Dissertation on
Plant Disease Identification using Deep Learning Technique

Thesis submitted towards partial fulfilment of the requirements
for the degree of

Master of Technology in IT (Courseware Engineering)

Submitted by
Subarna Das

EXAMINATION ROLL NO.: M4CWE24006
UNIVERSITY REGISTRATION NO.: 163776 of 2022-23

Under the guidance of
Prof. Dr. Matangini Chattopadhyay

School of Education Technology
Jadavpur University

Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata-700032
India

2024

M.Tech. IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled **"Plant Disease Identification using Deep Learning Technique"** is a bonafide work carried out by Subarna Das under our supervision and guidance for partial fulfillment of the requirements for the degree of Master of Technology in IT (Courseware Engineering) in School of Education Technology, during the academic session 2023-2024.

____

SUPERVISOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

____

DIRECTOR
School of Education Technology
Jadavpur University,
Kolkata-700 032

____

DEAN - FISLM
Jadavpur University,
Kolkata-700 032

M.Tech. IT (Courseware Engineering)
Course affiliated to
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

<u>CERTIFICATE OF APPROVAL</u> **

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactory to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the thesis only for purpose for which it has been submitted.

Committee of final examination
for evaluation of Thesis

** Only in case the thesis is approved.

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains literature survey and original research work by the undersigned candidate, as part of her **Master of Technology in IT (Courseware Engineering)** studies.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by this rule and conduct, I have fully cited and referenced all materials and results that are not original to this work.

NAME: Subarna Das

EXAMINATION ROLL NUMBER: M4CWE24006

THESIS TITLE: Plant Disease Identification using Deep Learning Technique

SIGNATURE: DATE:

Acknowledgement

I hereby take this opportunity to express my heartfelt gratitude to my supervisor, **Prof. Dr. Matangini Chattopadhyay, Director of the School of Education Technology** for her invaluable guidance, support, constructive criticism, and inspiring advice. I am highly indebted to madam for her guidance. Her illuminating perspectives on various issues related to this dissertation have been instrumental in shaping my work. I could not have a better mentor or advisor other than her for my research work.

I would also like to extend my sincere appreciation to **Dr. Saswati Mukherjee**, for her constant motivation and advice. I would also like to thank **Mr. Joydeep Mukherjee** for his support during my entire course of work.

I am deeply thankful for the academic resources provided by Jadavpur University during my entire period of study at the School of Education Technology. My gratitude goes to all the faculty members, staff, lab assistants, fellow research scholars, and my classmates who were always ready to help and offer suggestions whenever needed.

Lastly, I am profoundly grateful to my parents, sister, family members, friends, and well-wishers. Their unwavering faith in me and their support, encouragement, and enthusiasm have been a source of strength throughout my academic career, especially during my project work.

Subarna Das
Examination Roll No. M4CWE24006
Univ. Registration No. 163776 of 2022-23
M.Tech IT(Courseware Engineering)
School of Education Technology
Jadavpur University, Kolkata - 700032

DEDICATED TO,

My Parents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ML: Machine Learning

DL: Deep Learning

RF: Random Forest

KNN: K-Nearest Neighbor

SVC: Support Vector Classifier

GAC: Geodesic Active Contour

CNN: Convolution Neural Network

PCA: Principal Component Analysis

SMOTE: Synthetic Minority Over-sampling Technique

VGG: Visual Geometry Groups

RNNs: Recurrent Neural Networks

CV: Computer Vision

GA: Genetic Algorithm

KPCA: Kernel Principal Component Analysis

SCA: Sine Cosine Algorithm

ROA: Remora Optimization Algorithm

FFNN: Feed Forward Neural Network

HMFS: Hybrid Metaheuristic Feature Selection

MCNN: Multilayer CNN

GAP: Global Average Pooling

CBAM: Convolution Block Attention Module

CAM: Class Activation Map

2DCNN: 2-Dimensional CNN

RBF: Radial Basis Function

TL: Transfer Learning

LR: Logistic Regression

EXECUTIVE SUMMARY

Agriculture is extremely important to any country's economy. Every country's population is fully dependent on food, which is primarily produced through agriculture. However, plant diseases mostly affect agriculture, reducing yield and causing financial hardship for farmers. The time they spent producing those crops has also been wasted as a result of the sickness that has infected them. Because of the lack of sufficient infrastructure and procedures for detecting the sickness in a timely manner, curing the condition becomes difficult. However, expert farmers manually identify the diseases, which is a time-consuming operation. This study describes a deep learning approach for accurately identifying illnesses in apple tree and potato plants. To accomplish robust categorization, the system takes a multistage technique.

This work provides a new pipeline for detecting apple and potato diseases that combines GAC segmentation, EfficientNetB0 for feature extraction, PCA for dimensionality reduction, SMOTE for class imbalance, and a stacked ensemble classifier. The system uses pre-trained deep learning models, such as EfficientNetB0, to extract features, resulting in efficient and effective disease pattern learning. PCA improves computational efficiency while potentially reducing overfitting, resulting in a more robust model. SMOTE addresses class imbalance, guaranteeing that all disease classes have an equal chance of being accurately categorised. The stacked ensemble classifier combines the strengths of numerous models, perhaps leading to higher illness diagnosis accuracy. And last but not the least K-fold cross-validation to ensure model reliability and generalization. The aim is to enhance the accuracy, efficiency, and reliability of plant disease identification.

There is the possibility that this system could be utilised in applications related to precision agriculture. The ability to detect diseases at an early stage not only gives farmers the ability to take prompt action, but it also helps reduce crop losses, which ultimately leads to an increase in overall production.

CHAPTER 1

1.0 INTRODUCTION

1.1 Overview

Plant diseases significantly threaten global agriculture, impacting crop yield and quality. Traditional methods for disease detection are reliant on expert knowledge and manual inspection. Thus, the process is time-consuming and prone to error. This research work presents a system for identifying diseases in apple tree and potato plants using deep learning (DL) technique.

Agriculture has faced significant challenges in recent years due to increasing prevalence of plant diseases, which threaten food security, economic stability, and environmental sustainability. As the global demand for agricultural products continues to rise, there is an urgent need for efficient, accurate, and scalable solutions to identify and manage plant diseases.

Advancements in Machine Learning (ML) and Deep Learning (DL) have revolutionised numerous fields such as pattern recognition, image analysis, and predictive modelling. These technologies promise to transform agricultural practices by providing automated, precise and rapid disease detection systems. ML and DL models can analyse vast amount of data such as images of plant leaves, stems, and fruits, to identify symptoms of various diseases at an early stage. Thus enables timely intervention and reduction of crop losses.

1.2 Problem Statement

To design an efficient system for identifying diseases in plants using deep learning technique.

1.3 Objectives

The objectives of this research work are as follows:

- To study already published research work on identifying plant diseases using various AI techniques.
- Implementing already published research works to gain insight of the domain.
- To gain proficiency in Python and become acquainted with many libraries (e.g NumPy, SciPy, matplotlib, scikit-learn) that will be utilised to create the suggested system.

- To design & develop an efficient system for identification of plant diseases using deep learning technique.

1.4 Assumptions and Scope

1.4.1 Assumptions

- Quality of Input Images: The input leaf images are of sufficient quality and resolution to allow accurate segmentation and feature extraction.
- Consistency in Data: The training and test datasets are representative of real-world conditions and cover a variety of disease manifestations.
- Availability of Labels: Ground truth labels for the diseases are accurate and available for supervised learning.
- Computational Resources: Adequate training resources should be available to train deep learning models and for processing large datasets.

1.4.2 Scope

- To perform accurate delineation of diseased regions in leaf images
- To extract high-quality feature vectors from segmented images
- To perform dimensionality reduction for enhancing computational efficiency
- To handle class imbalance by generating synthetic minority class samples
- To achieve robust classification, perform Ensemble classification by combining Random Forest (RF), K-Nearest Neighbour (KNN) and Support Vector Classifier (SVC) with XGBoost
- To perform cross-validation which validates the model's performance across multiple folds of the dataset, ensuring generalizability of the model

1.5 Concept and Problem Analysis

Plant diseases have a significant impact on agricultural production and food security by destroying crops and reducing yields. Accurate and early detection of plant diseases is critical for effective management and mitigation, relying on advanced strategies such as deep learning for precise diagnosis.

The proposed system integrates several state-of-the-art techniques to enhance the accuracy and efficiency of identification. Different steps involved in the system are image segmentation, feature extraction, dimensionality reduction, class balancing, classification, and model validation. The proposed system aims to create a reliable tool for precision agriculture. The steps are elaborated in the following:

- Image Segmentation: The morphological Geodesic Active Contour (GAC) method is employed to improve the quality of feature extraction by accurately segmenting leaf images to isolate diseased regions.
- Feature Extraction: EfficientNetB0, a convolutional neural network (CNN), extracts relevant and detailed features from segmented images, balancing both accuracy and efficiency.
- Dimensionality Reduction: Principal Component Analysis (PCA) is employed to reduce the dimensionality of the extracted features, thereby improving computational efficiency and reducing the risk of overfitting.
- Class Balancing: Synthetic Minority Over-sampling Technique (SMOTE) has been implemented to generate synthetic samples in order to balance the dataset, thereby guaranteeing that the classifier performs optimally across all classes.
- Classification and Model Validation: A stacked Ensemble Classifier that integrates the capabilities of multiple classifiers (Random Forest, KNN, and SVC) with a meta-classifier (XGBoost) is used with K-fold cross-validation for robust disease classification and validation of the model.

1.6 Organization of the Thesis

- Chapter 1: This chapter contains an introduction of the thesis which includes an overview, problem statement, objectives, assumptions, scope, concept and problem analysis.
- Chapter 2: It covers all the literature surveys done to carry out the research work.
- Chapter 3: Proposed Methodology, detailed description of the implementation and overview of the proposed method have been discussed.
- Chapter 4: Throughout this chapter, the implementation, results, and comparison of the outcomes are detailed.
- Chapter 5: This chapter describes conclusion and future scope of the research work.
- References: All the references have been listed here.
- Appendix: Code snippets have been included here.

CHAPTER 2

2.0 LITERATURE SURVEY

The application of ML and DL techniques in agricultural disease identification has seen significant advancements, driven by the increasing availability of large datasets and powerful computational resources. This chapter provides an overview of notable research contributions in plant diseases identification.

Plant diseases are an existing problem which have long been affecting the yield of agricultural production by quality and quantity[1]. Sladojevic et al. [2] demonstrated the use of CNNs for plant disease recognition. Their model accurately identified 13 different plant diseases from images of healthy and diseased leaves. The study highlighted the potential of deep learning to automate and enhance disease diagnosis in plants.

Mohanty et al. [3] applied DL techniques to identify 26 diseases in 14 different crop species using a dataset of over 50,000 images. Their approach utilized a pre-trained AlexNet model, fine-tuned on the plant disease dataset, achieving an accuracy of over 99% on a held-out test set.

Ferentinos et al.[4] explored the use of transfer learning with deep CNNs for the detection of plant diseases. By leveraging pre-trained models like VGG(Visual Geometry Group) and Inception, they have demonstrated improved accuracy and reduced training time, making it feasible for practical agricultural applications.

Kamilaris and Prenafeta-Boldú [5] provided a comprehensive review of deep learning techniques for agricultural applications, emphasizing the effectiveness of ensemble methods. By combining multiple models, ensemble approaches can enhance robustness and accuracy in disease classification tasks.

Patil and Kumar [6] utilized SVM and decision tree classifiers for the identification of fungal diseases in soybean crops. Their study highlighted the effectiveness of traditional ML methods, especially when combined with image preprocessing techniques, to enhance feature extraction.

Liu et al. [7] applied Recurrent Neural Networks (RNNs) for the prediction of disease outbreaks in crops based on temporal data from environmental sensors. This approach enabled early warning systems, providing farmers with timely interventions to mitigate disease spread.

Anagnostis et al. [8] explored unsupervised learning techniques, such as autoencoders, for detecting anomalies in plant health. By learning the normal patterns of plant features, the model could identify deviations indicative of disease even without labelled data.

Zhang et al.[9] reviewed the integration of ML and DL in precision agriculture focusing on disease detection and management. The study emphasized the role of sensor networks, UAV imagery, and IoT devices in collecting high-resolution data which, when combined with ML/DL models, significantly enhances disease monitoring and decision-making processes.

Benos et al.[10] discussed the challenges of deploying ML and DL models in real-world agricultural settings such as the need for large annotated datasets, model interpretability and computational resource constraints. The paper also suggested future research directions including the development of lightweight models suitable for edge computing devices used in the field.

Siddique et al.[11] proposes a deep-learning approach using CNNs to classify diseases such as early blight, late blight and bacterial spots. It explores the impact of different CNN architectures and data augmentation techniques on the performance on disease recognition.

Barbedo [12] provides insights into the challenges and opportunities of using deep-learning approaches for crop disease detection and diagnosis. It discusses the importance of feature selection, dataset quality, and model interpretability in developing effective disease detection systems.

Huang et al. [13] investigates the use of deep learning algorithms including CNNs and generative adversarial networks (GANs), for detecting apple leaf diseases. It evaluates the performance of different deep learning models in accurately identifying diseases such as apple scab and apple rust.

AI technologies have shown promising results in identifying plant abnormalities and infestations. ML, DL, and CV(Computer Vision)-based systems are utilized for the classification and lesion segmentation of plant diseases from digital images and could change the method of discovering plant illnesses significantly [14]. However, these technologies need a considerable amount of annotated training data and may not be suitable for diseases that have not been seen before. Further research is needed to develop generalizable models that can be applied to different plant species and diseases and to make more datasets publicly available for training and evaluating the models.

Tian et al. [15] combined a GA (Genetic Algorithm) with the SVM classifier and performed feature selection based on kernel principal component analysis (KPCA) to identify the best features in the images. The proposed KPCA/GA-SVM recognition model achieved the following results: 98.14%, 94.05% and 97.96% accuracy for apple mosaic virus, apple rust and apple leaf spot, respectively.

Gulavnai et al. [16] proposed a ResNet-CNN (ResNet18, ResNet34 and ResNet50) combined with TL for automatic detection and identification of four mango leaf diseases named anthracnose, powdery mildew, red rust and golmich. Results show that ResNet50 gives better performance with an accuracy of 91.50%.

In recent years, many studies have employed detection networks to classify pathogens and pests [17]. It is expected that in the future, more advanced detection models will be utilized for the identification of plant maladies and infestations as object segmentation networks in computer vision continue to evolve.

Further, Shrivastava and Pradhan [18] gave a rice plant detection and classification system using a colour feature based on ML models. Out of 172 features, they considered 14 different colour spaces from which they used 4 characteristics for each. The result shows that it will help farmers enhance the quality & amount of their yield.

In addition to this paper, Kumar et al. [19] suggested a machine learning method to develop a system that predicts various fungal infections for plant disease detection. This study shows how sensors can provide insights into numerous abiotic variables that could support the detection of plant illnesses. Although several ML with IoT papers have been proposed to detect plant disease, the main disadvantages of ML with IoT techniques are overfitting problems, fine-tuning issues and Conventional approaches falling short in assessing the disease's severity.

Mishra et al. [20] presented the rider neural network with the sine-cosine algorithm as a unique illness classifier. In this case, the SCA (Sine Cosine Algorithm) model modifies the ROA (Remora Optimization Algorithm) algorithm to regulate the location update. For the experiment, the three Internet of Things systems are taken with the total number of nodes, such as 50,100,150. The overall work of the classifier is validated using criteria such as specificity, accuracy and efficiency of nodes on various Internet of Things systems. As a result, the suggested algorithm aids farmers in quickly identifying the infected plants on their property.

Pham et al. [21] proposed a Feed-Forward Neural Network (FFNN) with Hybrid Metaheuristic Feature Selection (HMFS) to classify 3 mango diseases named Anthracnose, Gall Midge, and Powdery Mildew.

While Singh et al. [22] used a multilayer convolutional neural network (MCNN) model to classify mango leaves infected with the fungal disease named as anthracnose. They pre-trained their images using histogram of equalization (for contrast enhancement) and central square crop method (for image resizing). Over the years, researchers have continued to develop deeper and more powerful CNN models.

Geetharamani et al. [23] developed a deep CNN model with nine layers to solve plant leaf disease identification problems using PlantVillage dataset. To achieve better performance and accuracy (96.46%), they had to improve the model training images using the following methods: image flipping, gamma correction, noise injection, color enhancement by principal component analysis (PCA), rotation and scaling. The authors believe that extending their database with new images of different plant species and from different sources would increase the performance and accuracy of their model.

Dai et al. [24] have presented a DL model (PPLCNet) that includes dilated convolution, a multi-level attention mechanism, and GAP (Global Average Pooling) layers. The model used novel weather data augmentation to expand the sample size to enhance the generalization and robustness of feature extraction. The feature extraction network uses saw-tooth dilated convolution with a configurable expansion rate to extend the perceptual field of the convolutional domain, effectively addressing the problems of insufficient data information extraction. The lightweight CBAM (Convolutional Block Attention Module) attention mechanism was located in the feature extraction network's middle layer. It was used to improve the model's information representation. By reducing the number and complexity of parameters computed by the network, the GAP layer prevents overfitting of the model. Furthermore, the proposed integrated CAM (Class Activation Map) visualization approach fully validates the efficiency of the proposed model.

According to the study, P.B.R and A.VV. et al. [25] proposed an effective CNN model to categorize tomato leaf diseases and detect the name of the disease affecting tomato leaves. An approach to a 2-dimensional Convolutional Neural Network (2DCNN) model with 2-Max Assembling covers and completely related layers has been proposed.

To extract different features, Anari [26] have used model engineering (ME). To improve feature discrimination and processing speed, several SVM models were used. In the training process, the kernel parameters of the radial basis function (RBF) were computed depending on the selected model. Six leaf image sets encompassing healthy and sick leaves of apple, corn, cotton, grape, pepper, and rice were analyzed using PlantVillage and UCI databases. Accordingly, the categorization procedure yielded almost 90,000 images. The findings of the experimental implementation phase reveal the potential of a powerful model in classification activities, which would be useful for a variety of future leaf disease diagnostic applications in the agricultural business. In terms of stability, the dilated learning model outperforms the typical ResNet-18 design.

Singh and Mishra [27] have presented an image segmentation algorithm for the automatic detection and classification of plant leaf diseases. It also includes an overview of various disease classification techniques that can be used to detect plant leaf disease. The genetic algorithm was used for image segmentation, which was vital for disease detection in plant leaf disease.

Saraswathi and FarithaBanu used ensemble classifiers (EC) in [28], which are developed by using various approaches to preparation, feature extraction, and classification. The performance of these multiple ensemble techniques was then compared to select the best ensemble classifiers. The suggested technique's precision and reliability were tested in both controlled laboratory settings and real-world conditions using two databases, namely PlantVillage and Taiwan tomato leaves.

Garg and Singh [29] have employed an aggregated loss function by combining triplet and cross-entropy loss with MobileNetV2 as a basis model for the effective classification of plant disease using small samples. For the evaluation of the proposed study, two publicly available datasets (PlantVillage with 54,303 leaf samples and Plantdoc with 2598 leaf samples) were used. To partition the dataset into the source and target domains, different domain splits were examined, and a large quantity of testing was conducted on the target dataset using various sample sizes. For the analysis of the PlantVillage dataset, four domain splits were considered, and it was found that using the proposed aggregated loss and the lightweight transfer learning (TL) model for the target domain data (K-ways, N-shot), an average improvement in accuracy was 1.49% for split-1, 16.25% for split-2, 2.9% for split-3, and 2.1% for split-4 when compared to previous work. For the plantdoc dataset, two domain splits were evaluated, yielding an accuracy of around 81% with 30 samples and more than 40% with only one sample. Using several evaluation measures such as loss functions, execution time, model size, and model parameters, the suggested work was compared to other state-of-the-art research works.

Kukadiya and Meva [30] have presented a DL-based CNN solution for automatically classifying and distinguishing cotton leaf diseases. There has been a lot of study done on leaf diseases that were common in many crops, but this work offered an effective and reliable method for identifying cotton leaf disease. The proposed method successfully classified and detected three significant cotton leaf diseases, which were difficult to control if not detected early. The proposed model for the identification and classification of cotton leaf diseases has used CNN.

Attallah [31] has proposed a pipeline for autonomous identification of tomato leaf diseases using three compact CNNs. The author has used TL to extract deep features from the CNNs' final fully connected layer for more condensed and high-level representation. Next, it merges elements from the three CNNs to take advantage of each CNN structure. Following that, a hybrid feature selection approach was used to select and build a comprehensive feature set of lower dimensions. The tomato leaf disease identification approach has been utilized for six classifiers. The proposed pipeline's experimental findings were also compared with existing research studies for tomato leaf disease classification, confirming its competitive potential.

Al-gaashani et al. [32] have proposed a tomato leaf disease classification method using TL and feature concatenation. The authors extract features from MobileNetV2 and NASNetMobile using pre-trained kernels (weights), then concatenate and reduce the dimensionality of these features using kernel principal component analysis. They then feed these features into a conventional learning algorithm. The experimental results confirmed the efficiency of concatenated features in improving classifier performance. The authors have tested the three most common traditional ML classifiers, RF, SVM, and multinomial LR (Logistic Regression) and found that multinomial LR performed the best.

After a comprehensive literature review, a multi-stage methodology is proposed for robust image classification. First, a meticulous pre-processing pipeline ensures all data is presented consistently. This involves converting images to grayscale, applying an inverse Gaussian gradient, performing Geodesic Active Contour (GAC) for potential noise reduction, and finally removing the background. Then, EfficientNetB0, a pre-trained CNN, extracts informative features from the pre-processed images. To optimize training efficiency, PCA reduces the dimensionality of the extracted features. And after this stage, to address the class imbalance, SMOTE has been used which artistically generates synthetic data points for under-represented classes. A stacked ensemble classifier with K-fold Cross-Validation integrates the power of Random Forest, K-Nearest Neighbour, and SVM by feeding their predictions into a final XGBoost model. K-fold cross-validation ensures the model's robustness and reliability, contributing to precision agriculture and better crop management.

CHAPTER 3

3.0 PROPOSED APPROACH

The proposed work starts by dividing the dataset images into distinct segmented images to better understand and process the data. Then utilizing the state-of-the-art EfficientNetB0 model(pre-trained), known for its robustness and efficiency, to extract highly detailed and exceptional features from the dataset. Following this, PCA (Principal Component Analysis), a powerful statistical method is applied to effectively reduce the dimensionality of the feature set while retaining critical information. Furthermore, the work aims to delve into the intricate details of the feature set and identify key patterns and relationships within the data. Also, one-hot encoding is performed. Then, leveraging the power of a stacked Ensemble Classifier, images are classified and finally cross-validation is performed to ensure the model's robustness and reliability, contributing to precision agriculture and better crop management.

3.1 Data Collection

The data utilised in this study are obtained from various publicly accessible datasets to guarantee a broad and inclusive compilation of photos including apple and potato plant diseases. This study exclusively focuses on the apple and potato leaf images from the PlantVillage dataset [33], which consists of a diverse range of leaf images exhibiting various plant diseases. Figure 1 shows some sample images from PlantVillage dataset. The Apple Disease dataset [34] contains images of apple leaves exhibiting various sorts of diseases, whereas the Potato Leaf Disease dataset [35] primarily concentrates on varied manifestations of diseases affecting potato leaves. In addition, this study examines the Potato Leaf (Healthy and Late Blight) dataset [36], which contains diverse images of potato plants, and the Apple Leaf diseases dataset [37], which is a comprehensive collection specifically focused on illnesses affecting apple leaves.

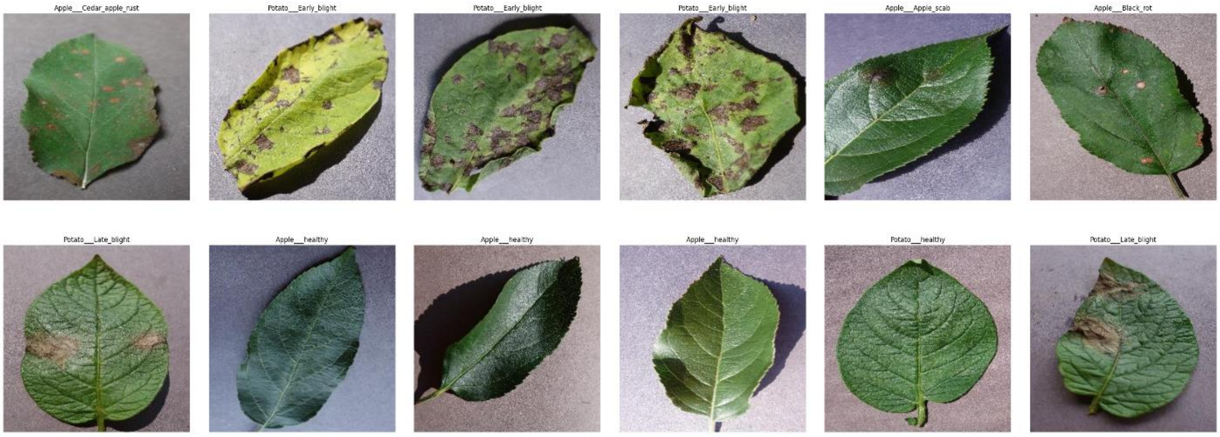


Figure 1: Sample images from PlantVillage dataset

3.2 Data Pre-processing

To prepare the collected data for analysis, the following pre-processing steps are performed –

- ✓ Convert all colour images to grayscale to reduce complexity and focus on intensity patterns.
- ✓ This step helps in highlighting disease symptoms which are often characterized by intensity changes.

3.3 Data Segmentation

The pre-processed images undergo segmentation to isolate diseased regions for more accurate feature extraction –

- ✓ Convert grayscale images to inverse gaussian gradient images to enhance edges and boundaries of the diseased areas.
- ✓ Apply Morphological GAC to the gradient images, which accurately segments the image by delineating the boundaries of diseased regions, ensuring precise feature extraction.
- ✓ Finally, the background of the images were removed.

3.4 Methodology

This work utilizes a multi-stage approach to achieve accurate classification of apples and potatoes. Figure 2 shows the proposed classification model.

First, raw images undergo pre-processing and segmentation using GAC to isolate potential disease areas. This focuses the analysis on the most relevant parts of the image.

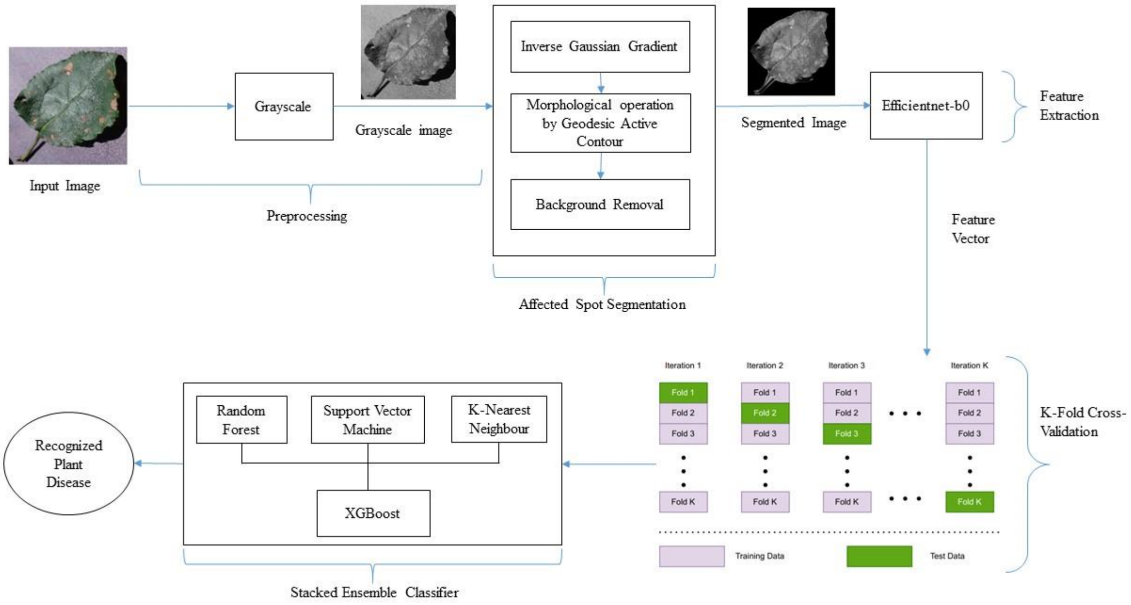


Figure 2: Proposed Classification Model

Next, these segmented regions are fed into a pre-trained deep learning model called EfficientNetB0. This powerful model acts as a feature extractor, automatically learning discriminative features from the image data that represent the presence or absence of disease. Importantly, features are extracted from the topmost convolutional layer of EfficientNetB0 (‘**top_conv**’), where high-level disease-relevant information is captured. Figure 3 shows the baseline model of EfficientNetB0.

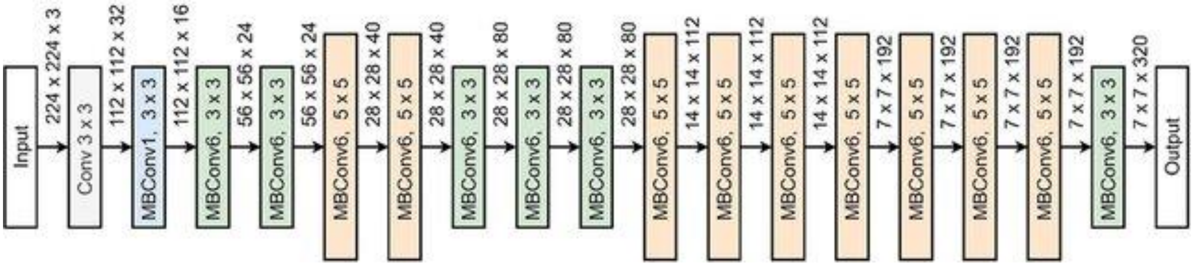


Figure 3: EfficientNetB0 baseline model

However, the extracted features might be high-dimensional. To improve computational efficiency and potentially prevent overfitting, this work employs Principal Component Analysis (PCA). PCA reduces the complexity of the data by identifying the most informative features.

Another challenge the system addresses is class imbalance, where some diseases might be less frequent. To ensure all disease categories have a fair chance of being classified correctly, this work utilizes Synthetic Minority Over-sampling Technique (SMOTE). SMOTE creates synthetic data points for under-represented disease classes, resulting in a more balanced training dataset.

Lastly, this work leverages the power of ensemble learning with a stacked approach. Here, three base models – Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbor (KNN) – are trained on the data first. The predictions from these models are then used to train a final estimator, an XGBoost model. This final model leverages the combined knowledge of the base models, potentially leading to superior disease identification accuracy and robustness. The Ensemble classifier used in this work is depicted below in the Figure

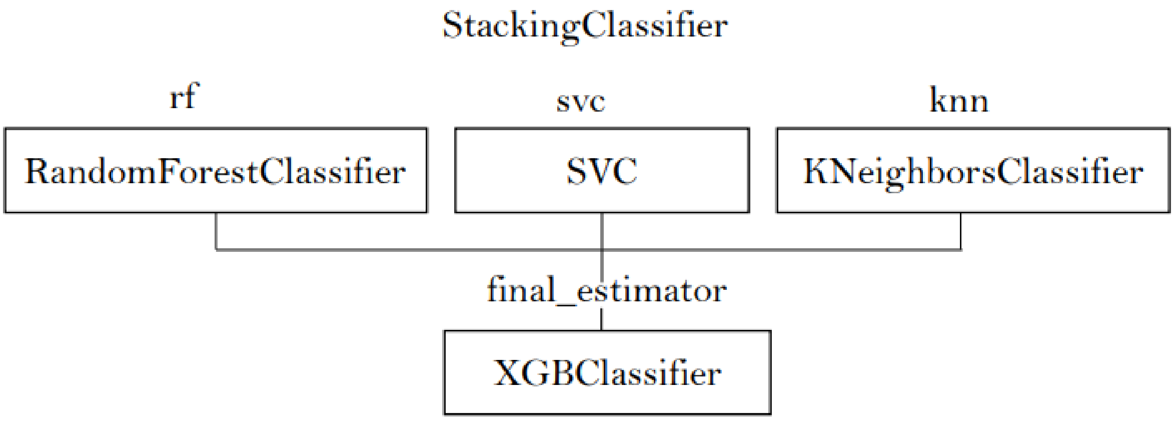


Figure 4: Deployed Ensemble Classifier

To assess the generalizability of the model and prevent overfitting, this work employs K-fold cross-validation with 10 folds which signifies that the entire dataset is divided into 10 folds. The model is trained and validated 10 times, each time using a different fold as the validation set and the remaining folds as the training set. This process ensures that the model's performance is thoroughly evaluated across different subsets of the data and also ensures the model performs well on unseen data.

CHAPTER 4

4.0 EXPERIMENTATIONS AND RESULTS

In this work, Python (version 3.11.7) is used with the required built-in libraries.

The proposed work begins with preprocessing the datasets by converting images to grayscale. Figure 5 shows a sample of pre-processed image.

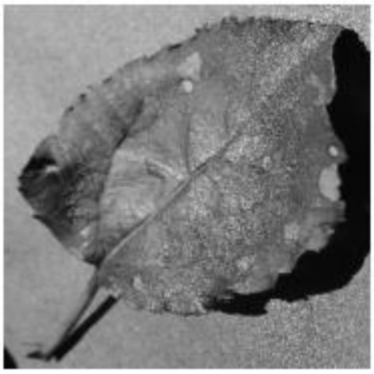


Figure 5: Pre-processed image

This is followed by segmenting the pre-processed images using inverse Gaussian gradient conversion and Morphological Geodesic-based Active Contour (GAC) to accurately delineate diseased regions. Figure 6 shows a sample of segmented images.

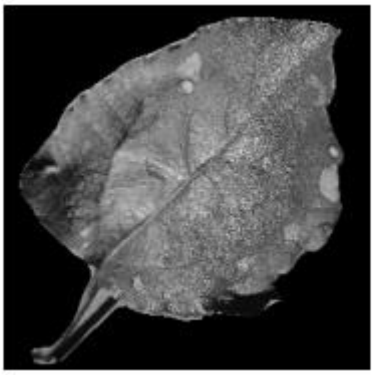


Figure 6: Segmented image

EfficientB0 has been used in this work to extract the features from the dataset of segmented images. Table 1 shows the parameters of EfficientNetB0 architecture.

Table 1: Parameters of EfficientNetB0 architecture

Stage i	**Operator fi**	**Resolution $\hat{H} \times \hat{W}$**	**#Channels $\hat{C}\,i$**	**#Layers $\hat{l}_i$**
1	Conv3 × 3	224 × 224	32	1
2	MBConv1, k3 × 3	112 × 112	16	1
3	MBConv6, k3 × 3	112 × 112	24	2
4	MBConv6, k5 × 5	56 × 56	40	2
5	MBConv6, k3 × 3	28 × 28	80	3
6	MBConv6, k5 × 5	28 × 28	112	3
7	MBConv6, k5 × 5	14 × 14	192	4
8	MBConv6, k3 × 3	7 × 7	320	1
9	Conv1 × 1&Pooling&FC	7 × 7	1280	1

All the dataset utilized in this work has been divided into two directories, namely train and test. The train directory contains 80% of the images from the dataset, while the test directory has the remaining 20%. For object detection and classification tasks, many performance metrices have been used. Some of them are precision, recall, classification accuracy, F1-score, and Cohen-Kappa score. And all of this can be calculated using the following formulas:

$$Precision(i) = \frac{\#TP(i)}{\#TP(i) + \#FP(i)}$$

$$Recall(i) = \frac{\#TP(i)}{\#TP(i) + \#FN(i)}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$Classification\ Accuracy = \frac{\#TP(i) + TN(i)}{\#TP(i) + \#FP(i) + TN(i) + FN(i)}$$

$$Cohen - Kappa\ score(k) = \frac{p_0 - p_e}{1 - p_e}$$

Where, i is the number of classes, TP represents the number of true positives, FP denotes the number of false positives, TN represents the number of true negatives, FN represents the number of false negatives, p_0 is the overall accuracy of the model and p_e is the measure of the agreement between the model predictions and the actual class values.

The performance of the proposed model on each dataset has been illustrated using the Cross-validation curve, ROC curve, and Precision-Recall curve.

For each dataset, the result produced by the proposed model are given below:

4.1 PlantVillage dataset

Table 2 provides a description of the dataset.

Table 2: Description of PlantVillage Dataset

Disease Types	**Assigned Class label**	**Number of Images**	**Total Training Images**	**Total Testing Images**
Potato_healthy	0	152	121	31
Apple_healthy	1	1645	1316	329
Apple_Black_rot	2	621	497	124
Apple_Apple_scab	3	630	504	126
Potato_Early_blight	4	1000	800	200
Apple_Cedar_apple_rust	5	275	220	55
Potato_Late_blight	6	1000	800	200

The Original class distribution before applying SMOTE: Counter({1: 1316, 6: 800, 4: 800, 3: 504, 2: 497, 5: 220, 0: 121}). Figure 7 displays the class distribution before SMOTE.

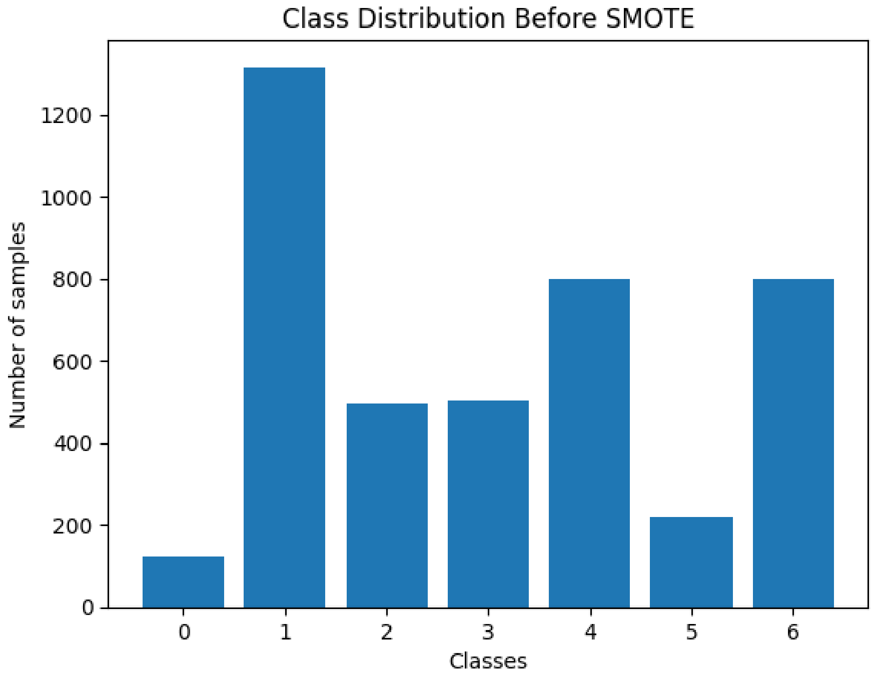


Figure 7: Class Distribution Before SMOTE (PlantVillage dataset)

Balanced class distribution after applying SMOTE: Counter({6: 1316, 4: 1316, 2: 1316, 5: 1316, 1: 1316, 3: 1316, 0: 1316}). Figure 8 displays the class distribution after SMOTE.

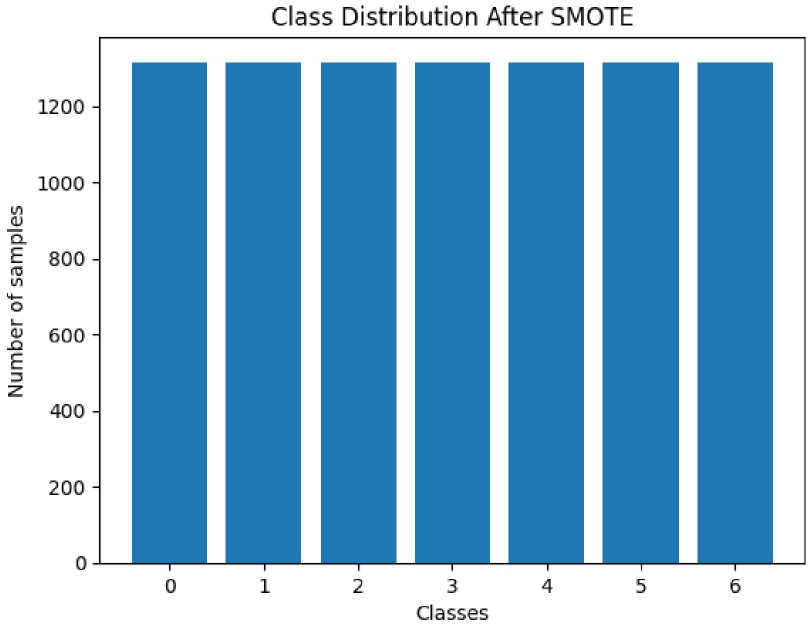


Figure 8: Class Distribution After SMOTE (PlantVillage dataset)

Table 3 describes the cross-validation report for each fold, where the average metrics (precision, recall, and F1-score) are calculated using the macro average. This ensures that each class contributes equally to the average. Figure 9 shows the cross-validation curve.

Table 3: Cross-validation report for each fold (PlantVillage dataset)

Fold No.	**Accuracy Score**	**F1-Score**	**Cohen Kappa Score**
1	98.8	98.81	98.60
2	98.04	98.01	97.72
3	97.82	97.91	97.46
4	98.15	98.08	97.84
5	98.47	98.35	98.22
6	98.04	98.08	97.71
7	99.23	99.24	99.11
8	97.17	97.25	96.70
9	98.26	98.29	97.97
10	98.37	98.30	98.09

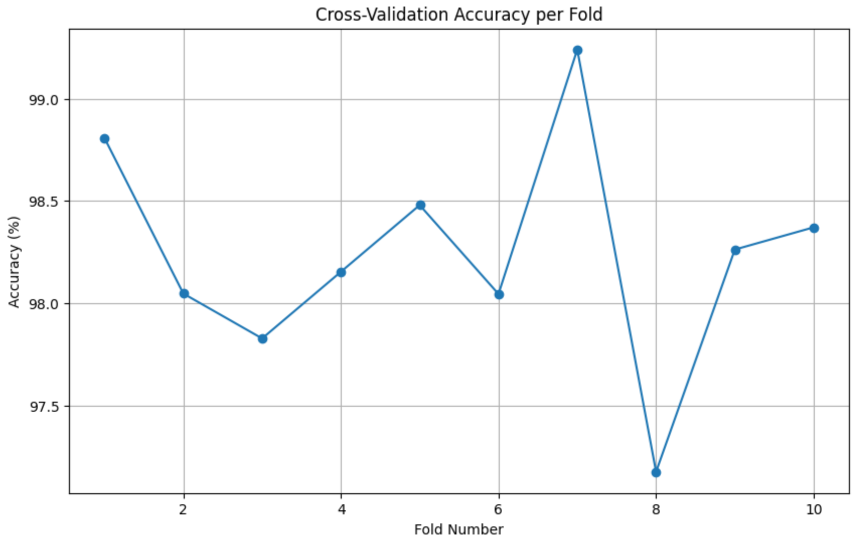


Figure 9: Cross-validation curve (PlantVillage dataset)

Table 4 describes the classification report generated for PlantVillage dataset.

Table 4: Classification Report (PlantVillage dataset)

	Precision	Recall	F1-Score	Support
Potato_healthy	1.00	1.00	1.00	1316
Apple_healthy	0.96	0.96	0.96	1316
Apple_Black_rot	0.99	0.99	0.99	1316
Apple_Apple_scab	0.97	0.97	0.97	1316
Potato_Early_blight	0.99	0.99	0.99	1316
Apple_Cedar_apple_rust	1.00	1.00	1.00	1316
Potato_Late_blight	0.97	0.97	0.97	1316
accuracy			0.98	9212
Macro-average	0.98	0.98	0.98	9212
Weighted-Average	0.98	0.98	0.98	9212

Figure 10 illustrates the receiver-operating characteristic curve for each class in the PlantVillage dataset.

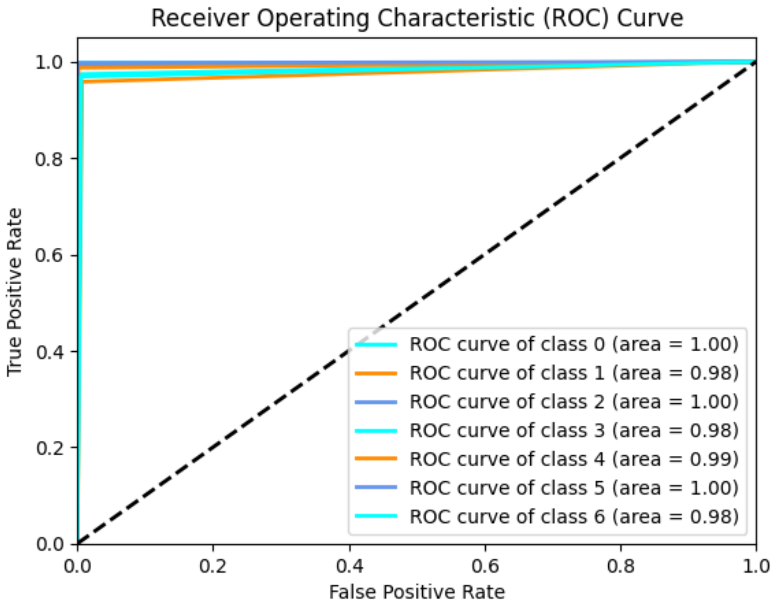


Figure 10: ROC curve (PlantVillage dataset)

Figure 11 exhibits the Precision-Recall curve for each class in the PlantVillage dataset.

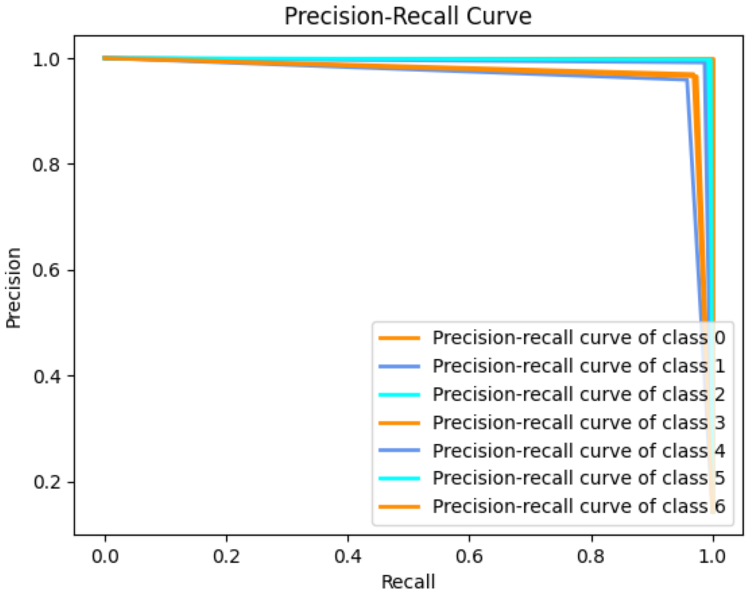
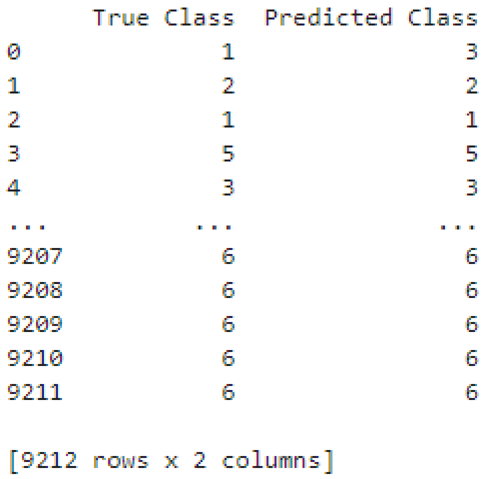


Figure 11: Precision-Recall curve (PlantVillage dataset)

Figure 12 depicts the model prediction report with true and predicted classes.



```
      True Class  Predicted Class
0              1                3
1              2                2
2              1                1
3              5                5
4              3                3
...          ...              ...
9207           6                6
9208           6                6
9209           6                6
9210           6                6
9211           6                6

[9212 rows x 2 columns]
```

Figure 12: Model prediction Report (PlantVillage dataset)

4.2 Apple Diseases Dataset

Table 5 provides a description of the dataset.

Table 5: Description of Apple Diseases Dataset

Disease Types	**Assigned Class label**	**Number of Images**	**Total Training Images**	**Total Testing Images**
Apple_Black_rot	0	2484	1987	497
Apple_Apple_scab	1	2520	2016	504
Apple_Cedar_apple_rust	2	2200	1760	440
Apple_healthy	3	2510	2008	502

The Original class distribution before applying SMOTE: Counter({1: 2016, 3: 2008, 0: 1987, 2: 1760}) Figure 13 displays the class distribution before SMOTE.

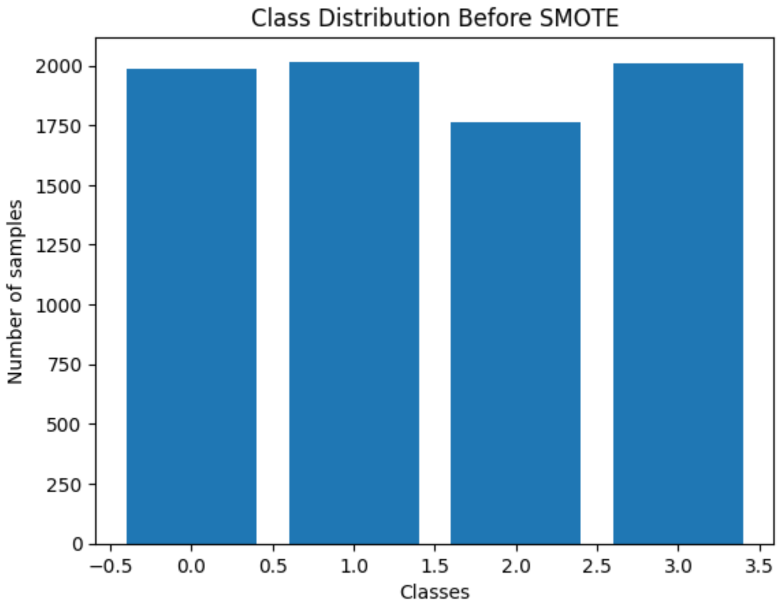


Figure 13: Class Distribution Before SMOTE (Apple Diseases dataset)

Balanced class distribution after applying SMOTE: Counter({0: 2016, 3: 2016, 1: 2016, 2: 2016}) . Figure 14 displays the class distribution after SMOTE.

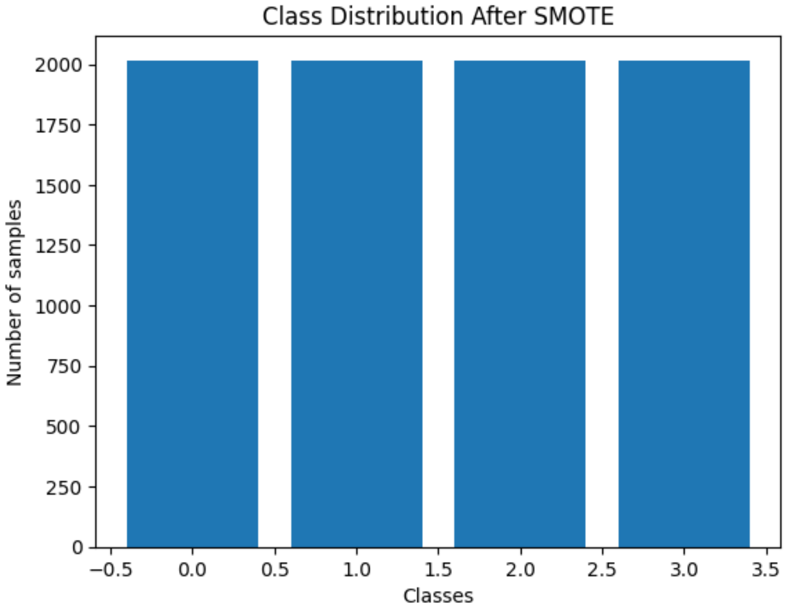


Figure 14: Class Distribution After SMOTE (Apple Diseases dataset)

Table 6 describes the cross-validation report for each fold, where the average metric s (precision, recall, and F1-score) are calculated using the macro average. This ensur es that each class contributes equally to the average. Figure 15 shows the cross-valid ation curve.

Table 6: Cross-validation report for each fold (Apple Diseases dataset)

Fold No.	Accuracy Score	F1-Score	Cohen Kappa Score
1	95.91	95.96	94.54
2	96.28	96.34	95.03
3	94.75	94.83	93.05
4	96.03	95.94	97.84
5	96.65	96.58	94.70
6	95.75	95.79	95.52
7	96.27	96.07	94.97
8	97.02	97.03	96.02
9	96.40	96.41	95.20
10	97.02	97.03	96.02

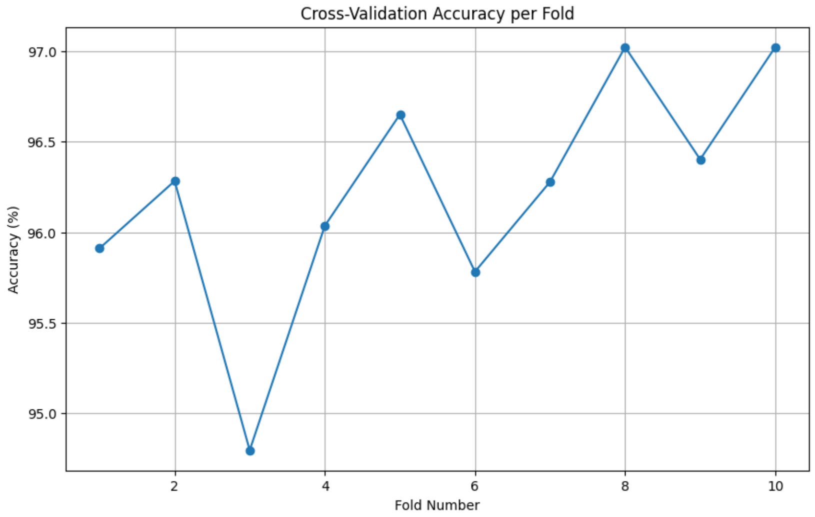


Figure 15: Cross-validation curve (Apple Diseases dataset)

Table 7 describes the classification report generated for Apple Diseases dataset.

Table 7: Classification Report (Apple Diseases dataset)

	Precision	Recall	F1-Score	Support
Apple_Black_rot	0.97	0.97	0.97	2016
Apple_Apple_scab	0.95	0.94	0.94	2016
Apple_Cedar_apple_rust	0.98	0.99	0.98	2016
Apple_healthy	0.96	0.95	0.95	2016
accuracy			0.96	8064
Macro-average	0.96	0.96	0.96	8064
Weighted-Average	0.96	0.96	0.96	8064

Figure 16 illustrates the receiver-operating characteristic curve for each class in the Apple Diseases dataset.

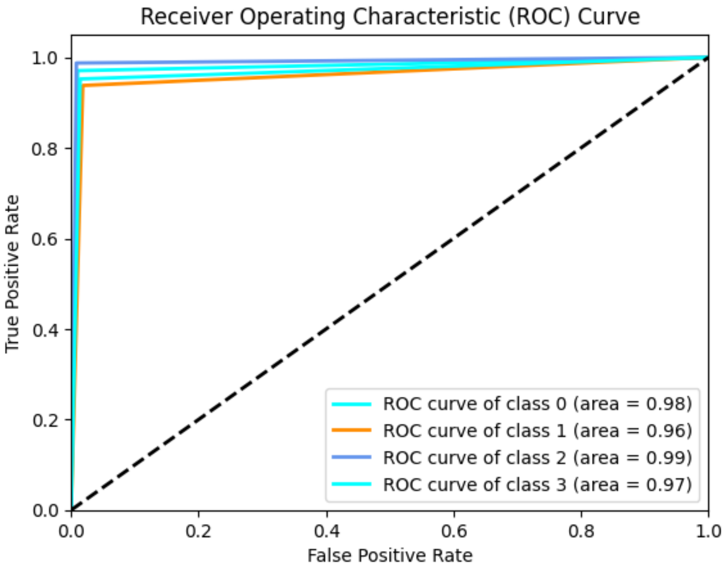


Figure 16: ROC curve (Apple Diseases dataset)

Figure 17 exhibits the Precision-Recall curve for each class in the Apple Diseases dataset.

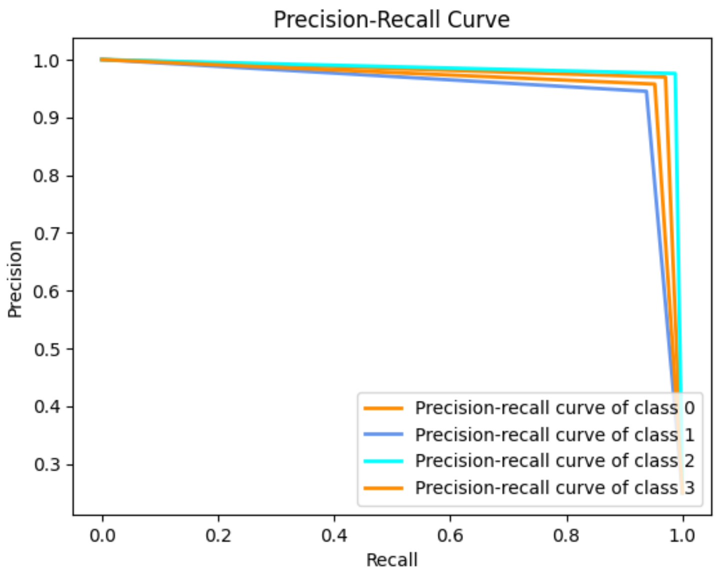
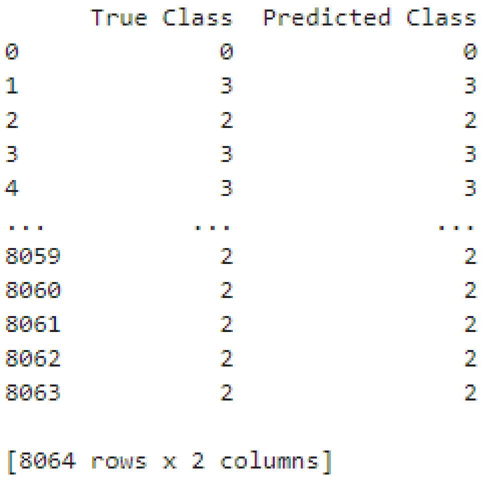


Figure 17: Precision-Recall curve (Apple Diseases dataset)

Figure 18 depicts the model prediction report with true and predicted classes.



```
      True Class  Predicted Class
0              0                0
1              3                3
2              2                2
3              3                3
4              3                3
...          ...              ...
8059           2                2
8060           2                2
8061           2                2
8062           2                2
8063           2                2

[8064 rows x 2 columns]
```

Figure 18: Model prediction Report (Apple Diseases Dataset)

4.3 Potato Leaf Disease Dataset

Table 8 provides a description of the dataset.

Table 8: Description of Potato Leaf Disease Dataset

Disease Types	Assigned Class Label	Number of Images	Total Training Images	Total Testing Images
Potato_healthy	0	500	400	100
Potato_Early_Bl ight	1	500	400	100
Potato_Late_Bli ght	2	500	400	100

The Original class distribution before applying SMOTE: Counter({0: 400, 2: 40 0, 1: 400}). Figure 19 displays the class distribution before SMOTE.

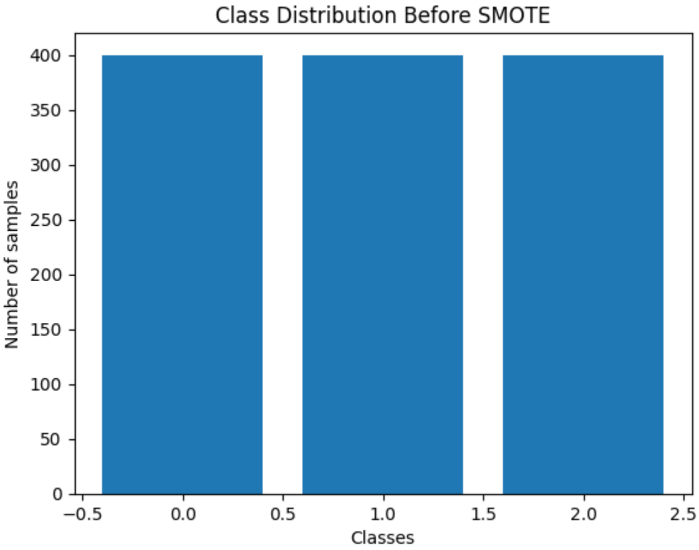


Figure 19: Class Distribution Before SMOTE (Potato Leaf Disease dataset)

Balanced class distribution after applying SMOTE: Counter({0: 400, 2: 400, 1: 4 00}). Figure 20 displays the class distribution after SMOTE.

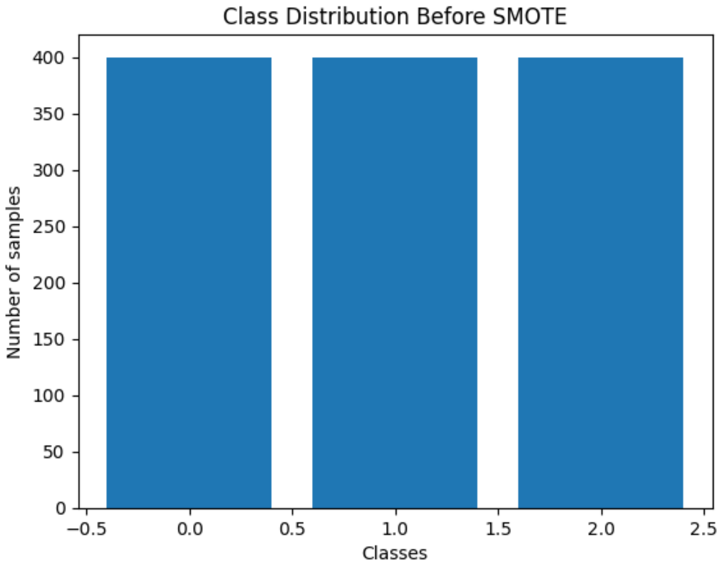


Figure 20: Class Distribution After SMOTE (Potato Leaf Disease dataset)

Table 9 describes the cross validation report for each fold, where the average metric s (precision, recall, and F1-score) are calculated using the macro average. This ensu res that each class contributes equally to the average. Figure 21 shows the cross-vali dation curve.

Table 9: Cross-validation report for each fold (Potato Leaf Disease dataset)

Fold No.	Accuracy Score	F1-Score	Cohen Kappa Score
1	97.50	97.51	96.25
2	94.16	93.83	91.23
3	95	94.91	92.49
4	95	94.70	91.42
5	90	94.11	84.94
6	95	94.88	92.45
7	94.16	93.89	91.11
8	96.66	96.75	94.29
9	94.83	95.79	93.74
10	91.66	91.81	87.47

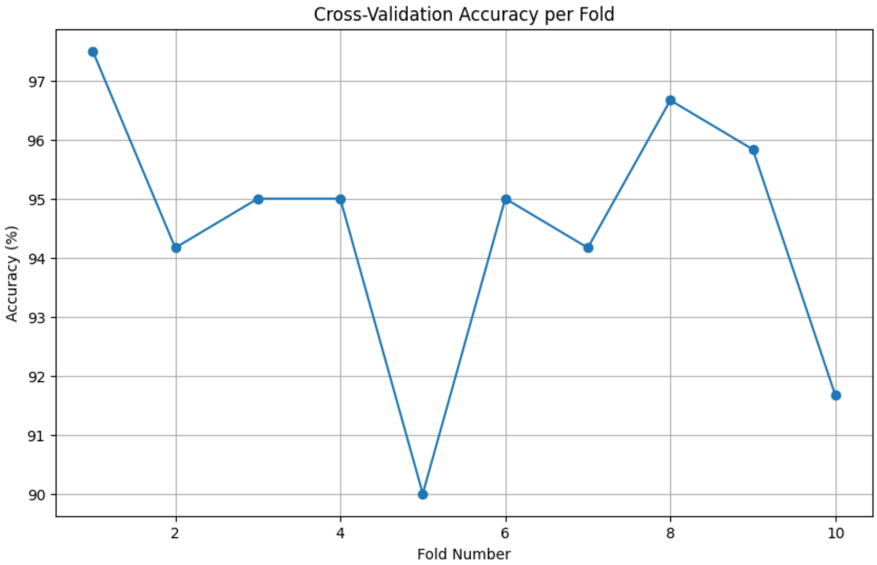


Figure 21: Cross-validation curve (Potato Leaf Disease dataset)

Table 10 describes the classification report generated for Potato Leaf Disease dataset.

Table 10: Classification Report (Potato Leaf Disease dataset)

	Precision	**Recall**	**F1-Score**	**Support**
Potato_healthy	0.96	0.95	0.96	400
Potato_Early_blight	0.96	0.94	0.95	400
Potato_Late_blight	0.92	0.94	0.93	400
accuracy			0.94	1200
Macro-average	0.95	0.94	0.95	1200
Weighted-Average	0.95	0.94	0.95	1200

Figure 22 illustrates the receiver-operating characteristic curve for each class in the Potato Leaf Disease dataset.

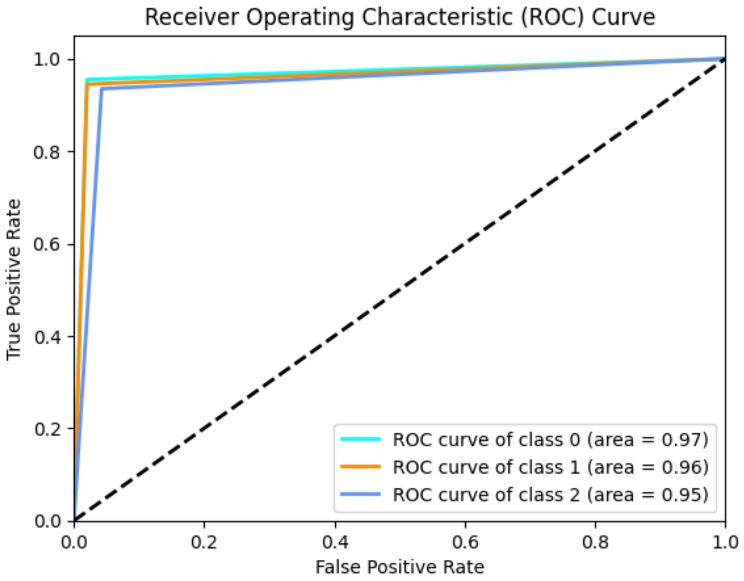


Figure 22: ROC curve (Potato Leaf Disease dataset)

Figure 23 exhibits the Precision-Recall curve for each class in the Potato Leaf Disease dataset.

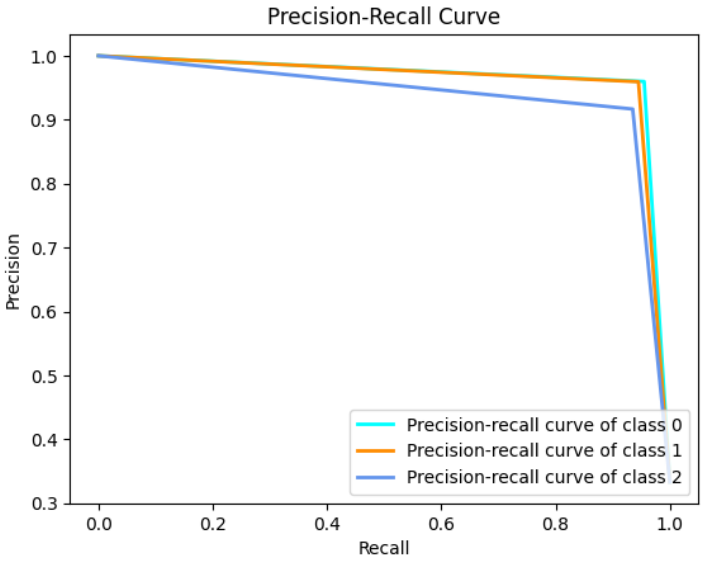
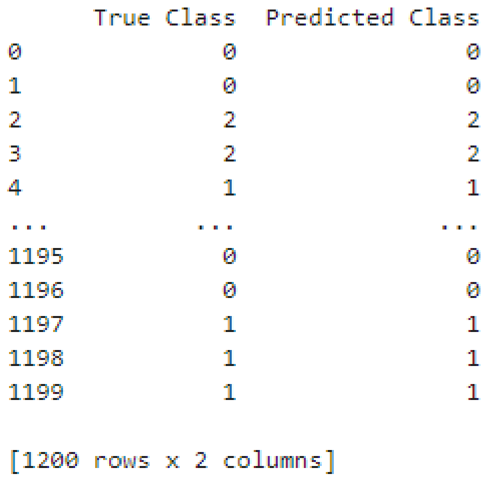


Figure 23: Precision-Recall curve (Potato Leaf Disease dataset)

Figure 24 depicts the model prediction report with true and predicted classes.



```
      True Class  Predicted Class
0              0                0
1              0                0
2              2                2
3              2                2
4              1                1
...          ...              ...
1195           0                0
1196           0                0
1197           1                1
1198           1                1
1199           1                1

[1200 rows x 2 columns]
```

Figure 24: Model prediction Report (Potato Leaf Disease dataset)

4.4 Potato (Healthy & Late Blight) Dataset

Table 11 provides a description of the dataset.

Table 11: Description of Potato (Healthy & Late Blight) Dataset

Disease Types	**Assigned Class Label**	**Number of Images**	**Total Training Images**	**Total Testing Images**
Potato_healthy	0	363	290	73
Potato_Late_Blight	1	67	54	13

The Original class distribution before applying SMOTE: Counter({0: 290, 1: 54 }). Figure 25 displays the class distribution before SMOTE.

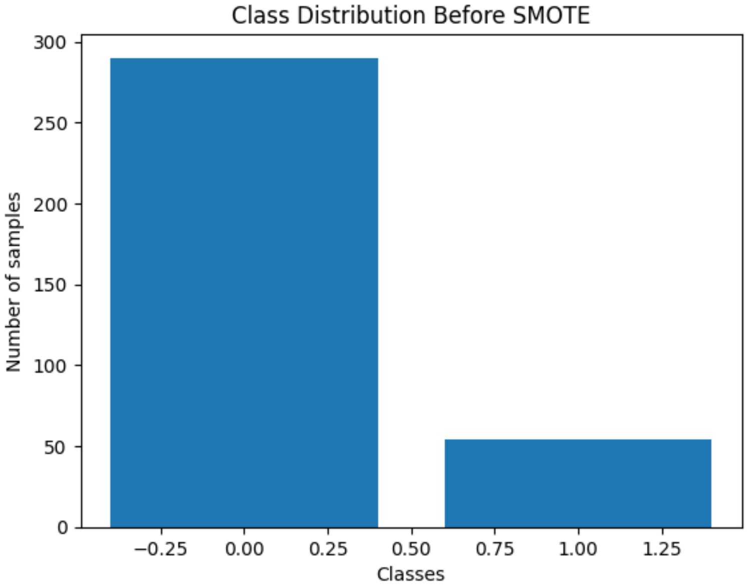


Figure 25: Class Distribution Before SMOTE(Potato (Healthy & Late Blight) dataset)

Balanced class distribution after applying SMOTE: Counter({1: 290, 0: 290}). Figure 26 displays the class distribution after SMOTE.

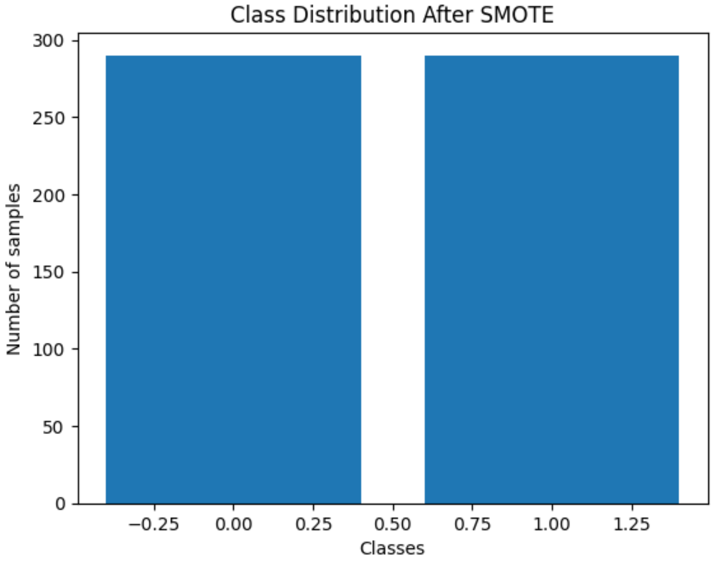


Figure 26: Class Distribution After SMOTE(Potato (Healthy & Late Blight) dat aset)

Table 12 describes the cross-validation report for each fold, where the average metri cs (precision, recall, and F1-score) are calculated using the macro average. This ensu res that each class contributes equally to the average. Figure 27 shows the cross-vali dation curve.

Table 12: Cross-validation report for each fold (Potato (Healthy & Late Blight) data set)

Fold No.	Accuracy Score	F1-Score	Cohen Kappa Score
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	1.00	1.00	1.00
6	96.55	96.55	93.10
7	1.00	1.00	1.00
8	1.00	1.00	1.00
9	1.00	1.00	1.00
10	1.00	1.00	1.00

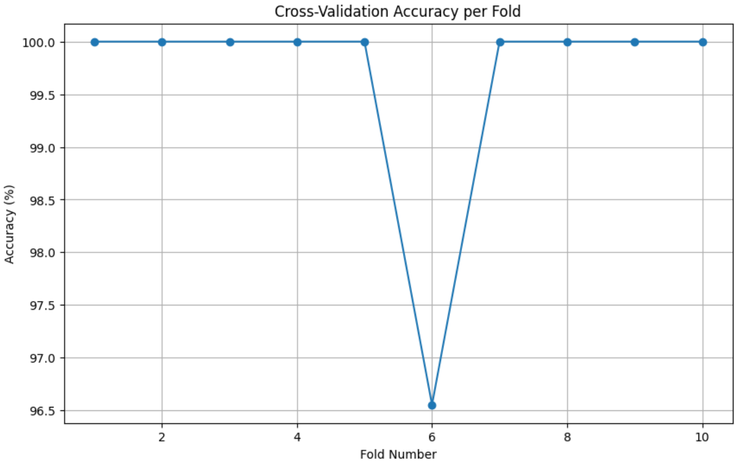


Figure 27: Cross-validation curve (Potato (Healthy & Late Blight) dataset)

Table 13 describes the classification report generated for Potato (Healthy & Late Blight) dataset.

Table 13: Classification Report (Potato (Healthy & Late Blight) dataset)

	Precision	Recall	F1-Score	Support
Potato_healthy	1.00	1.00	1.00	290
Potato_Late_blight	1.00	1.00	1.00	290
accuracy			1.00	580
Macro-average	1.00	1.00	1.00	580
Weighted-Average	1.00	1.00	1.00	580

Figure 28 illustrates the receiver-operating characteristic curve for each class in the Potato (Healthy & Late Blight) dataset.

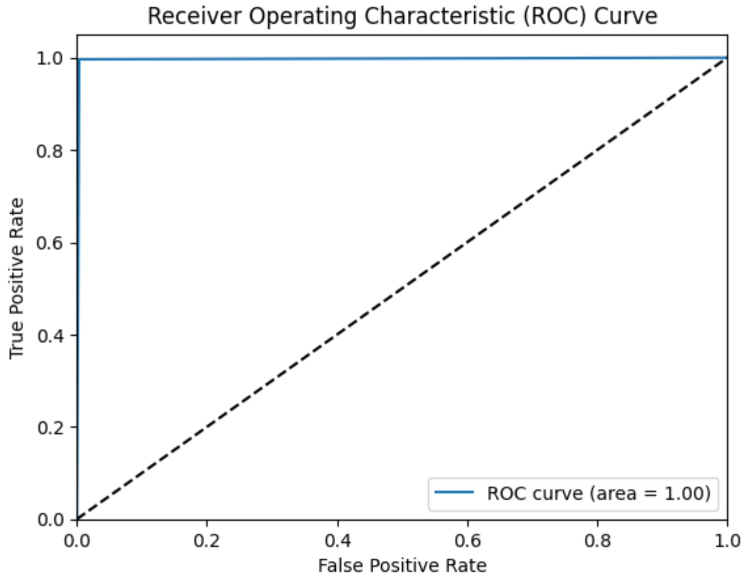


Figure 28: ROC curve (Potato (Healthy & Late Blight) dataset)

Figure 29 exhibits the Precision-Recall curve for each class in the Potato (Healthy & Late Blight) dataset.

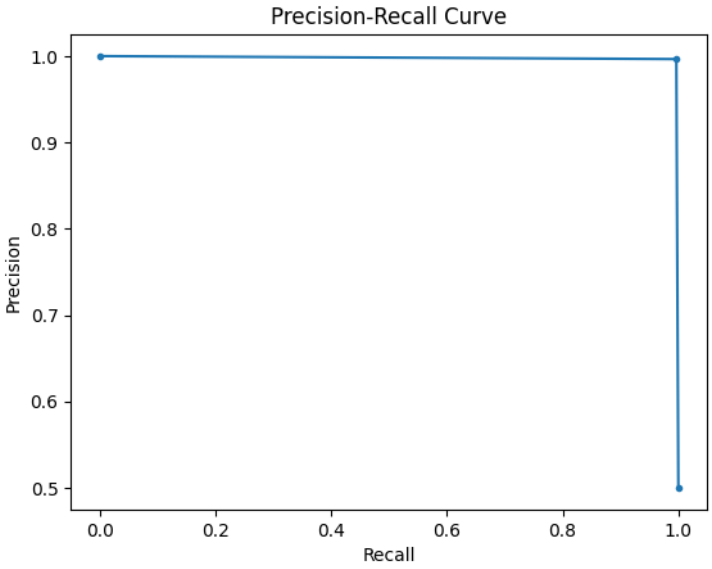
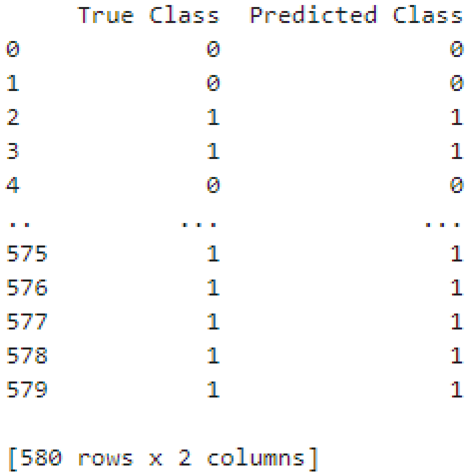


Figure 29: Precision-Recall curve (Potato (Healthy & Late Blight) dataset)

Figure 30 depicts the model prediction report with true and predicted classes.



```
     True Class  Predicted Class
0             0                0
1             0                0
2             1                1
3             1                1
4             0                0
..          ...              ...
575           1                1
576           1                1
577           1                1
578           1                1
579           1                1

[580 rows x 2 columns]
```

Figure 30: Model prediction Report (Potato (Healthy & Late Blight) dataset)

4.5 Apple Leaf Diseases Dataset

Table 14 provides a description of the dataset.

Table 14: Description of Apple Leaf Diseases Dataset

Disease Types	**Assigned Class label**	**Number of Images**	**Total Training Images**	**Total Testing Images**
Apple_Black_rot	0	170	136	34
Apple_Cedar_apple_rust	1	160	128	32
Apple_Apple_scab	2	150	120	30

The Original class distribution before applying SMOTE: Counter({0: 136, 1: 12 8, 2: 120}). Figure 31 displays the class distribution before SMOTE.

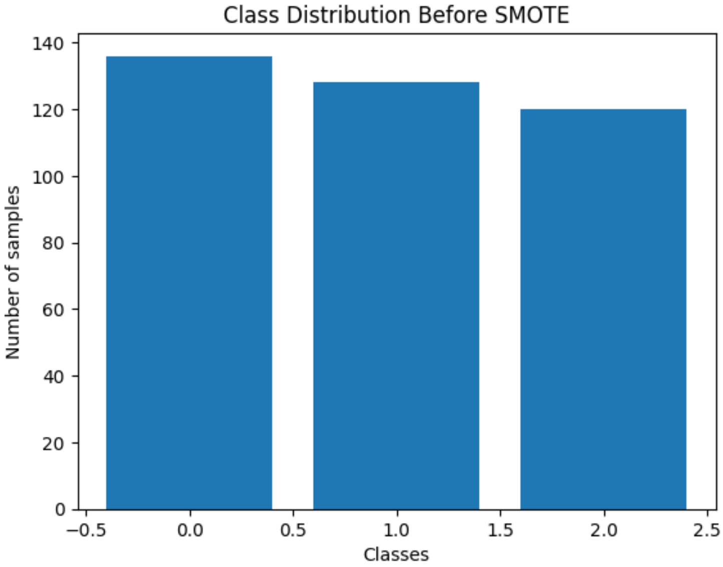


Figure 31: Class Distribution Before SMOTE (Apple Leaf Diseases dataset)

Balanced class distribution after applying SMOTE: Counter({1: 136, 2: 136, 0: 1 36}). Figure 32 displays the class distribution after SMOTE.

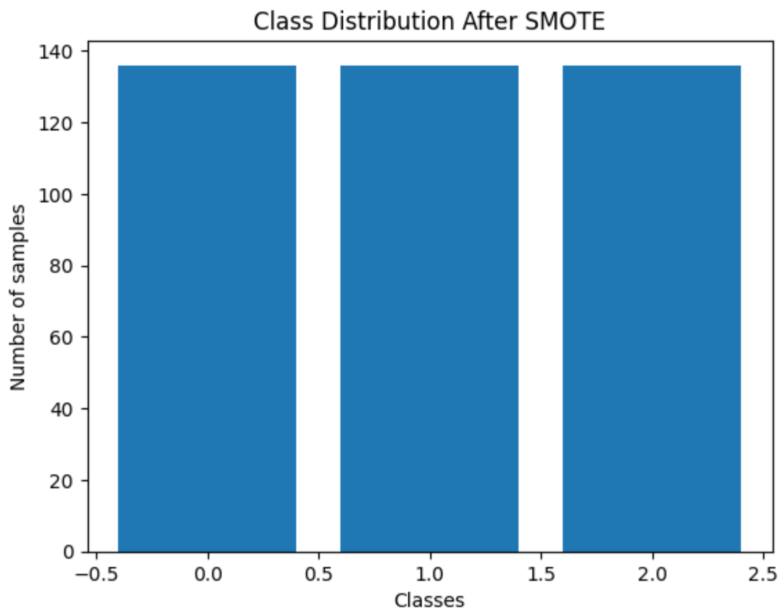


Figure 32: Class Distribution After SMOTE (Apple Leaf Diseases dataset)

Table 15 describes the cross-validation report for each fold, where the average metrics are calculated using the macro average. This ensures that each class contributes equally to the average. Figure 33 shows the cross-validation curve.

Table 15: Cross-validation report for each fold (Apple Leaf Diseases dataset)

Fold No.	Accuracy Score	F1-Score	Cohen Kappa Score
1	87.80	88.24	81.19
2	97.56	97.84	96.27
3	92.68	92.13	88.56
4	95.12	94.87	92.66
5	97.56	96.96	92.70
6	95.12	95.24	96.19
7	92.68	92.79	92.49
8	95.12	94.81	88.80
9	95	93.73	92.46
10	90	89.98	92.21

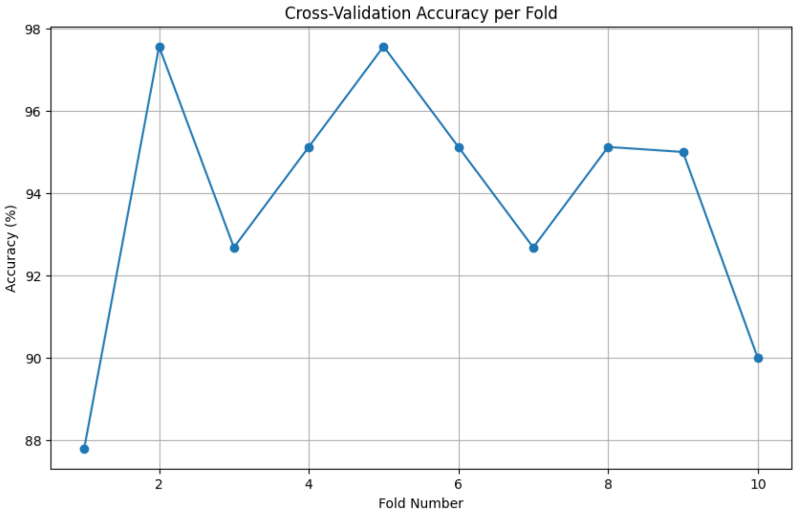


Figure 33: Cross-validation curve (Apple Leaf Diseases dataset)

Table 16 describes the classification report generated for Apple Leaf Diseases dataset.

Table 16: Classification Report (Apple Leaf Diseases dataset)

	Precision	**Recall**	**F1-Score**	**Support**
Apple_Black_rot	0.90	0.95	0.92	136
Apple_Cedar_apple_rust	0.98	0.96	0.97	136
Apple_Apple_scab	0.93	0.90	0.92	136
accuracy			0.94	408
Macro-average	0.94	0.94	0.94	408
Weighted-Average	0.94	0.94	0.94	408

Figure 34 illustrates the receiver-operating characteristic curve for each class in the Apple Leaf Diseases dataset.

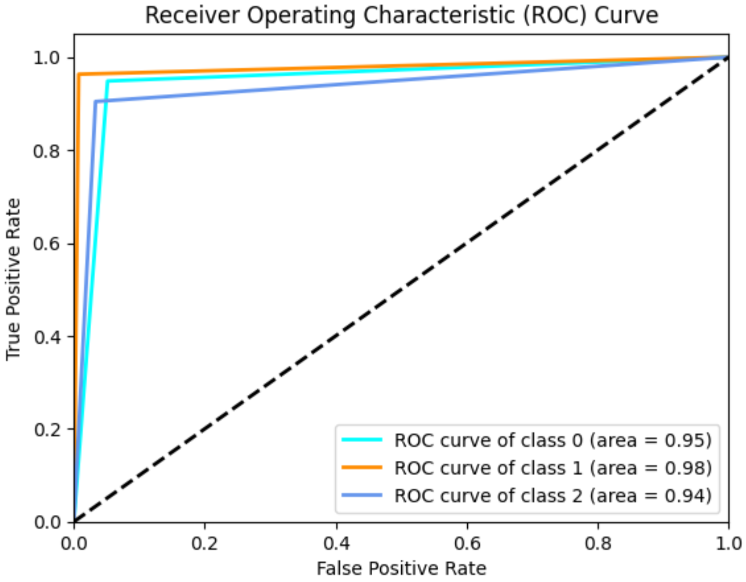


Figure 34: ROC curve (Apple Diseases dataset)

Figure 35 exhibits the Precision-Recall curve for each class in the Apple Leaf Diseases dataset.

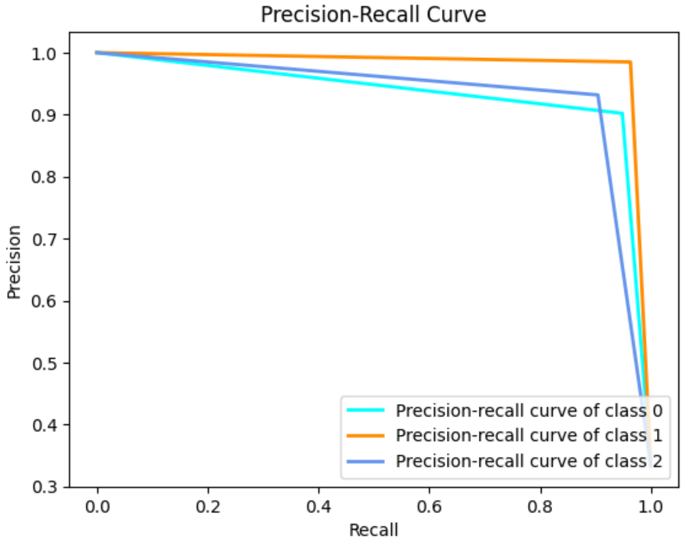
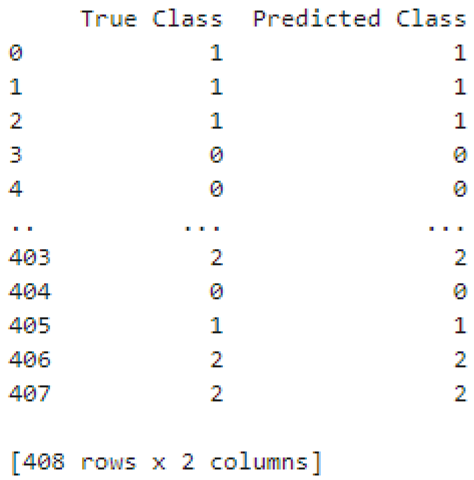


Figure 35: Precision-Recall curve (Apple Leaf Diseases dataset)

Figure 36 depicts the model prediction report with true and predicted classes.



```
     True Class  Predicted Class
0             1                1
1             1                1
2             1                1
3             0                0
4             0                0
..          ...              ...
403           2                2
404           0                0
405           1                1
406           2                2
407           2                2

[408 rows x 2 columns]
```

Figure 36: Model prediction Report (Apple Leaf Diseases Dataset)

4.6 COMPARATIVE ANALYSIS

This section analyses the performance of the proposed model on five distinct datasets given in Table 17: the PlantVillage dataset, the Potato Leaf Disease dataset, the Apple Disease dataset, the Apple Leaf Diseases dataset, and another Potato (Healthy & Late Blight) dataset.

Table 17: Model performance over all datasets

Dataset Name	**Mean Accuracy over all folds**
PlantVillage	98.24
Apple Disease	96.21
Potato Leaf Disease	94.5
Potato (Healthy & Late Blight)	99.65
Apple Leaf Diseases	93.86

The comparative analysis reveals that the proposed methodology is effective and reliable across different datasets, achieving good accuracy consistently. The slight variations in accuracy can be attributed to differences in dataset characteristics, such as image quality, disease variability, and class distribution.

CHAPTER 5

5.0 CONCLUSION & FUTURE SCOPE

5.1 Conclusion

This work incorporates a multi-stage methodology for robust image classification, particularly suited for imbalanced datasets. A meticulous preprocessing pipeline ensures data consistency through grayscale conversion, inverse Gaussian gradient, Geodesic Active Contour (GAC), and background removal. EfficientNetB0, a pre-trained CNN, extracts informative features from the pre-processed images. Dimensionality reduction is achieved using PCA to optimize training efficiency. SMOTE tackles class imbalance by generating synthetic data for under-represented classes. A stacked ensemble classifier leverages the strengths of Random Forest, K-Nearest Neighbour, and SVM, with a final XGBoost layer for enhanced classification accuracy.

The findings presented in Chapter 4 showcases the efficacy and resilience of the proposed methodology, emphasising its capacity to precisely detect plant diseases across various segments of the dataset. Macro averaging is employed to ensure that the performance measurements accurately represent the classifier's capacity to handle all classes, even ones that may be underrepresented. Therefore, it tackles important obstacles such as accurate division, complex data representation, uneven distribution of classes, and the requirement for reliable categorization.

5.2 Future Scope

Building on the success of this research, future work will explore the following things:

- Extend the methodology to cover additional plant species and diseases.
- Implement real-time disease detection capabilities using mobile and IoT devices.
- Explore the integration of environmental and sensor data to further enhance disease prediction accuracy.

REFERENCES

[1] Ait Elkadi, K., Bakouri, S., Belbrik, M., Hajji, H., Chtaina, N.: Experimentation of model for early detection of tomato diseases by deep learning. Rev. Marocain. Protect. Plant. 14, 19–30 (2020)

[2] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification,” Computational Intelligence and Neuroscience, vol. 2016, pp. 1–11, 2016, doi: https://doi.org/10.1155/2016/3289801.

[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using Deep Learning for Image-Based Plant Disease Detection,” Frontiers in Plant Science, vol. 7, Sep. 2016, doi: https://doi.org/10.3389/fpls.2016.01419.

[4] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” Computers and Electronics in Agriculture, vol. 145, pp. 311–318, Feb. 2018, doi: https://doi.org/10.1016/j.compag.2018.01.009.

[5] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” Computers and Electronics in Agriculture, vol. 147, pp. 70–90, Apr. 2018, doi: https://doi.org/10.1016/j.compag.2018.02.016.

[6] J. K. Patil and R. Kumar, “Analysis of content based image retrieval for plant leaf diseases using color, shape and texture features,” Engineering in Agriculture, Environment and Food, vol. 10, no. 2, pp. 69–78, Apr. 2017, doi: https://doi.org/10.1016/j.eaef.2016.11.004.

[7] B. Liu, Y. Zhang, D. He, and Y. Li, “Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks,” Symmetry, vol. 10, no. 1, p. 11, Jan. 2018, doi: https://doi.org/10.3390/sym10010011.

[8] A. Anagnostis, P. Remagnino and P. Wilkin, “Unsupervised learning for plant health anomaly detection,” Computers and Electronics in Agriculture, vol. 11, no. 2, pp. 69–78, Apr. 2020.

[9] L. Zhang, Z. Zhang, Y. Luo, J. Cao, R. Xie, and S. Li, “Integrating satellite-derived climatic and vegetation indices to predict smallholder maize yield using deep learning,” Agricultural and Forest Meteorology, vol. 311, p. 108666, Dec. 2021, doi: 10.1016/j.agrformet.2021.108666.

[10] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, “Machine Learning in Agriculture: A Comprehensive Updated review,” *Sensors*, vol. 21, no. 11, p. 3758, May 2021, doi: 10.3390/s21113758.

[11] M. M. Siddique, M. J. Nazar, and K. Farooq, “Analysis of deep learning algorithms for detection and classification of tomato leaf diseases,” Mar. 29, 2023. https://www.jcbi.org/index.php/Main/article/view/237

[12] J. G. A. Barbedo, “Factors influencing the use of deep learning for plant disease recognition,” Biosystems Engineering, vol. 172, pp. 84–91, Aug. 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[13] J. Huang, W. Sun, and L. Huang, “Deep neural networks compression learning based on multiobjective evolutionary algorithms,” Neurocomputing, vol. 378, pp. 260–269, Feb. 2020, doi: 10.1016/j.neucom.2019.10.053.

[14] M. Akbar, M. Ullah, B. Shah, R.U. Khan, T. Hussain, F. Ali, F. Alinezi, I. Syed, K.S.Kwak, “An effective deep learning approach for the classification of Bacteriosis in peach leave,” Frontiers in Plant Science, vol. 13, Nov. 2022, doi: 10.3389/fpls.2022.1064854.

[15] J.Tian, Q. Hu, X.X Ma, M.Han, “An improved kpca/ga-svm classification model for plant leaf disease recognition” vol. 8(18), pp. 7737 - 7745, Aug 2012.

[16] S. Gulavnai, R. Patil, “ Deep learning for image based mango leaf disease detection” Int. J. Recent Technol. Eng. vol. 8(3S3), pp. 54–56, Sep 2019.

[17] A. Fuentes, J. Lee, Y. Lee, S. Yoon and D. S. Park, “Anomaly detection of plant diseases and insects using convolutional neural networks,” Proceedings of the International Society for Ecological Modelling Global Conference, Apr 2017.

[18] V. K. Shrivastava and M. K. Pradhan, “Rice plant disease classification using color features: a machine learning paradigm,” Journal of Plant Pathology, vol. 103, no. 1, pp. 17–26, Oct. 2020, doi: https://doi.org/10.1007/s42161-020-00683-3.

[19] M. Kumar, A. Kumar, and V. S. Palaparthy, “Soil Sensors Based Prediction System for Plant Diseases using Exploratory Data Analysis and Machine Learning,” IEEE Sensors Journal, pp. 1–1, 2020, doi: https://doi.org/10.1109/jsen.2020.3046295.

[20] M. Mishra, P. Choudhury, and B. Pati, “Modified ride-NN optimizer for the IoT based plant disease detection,” Journal of Ambient Intelligence & Humanized Computing/Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 1, pp. 691–703, Jun. 2020, doi: 10.1007/s12652-020-02051-6.

[21] T.N. Pham, L.V. Tran, S.V.T. Dao, “Early disease classification of mango leaves using Feed-Forward neural network and hybrid metaheuristic feature

selection,” IEEE Journals & Magazine | IEEE Xplore, 2020. https://ieeexplore.ieee.org/abstract/document/9229136/

[22] U.P. Singh, S.S. Chouhan, S. Jain, S. Jain,“Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease,” IEEE Journals & Magazine | IEEE Xplore, 2019. https://ieeexplore.ieee.org/abstract/document/8675730/

[23] G. G and A. P. J, “Identification of plant leaf diseases using a nine-layer deep convolutional neural network,” Computers & Electrical Engineering, vol. 76, pp. 323–338, Jun. 2019, doi: 10.1016/j.compeleceng.2019.04.011.

[24] G. Dai, J. Fan, Z. Tian, and C. Wang, “PPLC-Net:Neural network-based plant disease identification model supported by weather data augmentation and multi-level attention mechanism,” Journal of King Saud University. Computer and Information Sciences/Mağalaẗ Ğam’aẗ Al-malīk Saud : Ùlm Al-ḥasib Wa Al-ma’lumat, vol. 35, no. 5, p. 101555, May 2023, doi: 10.1016/j.jksuci.2023.101555.

[25] P. B. R., Aiswarya V. V., “Tomato leaf disease detection and classification using CNN,” *philstat.org*, Sep. 2022, doi: 10.17762/msea.v71i4.853.

[26] M. Saberi Anari, “A Hybrid Model for Leaf Diseases Classification Based on the Modified Deep Transfer Learning and Ensemble Approach for Agricultural AIoT-Based Monitoring,” Computational Intelligence and Neuroscience, vol. 2022, pp. 1–15, Apr. 2022, doi: https://doi.org/10.1155/2022/6504616.

[27] V. Singh and A. K. Misra, “Detection of plant leaf diseases using image segmentation and soft computing techniques,” Information Processing in Agriculture, vol. 4, no. 1, pp. 41–49, Mar. 2017, doi: https://doi.org/10.1016/j.inpa.2016.10.005.

[28] E. Saraswathi and J. FarithaBanu, “A novel ensemble classification model for plant disease detection based on Leaf images,” IEEE Conference Publication | IEEE Xplore, Jan. 05, 2023. https://ieeexplore.ieee.org/abstract/document/10083902

[29] S. Garg and P. Singh, “An aggregated loss function based lightweight few shot model for plant leaf disease classification,” Multimedia Tools and Applications, vol. 82, no. 15, pp. 23797–23815, Feb. 2023, doi: 10.1007/s11042-023-14372-7.

[30] Hirenkumar Kukadiya and Divyakant Meva, “Automatic Cotton Leaf Disease Classification and Detection by Convolutional Neural Network,”

Communications in computer and information science, pp. 247–266, Jan. 2022, doi: https://doi.org/10.1007/978-3-031-23092-9_20.

[31] O. Attallah, “Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection,” Horticulturae, vol. 9, no. 2, pp. 149, Jan. 2023, doi: 10.3390/horticulturae9020149.

[32] M. S. A. M. Al-gaashani, F. Shang, M. S. A. Muthanna, M. Khayyat, and A. A. Abd El-Latif, “Tomato leaf disease classification by exploiting transfer learning and feature concatenation,” IET Image Processing, vol. 16, no. 3, pp. 913–925, Jan. 2022, doi: https://doi.org/10.1049/ipr2.12397.

[33] David. P. Hughes and M. Salathe, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” arXiv.org, Nov. 25, 2015. https://arxiv.org/abs/1511.08060

[34] “Apple Disease Dataset,” *Kaggle*, Apr. 09, 2022. https://www.kaggle.com/datasets/ludehsar/apple-disease-dataset

[35] “Potato Leaf,” *Kaggle*, Dec. 08, 2019. https://www.kaggle.com/datasets/abbasataiemontazer/potato-leaf

[36] N. Tilahun, “Potato leaf (Healthy and late blight),” *Mendeley Data*, May 2022, doi: 10.17632/v4w72bsts5.2.

[37] “Apple leaf diseases,” *Kaggle*, Mar. 29, 2021. https://www.kaggle.com/datasets/mhantor/apple-leaf-diseases

APPENDIX

```
print(X.shape)
```

```
(7890, 224, 224, 3)
```

```
print(Y.shape)
```

```
(7890,)
```

```
# Encode labels to one-hot vectors
label_encoder = LabelEncoder()
labels_encoded = label_encoder.fit_transform(Y)
labels_one_hot = to_categorical(labels_encoded)

print("X_train shape after adding channel dimension:", X_train.shape)
print("X_test shape after adding channel dimension:", X_test.shape)
```

```
X_train shape after adding channel dimension: (6312, 224, 224, 1)

X_test shape after adding channel dimension: (1578, 224, 224, 1)
```

```
# Create a new model using only the feature extraction part

feature_extractor           =           Model(inputs=efficientnet_b0.inputs,
outputs=efficientnet_b0.get_layer('top_conv').output)

# Extract features directly from the preprocessed array

features_train = feature_extractor.predict(X_train)

print("Feature_train vector shape:", features_train.shape)

print("Feature_test vector shape:", features_test.shape)
```

```
Feature_train vector shape: (6312, 7, 7, 1280)

Feature_test vector shape: (1578, 7, 7, 1280)
```

```
print("Reshaped training features:", features_train_reshaped.shape)

print("Reshaped test features:", features_test_reshaped.shape)
```

```
Reshaped training features: (6312, 62720)

Reshaped test features: (1578, 62720)
```

```
# Explained Variance:
print("Explained variance ratio:", pca.explained_variance_ratio_)
print("Total variance explained:", np.sum(pca.explained_variance_ratio_))
```

```
Explained variance ratio: [8.63142908e-01 2.89842561e-02 2.33939085e-02
1.46710556e-02
1.29938591e-02 8.54799431e-03 4.98395134e-03 3.60348891e-03
3.23499111e-03 2.42885528e-03 2.30272068e-03 1.83549360e-03
1.68774545e-03 1.57788233e-03 1.45881460e-03 1.31289009e-03
1.17722619e-03 1.15121331e-03 9.71494999e-04 9.48079920e-04
8.51398217e-04 8.17020948e-04 8.04377429e-04 7.07916683e-04
6.65630156e-04 6.49353431e-04 5.98697923e-04 5.92905970e-04
5.40933630e-04 5.26225718e-04 4.87493438e-04 4.60755604e-04
4.32121829e-04 4.07343119e-04 3.91527690e-04 3.88322427e-04
3.76310752e-04 3.42323765e-04 3.23737506e-04 3.03140900e-04
2.94729194e-04 2.86067370e-04 2.67501106e-04 2.51608522e-04
2.48926139e-04 2.34409294e-04 2.27756580e-04 2.19805384e-04
2.13151667e-04 2.04440061e-04 2.00880007e-04 1.91724670e-04
1.81768410e-04 1.75961773e-04 1.65579622e-04 1.55175279e-04
1.51285189e-04 1.47030994e-04 1.38691335e-04 1.35447335e-04
1.26221144e-04 1.26068786e-04 1.18436961e-04 1.15118572e-04
1.11446090e-04 1.08794477e-04 1.02900216e-04 9.75499279e-05
9.49977257e-05 9.36616852e-05 9.06513524e-05 8.83751854e-05
8.56866827e-05 8.34864331e-05 7.97027315e-05 7.72369240e-05
7.62303971e-05 7.32467088e-05 7.09321175e-05 6.91067762e-05
6.71951930e-05 6.43380263e-05 6.20231876e-05 6.16146281e-05
5.81223649e-05 5.55184706e-05 5.48545868e-05 5.35243162e-05
5.07982732e-05 5.03687588e-05 4.81365569e-05 4.77141439e-05
4.70259911e-05 4.59364092e-05 4.39888026e-05 4.30519540e-05
4.11141409e-05 3.95436800e-05 3.94207636e-05 3.87456603e-05]
Total variance explained: 0.9980712
```

```
print("Reduced shape of training features:", features_train_reduced.shape)
print("Reduced shape of test features:", features_test_reduced.shape)
```

```
Reduced shape of training features: (6312, 100)
Reduced shape of test features: (1578, 100)
```

```
# Check the distribution of the classes

print(f"Original class distribution: {counter}")
```

```
Original class distribution: Counter({31: 182, 30: 178, 18: 178, 35: 176, 27:
175, 26: 174, 12: 173, 10: 173, 14: 172, 33: 172, 37: 171, 34: 169, 4: 169, 11:
168, 20: 168, 29: 168, 6: 168, 23: 167, 17: 167, 1: 167, 9: 167, 15: 166, 5:
165, 22: 164, 19: 164, 8: 164, 16: 164, 21: 163, 13: 163, 36: 162, 25: 161, 0:
160, 3: 160, 38: 159, 28: 158, 7: 158, 32: 156, 24: 121, 2: 2})
```

```
# Check the distribution of the classes after SMOTE

print(f"Balanced class distribution: {counter_balanced}")
```

```
Balanced class distribution: Counter({38: 182, 23: 182, 0: 182, 22: 182, 17:
182, 1: 182, 11: 182, 35: 182, 27: 182, 3: 182, 14: 182, 20: 182, 36: 182, 29:
182, 6: 182, 30: 182, 19: 182, 8: 182, 12: 182, 31: 182, 5: 182, 28: 182, 25:
182, 15: 182, 26: 182, 9: 182, 37: 182, 21: 182, 13: 182, 7: 182, 32: 182, 34:
182, 16: 182, 33: 182, 10: 182, 18: 182, 4: 182, 24: 182, 2: 182})
```

```
# Define base classifiers

classifier1 = RandomForestClassifier(n_estimators=100, random_state=42)

classifier2 = KNeighborsClassifier()
classifier3 = SVC(probability=True, random_state=42)
```

```
# Define the stacking classifier

stacking_classifier = StackingClassifier(

    estimators=[

        ('rf', classifier1),

        ('knn', classifier2),

        ('svc', classifier3)

    ],

    final_estimator=XGBClassifier(objective='binary:logistic',
                                  n_estimators=100
                                  learning_rate=0.1)


)
```