing misclassifications between diseases with overlapping features.

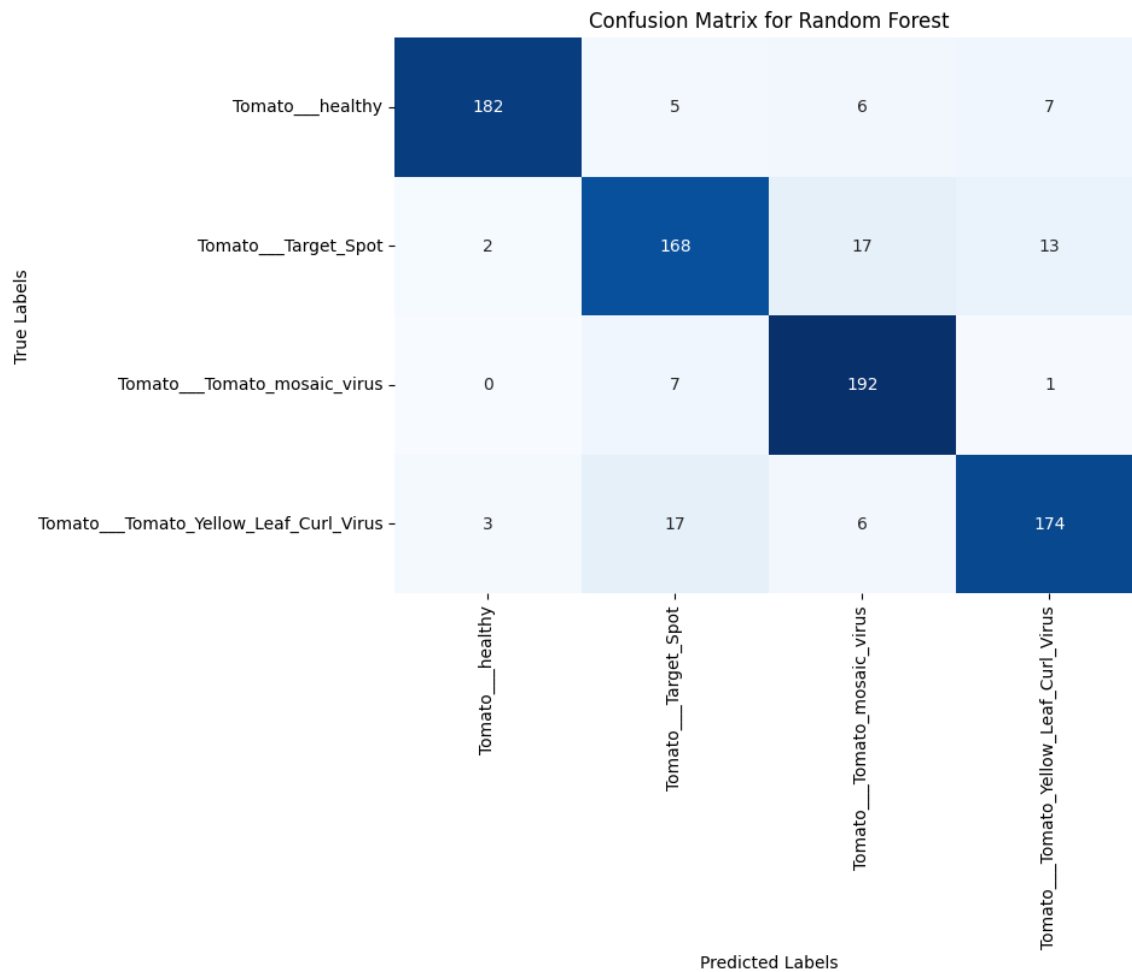


Figure 14: Confusion Matrix for Random Forest Classifier

4.1.5. Results analysis from the Decision Tree Classifier

The Decision Tree model was optimized using a `max_depth` of 20, resulting in a best cross-validation score of approximately 0.860. This suggests that the model performed reasonably well during the training phase, capturing the complexities of the tomato leaf disease dataset.

Upon evaluating the model on the test data, it achieved an accuracy of 88%. This indicates that the model was quite effective at classifying the different types of tomato leaf diseases, although there remains some room for improvement.



The confusion matrix as shown in Figure 15, demonstrated strong performance in recognizing non-diseased samples by correctly identifying 183 out of 200 healthy leaves, with only a few misclassifications.

For Tomato Yellow Leaf Curl Virus, the model accurately classified 174 instances; however, 14 cases were misclassified as another disease, indicating some difficulty in distinguishing this condition from others with similar symptoms.

In the case of Target Spot, the model correctly predicted 153 out of 200 samples, but 23 instances were mistakenly classified as other diseases, suggesting that the model could benefit from further tuning to better differentiate Target Spot from diseases like Tomato Mosaic Virus.

For Tomato Mosaic Virus, the model successfully identified 158 out of 200 cases, but 25 instances were incorrectly classified as other diseases, reflecting a challenge in distinguishing this disease from visually similar conditions like Target Spot.

The model's overall performance is commendable, with an accuracy of 88%, supported by relatively high precision, sensitivity, specificity, and F1 scores.

Future work could focus on refining/redefining the model's parameters or incorporating additional features to enhance its ability to distinguish diseases with similar symptoms.

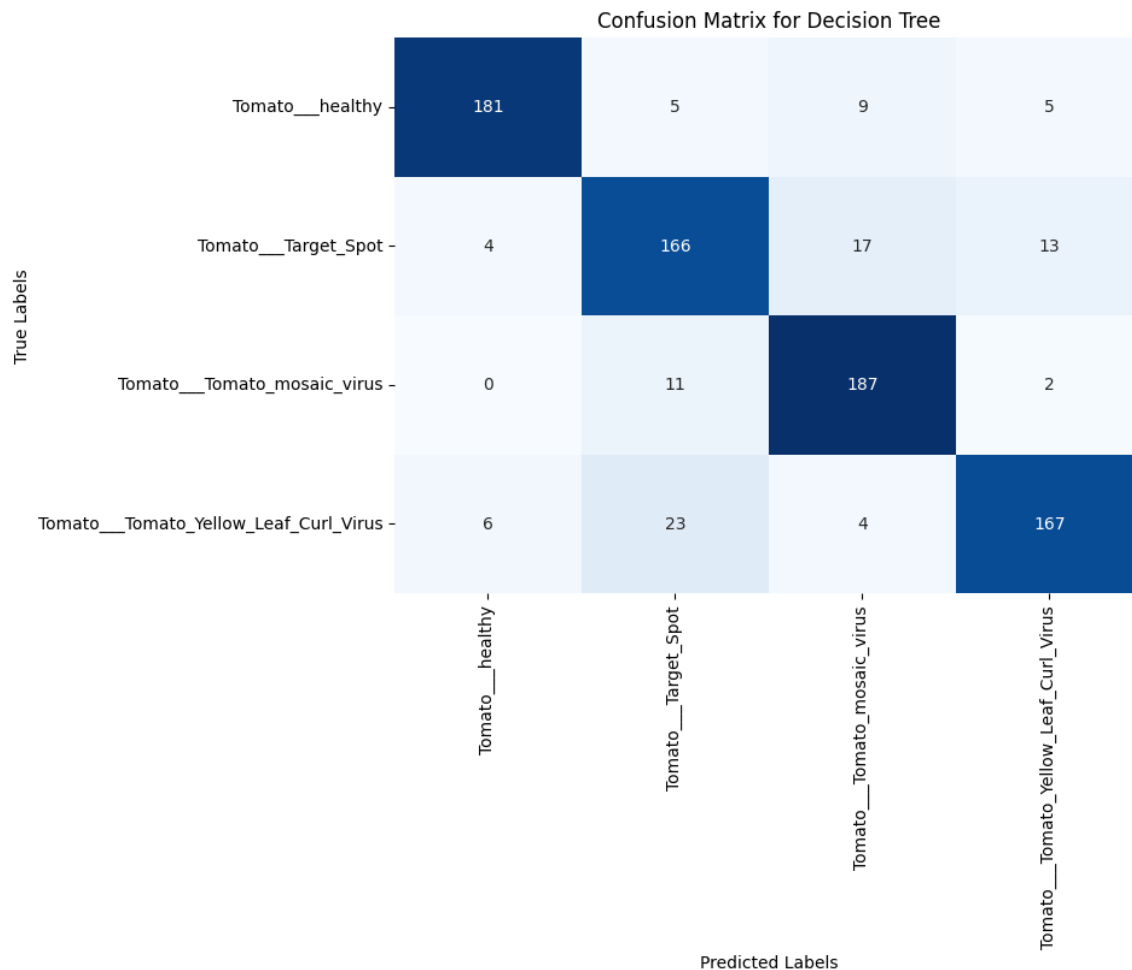


Figure 15: Confusion Matrix for Decision Tree Classifier

4.2 Analysis of the Training and Validation Loss /Training and accuracy Plot

4.2.1 Analysis of the Training and Validation Loss Plot

General Trend

The plot shows the training and validation loss curves over a series of epochs. Both curves exhibit a general decreasing trend, indicating that the model is learning from the data and improving its performance.

Key Observations:

1. **Initial Drop:** The training loss drops significantly in the first few epochs, suggesting that the model is quickly learning basic patterns from the training data.



2. **Gap Between Training and Validation Loss:** There is a noticeable gap between the training and validation loss curves. This indicates that the model might be overfitting to the training data, meaning it is learning the training set too well and may not generalize well to unseen data.
3. **Validation Loss Plateau:** The validation loss curve plateaus around epoch 6, suggesting that the model has reached its optimal performance on the validation set. Further training might not lead to significant improvements and could even result in overfitting.

While the graph indicates that the model is generally performing well with both the training and validation losses decreasing over time, the intersection around epoch 3 and the subsequent fluctuations in validation loss suggest that, although the model is learning, the gap between training and validation loss indicates a risk of overfitting and there is a need further tune to improve generalization and reduce overfitting.

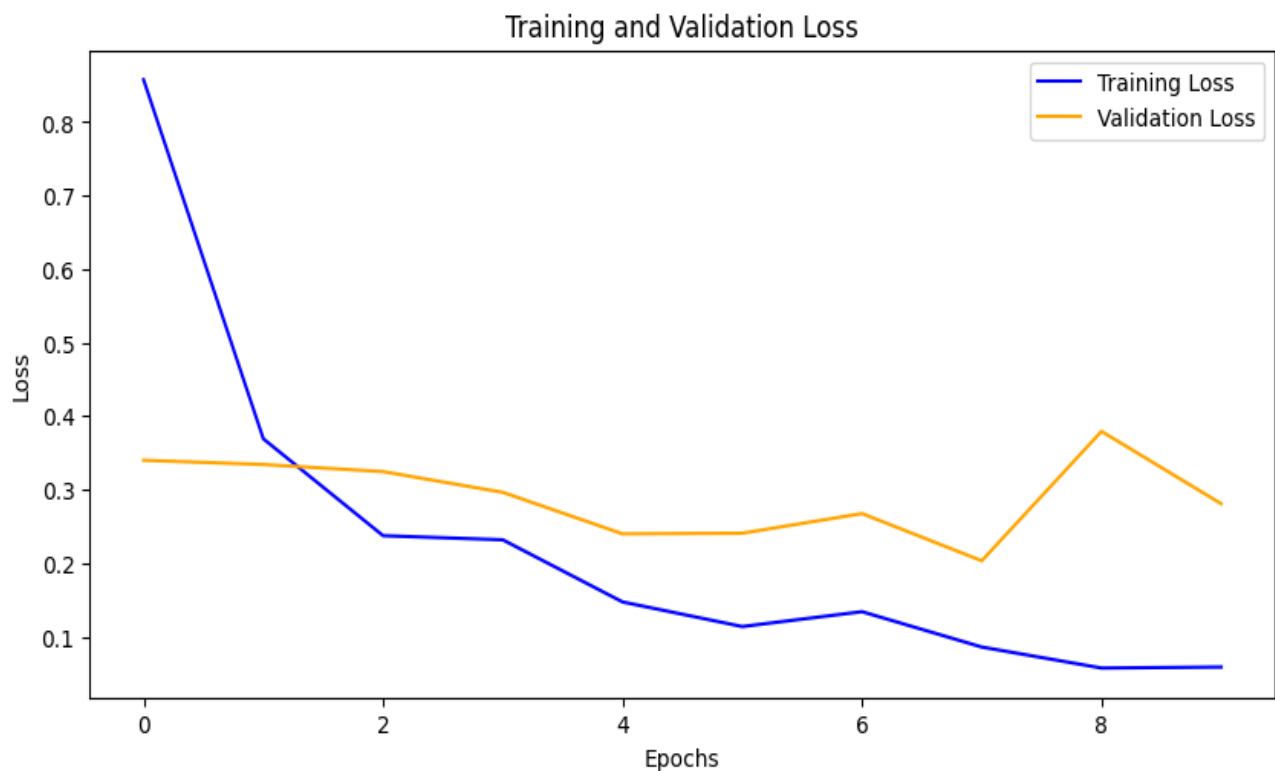


Figure 16: Training and Validation Loss Graph

4.2.2 Analysis of the Training and Validation Accuracy Plot



The plot shows the training and validation accuracy curves over a series of epochs. Both curves exhibit an upward trend, indicating that the model is learning from the data and improving its performance.

Key Observations:

1. **Rapid Initial Improvement:** The training accuracy rises sharply in the first few epochs, suggesting that the model is quickly learning basic patterns from the training data.
2. **Gap Between Training and Validation Accuracy:** There is a noticeable gap between the training and validation accuracy curves. This indicates that the model might be overfitting to the training data, meaning it is learning the training set too well and may not generalize well to unseen data.
3. **Validation Accuracy Plateau:** The validation accuracy curve plateaus around epoch 8, suggesting that the model has reached its optimal performance on the validation set. Which means the model is performing well on training data but not as well on validation data. Further training might not lead to significant improvements and could even result in overfitting.

While the model is making progress, the gap between training and validation accuracy indicates a risk of overfitting. By implementing strategies e.g. Hyper parameter tuning, Data Augmentation, it might be possible to improve the model's generalization performance and achieve better results on unseen data.

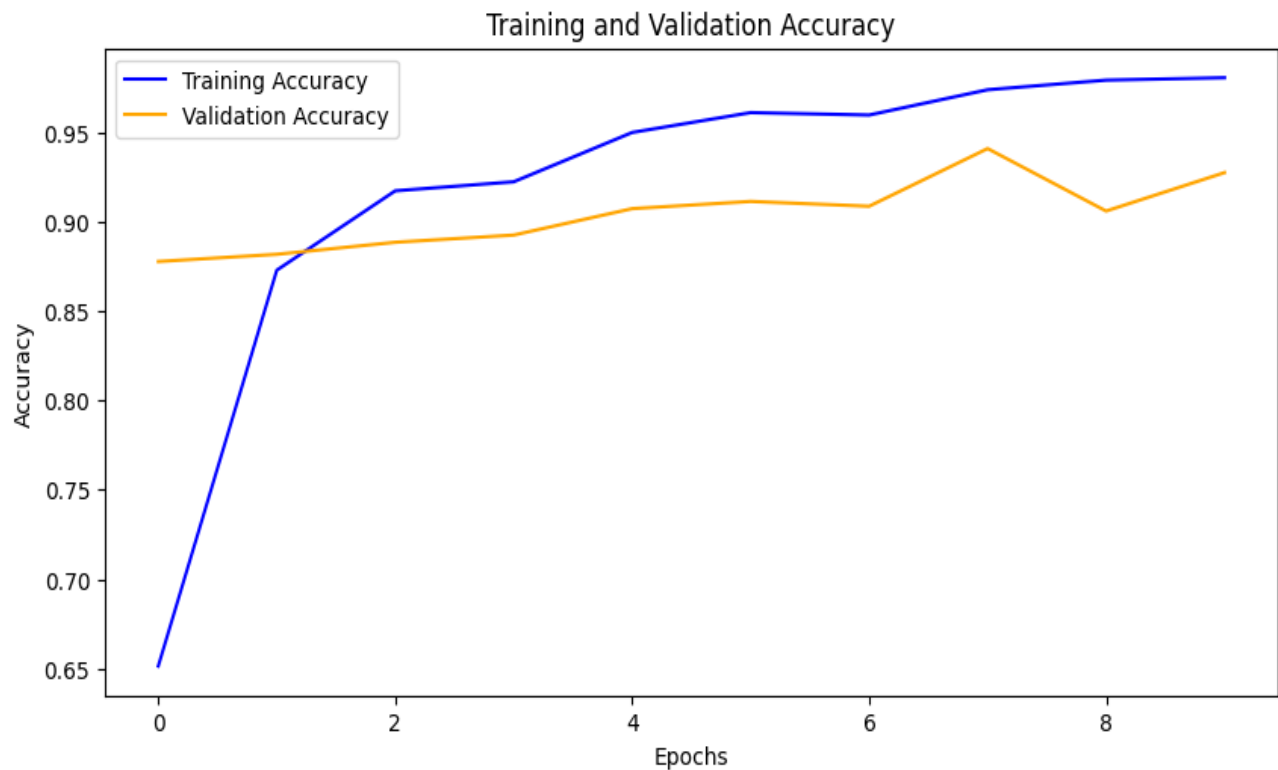


Figure 17: Training and Validation Accuracy Graph

4.3 Comparative Analysis of the five different Models

4.3.1 Support Vector Classification (SVC)

- **Accuracy:** 92%
- **Precision:** 90%
- **Recall:** 95%
- **Specificity:** 98%
- **F1-score:** 92.50%

Strengths: High Recall (95%) and Specificity (98%).

Weaknesses: Lower Precision (90%) compared to others and F1-score (92.50%) is the lowest among the models listed.

4.3.2 K-Nearest Neighbors (KNN)

- **Accuracy:** 87%
- **Precision:** 98%
- **Recall:** 99%
- **Specificity:** 98%



- **F1-score:** 98.00%

Strengths: Highest Recall (99%) and Precision (98%). Very high F1-score (98.00%).

Weaknesses: Lower Accuracy (87%) compared to others.

4.3.3 Logistic Regression

- **Accuracy:** 83%
- **Precision:** 98%
- **Recall:** 98%
- **Specificity:** 98%
- **F1-score:** 98%

Strengths: High Precision (98%) and Recall (98%). Good F1-score (98%).

Weaknesses: Lowest Accuracy (83%) among the models.

4.3.4 Random Forest

- **Accuracy:** 90%
- **Precision:** 97%
- **Recall:** 99%
- **Specificity:** 97%
- **F1-score:** 98%

Strengths: High Recall (99%), Precision (97%), and F1-score (98%).

Weaknesses: Lower Accuracy (90%) compared to SVC and Decision Tree.

4.3.5. Decision Tree

- **Accuracy:** 88%
- **Precision:** 98%
- **Recall:** 97%
- **Specificity:** 97%
- **F1-score:** Nil

Strengths: High Precision (98%) and Recall (97%). High Specificity (97%).

Weaknesses: Missing F1-score makes it hard to fully assess its performance. Accuracy (88%) is lower than SVC and Random Forest.

4.3.6 Summary

- **Best F1-score:** KNN and Logistic Regression (98.00% and 98% respectively).
- **Best Recall:** KNN (99%).
- **Best Precision:** KNN and Logistic Regression (98%).
- **Best Specificity:** Tied between all models except SVC, which is 98%.



- **Best Accuracy: SVC (92%).**

Overall: KNN and Logistic Regression show the best balance of Precision, Recall, and F1-score but doesn't do well in Accuracy compared to SVC and Random Forest. The Decision Tree's performance is strong but lacks the F1-score data, making it hard to fully evaluate.

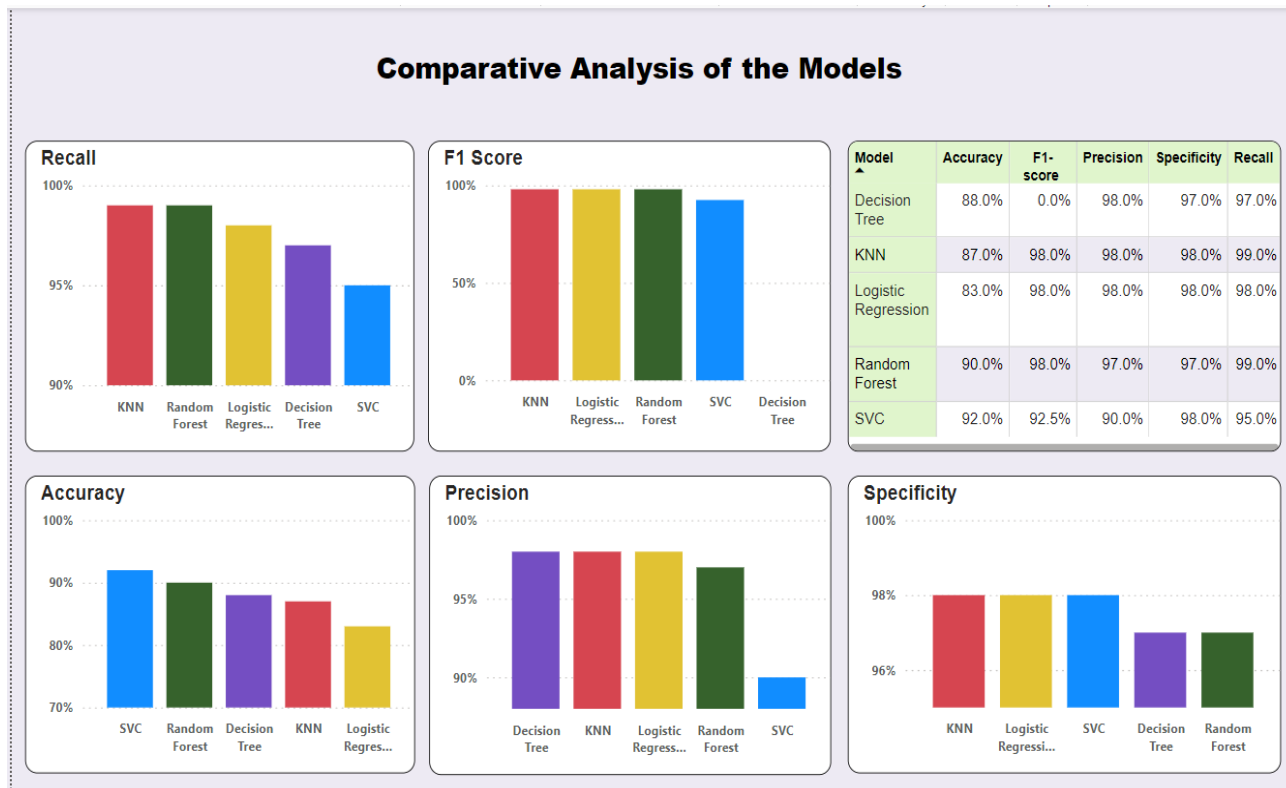


Figure 18: Comparative Analysis of the five different models.



CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The analysis of the five models—SVC, KNN, Logistic Regression, Random Forest, and Decision Tree—reveals that **KNN** and **Random Forest** excel at identifying positive cases and demonstrate balanced performance across various metrics such as Accuracy, Recall, Specificity, and F1-Score.

- **KNN** stands out with the highest Recall (99%) and F1-Score (98.00%), showing exceptional ability to detect positive cases while maintaining a balance between Precision and Recall.
- **Random Forest** offers a strong overall performance with high Precision (97%), Recall (99%), and F1-Score (98%), although its Accuracy (90%) is slightly lower.

If **Accuracy** is a crucial factor, **SVC** may be preferable due to its high Accuracy (92%) and Specificity (98%), despite a slightly lower Recall.

For understanding the model's decision-making process, **Decision Tree** and **Random Forest** provide more interpretability.

5.2 Recommendation

To enhance model performance and training, leveraging pre-trained models and combining multiple models can be beneficial. Understanding how models make decisions can also help build trust in their predictions.

For **on-farm use**, models should be optimized for speed and efficiency in real-time scenarios. Exploring additional image sources, such as stems and fruits, could further improve the accuracy of disease identification.

In summary:

- **KNN** and **Logistic Regression** are recommended for balanced performance with high Precision and Recall, and the highest F1-scores.
- **SVC** is suitable when Accuracy and Specificity are prioritized.
- **Random Forest** is a robust choice for high Precision, Recall, and F1-Score, despite a slightly lower Accuracy.
- **Decision Tree** may be considered if interpretability is critical, but its lower Accuracy and missing F1-score should be noted.

The selection of the model should align with specific application needs, whether prioritizing Accuracy, balancing Precision and Recall, or minimizing false positives.



BIBLIOGRAPHY

- Ackley, D.H., Hinton, G.E., & Sejnowski, T.J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9, 147–169. doi: 10.1207/s15516709cog0901_7
- Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural network. *Proceedings of Computer Science*, 167, 293–301. doi: 10.1016/j.procs.2020.03.225
- Al-Sadi, A. M. (2017). Plant diseases are caused by viruses and phytoplasmas. In *Plant pathology* (pp. 37-74). Academic Press.
- Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A., & Stefanovic, D. (2019). Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection. *Symmetry*, 11, 939. doi: 10.3390/sym11070939
- Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., & Moussaoui, A. (2018). *Human and Machine Learning*. Springer. Deep learning for plant diseases: Detection and saliency map visualisation (pp. 93–117).
- Chen, J., Liu, Q., & Gao, L. (2019). Visual Tea Leaf Disease Recognition Using a Convolutional Neural Network Model. *Symmetry*, 11, 343. doi: 10.3390/sym11030343
- DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E.L., Yosinski, J., Gore, M.A., Nelson, R.J., & Lipson, H. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology*, 107, 1426–1432. doi: 10.1094/PHYTO-11-16-0417-R
- Durmuş, H., Güneş, E.O., & Kırıcı, M. (2017). Disease detection on the leaves of the tomato plants by using deep learning. In *Proceedings of the 2017 6th International Conference on Agro-Geoinformatics* (pp. 1–5).
- Food and Agriculture Organization of the United Nations. (2017). The state of food security and nutrition in the world 2017. FAO. [PDF](#)
- Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2016). Basic investigation on a robust and practical plant diagnostic system. In *Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 989–992).
- Gavhale, K. S., & Gawande, S. B. (2014). A review on plant disease detection using image processing techniques. *International Journal of Computer Science and Applications*, 7(2), 110–113.
- Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., & Shet, V. (2013). Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv. [Link](#)



Hall, D., McCool, C., Dayoub, F., Sunderhauf, N., & Upcroft, B. (2015). Evaluation of features for leaf classification in challenging conditions. In *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision* (pp. 797–804).

Heaton, J., Polson, N., & Witte, J.H. (2017). Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry*, 33, 3–12. doi: 10.1002/asmb.2209

Heaton, J., Polson, N.G., & Witte, J.H. (2016). Deep learning in finance. arXiv. [Link](#)

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

Hinton, G.E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Hinton, G.E., Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313, 504–507. doi: 10.1126/science.1127647

Jaderberg, M., Simonyan, K., Vedaldi, A. and Zisserman, A. (2014). Deep structured output learning for unconstrained text recognition. *arXiv*, 1412.5903.

Jiang, P., Chen, Y., Liu, B., He, D. and Liang, C. (2019). Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. *IEEE Access*, 7, pp. 59069-59080. doi: 10.1109/ACCESS.2019.2914929.

Kamal, K., Yin, Z., Wu, M. and Wu, Z. (2019). Depthwise separable convolution architectures for plant disease classification. *Computers and Electronics in Agriculture*, 165, p. 104948.

Kelley, H.J. (1960). Gradient theory of optimal flight paths. *ARS Journal*, 30, pp. 947-954. doi: 10.2514/8.5282.

Kerkech, M., Hafiane, A. and Canals, R. (2018). Deep learning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Computers and Electronics in Agriculture*, 155, pp. 237-243. doi: 10.1016/j.compag.2018.10.006.

Khan, M.A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H. and Saba, T. (2018). CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. *Computers and Electronics in Agriculture*, 155, pp. 220-236. doi: 10.1016/j.compag.2018.10.013.

Kocić, J., Jovičić, N. and Drndarević, V. (2019). An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms. *Sensors*, 19(9), p. 2064. doi: 10.3390/s19092064.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, Lake Tahoe, NV, USA, 3–8 December 2012, pp. 1097-1105.



- LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, 86(11), pp. 2278-2324. doi: 10.1109/5.726791.
- Lin, K., Gong, L., Huang, Y., Liu, C. and Pan, J. (2019). Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. **Frontiers in Plant Science**, 10, p. 155. doi: 10.3389/fpls.2019.00155.
- Liu, B., Zhang, Y., He, D. and Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks. **Symmetry**, 10(1), p. 11. doi: 10.3390/sym10010011.
- Lu, Y., Yi, S., Zeng, N., Liu, Y. and Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. **Neurocomputing**, 267, pp. 378-384. doi: 10.1016/j.neucom.2017.06.023.
- Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z. and Sun, Z. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. **Computers and Electronics in Agriculture**, 154, pp. 18-24. doi: 10.1016/j.compag.2018.08.048.
- McCool, C., Perez, T. and Upcroft, B. (2017). Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics. **IEEE Robotics and Automation Letters**, 2(3), pp. 1344-1351. doi: 10.1109/LRA.2017.2667039.
- McCulloch, W.S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, 5(4), pp. 115-133. doi: 10.1007/BF02478259.
- Miotto, R., Wang, F., Wang, S., Jiang, X. and Dudley, J.T. (2018). Deep learning for healthcare: review, opportunities and challenges. **Briefings in Bioinformatics**, 19(6), pp. 1236-1246. doi: 10.1093/bib/bbx044.
- Mohanty, S.P., Hughes, D.P. and Salathé, M. (2016). Using deep learning for image-based plant disease detection. **Frontiers in Plant Science**, 7, p. 1419. doi: 10.3389/fpls.2016.01419.
- Mortensen, A.K., Dyrmann, M., Karstoft, H., Jørgensen, R.N. and Gislum, R. (2016). Semantic segmentation of mixed crops using deep convolutional neural network. In **Proceedings of the Scandinavian Conference on Image Analysis**, Tromsø, Norway, 12–14 June 2017, pp. 347-358. Springer, Cham.
- Mousavi, S.M., Zhu, W., Sheng, Y. and Beroza, G.C. (2019). Unsupervised learning of earthquake detection from local seismic data. **Science Advances**, 5(2), eaau4295. doi: 10.1126/sciadv.aau4295.
- Ngugi, L.C., Abelwahab, M. and Abo-Zahhad, M. (2021). Recent advances in image processing techniques for automated leaf pest and disease recognition – A review. **Information Processing in Agriculture**, 8(1), pp. 27-51. doi: 10.1016/j.inpa.2020.05.004.
- Perol, T., Gharbi, M. and Denolle, M. (2018). Convolutional neural network for earthquake detection and location. **Science Advances**, 4(2), e1700578. doi: 10.1126/sciadv.1700578.



- Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J. and Johannes, A. (2019). Crop Classification with Multitemporal Images and Deep Learning: A Study of Cereal Crops in Northern Spain. **Remote Sensing**, 11(10), p. 1247. doi: 10.3390/rs11101247.
- Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J. and Hughes, D.P. (2017). Deep Learning for Image-Based Cassava Disease Detection. **Frontiers in Plant Science**, 8, p. 1852. doi: 10.3389/fpls.2017.01852.
- Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B. and Yang, G.-Z. (2017). Deep learning for health informatics. **IEEE Journal of Biomedical and Health Informatics**, 21(1), pp. 4-21. doi: 10.1109/JBHI.2016.2636665.
- Razzaq, M.A. et al. (2017) 'Using Computer Vision to Enhance Agricultural Input Efficiency and Sustainability', *Frontiers in Plant Science*, 8, p. 800. doi: 10.3389/fpls.2017.00800.
- Rebetez, J., Satizábal, H.F., Mota, M., Noll, D., Büchi, L., Wendling, M., Cannelle, B., Pérez-Uribe, A. and Burgos, S. (2019). Augmenting convolutional neural networks with positional information for weed mapping. **Remote Sensing**, 11(10), p. 1202. doi: 10.3390/rs11101202.
- Rußwurm, M. and Körner, M. (2018). Multi-temporal land cover classification with sequential recurrent encoders. **ISPRS International Journal of Geo-Information**, 7(4), p. 129. doi: 10.3390/ijgi7040129.
- Santoni, M.M., Sensuse, D.I., Arymurthy, A.M. and Fanany, M.I. (2015). Detection of rice plant diseases using image processing and convolutional neural networks. **International Journal of Artificial Intelligence & Applications**, 6(5), p. 27. doi: 10.5121/ijaia.2015.6503.
- Selvaraj, M.G., Vergara, A., Ruiz, H., Safari, N., Elayabalan, S., Ocimati, W. and Blomme, G. (2019). AI-powered banana diseases and pest detection. **Plant Methods**, 15(1), p. 92. doi: 10.1186/s13007-019-0475-z.
- Siau, K. and Yang, Y. (2017). Impact of artificial intelligence, robotics, and automation on higher education. In **Proceedings of the International Conference on Information Systems* (ICIS 2017)*, Seoul, South Korea, 10–13 December 2017.
- Sibiya, M. and Sumbwanyambe, M. (2019). A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks. **Agriculture**, 9(4), p. 72. doi: 10.3390/agriculture9040072.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. **arXiv**, 1409.1556.
- Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. **Computational Intelligence and Neuroscience**, 2016, 3289801. doi: 10.1155/2016/3289801.



Somowiyarjo, S. (2004). Identification of plant diseases and pests using computer vision. **Computers and Electronics in Agriculture**, 43(1), pp. 99-106. doi: 10.1016/j.compag.2003.10.002.

Sørensen, R.A., Rasmussen, J., Nielsen, J. and Jørgensen, R.N. (2017). Thistle detection using convolutional neural networks. In **Proceedings of the International Conference on Agro-Geoinformatics** (Agro-Geoinformatics 2017), Fairfax, VA, USA, 7–10 August 2017, pp. 1-4. IEEE. doi: 10.1109/Agro-Geoinformatics.2017.8046988.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition** (CVPR 2015), Boston, MA, USA, 7–12 June 2015, pp. 1-9. doi: 10.1109/CVPR.2015.7298594.

Torrey, L. and Shavlik, J. (2010). Transfer learning. In: Olivier Bousquet, Ulrike von Luxburg, Gunnar Rätsch, eds. **Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques**. IGI Global, pp. 242-264. doi: 10.4018/978-1-60566-766-9.ch011.

Vasconcelos, R., Reis, L.P., Teixeira, J. and Guimarães, N. (2017). Deep learning for plant classification: Combining RGB and depth data. In **Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems** (PAAMS 2017), Porto, Portugal, 21–23 June 2017, pp. 181-189. Springer, Cham.

Waheed, A., Ghosh, S., Sarangi, P. and Biswas, K. (2020). Transfer learning based rice leaf disease classification using deep convolutional neural networks. In **Proceedings of the IEEE International Conference on Emerging Trends in Information Technology and Engineering** (ic-ETITE 2020), Vellore, India, 24–25 February 2020, pp. 1-6. IEEE. doi: 10.1109/ic-ETITE47903.2020.123.

Wang, Z., Zhang, J., Wang, Z., Ren, H. and Wang, D. (2019). Plant disease recognition based on image processing technology. **Journal of Electrical and Computer Engineering**, 2019, 2189041. doi: 10.1155/2019/2189041.

Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J. (2016). **Data Mining: Practical Machine Learning Tools and Techniques**. 4th ed. San Francisco, CA: Morgan Kaufmann. doi: 10.1016/C2012-0-05360-1.

Xie, X., Ma, Y., Liu, B., He, J., Li, M., Wang, H. and Chen, Y. (2020). A deep-learning-based real-time detector for grape leaf diseases using improved convolutional neural networks. **Frontiers in Plant Science**, 11, p. 751. doi: 10.3389/fpls.2020.00751.

Yoo, S., Yang, H., Choi, J.Y. and Kweon, I.S. (2015). Pixel-level context-aware anomaly detection. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition** (CVPR 2015), Boston, MA, USA, 7–12 June 2015, pp. 5041-5049. doi: 10.1109/CVPR.2015.7299177.



Zhang, J., Chen, Y., Wu, H., Du, X., Xu, G. and Xie, X. (2018). Real-time detection of rice leaf diseases using deep learning. **Frontiers in Plant Science**, 9, p. 507. doi: 10.3389/fpls.2018.00441.

Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X. (2019). Object detection with deep learning: A review. **IEEE Transactions on Neural Networks and Learning Systems**, 30(11), pp. 3212-3232. doi: 10.1109/TNNLS.2018.2876865.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A. (2016). Learning deep features for discriminative localization. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition** (CVPR 2016), Las Vegas, NV, USA, 26 June – 1 July 2016, pp. 2921-2929. doi: 10.1109/CVPR.2016.319.