

A STUDY ON PLANT DISEASE CLASSIFICATION USING AN ENSEMBLE OF DEEP LEARNING MODELS

Project submitted in partial fulfillment of the requirements for the degree of
Master of Computer Application
Jadavpur University, 2024

by

Anupama Halder

Exam Roll Number - MCA244006

Class Roll Number - 002210503006

Registration Number - 163598 of 2022-2023

under the guidance of

Prof. (Dr.) Kamal Sarkar

in the

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University

Kolkata - 700032, India

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF RECOMMENDATION

TO WHOM IT MAY CONCERN

I hereby recommended that the project report entitled '**A Study On Plant Disease Classification Using An Ensemble Of Deep Learning Models**' prepared by Anupama Halder (Reg. No - 163598 of 2022-2023, Roll No - 002210503006, Exam Roll - MCA244006), under my guidance and supervision to be accepted in partial fulfillment of the degree of Master of COMPUTER APPLICATION in DEPARTMENT of COMPUTER SCIENCE and ENGINEERING, JADAVPUR UNIVERSITY during the academic session 2022-2024.

Prof. (Dr.) Kamal Sarkar
Project Supervisor
Dept. of Computer Science and Engineering
Jadavpur University, Kolkata - 700032

Prof. (Dr.) Nandini Mukhopadhyay
Head of the Department
Dept. of Computer Science and Engineering
Jadavpur University, Kolkata - 700032

Prof. Dipak Laha
Dean, Faculty Council
of Engineering and Technology
Jadavpur University, Kolkata - 700032

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the project report entitled '**A Study On Plant Disease Classification Using An Ensemble Of Deep Learning Models**' is bonafide record of work out by carried out by ANUPAMA HALDER, in fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted.

Signature of Examiner 1

Date:

Signature of Examiner 2

Date:

Faculty of Engineering and Technology
Jadavpur University

DECLARATION OF COMPLIANCE WITH ACADEMIC ETHICS

I hereby declare that this project report entitled "A Study On Plant Disease Classification Using An Ensemble Of Deep Learning Models" is an outcome of my genuine and independent work, as part of my Master of Computer Application program.

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

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

Name: Anupama Halder

Class Roll Number: 002210503006

Exam Roll Number: MCA244006

Registration Number: 163598 of 2022-2023

Project Title: A Study On Plant Diseases Classification Using An Ensemble Of Deep Learning Models

Signature with Date

ACKNOWLEDGEMENT

I would first like to thank my thesis advisor Dr. Kamal Sarkar, Professor of the Department of Computer Science and Engineering, Jadavpur University for his guidance and support during the course of this project. It has been a constant learning experience working under Dr. Sarkar's guidance. I am immensely grateful to him for motivating me to study Machine Learning and providing his invaluable inputs in this thesis.

I would like to express my sincere gratitude to all the professors of the Department of Computer Science and Engineering, Jadavpur University, Kolkata for the guidance and knowledge they provided me with throughout the duration of the Master of Computer Applications course.

I am also indebted to my fellow classmates and my family for their constant support and understanding.

Last but not the least, I must acknowledge the academic resources that I have received from Jadavpur University. I would like to thank the administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

(Anupama Halder)

Date:

INDEX

	<i>Page Number</i>
Certificate of Recommendation	i
Certificate of Approval	ii
Declaration of Compliance with Academics Ethics	iii
Acknowledgement	iv
1. Introduction	1
1.1. Applications.....	2
1.2. Key Objectives and Contributions of this Project.....	2
1.3. Problem Definition.....	2
1.4. Individual Classifier Vs Ensemble Classifier.....	3
1.5. Contributions of this Project.....	4
1.6. Organizations of this Project.....	4
2. Literature Survey	5
2.1. Single Deep Learning Model based Approaches.....	5
2.2. Ensemble Model based Approaches.....	5
2.3. Gap Analysis.....	5
3. Proposed Methodology	10
3.1. Pre-trained Model.....	10
3.2. Summary of the work.....	11
3.3. VGG16 Model.....	13
3.3.1. Architecture of VGG16.....	13
3.4. ResNet50 Model.....	14
3.4.1. Architecture of ResNet50.....	14

3.5. DenseNet121 Model.....	15
3.5.1. Architecture of DenseNet121.....	15
3.6. Model Development.....	16
3.6.1. Transfer Learning.....	16
3.6.2. Data Augmentation.....	16
3.6.3. Model 1: VGG16-Based Model.....	17
3.6.4. Model 2: ResNet50-Based Model.....	17
3.6.5. Model 3: Densenet121-Based Model.....	17
3.7. Non-trainable Ensemble Strategies.....	18
3.7.1. Majority Voting Rule.....	19
3.7.2. Max Rule.....	19
3.7.3. Average Rule.....	20
3.7.4. Product Rule.....	20
4. Evaluation and Results.....	21
4.1. Classification Report Analysis.....	21
4.2. Dataset Description.....	23
4.2.1. Dataset-1.....	23
4.2.2. Dataset-2.....	26
4.2.3. Data Preprocessing.....	29
4.3. Results.....	29
4.3.1. Performance on Dataset-1.....	29
4.3.2. Performance on Dataset-2.....	30
4.4. Hyperparameter Tuning.....	31
4.5. Comparison with existing work.....	31
5. Conclusion and Future Work.....	33

LIST OF FIGURES

Page Number

1. Architecture of Proposed Ensemble of Deep Learning Models for Plant Disease classification.....	12
2. Architecture of VGG16 Model.....	13
3. Architecture of ResNet50 Model.....	14
4. Architecture of DenseNet121 Model.....	15
5. Class-wise Distribution of the Training Samples of Dataset 1.....	25
6. Class-wise Distribution of the Validation Set of Dataset 1.....	25
7. Class-wise Distribution of the Test Set of Dataset 1.....	26
8. Class-wise Distribution of the Training Samples of Dataset 2.....	27
9. Class-wise Distribution of the Validation Set of Dataset 2.....	28
10. Class-wise Distribution of the Test Set of Dataset 2.....	28

LIST OF TABLES

	<i>Page Number</i>
1. Summary of related work.....	6
2. Results of each Pre-trained Deep Learning Model on Dataset-1.....	29
3. Result of each proposed ensemble method on Dataset-1.....	29
4. Results of each Pre-trained Deep Learning Model on Dataset-2.....	30
5. Result of each proposed ensemble method on Dataset-2.....	30
6. Hyperparameter Settings for Various Model Configurations.....	31
7. Comparison with existing and related works.....	32

Abstract

This project presents a comprehensive study on the classification of plant diseases using an ensemble of deep learning models. Timely and accurate identification of plant diseases is critical for maintaining crop health and maximizing agricultural productivity. Leveraging pre-trained convolutional neural networks (CNNs) such as VGG16, ResNet50, and DenseNet121, this research aims to construct a robust ensemble classifier. The study investigates the influence of different activation functions (sigmoid and softmax) in the final layers of individual models and explores various ensemble combination rules (majority voting, average, maximum, and product). The proposed approach is evaluated on two diverse plant disease datasets: New Plant Disease Dataset and Tomato Leaf Disease Dataset. Experimental results demonstrate the effectiveness of the ensemble model in achieving superior classification performance compared to individual models, underscoring the potential of deep learning ensembles in advancing plant disease diagnostics.

Chapter 1

Introduction

Plant diseases constitute a significant threat to global food security, inflicting substantial yield losses and economic hardships on farmers [19]. The imperative for early and precise disease detection is underscored by Mohanty et al. (2016), highlighting the pivotal role of timely interventions in mitigating their impact [20]. Traditional diagnostic methods, reliant on labor-intensive visual inspection by experts, are fraught with subjectivity and time inefficiency [21]. However, the advent of deep learning, a subset of artificial intelligence, has heralded a paradigm shift in image recognition and classification, offering novel avenues for automated plant disease detection [22].

Deep learning models, particularly convolutional neural networks (CNNs), have exhibited remarkable efficacy in various image-based tasks owing to their capacity to autonomously discern intricate features from raw data [23]. In the realm of plant disease identification, these models have demonstrated promising outcomes, often surpassing conventional methods in terms of accuracy and efficiency [24]. Nonetheless, individual deep learning models may harbor limitations such as susceptibility to overfitting or vulnerability to specific error types [25].

To surmount these challenges and augment the efficacy of plant disease classification, this thesis advocates for the adoption of an ensemble approach. Ensembles amalgamate predictions from multiple models, harnessing their diverse strengths while mitigating individual weaknesses [26]. By synthesizing outputs from distinct models, ensembles frequently yield superior accuracy and robustness compared to standalone models [27].

1.1. Applications:

This chapter commences with an elucidation of the project's applicability in agricultural disease management. The delineation of the problem underscores the challenges associated with precise disease diagnosis and their ramifications for crop vitality and productivity. The significance of ensemble classifiers over individual counterparts is expounded upon, accentuating their capacity to amalgamate diverse perspectives and enhance overall performance.

1.2. Key Objectives and Contributions of this Project:

- **Comprehensive Literature Review:** Conduct a thorough review of existing research on plant disease classification using deep learning, identifying current trends, challenges, and opportunities.
- **Ensemble Model Design:** Design and implement an ensemble framework that effectively combines the outputs of multiple deep learning models, exploring various ensemble techniques such as voting, stacking, and bagging (Zhou, 2012).
- **Model Evaluation and Comparison:** Evaluate the performance of the proposed ensemble model on the aforementioned dataset, comparing its accuracy, precision, recall, and F1 score with individual models and other state-of-the-art methods.
- **Robustness Analysis:** Assess the robustness of the ensemble model to variations in image quality, lighting conditions, and disease severity, investigating its ability to generalize to real-world scenarios.
- **Practical Implementation:** Develop a user-friendly software tool or web application that incorporates the ensemble model, enabling farmers and agricultural practitioners to easily upload images of diseased plants and obtain accurate diagnoses.

1.3. Problem Definition:

Plant diseases are a major threat to agricultural productivity and food security worldwide. Timely and accurate identification of these diseases is essential for effective disease management and minimizing crop losses. However, traditional manual inspection methods are time-consuming, labor-intensive, and prone to human error.

1.4. Individual Classifier Vs Ensemble Classifier:

- **Number of Models:** Individual classifiers employ a **single model** for prediction whereas ensemble classifiers combine multiple models for prediction.
- **Complexity:** Individual classifiers are generally simpler to implement and train, as only one model needs to be optimized. Ensemble classifiers are more complex due to the need to train and manage multiple models, and then find an effective way to combine their predictions.
- **Interpretability:** Individual classifiers are often easier to interpret, as the decision-making process of a single model is more transparent and can be analyzed directly. Ensemble classifiers are Less interpretable, as the final prediction is a combination of multiple models' decisions, making it difficult to pinpoint the exact reasoning behind a specific classification.
- **Computational Cost:** Individual classifiers typically require less computational power for training and prediction due to the use of a single model. Ensemble classifiers are more computationally expensive due to the need to train and run multiple models, and then aggregate their predictions.
- **Accuracy:** Individual classifiers may have limited accuracy, as it relies on the performance of a single model, which might not capture all the complexities of the data. Ensemble classifiers often achieve higher accuracy than any individual model, as it leverages the diverse strengths of multiple models and reduces the impact of individual errors.
- **Robustness:** Individual classifiers can be less robust to variations in the data or noise, as it relies on a single model's ability to generalize. Ensemble classifiers are more robust to data variations and noise, as the combined predictions of multiple models can mitigate the impact of individual errors and improve generalization.
- **Bias:** Individual classifiers may be biased towards the specific training data used, leading to poor performance on unseen data. Ensemble classifiers can help reduce bias by averaging or voting the predictions of multiple models trained on different subsets of data or using different algorithms.

1.5. Contributions of this Project:

This thesis makes the following significant contributions to the field of plant disease classification:

- Utilization of Two Datasets:** We leveraged two distinct datasets for our study. The first dataset, termed as "New Plant Diseases Dataset" (dataset-1) from kaggle [13], consists of a more extensive collection of 87,000 images spanning 38 classes, encompassing a broader spectrum of plant diseases. The second dataset, referred to as "Tomato Leaf Disease Detection" (dataset-2) from kaggle [12], comprises 10,000 images categorized into 10 classes representing various tomato leaf diseases.
- Integration of Three Pre-trained Models:** We incorporated three widely-used pre-trained convolutional neural network (CNN) architectures - VGG16, ResNet50, and DenseNet121 - into our classification framework. These models were fine-tuned using both sigmoid and softmax activation functions across both dataset 1 and dataset 2. This approach allowed us to exploit the rich feature representations learned by these models on large-scale image datasets, enhancing the performance of our classification system.
- Ensemble Model with Multiple Combiner Rules:** We employed an ensemble learning approach to combine the predictions of individual models, investigating four different combiner rules: Majority Voting, Max Rule, Average Rule, and Product Rule. Our experimental results revealed that the highest accuracy for both Dataset-1 and Dataset-2 was achieved using the Majority Voting rule with the sigmoid activation function, with Dataset-1 reaching an impressive 100% accuracy. This comprehensive analysis highlights the efficacy of the Majority Voting ensemble strategy in significantly improving classification accuracy across diverse datasets and model configurations.

1.6. Organizations of this Project:

In the remainder of the project, chapter 2 gives the literature survey to present the existing and related works, chapter 3 gives a description of 3 pre-trained deep CNN models and model development. Chapter 4 explains the experimental setup and results. Chapter 5 comprises the conclusion.

Chapter 2

Literature Survey

A thorough literature survey is conducted to analyze existing research on plant disease classification using deep learning models. Two subsections are dedicated to individual and ensemble models, covering studies published between 2019 and 2024 in reputable journals and conferences.

2.1. Single Deep Learning Model based Approaches: This section reviews recent research (2019-2024) on plant disease classification using individual deep learning models. It discusses the strengths and limitations of VGG16, ResNet50, DenseNet121, and other relevant architectures. A summary of the existing works that use single deep learning models in Plant disease classification is given in Table 1. In this table, we compare various models depending upon the type of deep learning model used, data size, model performance.

2.2. Ensemble Model based Approaches: This section examines the existing work on ensemble classifiers for plant disease identification. It analyzes various ensemble techniques, such as majority voting, bagging, boosting, and stacking, and discusses their advantages and disadvantages. In Table 1, we present an overview of the various ensemble models applied to plant disease classifications. In this table, models are compared based on the type of ensemble method used, data size and model performance.

2.3. Gap Analysis: The survey highlights the gaps in the existing literature, such as the limited exploration of specific model combinations, activation functions, and ensemble strategies. This thesis aims to address these gaps by investigating the ensemble of VGG16, ResNet50, and DenseNet121 with different activation functions and combination rules on diverse datasets.

Table-1: Summary of related work

Author Name	Short Model Description	Name of Dataset with dataset size	Performance
Akshai KP, et. al., (Year - 2021) [1]	A deep learning model utilizing CNN and pre-trained models (VGG, ResNet, DenseNet) for accurate and fast plant disease classification.	Plant village dataset [4], The dataset consists of 54,305 images of 38 different disease classes of 14 different plant species.	(Best performance by DenseNet model) Precision: 98.31 Recall: 98.27 F1-Score: 98.28 Overall Accuracy: 98.27
Hassan, S.M., Maji, et. al., (Year - 2022) [2]	A deep CNN-based model for plant disease identification using ensemble classifiers, including SVM, KNN, Random Forest, Naive Bayes, and Logistic Regression, demonstrating superior accuracy and efficiency over existing methods.	Plant village dataset [4], The dataset consists of 54,305 images of 38 different disease classes of 14 different plant species.	(Best performance by SVM model) Precision: 93.76 Recall: 96.75 F1-Score: 94.18 Overall Accuracy: 93.80
Chen, J., Zeb, et. al., (Year - 2023) [3]	A convolutional ensemble network, Es-MbNet, integrating SE-MobileNet, Mobile-DANet, and MobileNet V2, uses transfer learning and a two-stage training strategy for superior plant disease identification.	1000 plant disease images containing 466 maize and 500 rice plant disease samples. Along with the Plant village dataset [4], The dataset consists of 54,305 images of 38 different disease classes of 14 different plant species.	Precision: ____ Recall: 98.08 F1-Score: 98.08 Overall Accuracy: 99.61
H. Amin, et. al., (Year - 2022) [5]	An end-to-end deep learning model using EfficientNetB0 and DenseNet121 for corn plant disease identification, achieving 98.56% accuracy with fewer parameters compared to ResNet152 and InceptionV3.	The dataset used in this experiment is a subset of a PlantVillage dataset hosted on Kaggle [6]. It consists of approximately 217,000 images consisting of 38 different categories of both healthy and diseased plant images. Training: 80% Testing: 20%	Precision: 95.00 Recall: 94.00 F1-Score: 95.00 Overall Accuracy: 98.56

Author Name	Short Model Description	Name of Dataset with dataset size	Performance
R. H. Hridoy and M. R. A. Tuli, (Year - 2021) [15]	The model uses an ensemble of three fine-tuned EfficientNet models (B5, B7, and B6) for papaya disease recognition.	A dataset of 138980 images of papaya leaves and fruits is created by utilizing image augmentation techniques from 13898 collected images	Precision: ____ Recall: ____ F1-Score: ____ Overall Accuracy: 99.74
Sachar, S., et. al., (Year - 2022) [7]	Proposing EDL-AMLI, an ensemble classifier integrating MobileNetV2, InceptionV3, and ResNet50 pre-trained models for medicinal leaf identification. Achieves 99.66% accuracy on test set and 99.9% average accuracy with threefold and fivefold cross-validation, surpassing state-of-the-art models.	The medicinal leaf dataset is published in Mendeley and consists of 30 classes.	Precision: ____ Recall: ____ F1-Score: ____ Overall Accuracy: 99.94
Basavaiah, J., et.al., (Year - 2020) [9]	The proposed method uses a fusion of color histograms, Hu Moments, Haralick, and Local Binary Pattern features for leaf disease classification in tomato plants. Random forest and decision tree algorithms are employed.	The dataset includes 300 training images across 5 classes ("Bacterial," "Healthy," "Septoria," "Mosaic," "Yellow Curl") with 60 images each, and 200 test images with 40 images per class, totaling 500 images. Splitted dataset into 60% for training and 40% for testing.	(Best Performance by Random Forest classifier) Precision: ____ Recall: ____ F1-Score: ____ Overall Accuracy: 94.00
Turkoglu, M., et. al., (Year - 2022) [11]	The models use deep feature extraction from six pre-trained CNNs AlexNet, GoogleNet, ResNet18, ResNet50, ResNet10 and DenseNet201, combined with a support vector machine (SVM) classifier.	Turkey-PlantDataset called as Turkey Plant Diseases and Pests Dataset [14], consists of a total of 4,447 images in 15 classes.	(Best Performance by AlexNet+GoogleNet +DenseNet201+ResNet50+ResNet101) Precision: ____ Recall: ____ F1-Score: 97.07 Overall Accuracy: 97.56

Author Name	Short Model Description	Name of Dataset with dataset size	Performance
M. S. H. Shovon, et. al., (Year - 2023) [29]	The model, PlantDet (Proposed Model), is a deep ensemble model integrating InceptionResNetV2, EfficientNetV2L, and Xception.	Rice Leaf dataset consists of five diseases [30], and Betel Leaf [31] consists of two categories. A total of 2710 images of rice leaves, 2167 (80%) for training, and 543 (20%) for testing. From a total 1000 images of Betel leaf, taking 400 images of each category as the training data and 100 images of each category as the validation data.	Precision: 99.00 Recall: 97.50 F1-Score: 98.14 Overall Accuracy: 97.50
P. Nancy, et. al., (Year - 2022) [33]	Performing Classification of Preprocessed Input data set using SVM, CNN, KNN Algorithms. To extract features from the input data the PCA approach is used.		(Best Performance by SVM Algorithm) Precision: ____ Recall: ____ F1-Score: ____ Overall Accuracy: 95.00 (Approx.)
P. S. Kanda, et. al., (Year - 2021) [34]	A plant classification model combining a Conditional Generative Adversarial Network for synthetic data generation, a Convolutional Neural Network for feature extraction, and a Logistic Regression classifier, achieving high accuracy across multiple plant leaf image datasets.	Used seven original datasets of plant leaves. 1. Flavia Dataset (1703 plant leaf images of 32 different species) [35] 2. MalayaKew Dataset (Having 2288 training and 528 testing images) [36] 3. Swedish Dataset (75 samples from each of 15 species of Swedish trees)[37] 4. Middle European Woody Plants Dataset [38] 5. Folio Dataset [39] 6. Amazon Forestry dataset [40][41] 7. LeafSnap Dataset [42]	Precision: ____ Recall: ____ F1-Score: ____ Overall Accuracy: 96.10

Author Name	Short Model Description	Name of Dataset with dataset size	Performance
M. S. H. Shovon, et. al., (Year - 2023) [29]	The model, PlantDet (Proposed Model), is a deep ensemble model integrating InceptionResNetV2, EfficientNetV2L, and Xception.	Rice Leaf dataset consists of five diseases [30], and Betel Leaf [31] consists of two categories. A total of 2710 images of rice leaves, 2167 (80%) for training, and 543 (20%) for testing. From a total 1000 images of Betel leaf, taking 400 images of each category as the training data and 100 images of each category as the validation data.	Precision: 99.00 Recall: 97.50 F1-Score: 98.14 Overall Accuracy: 97.50
Dubey, R.K., et. al., (Year - 2024) [43]	The model uses SV-RFE and ARO for feature selection and employs an ABi-LSTM classifier to identify paddy leaf diseases.	Plant Village Dataset[4]. The dataset consists of 54,305 images of 38 different disease classes of 14 different plant species.	Precision: 97.73 Recall: 97.30 F1-Score: 97.50 Overall Accuracy: 98.86
Alguliyev, et. al., (Year - 2021) [44]	The model combines a deep convolutional neural network (CNN) with gated recurrent units (GRUs) to identify plant leaf diseases.	Plant Village Dataset[4]. The dataset consists of 54,305 images of 38 different disease classes of 14 different plant species.	Precision: __ Recall: __ F1-Score: __ Overall Accuracy: 91.19
M. J. Hasan, et. al., (Year - 2019) [45]	The model integrates a deep convolutional neural network (DCNN) with a Support Vector Machine (SVM) classifier.	Collected images of rice plant diseases from resources such as IRRI [46], BRRI, BRKB [47], and Plantix [48]. There are nine different types of rice diseases.	Precision: __ Recall: __ F1-Score: __ Overall Accuracy: 97.50

Chapter 3

Proposed Methodology

In this project, we have proposed a novel ensemble method for plant disease classification. The proposed method combines multiple models where each model is a deep learning classifier that uses a pretrained model which is finetuned by the plant disease dataset.

3.1. Pre-trained Model

Pre-trained models are utilized in deep learning for several compelling reasons:

- **Leveraging Prior Knowledge:** Pre-trained models have been trained on massive datasets, often containing millions of images. This training allows them to learn general features and patterns that are relevant to a wide range of image recognition tasks. By using a pre-trained model, we can leverage this prior knowledge and avoid starting from scratch.
- **Faster Training:** Training a deep learning model from scratch can be computationally expensive and time-consuming. Pre-trained models already have learned weights and biases, significantly reducing the training time required for new tasks.
- **Improved Performance:** Pre-trained models often exhibit superior performance compared to models trained from scratch, especially when the target dataset is small. The pre-trained model's prior knowledge helps it generalize better to unseen data, leading to higher accuracy and robustness.
- **Overcoming Data Limitations:** In some cases, obtaining a large, labeled dataset for a specific task may be difficult or expensive. Pre-trained models can be fine-tuned on smaller datasets, making deep learning accessible for tasks with limited data availability.
- **Transfer Learning:** Pre-trained models enable transfer learning, where knowledge learned from one task is transferred to another related task. This is particularly useful when the target task has limited data or is similar to the task on which the pre-trained model was originally trained.
- **Reduced Risk of Overfitting:** Pre-trained models are less likely to overfit the training data compared to models trained from scratch. This is because they have already

learned general features from a large dataset, making them more robust to variations in the new data.

3.2. Summary of the work:

In the context of plant disease classification, we have used the three pre-trained models like VGG16, ResNet50, and DenseNet121. So, Those classification models have been developed separately using each pre-trained model and the model is tuned on the training dataset. So after tuning those models, models are actually combined to form an ensemble. An architecture of the proposed model for plant disease classification is shown in Figure 1. In the rest of this section, we will discuss the various components of the proposed model shown in Figure 1.

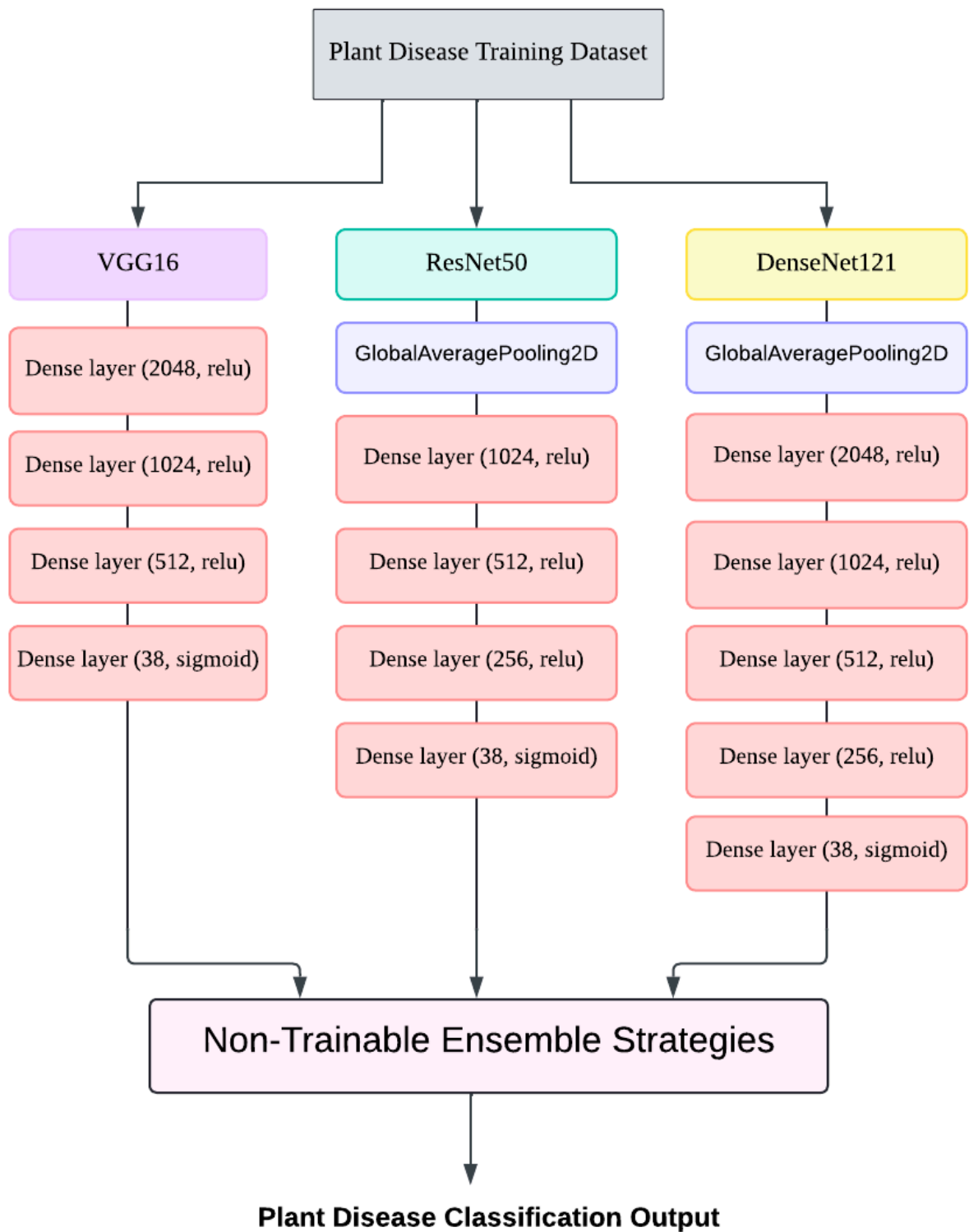


Figure 1: Architecture of Proposed Ensemble of Deep Learning Models for Plant Disease classification

3.3. VGG16 Model:

VGG16 is a convolutional neural network trained on a massive dataset (ImageNet) for image classification tasks. Its pre-trained model is available for use, offering a powerful feature extractor or a starting point for transfer learning in various image-related applications.

3.3.1. Architecture of VGG16:

The architecture of the VGG16 model is composed of several layers that systematically extract features from input images:

- **Input Layer:** 224x224 pixel RGB image (3 channels).
- **Convolutional Layers:** 13 layers with 3x3 filters, increasing in depth (64, 128, 256, 512) across 5 blocks.
- **Max Pooling Layers:** 5 layers with 2x2 filters interspersed with convolutional layers for downsampling.
- **Fully Connected Layers:** 3 layers (with 4096, 4096 neurons, and a final layer corresponding to the number of classes) for classification.
- **Output Layer:** Utilises Softmax activation for multi-class probability prediction.

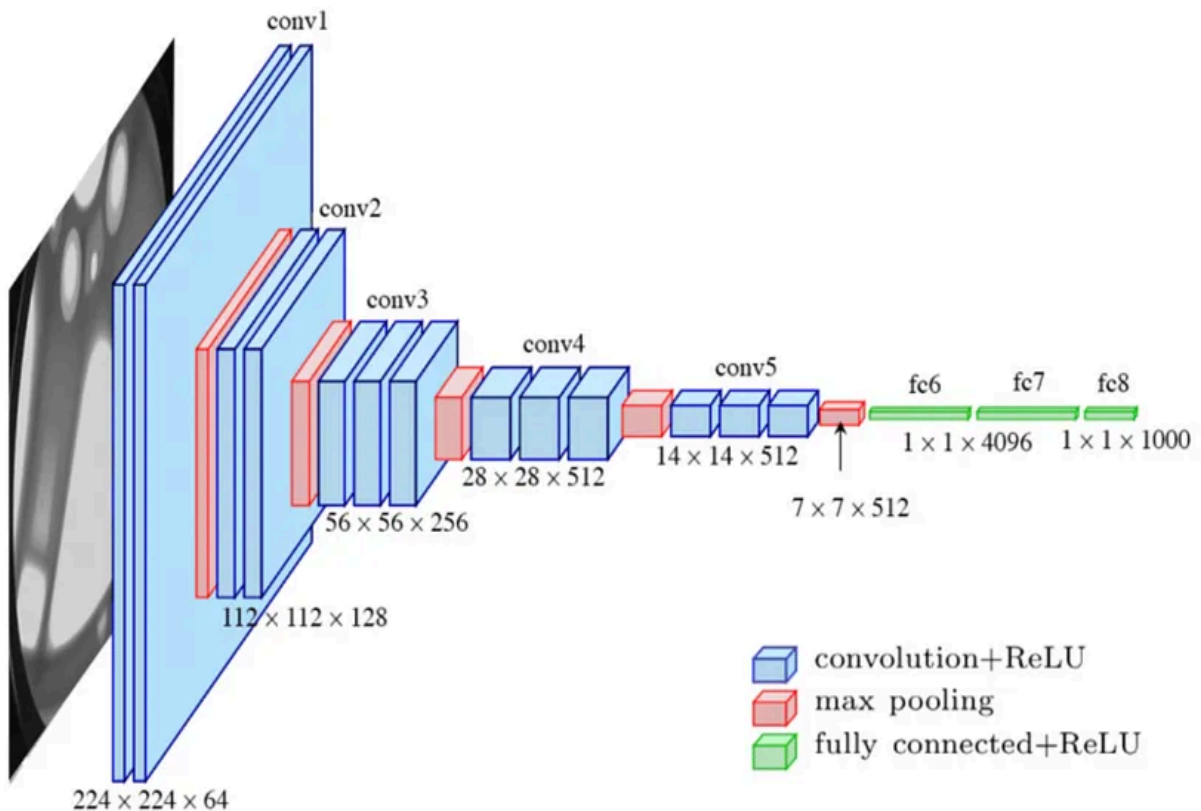


Figure 2: Architecture of VGG16 Model [16]

3.4. ResNet50 Model:

ResNet50 is a deep residual network designed to address the vanishing gradient problem in deep networks, allowing for the training of much deeper models. It is pre-trained on ImageNet and is widely used for transfer learning in various image classification tasks.

3.4.1. Architecture of ResNet50:

- **Input Layer:** 224x224 pixel RGB image.
- **Convolutional Layers:** Multiple convolutional layers (mostly 3x3 filters) organized into residual blocks.
- **Residual Blocks:** Each block contains two or more convolutional layers with skip connections, allowing information to flow directly from earlier layers to later layers.
- **Max Pooling Layers:** A few strategically placed layers for downsampling.
- **Global Average Pooling:** Averages feature maps before the final fully connected layer.
- **Fully Connected Layer:** One fully connected layer with the number of classes as output units.
- **Output Layer:** Softmax activation.

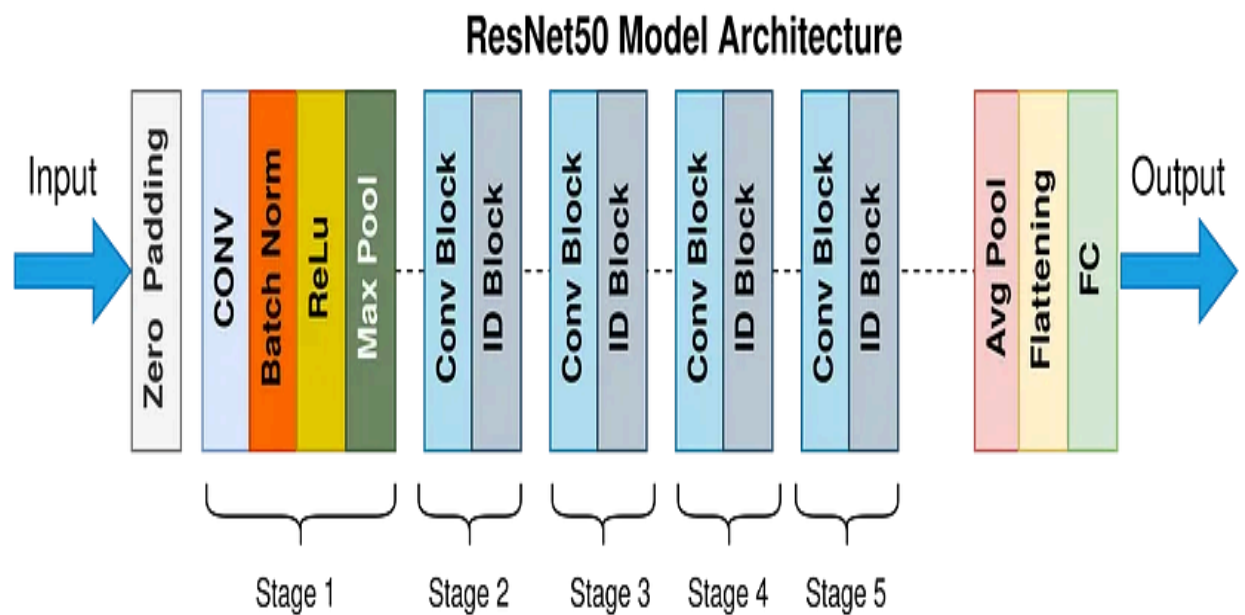


Figure 3: Architecture of ResNet50 Model [17]

3.5. DenseNet121 Model:

DenseNet121 is a densely connected convolutional network that connects each layer to every other layer in a feed-forward fashion. It is pre-trained on ImageNet and is highly efficient in terms of parameter usage, making it effective for various image classification tasks [10].

3.5.1. Architecture of DenseNet121:

- **Input Layer:** 224x224 pixel RGB image.
- **Dense Blocks:** Multiple blocks where each layer is densely connected to all subsequent layers within the block.
- **Transition Layers:** Layers between dense blocks that reduce the feature map size (downsampling).
- **Convolutional Layers:** 1x1 convolutions for feature map reduction within dense blocks.
- **Global Average Pooling:** Before the final fully connected layer.
- **Fully Connected Layer:** One fully connected layer with the number of classes as output units.
- **Output Layer:** Softmax activation.

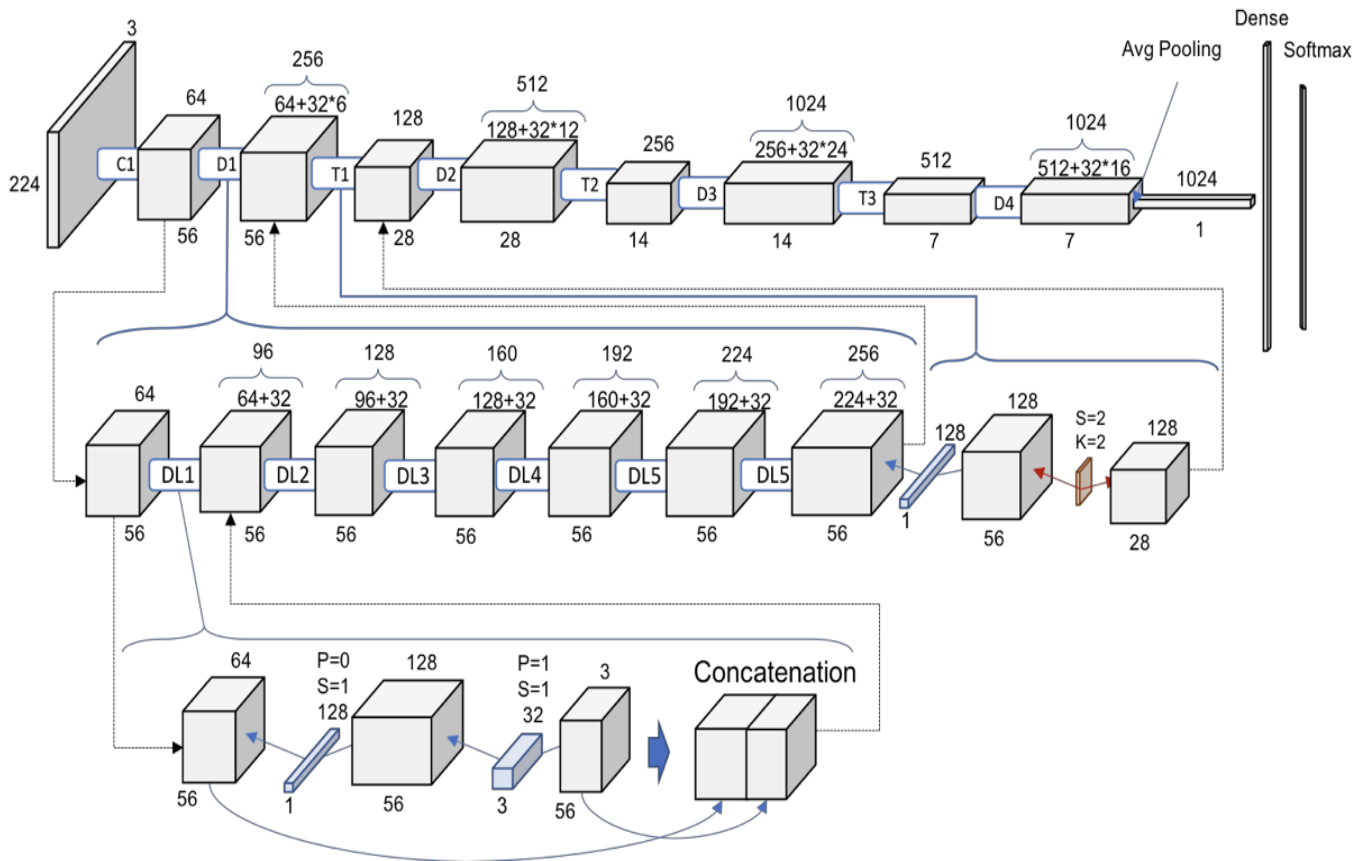


Figure 4: Architecture of DenseNet121 Model [18]

3.6. Model Development:

We have used transfer learning for model development. For developing each model we have used a pre-trained model.

3.6.1. Transfer Learning:

Transfer learning is a technique where knowledge gained from solving one problem is applied to a different but related problem. In deep learning, it involves using pre-trained models and fine-tuning them on new data to improve performance on a specific task. This often begins by initializing the model with weights pre-trained on a large dataset like ImageNet and then adjusting these weights during training on the new dataset.

- **Initialization:** Load weights pre-trained on ImageNet.
- **Fine-tuning:** Optionally freeze initial layers, then train the remaining layers (or all layers) on the plant disease dataset with a lower learning rate.

3.6.2. Data Augmentation:

Data augmentation is a technique used to artificially increase the size of a dataset by applying various transformations to the existing data samples. These transformations can include techniques such as rotation, scaling, cropping, flipping, translation, brightness adjustment, and noise addition, among others.

The primary purpose of data augmentation is to increase the diversity and variability of the training data without actually collecting new data. By introducing these variations, the model becomes more robust and generalizes better to unseen data. It helps to reduce overfitting, as the model learns to recognize the same objects or patterns in different forms. Additionally, data augmentation can help address class imbalance issues by generating synthetic examples of minority classes.

- **Techniques:** Random rotation, horizontal/vertical flips, random crops, brightness / contrast adjustments, etc.
- **Purpose:** Increase data variability, prevent overfitting, and improve model generalization.

This section outlines the development of the deep learning models used in this study. We utilized three pre-trained architectures: VGG16, ResNet50, and DenseNet121, integrating them with additional dense layers to adapt them for plant disease classification.

3.6.3. Model 1: VGG16-Based Model

Model 1 leverages the VGG16 architecture, a well-established convolutional neural network pre-trained on the ImageNet dataset. The base VGG16 model is connected with additional dense layers to enhance its performance for the specific task of plant disease classification. The model architecture includes:

- **Base Model:** VGG16 pre-trained on ImageNet.
- **Additional Layers:**
 - Three dense layers with 2048, 1024, and 512 neurons respectively, all using ReLU activation functions.
 - An output layer with 38 neurons using a sigmoid activation function to handle multi-label classification.

3.6.4. Model 2: ResNet50-Based Model

Model 2 employs the ResNet50 architecture, another highly regarded convolutional neural network pre-trained on ImageNet. This model includes additional layers to adapt it for plant disease classification:

- **Base Model:** ResNet50 pre-trained on ImageNet.
- **Additional Layers:**
 - A GlobalAveragePooling2D layer to reduce the spatial dimensions of the feature maps.
 - Three dense layers with 1024, 512, and 256 neurons respectively, all using ReLU activation functions.
 - An output layer with 38 neurons using a sigmoid activation function for multi-label classification.

3.6.5. Model 3: DenseNet121-Based Model

Model 3 is based on the DenseNet121 architecture, known for its dense connectivity pattern and efficient feature extraction. This model includes:

- **Base Model:** DenseNet121 pre-trained on ImageNet.
- **Additional Layers:**
 - A GlobalAveragePooling2D layer to reduce the spatial dimensions of the feature maps.

- Four dense layers with 2048, 1024, 512, and 256 neurons respectively, all using ReLU activation functions.
- An output layer with 38 neurons using a sigmoid activation function for multi-label classification.

In this study, the sigmoid activation function is employed in the output layer instead of the softmax activation function. This choice aligns with the nature of the classification problem, which is multi-label rather than multi-class.

- **Multi-Label Classification:** In plant disease classification, a single plant can exhibit symptoms of multiple diseases simultaneously. This necessitates a model capable of predicting multiple classes for a single input.
- **Sigmoid Activation Function:** Each neuron in the output layer produces an independent probability between 0 and 1, allowing the model to predict multiple classes (diseases) for a single instance. This is essential for accurately diagnosing plants that may be affected by more than one disease.
- **Softmax Activation Function:** Typically used for multi-class classification, where the sum of the output probabilities is constrained to 1. This is suitable for problems where each instance belongs to only one class, which does not align with the requirements of our multi-label classification task.

Therefore, the use of the sigmoid activation function in the output layer is crucial for handling the complexity of multi-label classification problems inherent in plant disease diagnosis. This allows for a more accurate and comprehensive diagnostic tool, capable of predicting the presence of multiple diseases in a single plant instance.

3.7. Non-trainable Ensemble Strategies:

Non-trainable ensemble methods are techniques used to combine the predictions of multiple individual models without further training on the combined predictions. Instead of adjusting the parameters of the ensemble itself based on the training data, these methods rely on simple rules or algorithms to aggregate predictions. Examples include Majority Voting, Max Rule, Average Rule, and Product Rule. These methods are valuable for leveraging the diversity of individual models and enhancing overall predictive performance without introducing additional training complexity.

Model combination can be done in two ways –

1. Trainable Methods (Stacked Ensemble)
2. Non-trainable method

In our work, we have used non-trainable four non-trainable strategies

- Majority Voting Rule
- Max Rule
- Average Rule
- Product Rule

3.7.1. Majority Voting Rule:

Majority Voting is an ensemble strategy where each model in the ensemble makes a prediction for each instance. The final output is determined by the class that receives the most votes across the models.

- **Process:**
 - Each of the three models (VGG16, ResNet50, DenseNet121) predicts the class probabilities for a given input.
 - For each class, we count the number of models that predict this class as the most probable.
 - The class with the highest count is selected as the final prediction.

- **Output:**

This method is particularly useful when the individual models have varying strengths, as it leverages the consensus among them to improve the overall accuracy. The result is a robust prediction that typically reflects the most likely class for each instance.

3.7.2. Max Rule:

The Max Rule combines the outputs by taking the maximum probability predicted by any model for each class.

- **Process:**
 - Each model outputs a probability score for each class.
 - For each class, the highest probability score across all models is selected as the final probability.

- **Output:**

This method ensures that the strongest signal for each class, from any model, is taken into account. It can be beneficial when one model is particularly confident in its prediction for certain classes, thereby influencing the final decision with its strong prediction.

3.7.3. Average Rule:

The Average Rule calculates the mean probability of each class across all models to make the final prediction.

- **Process:**

- Each model provides a probability score for each class.
- The probabilities for each class are averaged across all models.
- The class with the highest average probability is selected as the final prediction.

- **Output:**

By averaging the probabilities, this method smoothes out the predictions, reducing the impact of any single model's uncertainty. It provides a balanced approach, leveraging the strengths of all models to produce a consensus prediction.

3.7.4. Product Rule:

The Product Rule multiplies the probability scores of each class from all models to determine the final prediction.

- **Process:**

- Each model outputs a probability score for each class.
- The probability scores for each class are multiplied across all models.
- The class with the highest product of probabilities is selected as the final prediction.

- **Output:**

This method emphasizes the agreement among models. A class will only have a high final probability if all models assign it a high probability. It can be effective in reinforcing the confidence in classes where all models consistently agree, thus providing a strong consensus-based prediction.

Chapter 4

Evaluation and Results

In this section, we present the evaluation metrics and results obtained from the ensemble of pre-trained models, namely VGG16, ResNet50, and DenseNet121. The performance of the ensemble is analyzed using the classification report from the scikit-learn library, which provides a comprehensive overview of various metrics for evaluating the model's classification performance.

4.1. Classification Report Analysis:

The classification report generated from the scikit-learn library consists of several key metrics that assess the performance of the ensemble model across different classes. These metrics include:

- **Accuracy:**

Accuracy is a fundamental metric used to evaluate the performance of a classification model. It measures the proportion of correctly predicted instances out of the total instances.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:**

Precision is calculated as the ratio of true positive predictions (TP) to the sum of true positive and false positive predictions (FP).

Mathematically, it can be expressed as:

$$precision = \frac{TP}{TP + FP}$$

- **Recall:**

Recall, also known as sensitivity or true positive rate, is calculated as the ratio of true positive predictions (TP) to the sum of true positive predictions and false negative predictions (FN).

Mathematically, it can be expressed as:

$$recall = \frac{TP}{TP + FN}$$

- **F1-Score:**

The F1-score is the harmonic mean of precision and recall, calculated as:

$$f1\ score = \frac{2 \times precision \times recall}{precision + recall}$$

It balances precision and recall, providing a single metric that considers both false positives and false negatives.

A classification report typically provides several types of F1 scores:

1. Micro F1-score:

The micro F1 score is calculated globally by counting the total true positives, false negatives, and false positives across all classes. It treats each individual instance equally, regardless of its class.

Mathematically, it can be expressed as:

$$\text{Micro Precision} = \text{Total True Positives} / (\text{Total True Positives} + \text{Total False Positives})$$

$$\text{Micro Recall} = \text{Total True Positives} / (\text{Total True Positives} + \text{Total False Negatives})$$

$$\text{Micro F1} = 2 * (\text{Micro Precision} * \text{Micro Recall}) / (\text{Micro Precision} + \text{Micro Recall})$$

2. Macro F1-score:

The macro F1 score is calculated as the unweighted average of the F1 scores of each class. It treats all classes equally, regardless of their support (number of samples).

Mathematically, it can be expressed as:

$$\text{Macro F1} = (\text{F1_Class1} + \text{F1_Class2} + \dots + \text{F1_ClassN}) / N$$

3. Weighted F1-score:

The weighted F1 score is calculated as the weighted average of the F1 scores of each class, where the weight of each class is its support (number of samples). It gives more importance to classes with a larger number of samples.

Mathematically, it can be expressed as:

$$\text{Weighted F1} = (w1 * \text{F1_Class1} + w2 * \text{F1_Class2} + \dots + wN * \text{F1_ClassN})$$

[where w represents the support of each class]

- **Support:**

Support refers to the number of actual occurrences of each class in the dataset. It provides insight into the distribution of classes and helps interpret the significance of precision, recall, and F1-score for each class.

Term Descriptions:

True Positives (TP): Correctly predicted positive instances.

True Negatives (TN): Correctly predicted negative instances.

False Positives (FP): Incorrectly predicted positive instances (actually negative).

False Negatives (FN): Incorrectly predicted negative instances (actually positive).

4.2. Dataset description:

In this study, we utilized two comprehensive datasets to evaluate the performance of our proposed ensemble deep learning model for plant disease classification. The relevance and importance of these datasets lie in their diverse representation of plant diseases, which is critical for training robust and accurate models capable of generalizing well to real-world scenarios.

4.2.1. Dataset-1:

Dataset Name: New Plant Diseases Dataset

Source: This is an open-source dataset from Kaggle [13].

Description: This dataset consists of about 87K rgb images of healthy and diseased crop leaves which is categorized into 38 different classes. The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure. A new directory containing 33 test images is created later for prediction purposes. The class names are given below

- Apple___Apple_scab
- Apple___Black_rot
- Apple___Cedar_apple_rust
- Apple___healthy
- Blueberry___healthy
- Cherry_(including_sour)___Powdery_mildew
- Cherry_(including_sour)___healthy
- Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
- Corn_(maize)___Common_rust_
- Corn_(maize)___Northern_Leaf_Blight
- Corn_(maize)___healthy
- Grape___Black_rot
- Grape___Esca_(Black_Measles)
- Grape___Leaf_blight_(Isariopsis_Leaf_Spot)

- Grape___healthy
- Orange___Haunglongbing_(Citrus_greening)
- Peach___Bacterial_spot
- Peach___healthy
- _bell___Bacterial_spot
- Pepper,_bell___healthy
- Potato___Early_blight
- Potato___Late_blight
- Potato___healthy
- Raspberry___healthy
- Soybean___healthy
- Squash___Powdery_mildew
- Strawberry___Leaf_scorch
- Strawberry___healthy
- Tomato___Bacterial_spot
- Tomato___Early_blight
- Tomato___Late_blight
- Tomato___Leaf_Mold
- Tomato___Septoria_leaf_spot
- Tomato___Spider_mites Two-spotted_spider_mite
- Tomato___Target_Spot
- Tomato___Tomato_Yellow_Leaf_Curl_Virus
- Tomato___Tomato_mosaic_virus
- Tomato___healthy

A. Data Composition:

Total Images: 87,000 images

Classes: 38 classes

Class Distribution: Varies across classes to reflect real-world conditions.

B. Data Splitting:

Training Set: 69,600 images (80% of the dataset) allocated for training.

Validation Set: 17,400 images (20% of the dataset) used for validation.

Test Set: A separate set of 33 images created for testing and prediction purposes.

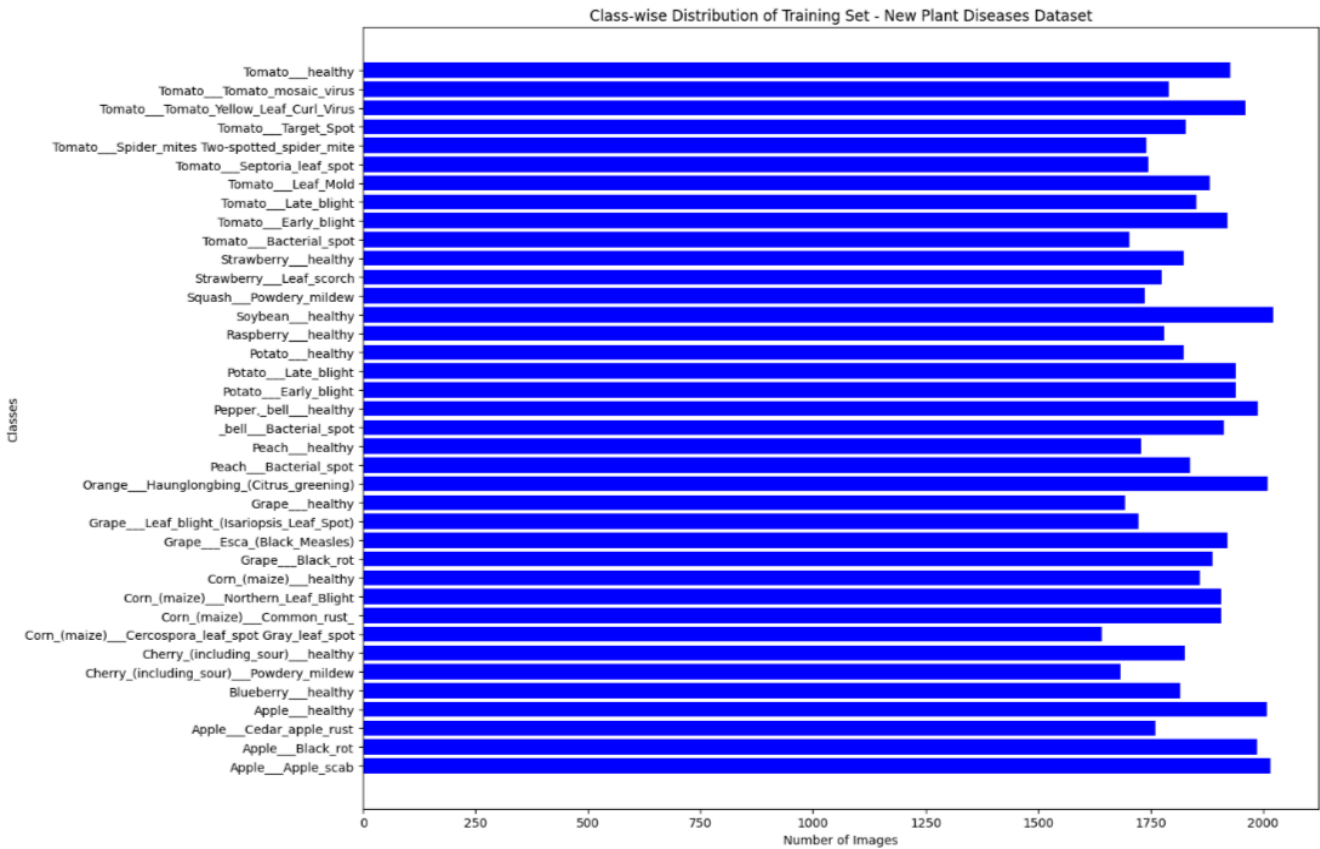


Figure 5: Class-wise Distribution of the Training Samples of Dataset 1

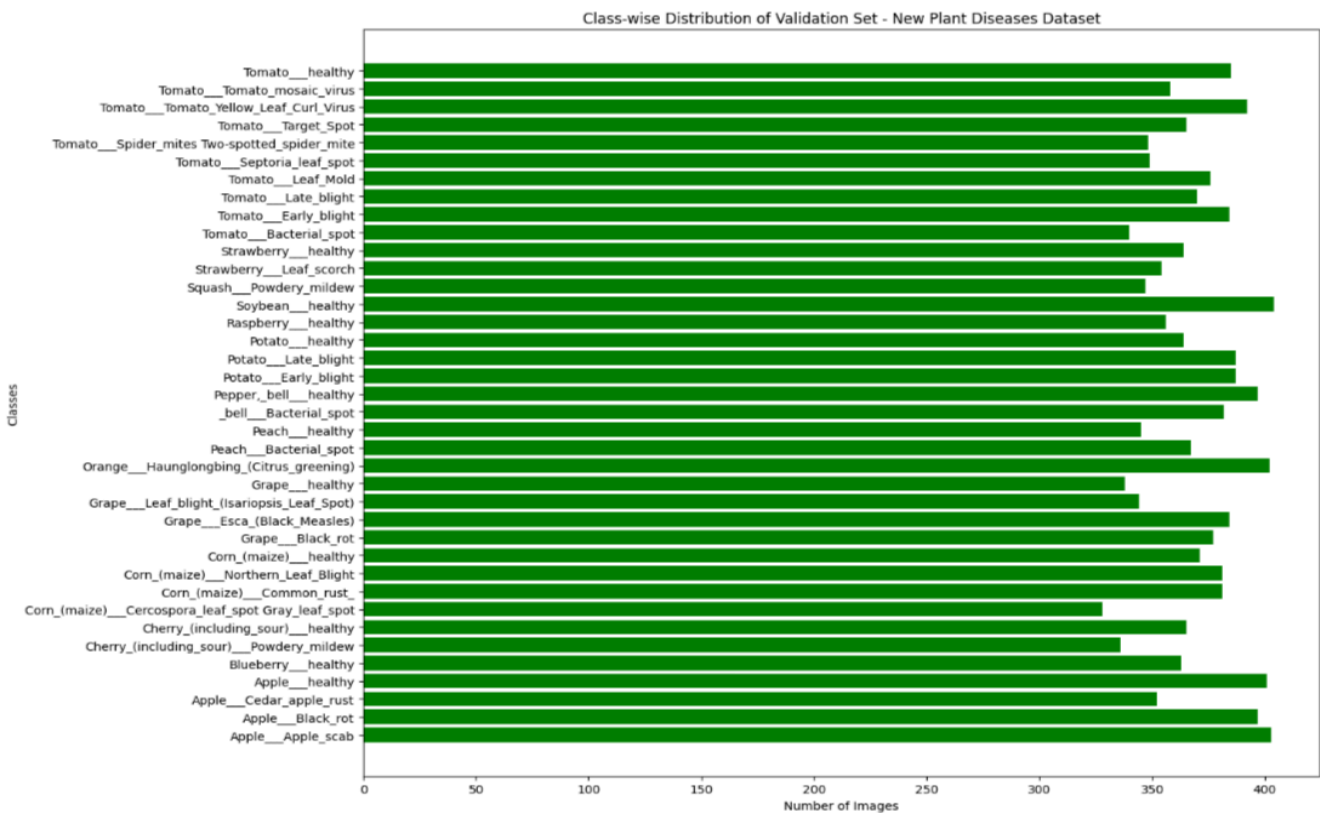


Figure 6: Class-wise Distribution of the Validation Set of Dataset 1

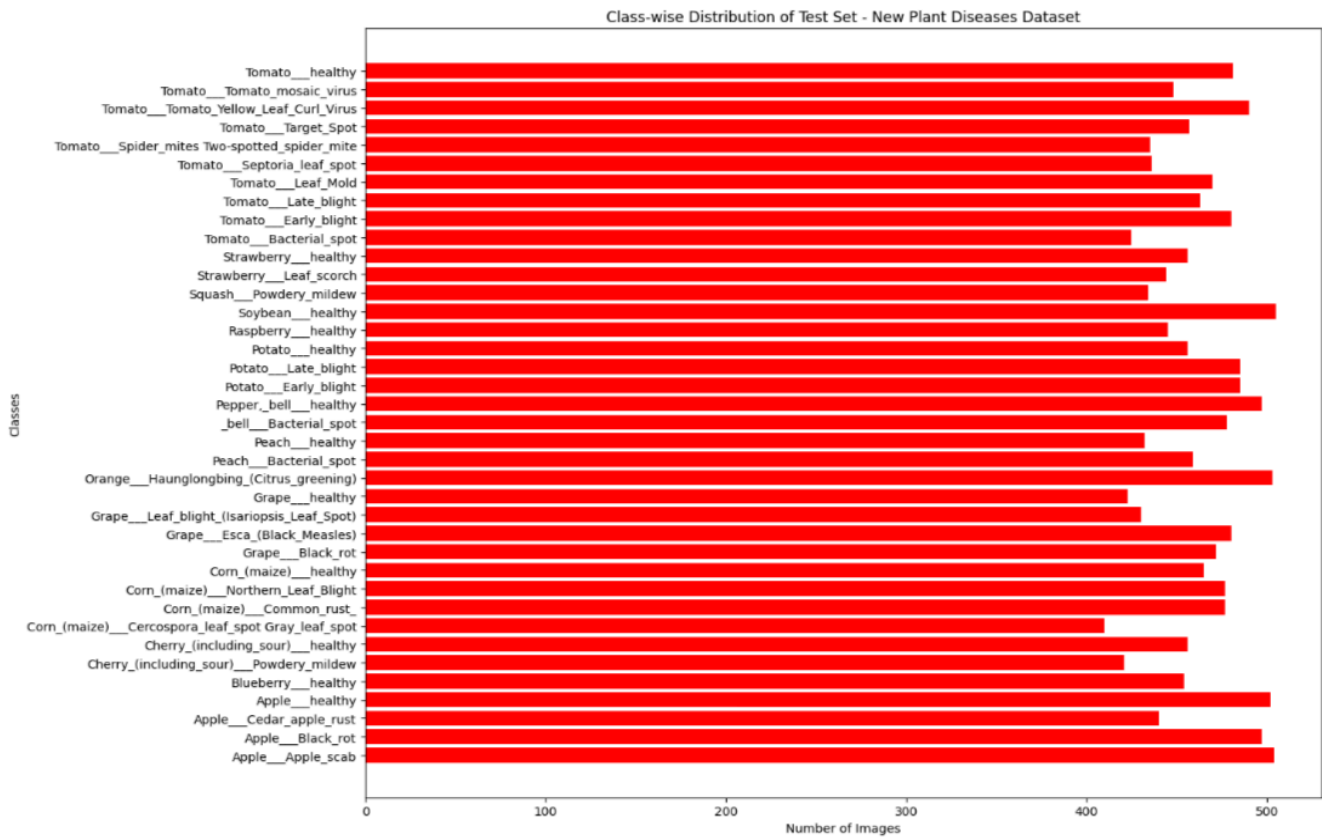


Figure 7: Class-wise Distribution of the Test Set of Dataset 1

4.2.2. Dataset-2:

Dataset Name: Tomato Leaf Disease Detection

Source: This is an open-source dataset from kaggle [12].

Description: The dataset contains images of tomato leaves affected by 10 different types of diseases. This diverse dataset is essential for training robust models capable of accurately identifying various diseases in tomato plants. The class names are given

- Tomato_mosaic_virus
- Target_Spot
- Bacterial_spot
- Tomato_Yellow_Leaf_Curl_Virus
- Late_blight
- Leaf_Mold
- Early_blight
- Spider_mites Two-spotted_spider_mite

- Tomato___healthy
- Septoria_leaf_spot

A. Data Composition:

Total Images: 10,000 images

Classes: 10 classes representing different diseases and healthy conditions.

Class Distribution: Approximately 1,000 images per class ensuring balanced representation.

B. Data Splitting:

Training Set: 8,000 images (80% of the dataset) used for training the models.

Validation Set: 2,000 images (20% of the dataset) used for validating the model performance.

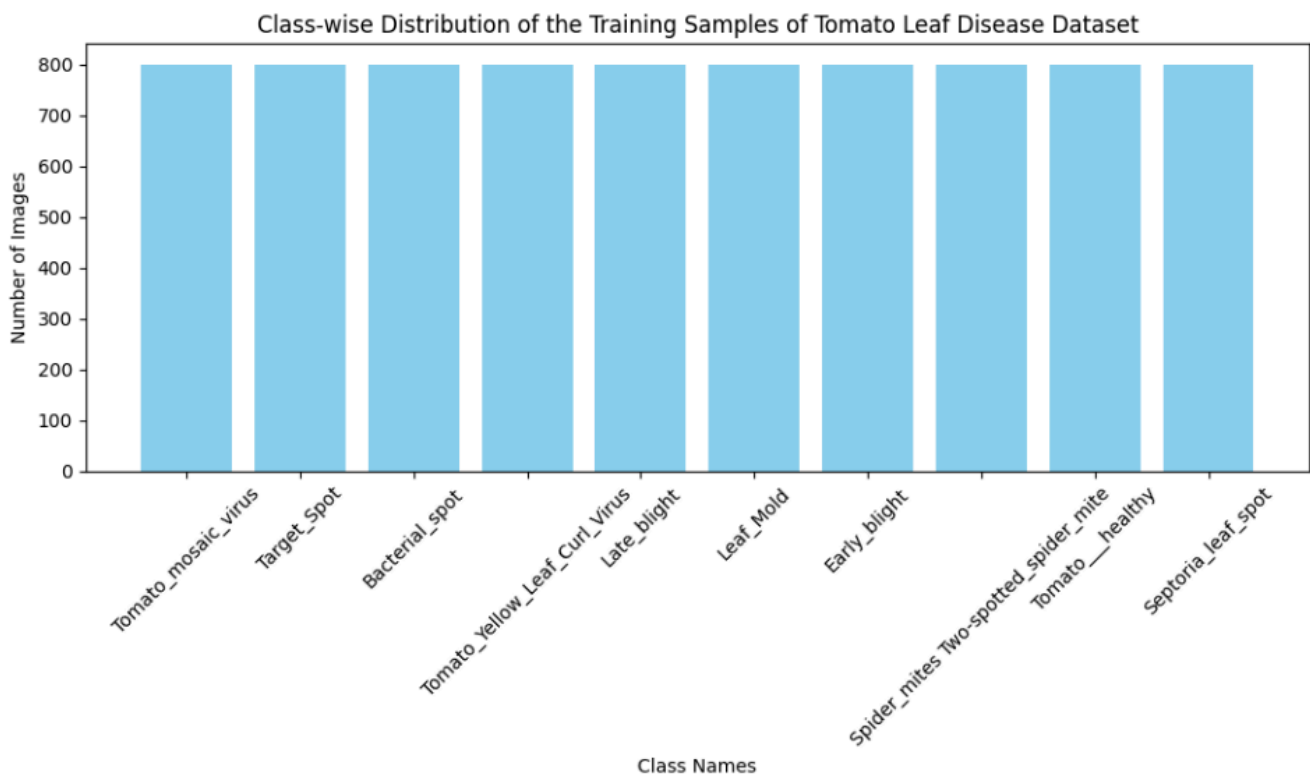


Figure 8: Class-wise Distribution of the Training Samples of Dataset 2

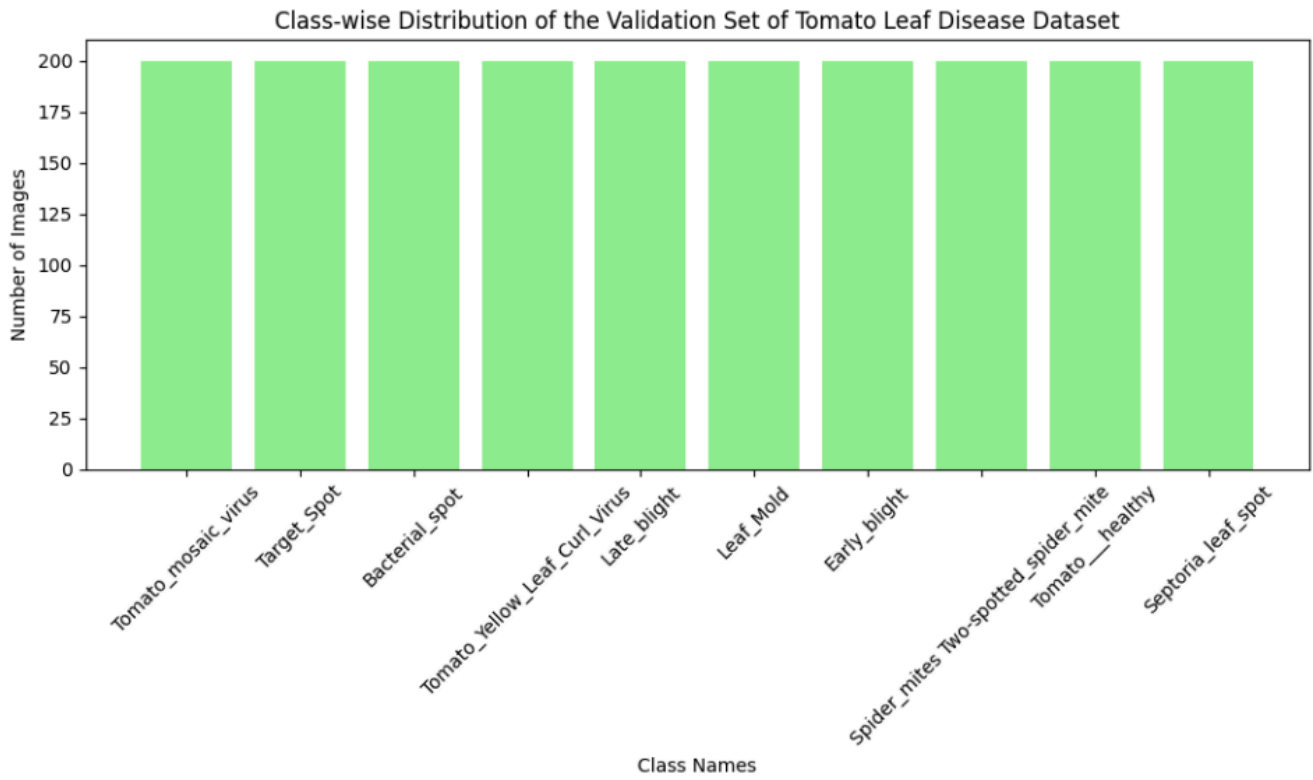


Figure 9: Class-wise Distribution of the Validation Set of Dataset 2

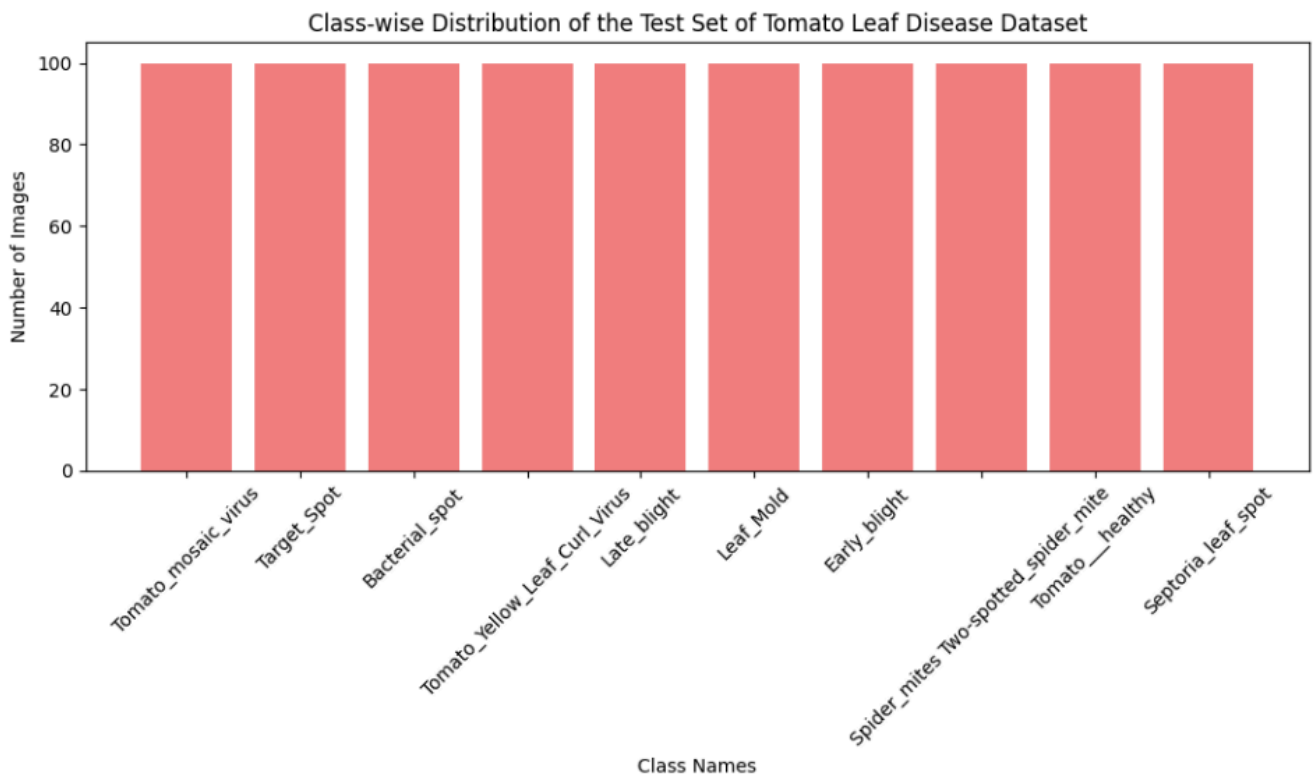


Figure 10: Class-wise Distribution of the Test Set of Dataset 2

4.3.2 Performance on Dataset-2:

Similarly, Table-4 illustrates the results of each pre-trained deep learning model on Dataset-2.

Table 4: Results of each Pre-trained Deep Learning Model on Dataset-2

Model Name	Macro Precision	Macro Recall	Macro F1-Score	Weighted Precision	Weighted Recall	Weighted F1-Score	Accuracy
VGG16	0.94	0.93	0.93	0.94	0.93	0.93	0.93
ResNet50	0.95	0.95	0.95	0.95	0.95	0.95	0.95
DenseNet121	0.97	0.97	0.97	0.97	0.97	0.97	0.97

Moreover, Table-5 provides the results of each proposed ensemble method on Dataset-2.

Table 5: Result of each proposed ensemble method on Dataset-2

Ensemble Method	Macro Precision	Macro Recall	Macro F1-Score	Weighted Precision	Weighted Recall	Weighted F1-Score	Accuracy
Majority Voting	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Average Rule	0.98	0.98	0.98	0.98	0.98	0.98	0.98
Max Rule	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Product Rule	0.98	0.98	0.98	0.98	0.98	0.98	0.98

Based on Table 3 and Table 5 we can conclude that the Majority Voting Rule based proposed ensemble method is performing better than other three ensemble methods based proposed methods. Both the Product Rule method based proposed ensemble model and Average Rule method based proposed ensemble model perform well after the Majority Voting based proposed method.

4.4. Hyperparameter Tuning:

In this section, we outline the major hyperparameters used to optimize our proposed method on Dataset 1. The parameters are crucial for enhancing the model's performance and include details about batch size, number of epochs, learning rate, optimizer, and dense layer sizes. The hyperparameters were specifically chosen for models based on VGG16, ResNet50, and DenseNet121 architectures, with both Sigmoid and Softmax output layers.

Table 6: Hyperparameter Settings for Various Model Configurations

Models	Batch Size	Epoch	Learning Rate	Optimizer	Dense Layer Sizes
VGG16 based model with Sigmoid output layer	64	7	0.001	RMSprop	2048, 1024, 512, 38
ResNet50 based model with Sigmoid output layer	64	13	0.001	RMSprop	2048, 1024, 512, 38
DenseNet121 based model with Sigmoid output layer	64	9	0.0001	RMSprop	2048, 1024, 512, 256, 38
VGG16 based model with Softmax output layer	64	9	0.0001	RMSprop	2048, 1024, 512, 38
ResNet50 based model with Softmax output layer	64	8	0.0001	RMSprop	1024, 512, 256, 38
DenseNet121 based model with Softmax output layer	64	15	0.0001	RMSprop	2048, 1024, 512, 38

4.5. Comparison with existing work:

As the Majority Voting Rule based proposed ensemble method is performing best among these four ensemble methods, we are going to compare the proposed ensemble model with some established improved models.

Table 7: Comparison with existing and related works

Model Description	Dataset Description	Precision	Recall	F1-Score	Accuracy
The model uses an ensemble of three fine-tuned EfficientNet models (B5, B7, and B6) for papaya disease recognition [1].	Plant village dataset [4], The dataset consists of 54,305 images of 38 different disease classes of 14 different plant species.	98.31	98.27	98.28	98.27
A deep CNN-based model for plant disease identification using ensemble classifiers, including SVM, KNN, Random Forest, Naive Bayes, and Logistic Regression [2]	Plant village dataset [4], The dataset consists of 54,305 images of 38 different disease classes of 14 different plant species.	93.76	96.75	94.18	93.8
The models uses six pre-trained CNNs AlexNet, GoogleNet, ResNet18, ResNet50, ResNet10 and DenseNet201, combined with a support vector machine (SVM) classifier [11]	Turkey-PlantDataset called as Turkey Plant Diseases and Pests Dataset [14], consists of a total of 4,447 images in 15 classes.	—	—	97.07	97.56
The model uses an ensemble of three fine-tuned EfficientNet models (B5, B7, and B6) for papaya disease recognition [15]	A dataset of 138980 images of papaya leaves and fruits is created by utilizing image augmentation techniques from 13898 collected images	—	—	—	99.74
The model, PlantDet (Proposed Model), is a deep ensemble model integrating InceptionResNetV2, EfficientNetV2L, and Xception. [29]	Rice Leaf dataset consists of five diseases [30], and Betel Leaf [31] consists of two categories.	99.00	97.50	98.14	97.50
Proposed Model Ensemble of 3 pre-trained models with Sigmoid output layer.	New plant disease dataset, dataset of 87,000 images has 38 classes.	1.00	1.00	1.00	1.00

The proposed ensemble model, leveraging Majority Voting and pre-trained models with a Sigmoid activation function, outperforms several existing models in the field of plant disease classification. Our model achieves perfect precision, recall, F1-score, and accuracy (1.00) on a new dataset consisting of 87,000 images across 38 classes. This performance not only highlights the efficacy of our ensemble approach but also underscores the potential of deep learning ensembles to significantly advance plant disease diagnostics. By achieving higher accuracy than established models, our proposed method sets a new benchmark for future research and practical applications in agriculture.

Chapter 5

Conclusion and Future Work

In this study on plant disease classification, three pre-trained models—VGG16, ResNet50, and DenseNet121—were employed. Each model was individually developed and fine-tuned using the training dataset. After tuning, these models were combined to form an ensemble classifier. The proposed ensemble model achieved an impressive accuracy of 1.00 on Dataset-1 using the Majority Voting method, demonstrating its effectiveness in accurately identifying plant diseases.

For future research, several enhancements and extensions can be explored. Implementing trainable ensemble methods, such as stacking and blending, which involve training a meta-model to combine the predictions of the base models, could potentially improve overall performance by leveraging the strengths of each individual model. Additionally, incorporating other state-of-the-art pre-trained models like EfficientNet, InceptionV3, or MobileNet could provide further performance gains and insights due to their efficiency and accuracy in feature extraction. Moreover, integrating multimodal data, such as combining image data with environmental and phenotypic data, could enhance the model's robustness and predictive power. Investigating different data augmentation techniques and their role in improving model generalizability, as well as applying transfer learning to fine-tune the ensemble model for specific subsets of plant diseases or varying agricultural conditions, could make the model more versatile and adaptive. By pursuing these advanced methodologies and expanding the scope of datasets and models, we can further enhance the accuracy, efficiency, and applicability of deep learning ensembles in plant disease diagnostics, ultimately contributing to more effective and sustainable agricultural practices.

References:

- [1] Akshai, K.P. and Anitha, J., 2021, May. Plant disease classification using deep learning. In *2021 3rd International Conference on Signal Processing and Communication (ICPSC)* (pp. 407-411). IEEE.
- [2] Hassan, S.M., Maji, A.K. Deep feature-based plant disease identification using machine learning classifier. *Innovations Syst Softw Eng* (2022). <https://doi.org/10.1007/s11334-022-00513-y>
- [3] Chen, J., Zeb, A., Nanekaran, Y.A. *et al.* Stacking ensemble model of deep learning for plant disease recognition. *J Ambient Intell Human Comput* 14, 12359–12372 (2023). <https://doi.org/10.1007/s12652-022-04334-6>
- [4] Hughes D, Salathé M, et al (2015) An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv:1511.08060
- [5] H. Amin, A. Darwish, A. E. Hassanien and M. Soliman, "End-to-End Deep Learning Model for Corn Leaf Disease Classification," in *IEEE Access*, vol. 10, pp. 31103-31115, 2022, doi: 10.1109/ACCESS.2022.3159678.
- [6] Plant Village Dataset. Accessed: Jul. 9, 2021. [Online]. Available: <https://www.kaggle.com/saroz014/plant-diseases>
- [7] Sachar, S., Kumar, A. Deep ensemble learning for automatic medicinal leaf identification. *Int. j. inf. tecnol.* 14, 3089–3097 (2022). <https://doi.org/10.1007/s41870-022-01055-z>
- [8] V. K. Shrivastava, M. K. Pradhan and M. P. Thakur, "Application of Pre-Trained Deep Convolutional Neural Networks for Rice Plant Disease Classification," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 1023-1030, doi: 10.1109/ICAIS50930.2021.9395813.
- [9] Basavaiah, J., Arlene Anthony, A. Tomato Leaf Disease Classification using Multiple Feature Extraction Techniques. *Wireless Pers Commun* 115, 633–651 (2020). <https://doi.org/10.1007/s11277-020-07590-x>
- [10] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, “DenseNet: Implementing efficient ConvNet descriptor pyramids,” 2014, arXiv:1404.1869.

- [11] Yanikoğlu, B. & Hanbay, D. PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection. *SIViP* 16, 301–309 (2022). <https://doi.org/10.1007/s11760-021-01909-2>
- [12] Tomato leaf disease detection. Available: <https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf>
- [13] New Plant Diseases Dataset. Available: <https://www.kaggle.com/datasets/vipooooo1/new-plant-diseases-dataset>
- [14] The Turkey-PlantDataset. Available: <https://github.com/mturkoglu23/PlantDiseaseNet>
- [15] R. H. Hridoy and M. R. A. Tuli, "A Deep Ensemble Approach for Recognition of Papaya Diseases using EfficientNet Models," 2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/ICEEICT53905.2021.9667825.
- [16] Available: <https://lekhuyen.medium.com/an-overview-of-vgg16-and-nin-models-96e4bf398484>
- [17] Available: <https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>
- [18] Available: <https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a>
- [19] Savary, S., Willocquet, L., Pethybridge, S. J., Esker, P., McRoberts, N., & Nelson, A. (2019). The global burden of pathogens and pests on major food crops. *Nature Ecology & Evolution*, 3(3), 430-439.
- [20] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7, 1419.
- [21] Barbedo, J. G. A. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, 144, 52-60.
- [22] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90.
- [23] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

- [24] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272-279.
- [25] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60.
- [26] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60.
- [27] Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple classifier systems*, 1857, 1-15.
- [28] Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11, 169-198.
- [29] M. S. H. Shovon, S. J. Mozumder, O. K. Pal, M. F. Mridha, N. Asai and J. Shin, "PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning For Plant Disease Detection," in *IEEE Access*, vol. 11, pp. 34846-34859, 2023, doi: 10.1109/ACCESS.2023.3264835.
- [30] A. Fiqri. Rice Leafs 5 Diseases. Accessed: Dec. 10, 2022. [Online]. Available: <https://www.kaggle.com/datasets/adeifiqri12/riceleafsv3>
- [32] Betel Leaf_2023 Google Drive. Accessed: Dec. 10, 2022. [Online]. Available: https://drive.google.com/drive/u/1/folders/1_v4f_5pdk00hmfJJ_QuADYneC1nHycqn
- [33] P. Nancy, H. Pallathadka, M. Naved, K. Kaliyaperumal, K. Arumugam and V. Garchar, "Deep Learning and Machine Learning Based Efficient Framework for Image Based Plant Disease Classification and Detection," *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, Coimbatore, India, 2022, pp. 1-6, doi: 10.1109/ICACTA54488.2022.9753623.
- [34] P. S. Kanda, K. Xia and O. H. Sanusi, "A Deep Learning-Based Recognition Technique for Plant Leaf Classification," in *IEEE Access*, vol. 9, pp. 162590-162613, 2021, doi: 10.1109/ACCESS.2021.3131726.

- [35] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, “A leaf recognition algorithm for plant classification using probabilistic neural network,” in Proc. IEEE Int. Symp. Signal Process. Inf. Technol., Dec. 2007, pp. 11–16, doi: 10.1109/ISSPIT.2007.4458016.
- [36] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, “Deep-plant: Plant identification with convolutional neural networks,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2015, pp. 452–456, doi: 10.1109/ICIP.2015.7350839.
- [37] O. J. O. Soderkvist, “Computer vision classification of leaves from Swedish trees,” M.S. thesis, Dept. Electron. Eng., Linköping Univ., Linköping, Sweden, 2001.
- [38] P. Novotný and T. Suk, “Leaf recognition of woody species in central Europe,” Biosyst. Eng., vol. 115, no. 4, pp. 444–452, 2013, doi: 10.1016/j.biosystemseng.2013.04.007.
- [39] T. Munisami, M. Ramsurn, S. Kishnah, and S. Pudaruth, “Plant leaf recognition using shape features and colour histogram with K-nearest neighbour classifiers,” Proc. Comput. Sci., vol. 58, pp. 740–747, Jan. 2015, doi: 10.1016/j.procs.2015.08.095.
- [40] G. Vizcarra, D. Bermejo, A. Mauricio, R. Zarate Gomez, and E. Dianderas, “The Peruvian Amazon forestry dataset: A leaf image classification corpus,” Ecol. Inf., vol. 62, May 2021, Art. no. 101268, doi: 10.1016/J.ECOINF.2021.101268.
- [41] Amazon Leaf Dataset. Accessed: May 25, 2021. [Online]. Available: <http://teledeteccion.iiap.gob.pe/dataset/en/>
- [42] N. Kumar, P. N. Belhumeur, A. Biswas, and D. W. Jacobs, “LeafSnap: A computer vision system for automatic plant species identification,” in Proc. Eur. Conf. Comput. Vis., Oct. 2012, pp. 502–516, doi: 10.1007/978-3-642-33709-3_36.
- [43] Dubey, R.K., Choubey, D.K. An efficient adaptive feature selection with deep learning model-based paddy plant leaf disease classification. *Multimed Tools Appl* 83, 22639–22661 (2024). <https://doi.org/10.1007/s11042-023-16247-3>
- [44] Alguliyev, R., Imamverdiyev, Y., Sukhostat, L. *et al.* Plant disease detection based on a deep model. *Soft Comput* 25, 13229–13242 (2021). <https://doi.org/10.1007/s00500-021-06176-4>

- [45] M. J. Hasan, S. Mahbub, M. S. Alom and M. Abu Nasim, "Rice Disease Identification and Classification by Integrating Support Vector Machine With Deep Convolutional Neural Network," *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, Dhaka, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ICASERT.2019.8934568.
- [46] Irri.org. (2019). Home. IRRI. [online] Available at: <https://irri.org/> [Accessed 18 Feb. 2019].
- [47] Knowledgebank-brrri.org. (2019). BRKB. [online] Available at: <http://knowledgebank-brrri.org/> [Accessed 18 Feb. 2019].
- [48] Plantix.net. (2019). [online] Available at: <https://plantix.net/en> [Accessed 18 Feb. 2019].