Java plum leaf disease detection using ML and DL techniques

By

Ashab Rahman
213-15-4479

MD Istiaq Nezoom Molla
213-15-4269

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Md Shohidul Islam Polash

Lecturer
Department of Computer Science and Engineering Daffodil International University

Co-Supervised by

Dr. Naznin Sultana
Associate Professor
Department of Computer Science and Engineering Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

This Project titled "**Java plum leaf disease detection using ML and DL techniques**," submitted by **Ashab Rahman** and **Md Istiaq Nezoom Molla** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17-09-2025**.

BOARD OF EXAMINERS

Dr. Arif Mahmud (AM)
Associate Professor & Associate Head
Chairman, Department of CSE, DIU

Dr. Fizar Ahmed (FZA)
Associate Professor, Internal Member
Department of CSE, DIU

Md. Mushfiqur Rahman (MUR)
Assistant Professor, Internal Member
Department of CSE, DIU

Dr. Md. Manowarul Islam
Professor, External Member
Department of Computer Science and Engineering, Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Md Shohidul Islam Polash, Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

17.9.25

Md Shohidul Islam Polash
Lecturer
Department of Computer Science and
Engineering Daffodil International University

Co-Supervised by:

Dr. Naznin Sultana
Associate Professor
Department of Computer Science and
Engineering Daffodil International University

Submitted by:

Ashab 17.09.25

Ashab Rahman
Student ID: 213-15-4479
Department of Computer Science and
Engineering Daffodil International University

Nezoom

MD Istiaq Nezoom Molla
Student ID: 213-15-4269
Department of Computer Science and
Engineering Daffodil University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Md Shohidul Islam Polash, Lecturer**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Deep Learning** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Java Plum (Syzygium cumini) is a valuable fruit crop in terms of nutritional, medicinal and economic value. Nevertheless, the leaves are very prone to disease that can lower yield and quality which can be a burden on growers. There is thus a need to detect the disease early and accurately to facilitate effective disease management and to maintain sustainability of cultivation. Although recent research on plant disease detection has been based on deep learning models, including convolutional neural networks (CNNs) and Vision Transformers, most methods do not go beyond raw classification performance. There have been only few approaches that use explainability techniques, and, therefore, end-users like farmers and agricultural experts do not have a clear picture on how and why a prediction is delivered.

We presented and tested two complementary approaches to Java Plum leaf disease detection in this work: transfer learning using fine-tuning of the previously trained CNNs and hybrid classifiers that combine CNN feature extraction with traditional machine learning classifiers. A publicly available Kaggle dataset of healthy and diseased Java Plum leaf images was used, with rigorous preprocessing and augmentation to enhance generalization. Pretrained models like ResNet50, VGG16, DenseNet121, MobileNetV2, variants of EfficientNet models were tested in pre- and post-fine-tuning. After the results, it was found that fine-tuning offered a strong performance improvement of most models, and EfficientNetB5 reached its highest classification accuracy of 97.9%, making it the most successful architecture with Grad-Cam visualizations showing accurate results as well. Competitive but slightly worse results were achieved with hybrid CNN and Machine Learning Classifier pipelines as compared to the fine-tuned deep learning models. Generally, the findings suggest that hybrid feature extraction models tend to be stronger, yet fine-tuning CNNs models using transfer learning prove to be more precise and addition of explainability methods like Grad-CAM builds confidence and reliability in real-world agriculture. This framework, besides achieving state of art performance, also takes a step toward some practical interpretable solutions to disease monitoring in Java Plum leaf.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The chapter provides an introduction to the research by outlining its significance, the rationale behind the creation of the research, and the aims of the work. It also includes the research methodology, anticipated results, and a clear roadmap of the work.

1.1 Introduction

A plant's growth and productivity are very important not only for agriculture but also for the sustainability of human life. Just as human health is essential for a good functioning plant health and is critical for food security and ecosystem balance. In today's modern era, where agricultural practices are becoming more and more intensive and climate conditions are unpredictable along with plants being more vulnerable to many diseases than ever before. One of the most significant threats to plants is leaf disease, which directly impacts photosynthesis, makes the plant weak, and eventually reduces yield. Furthermore, plant leaf diseases are becoming increasingly common across the globe.The factors that contribute to the spread are changing weather patterns, the absence of early detection and insufficiency of disease management. Plant diseases are now becoming a serious agricultural issue, especially in regions relying heavily on farming. These diseases if not diagnosed can lead to substantial economic losses and food shortages. Identifying new plant leaf diseases is becoming more difficult due to environmental changes. It is essential to determine the factors that contribute to the emergence and rising occurrences of these diseases. To reduce the damage caused by these damaged leaves early detection and identification during the infection process are necessary.

1.2 Motivation

Java Plum is a crop of economic significance, yet the crop is greatly affected by leaf diseases which in most cases are challenging to detect by hand. Conventional disease detection methods are time-consuming, labor-intensive and liable to errors causing delays in intervention and loss of crop. Prior studies on detection of Java Plum leaf disease are few and those that have been conducted are on major crops, such as tomatoes and potatoes.

1.3 Objectives

- To identify existing literature on the detection of plant leaf diseases and gaps in the existing literature in the way Java Plum disease is detected.

- To build a system of image classification that could be used to give an accurate and early diagnosis of the leaves of Java Plums as healthy or diseased based on visual symptoms.

- To compare the efficacy of various transfer learning models and hybrid feature extraction using machine learning classifiers.
- To evaluate models by evaluating model performance metrics
- To examine how fine-tuning and feature extraction methods enhance classification.
- To modify the model by adding explainability techniques like Grad-CAM in order to interpret the model outputs and guarantee transparency.

1.4 Methodology

Our research aims to discover a more trustworthy and smarter way of categorizing the plant leaf diseases with the help of machine learning and deep learning methods. In farming, timely detection of leaf diseases is the key to avoiding loss of produce and reducing the application of dangerous pesticides. Nevertheless, the symptoms of the disease may be very similar to those of other e.g. yellowing, spots, and curling could be present in different diseases. It renders the topic of manual inspection erroneous and irregular. In the development of our model we investigated transfer learning methods with several pre-trained models such as VGG16, ResNet50, MobileNetV2 and a hybrid model which performs feature extraction followed by machine learning classifiers to determine a suitable model of the images of a plum leaf disease in Java. Various performance measures were used to evaluate the models including the confusion matrix, ROC-AUC curve among other performance measures.
We implemented this research with Python programming language and operated mainly on the Kaggle environment. Python had easy access to deep learning and visualization libraries, such as TensorFlow, Keras, NumPy, Matplotlib, Seaborn, and Scikit-learn. Kaggle proved particularly useful because it is easier to use and offers access to GPU acceleration over other platforms as well as it helps to keep the dataset in a convenient place so that we can work with large image datasets easily.

1.5 Project Outcome

The main outcome of our research is to produce an effective detection model of identifying and classifying diseases on the Java Plum leaf using Machine Learning (ML) and Deep Learning (DL) algorithms. The model will properly classify and label healthy and diseased leaves using images and aid in early disease diagnosis in order to aid in proper management of crops. In addition, the paper will also compare the performance of ML and DL techniques and determine their accuracy, efficiency, and real-world applicability in agriculture. With such high technology, the project will also focus on providing farmers with a particular tool that will help them to be able to detect diseases that can inflict their crops before it is too late and eventually improve the health of the crops and the crop yield.

1.6 Organization of the Report

This report is organized into six chapters, each systematically detailing different aspects of the project.

Chapter 1 introduces the research topic, motivation, and objectives.

Chapter 2 presents the background and literature review. Section 2.1 defines the problem and motivation. Section 2.2 reviews related applications and research. Section 2.3 provides a gap analysis. Section 2.4 summarizes the chapter's insights.

Chapter 3 outlines the methodology. Section 3.1 describes the proposed AI framework and system design. Section 3.2 explains diagrams such as context diagrams and UI diagrams. Section 3.3 details the chosen methodology and design rationale. Section 3.4 outlines the project timeline, and Section 3.5 discusses task allocation. The chapter concludes with a summary in Section 3.6.

Chapter 4 focuses on implementation and experimentation. Section 4.1 describes the development environment. Section 4.2 covers model evaluation and comparative analysis. Section 4.3 presents results and interpretations.

Chapter 5 addresses engineering standards and societal impacts. Section 5.1 discusses compliance with software, hardware, and communication standards. Section 5.2 covers impacts on life, society, ethics, and sustainability.

Chapter 6 concludes the report. Section 6.1 summarizes the study. Section 6.2 discusses limitations. Section 6.3 proposes future work.

Chapter 2

Background

This chapter presents a critical need of developing timely and accurate detection of Java Plum leaf diseases to protect crop yield and to facilitate sustainable agriculture. It shows recent development of deep learning methods which greatly enhance accuracy of detecting diseases, and puts more emphasis on the practical usefulness of these methods in improving agriculture and gap study of past research.

2.1 Introduction

Java Plum also known as Jambolana or black plum is a healthy fruit commonly grown in Southeast Asia and is rich in iron, calcium and vitamins. Like most agricultural crops, the production of Java Plum has been at risk due to numerous leaf diseases that have the potential to affect the production and the overall economy of the areas where they are grown. Adequate and prompt detection of such diseases is essential to the health of crops and the security of food and over the last few years, machine learning and deep learning have provided hopeful solutions to enhancing the identification and control of plant disease in precision agriculture. The fact that the number of outbreak incidents of plant diseases is ever-increasing demonstrates that new effective methods of disease detection should be developed to be able to contribute to the preservation of agricultural production and to safe and sustainable farming practices.

In an effort to address these issues, researchers have vehemently researched machine learning and deep learning methods in an attempt to make detection and control of plant diseases in agriculture are more accurate. The classical approaches have been based on the application of convolutional neural networks (CNNs) and are believed to be capable of classifying plant leaf diseases, although CNNs are generally limited by disease symptoms complexity and variability. In recent research, more sophisticated and accurate detection methods with the ability to adapt to a wide range of situations in farming and provide immediate and accurate responses have been highlighted. Vision Transformer (ViT) is mostly a new image classification method that has not seen much use in plant leaf disease detection, but experiments suggest that well trained ViT models can attain the effect of very high accuracy[1]. It has also been shown, e.g., with federated learning models such as the FL-CNN model, that learning with high diagnostic accuracy is possible while maintaining the privacy of distributed datasets unviolated, which is a big step towards collective and safe disease diagnosis in agriculture[2][3]. Other comparative studies using conventional machine learning algorithms such as Naive Bayes (NB) and K-nearest neighbour (KNN) have determined that NB is far more effective in the accuracy of the choice of robust algorithms used during the practice of effective plant disease control[4]. Likewise, the increasing issue of other related plant leaf diseases where recent research through CNN models has attained remarkable accuracy of up to 95.61, has continued to demonstrate the transformative potential of deep learning in terms of protecting the welfare of farmers [5][6].

However, most recently, sophisticated deep learning models have been effectively used in which a two-step algorithm with a combination of YOLOv8 object detector and a more efficient U-Net segmentation model demonstrated to achieve 95.3% accuracy in leaf recognition [7]. YOLO (You Only Look Once) has become a very useful tool in detecting plant diseases because it can detect objects at high accuracy and in real time. The architecture of YOLO allows identifying and localizing symptoms of diseases on the leaves of plants even in changing and challenging backgrounds in a few seconds. Recent research has shown that by using YOLO models, including YOLOv8, detection of leaf diseases can attain a very high rate of accuracy. [8][9].

Besides individual model methods, hybrid methods have been proposed as an effective plant leaf disease detection method. These models integrate the advantages of several machine learning and deep learning algorithms, including convolutional neural networks (CNNs), support vector machines (SVMs) and ensemble models, to improve diagnostic precision and resilience in a wide range of agricultural scenarios [10][11][12][13].

Regardless of these encouraging developments, there is a lack of studies on Java Plum leaf disease with little dedicated datasets and limited investigation of more complex or hybrid deep learning models in this crop. Furthermore the majority of the currently available studies are not properly tackling the issues of generalization and interpretability and practical implementation, which are essential in the context of real-life agricultural implementation.

In an effort to address such gaps, this study suggests a comparative framework in which both transfer learning using fine-tuning and hybrid feature extraction using machine learning classifiers will be used in the detection of Java Plum leaf diseases. The framework also includes methods of explainability like Grad-CAM, to offer visual justification of predictions and increase the trust of users. Through a systematic study of several pretrained CNNs on a Java Plum leaf dataset, the study identifies which models are the most effective in addition to showing how deep learning can be applied to crops that are yet to be explored.

2.2 Literature Review

Deep learning redesigned the process of plant disease detection and has achieved advancements in recent years. In the case of Java plum (Syzygium cumini), the prevention of yield loss is dependent on detecting leaf diseases early. This review describes recent research in the area of deep learning-based plant disease detection.

Out of the limited studies conducted specifically on Java Plum leaf diseases, Bhowmik et al[1] is concerned with the detection and classification of Java Plum leaf diseases using a trained Vision Transformer (ViT) model. In the study, the researchers point out that optimised hyperparameters are crucial to model performance and attain high accuracy of 97.51%. Mehta et al[2] demonstrated a new approach in diagnosing Java plum leaf diseases based on a federated learning convolutional neural network (FL-CNN) model that reached an accuracy of 96%. Likewise Vats et al[3] proposed a federated learning convolutional neural network methodology in Java plum leaf disease diagnoses that showed good performance on local clients, with F1-Scores near 90% and accuracies of 95% to 96%.

The performance of K-Nearest Neighbours (KNN) versus Naive Bayes (NB) algorithms is given by Reddy et al[4] who introduce methods to detect disease, where the NB algorithm has reached an accuracy of 95% and K NN has reached an accuracy of 84% and show the higher predictive abilities of the NB method over K NN. Al Haque et al [5] proposed a deep learning method based on CNN to witness the fruit canker, anthracnose and fruit rot. This research on guava fruit leaf in Bangladesh achieved 95.61% accuracy.

Kannaiah et al[6], uses preprocessing methods such as contrast enhancement and noise removal to improve feature extraction and classification which helps detect plant leaf disease using AlexNet with 94.5% accuracy.

Yao et al[7] also adopted a variant of this by proposing a two-stage detection algorithm using deep learning, which employs YOLOv8 to extract leaves, U-Net that is enhanced to achieve more accurate segmentation, and loss functions. The model was found to have 95.3% accuracy in recognizing the leaf, and good performance in segmenting a disease. A study conducted by Righi et al[8] revealed a plant leaf disease detection framework based on YOLOv7 that allows real-time monitoring based on online leaf images taken by cameras. The model has attained a high accuracy rate of 96% in the detection of diseases in tomato leaves. Wen et al[9] proposed a modified YOLOv8 to detect plant leaf diseases with a maximum accuracy of 90.5% and increase Mean Average Precision to 73.6% by refined dynamic object detection and a sophisticated loss function.

Thai et al[10] proposed MobileH-Transformer, a hybrid network that uses CNNs along with Transformer to effectively identify diseases of leaves. In this model, a dual convolutional block is used to extract features and the size of the input is reduced to feed the Transformer component that balances accuracy and computational efficiency. Assessments conducted on publicly available data reveal that the model can attain F1-scores of 97.20% on corn and 96.80% on PlantVillage. Aboelenin et al[11] proposed a hybrid architecture that integrates Vision Transformer and Convolutional Neural Networks to improve the accuracy of disease classification. In this model CNNs are first combined to extract robust global features and then ViT is used to extract local features. When tested on data on Apple and Corn, the model demonstrated 99.24% and 98% accuracy, respectively, and outperforms other methods. Venkatraman et al[12] introduced hybrid deep learning that uses Squeeze-and-Excitation network architecture with channel attention and Swish ReLU activation. This design not only enhanced the choice of features, but also the frequent activation problems that were present in other designs, and achieved an astonishing F1-score of 99.76% and accuracy of 99.74%. The results indicate that superior DL techniques are effective in detecting plant diseases in a reliable and efficient manner.

Saleem et al[13] advances the classification of wheat disease, by a strategy adopted multi-scale feature extraction and image segmentation based on deep models such as Xception, Inception V3, and ResNet50, with ensemble classifiers. The accuracy of the proposed method was 99.75% and Xception was found to be the most effective in identifying diseases compared to the current methods. Deep feature extraction with SVM has been demonstrated by Sethy et al[14] to perform better than traditional transfer learning on rice leaf disease detection. When 5,932 field images were used, the ResNet50+SVM model demonstrated the highest F1 score of 98.3% with a higher performance than CNN-based and traditional feature-based classifiers.

Banerjee et al. [15] reported a study with a high-accuracy model to identify turmeric leaf diseases, reporting more than 97% of key performance indicators. El Akhal et al.[16] implemented a hybrid framework that incorporated convolutional neural networks and machine learning classifiers in a study, in which they trained 30 different deep hybrid models with six CNNs and five ML classifiers. They were based on evaluation metrics such as

accuracy, precision, recall, and F1-score, cross-validation, and the NPSK test. The most efficient model, which is the merger of EfficientNetB0 and logistic regression, achieved an accuracy of 96.14%.
Sundhar et al.[17] employs a model that involves superpixel segmentation of features and augmentation of edges to enhance robustness. Tested on apple, potato and sugarcane leaf diseases this hybrid model showed high performance with precision, recall and F1-scores greater than 0.97% in most categories. Ait Nasser et al.[18] present CTPlantNet, a hybrid deep learning algorithm that uses Convolutional Neural Networks and Vision Transformers to effectively classify plant diseases, on two open-access datasets, the CTPlantNet scored 98.28% and 95.96%. Pavan et al.[19] shows that the state-of-the-art accuracy of 99.2 in the classification of plant leaf disease can be shown using transfer learning using architectures like EfficientNetB5, which can be used to illustrate the potential of CNN-based models as useful tools in monitoring of plant health. Kaur et al.[20] used EfficientNetB5 which demonstrated the best accuracy of 98.52% among similar CNN models, which demonstrates that deep learning has the potential to revolutionize the world of plant disease detection and sustainable agriculture.

Table 2.1: Summary of Literature Review.

Author(s)	**Year**	**Methodology**	**Accuracy**
Bhowmik et al[1]	2024	Customized Vision Transformer (ViT) with optimized hyperparameters	97.5%
Mehta et al[2]	2023	Federated Learning with Convolutional Neural Network (FL-CNN)	96%
Vats et al[3]	2023	Federated Learning with Convolutional Neural Network (FL-CNN)	96%
Reddy et	2022	Machine learning algorithms (Naive Bayes vs	NB: 95%,

al[4]		KNN)	KNN: 84%
Al Haque et al[5]	2019	Convolutional Neural Network (CNN)	95.61%
Kannaiah et al[6]	2021	AlexNet	94.5%
Yao et al[7]	2023	Two-stage deep learning (YOLOv8 for detection + Enhanced U-Net for segmentation)	95.3%
Righi et al[8]	2023	YOLOv7	96%
Wen et al[9]	2023	Modified YOLOv8	90.5%
Thai et al[10]	2022	MobileH-Transformer (CNN + Transformer hybrid)	97.2% (Corn), 96.8% (PlantVillage)
Aboelenin et al[11]	2023	Hybrid CNN + Vision Transformer	99.2% (Apple), 98%

			(Corn)
Venkatraman et al[12]	2022	Hybrid deep learning (SENet + channel attention + Swish ReLU)	99.7%,
Saleem et al[13]	2020	Multi-scale feature extraction & segmentation with Xception, InceptionV3, ResNet50 + ensemble classifiers	99.7%
Sethy et al[14]	2020	Deep feature extraction with ResNet50 + SVM	98.3%
Banerjee et al[15]	2021	CNN-based turmeric leaf disease detection	97%
El Akhal et al[16]	2021	Hybrid CNN + ML classifiers	96.1%
Sundhar et al[17]	2022	Hybrid model with superpixel segmentation & edge augmentation	97%
Ait Nasser et al[18]	2023	CTPlantNet (CNN + Vision Transformer hybrid)	95.9%

Pavan et al[19]	2023	EfiicientNetB5	99.2
Kaur et al[20]	2024	EfficientNetB5	98.52

2.3 Gap Analysis

Although extensive studies have been conducted on the leaf diseases of commonly grown crops such as potatoes and rice, the leaf diseases of Java Plum have not received sufficient research attention and have therefore been underexploited and thus not many specialized datasets and models addressing this crop exist.

Deep learning has a great potential to detect diseases in Java Plum leaves, but there are still a number of challenges. Generalization is one of the main barriers: despite the high performance of pretrained models, like ResNet50 and EfficientNet, on well-curated datasets, their accuracy tends to decrease when dealing with real-world images since they have to work on a small dataset and on a small diversity of images. This causes overfitting whereby the models perform well in training but poorly in new leaf images with different levels of disease and environmental factors.

The other important gap is model interpretability. People working in the agricultural domain find it hard to understand based on what visualization the model is making, which is why visual interpretability such as Grad-CAM offer useful visualization that emphasizes the key parts of the leaf that affect the classification, thus enhancing transparency and user trust.

Table 2.3: Gap Analysis table

Authors	Data Augmentation	Pretrained Models	Hybrid Models	Visual Interpretability
Bhowmik et al[1]	Yes	No	No	No
Mehta et al[2]	Yes	No	No	No
Vats et al[3]	Yes	No	No	No
Pavan et al[19]	Yes	Yes	No	No
Kaur et al [20]	Yes	Yes	No	No

2.4 Summary

In this chapter, we provide an in-depth background review of java plum leaf disease and existing detection methods for java plum leaf disease and other plant leaf diseases. The chapter starts by introducing the significance of proper and timely detection of diseases in ensuring crop productivity and economic sustainability. Conventional methods based on machine learning implementations for disease detection and other methods of deep learning, in particular, Convolutional Neural Networks (CNNs) have yielded better results, with accuracy over 95% in certain cases. Recently, new examples such as Vision Transformers (ViT) and Federated Learning CNNs (FL-CNN) also show high diagnostic accuracy and have the advantage of collaborative learning and preservation of privacy of data.

Another area of the chapter that is worth noting is the hybrid models combining CNNs, transformers, and ensemble classifiers that boost feature extraction and resilience through diverse farming environments. YOLOv8 and enhanced U-Net are discussed as object detection and segmentation methods that can be used to detect and localize disease symptoms in real-time. Table 2.1 reports the results of comparative studies that demonstrate that deep learning models are always more effective than traditional ones.

Lastly, the gap analysis reveals that major challenges are the following: Java Plum leaf diseases are not investigated; the provided datasets are small and unreliable due to overfitting and poor generalization; and most of the models are not interpretable, so agricultural specialists do not trust predictions. Explainable AI methods, including Grad-CAM, are highlighted to deliver visualization and boost user trust. The chapter provides a preface to the present research by evaluating the current methodologies, comparing their performance, and identifying the areas in which additional innovation is required.

Chapter 3

Research Methodology

The chapter outlines an end-to-end approach to detect and classify Java Plum leaf diseases based on a deep learning framework. It outlines the steps of data collection and preprocessing to train a model by transfer learning with fine-tuning and hybrid feature extraction algorithms to guarantee a correct and reliable disease diagnosis.

3.1 Methodology

3.1.1 Overview

This research gives a generalized method of detection and classification of plant diseases, in this case, Java Plum leaf diseases, using deep learning techniques. The study design is well organized and presents the following steps: data preparation, preprocessing, model building, fine-tuning, hybrid modeling, evaluation, and interpretation improvement. The dataset which is composed of the images of different plant diseases are pre-processed rigorously to provide uniformity to all the input data. Data is categorized into training, validation, and testing subsets to guarantee an effective model training and effective performance testing. In this study two main approaches are used, transfer Learning with Fine-Tuning where Convolutional neural networks (CNNs) which have been pretrained on the ImageNet dataset are fine-tuned on the domain of the plant disease, and a specialized classification head is introduced to the model to detect the diseases. This method utilises the representational power of the pretrained models alongside enhancing performance on the targeted task. The other method is Hybrid Feature Extraction with Machine Learning Classifier where extraction of features based on the pretrained CNN followed by their input into classical machine learning based classifiers like Random Forest, Support Vector Machines (SVM). This method balances the strength of representing features with deep learning and the robustness and interpretability of classical ML algorithms.
Both approaches are evaluated by using standard measures of performance evaluation. Moreover, explainable AI methods like Grad-CAM are used to give visual information about the models prediction, improving understandability and reliability by agricultural professionals.

3.1.2 Proposed Methodology

The proposed methodology aims to develop a deep learning-based framework for the automated detection and classification of plant diseases in Java plant leafs from image data.

The entire process is designed into a set of consecutive steps, starting with data collection and preprocessing, followed by model selection, fine-tuning, evaluation, and deployment.
The data in this study is represented by labelled pictures of the healthy and sick Java plants. Agricultural datasets were obtained publicly and augmented further with methods of rotation, flipping, and scaling to increase model generalization. Preprocessing procedures such as image resizing, image normalization, and label encoding procedures were done to provide uniformity and consistency to all input images.
Two strategies were applied to the model development:
Transfer learning was done using pretrained convolutional neural networks (CNNs) including ResNet, VGG, and EfficientNet. Firstly, the models were pretrained using frozen layers to memorize their learned representations on large-scale datasets such as ImageNet. It was then fine-tuned by unfreezing the necessary layers to enable the networks to adapt to the specific task of plant disease classification. The measures of model performance were accuracy, precision, recall, F1-score, and confusion matrices.
In the second method, outcomes of pretrained CNNs were obtained as features and were fed into conventional machine learning classifiers, such as Random Forest, Support Vector Machines (SVM). These classifiers were trained and tested by the same performance measures as the transfer learning models.
To aid in transparency and interpretability, model Interpretability Grad-CAM visualizations were run on both strategies to give information on the parts of the plant images that contributed the most to the models prediction. According to the evaluation metrics and the visual analysis, the most effective model will be deployed in Java Plum leaf disease detection.
The general process of the methodology is shown in Figure 3.1, which demonstrates every step in the data preparation process all the way to the implementation of a model.

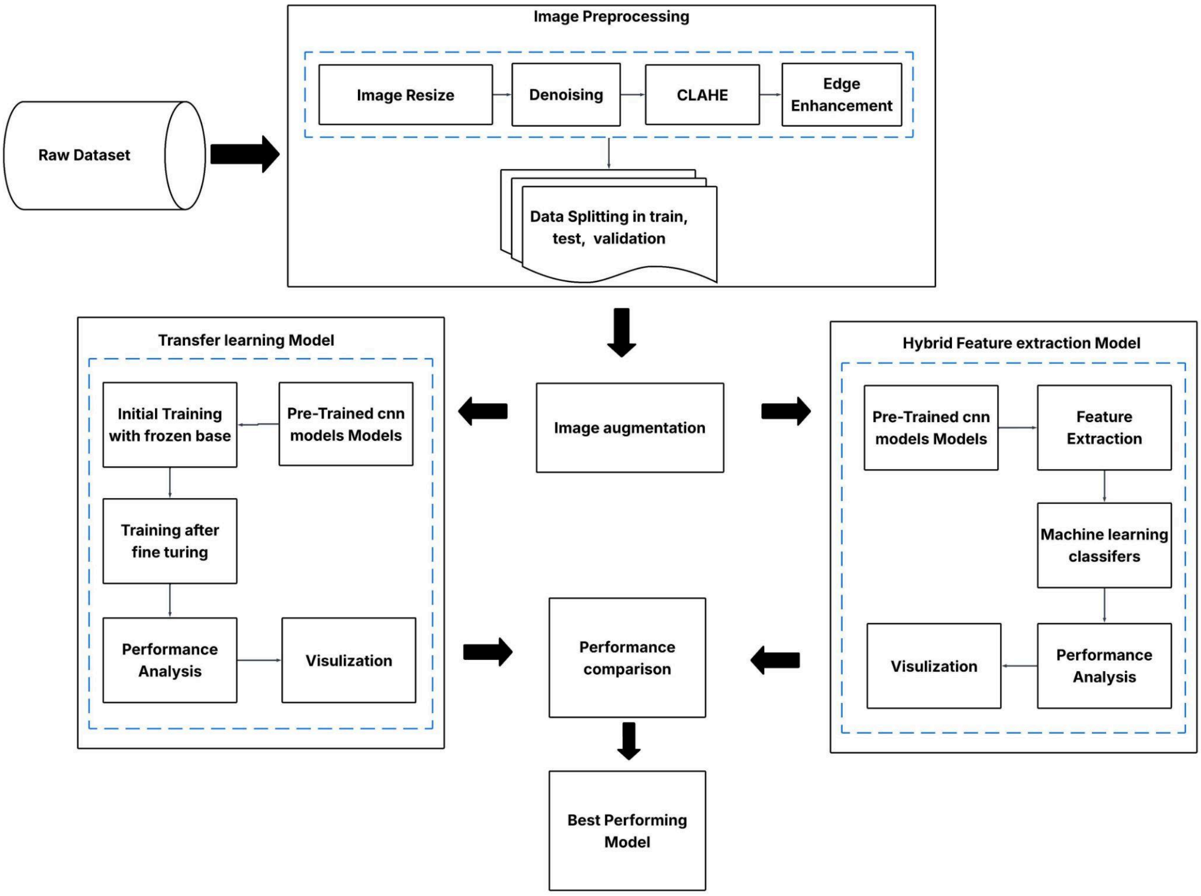


Figure 3.1: Proposed methodology

3.1.3 Dataset Collection

In order to make this research relevant and valid, we have limited ourselves to images of Java Plum leaves based on the Healthy vs. Diseased Leaf Image Dataset [21] available at Kaggle. The dataset contains 624 images from two classes healthy and diseased with 345 images in diseased class and 279 in healthy class. This selectively filtered data is a valuable addition to research on detecting and classifying plant diseases, as it will not only focus the analysis but also increase the accuracy of the machine learning and deep learning algorithms trained to identify the diseases specific to Java Plum. Focusing exclusively on this crop means that the findings and the models can be directly applied to the real-life agricultural problems of the Java Plum growers and, eventually, help to obtain more accurate and viable disease management solutions.

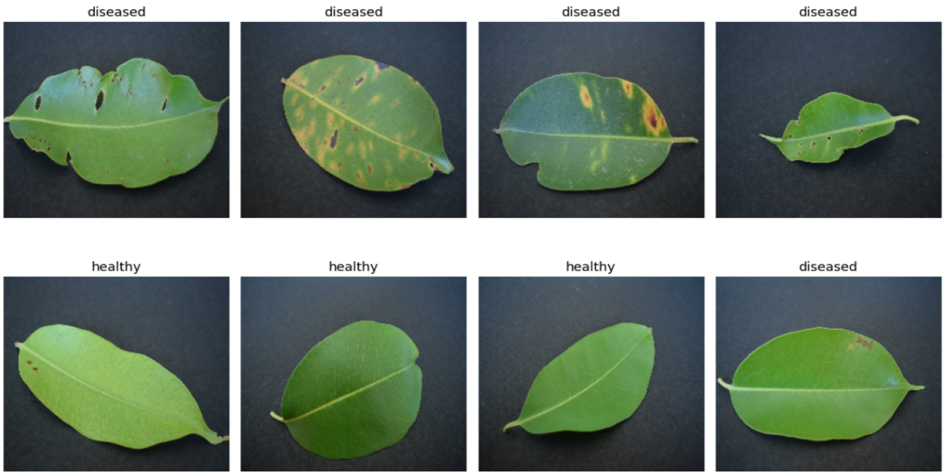


Figure 3.1.3: Healthy and Diseased Images

3.1.4 Preprocessing

As the first step before we trained the models the Java Plum leaf images were properly preprocessed to ensure the dataset used to train the model was clean, balanced, and representative of the task. To accommodate deep learning, all the images, healthy and diseased, were resized to the common 224 by 224 pixel size to get the images into a consistent format across the dataset. Bilateral filtering was applied to each image to reduce the noise, but not the useful information. We then used the CLAHE in the LAB color space, making the contrast higher, and disease patterns less obvious in the image. Edges were sharpened in order to highlight the features and boundaries of the most importance to classification. Then the images were rescaled to the range of 0 to 1 that proves beneficial in order to make the model learn more effectively. Then the dataset was split into 75% train, 15% test and 15% validation set which puts the image as 436 train and 94 in test and 94 valid split. All this is done to ensure that the data was in the optimum format to generate a valid Java Plum leaf disease detection model. To strengthen the dataset and to eliminate any bias we have used data augmentation methods including flipping and rotating the images and ensured that the count of healthy and diseased samples was equalized by generating new images in the small category.

Figure 3.1.4 : Image after preprocessing

3.1.5 Experimental Setup

This study experiment was implemented in the Keras library through the Kaggle platform, which provides an efficient deep learning research and experimentation platform. The basic framework around model development and training was TensorFlow, a popular Python library to perform machine learning and deep learning tasks. The cloud-based service provided by Kaggle also includes a high-performance GPU and plenty of computing resources, which include around 16 GB of RAM and specific GPU acceleration, which makes it possible to easily train deep learning models.

3.1.6 Model selection

To detect Java Plum leaf disease, we have chosen some convolutional neural networks (CNNs) based on their architectural advantages, their demonstrated generalization ability, and their recognition and classification of images. A total of two methods were used, which include transfer learning with fine-tuning and hybrid feature extraction using machine learning classifiers.
Transfer learning involved the following pre-trained models, where layers were frozen, followed by fine-tuning to adapt to the domain of plant diseases:
ResNet50: This network is a 50-layer deep residual network which has skip connections to reduce the vanishing gradient issue. It can capture hierarchical and complex features and is therefore best at performing detailed classification tasks like leaf disease detection.
MobileNetV2: This is a lightweight neural network that is optimized to run on a mobile device. It employs reversed residuals and linear bottlenecks to ensure accuracy as well as fewer parameters. It is efficient and can be used in real time to monitor agriculture.
DenseNet121: It is linked with high connectivity in which all the layers are supplied by all the previous layers. This architecture promotes reuse and feature gradient flow and fine gradient disease symptoms.

VGG16: A standard CNN structure, consisting of 16 layers, constructed using stacked 3 x 3 convolutions. Although it is relatively simple, it is efficient to extract fine local features like edges and textures and is uniform in structure, enabling transfer learning.
EfficientNetB0: Adds a compound scale technique to scale the depth, width and resolution in a uniform way. It balances well between accuracy and computational efficiency, which means it can be applied to moderately sized datasets.
EfficientNetV2B3: EfficientNet, now with a better version called EfficientNetV2B3, which is much faster to train, converges more easily and is more parameter efficient. It can capture finer details and can change to customized image areas.
EfficientNetB5: The larger implementation of EfficientNet which can make use of high-level and complex features. Its depth and scaling properties particularly allow its application in high accuracy on hard classification problems.
In addition to end-to-end transfer learning, we applied pre-trained CNNs to generate features, which were then fed into classical machine learning models to generate hybrid models namely Random Forest, SVM. The approach relies on the deep CNNs describing the features and the interpretability and robustness of the ML classifiers. The model of feature extraction are:
VGG16: Fine local features are extrapolated and may be used by classical ML classifiers.
ResNet50: The model presents progressive features and strong representation of slight variations in diseases.
MobileNetV2: This represents a tiny, low-weight, framework capable of generating small, but informative features to be utilized by ML classifiers.
DenseNet121: Effective use of features, and fine-scale detail is preserved in the input to the classifier.
EfficientNetB0: Lightweight feature extractor that has balanced scaling and provides effective embeddings.
EfficientNetV2B3: More parameter efficient and faster extractor.
EfficientNetB5: More powerful feature extractor exploiting fine grained, high level features.

3.1.7 Model Training Strategies

The training process in this study was performed in two, complementary to one another strategies to achieve the maximum potential of deep learning in the detection of leaf diseases. Both approaches exploited pretrained convolutional neural networks (CNNs), which were ImageNet weights that ensured effective convergence and superior generalization.
Strategy 1: Fine tuned Transfer Learning Models.
Transfer Learning ImageNet weights were used to initialize the pre-trained cnn backbones. To classify the two categories of diseases, the last classification head was substituted with a custom dense layer.
Freezing & Unfreezing: In the first stage, convolutional layers were frozen and the dense

layers were only trained over 50 epochs. The second stage involved unfreezing and fine-tuning of the chosen higher-level layers with 100 more epochs and letting the model specialize on features of Java plum leaf disease.

Learning Rate Scheduler: Cosine annealing was employed to dynamically adjust the learning rate, with an initial value of 1 × 10 -4. This served to stabilize convergence and prevent overfitting.

Hyperparameters:

Batch size: 32

Dropout rate: 0.5

Optimizer: Adam optimizer

Regularization and Early Stopping: L2 regularization was used, as well as dropout, and early stopping was used to stop training once the validation loss stopped decreasing and guaranteed improved generalization.

Strategy 2: CNN + ML Classifier feature extraction.

Feature Extraction: Pretrained CNNs such as EfficientNet, DenseNet, ResNet were used as fixed feature extractors not trained end-to-end. Features were extracted using the last-two layers of the CNN.

Machine Learning Classifiers: Extracted features were fed poisoned with a few standard ML classifiers, such as:

Support Vector Machine

Random Forest

Decision Tree

AdaBoost

Comparison: The accuracy, precision, recall, F1-score and ROC-AUC of each classifier were used to make a fair comparison between various hybrid classifications using deep learning + ML with the final fine-tuning.

Using this methodology, the research found a compromise in the strength of the deep features representation, as well as strong decision-making ability of the ML classifiers, which ensured that Java plum leaf disease detection approaches will be thoroughly analyzed.

3.1.8 Performance metrics

The proposed model is tested with widely used measures like precision (P), recall (R) and mean average precision (mAP) in this research. The recall of a model suggests its capability to identify every real appearance of an object inside an image. Accuracy is the percentage of correctly classified cases. It is simple but may not accurately show model performance in imbalanced datasets. Common accuracy equation.Recall is calculated by dividing the sum of real objects by the number of real objects detected. Greater value of recall suggests that even though the model may show a large number of false positives, it is less likely that it will skip over real objects. Precision is a measurement of detection accuracy. The number is obtained by dividing the number of positive detections with the actual detections. High precision implies that even though the model may fail to detect some existing real objects, it is less likely to create large numbers of false positives. Mean average precision (mAP) is a statistic, which is considered to be a measure of the ability to detect items, based on the measure of accuracy at various recall levels. It assesses the item detection accuracy of the model with a fixed range of recall limits. It compares the area under the Precision-Recall curve and averages such areas over all the classes in the data set. And lastly the final product mAP is obtained. mAP is one of the more desirable measures, since it provides an improved reflection of the model performance at different classes and different degrees of complexity in the object identification. By viewing these three measures precision, recall and mean average percentage, it is possible to better understand the overall object identification performance of the model.

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

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

$$\text{Recall}= \frac{TP}{TP+FN}$$

$$\text{mAP}=\frac{1}{n}\sum_{c=1}^{n}\frac{TPc}{TPc+FPc}$$

3.1.9 Data Flow Diagram

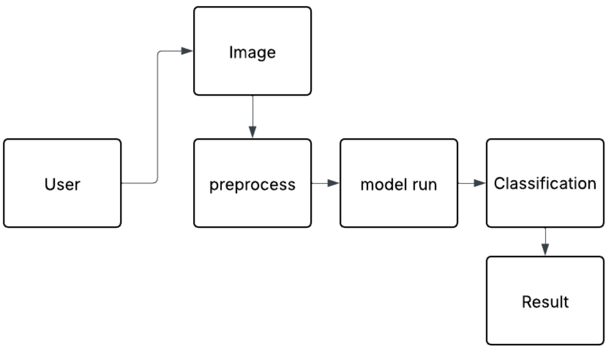


Figure 3.1.10: Data flow diagram

The figure 3.1.10 represents the data flow diagram, it shows how the system would work. Users upload a picture, which is preprocessed by the system then the model is run and as output the classification result is shown to the user

3.1.10 UI Design

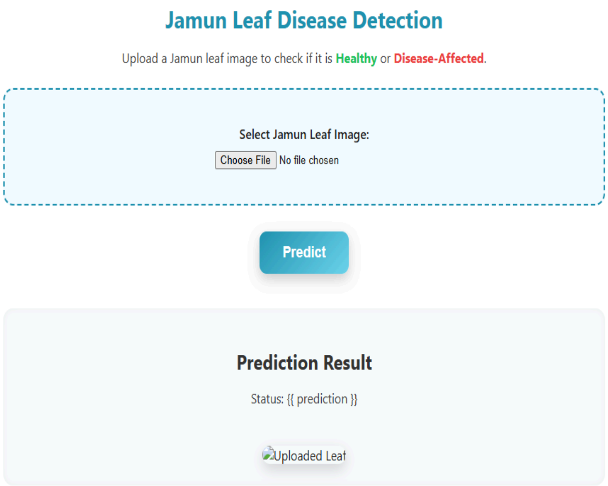


Figure 3.1.11: UI Design

3.2 Detailed Methodology and Design

This section describes the design rationale for the Java Plum leaf disease detection methodology, explaining the alternatives considered and the reasoning behind the final choices.

Alternative Approaches Considered:

Model Architectures: Initially, basic CNN models were trialed but showed limited accuracy and lacked robustness on the leaf dataset. Therefore, pretrained deep networks like ResNet50, EfficientNetB3, and VGG16 were adopted due to their effective feature extraction capability and transferability from large-scale datasets such as ImageNet. In addition to end-to-end transfer learning, a hybrid approach was explored where features were extracted from pretrained CNNs and then fed into classical machine learning classifiers . This hybrid strategy leverages the representational power of deep networks while benefiting from the interpretability and robustness of ML classifiers.

Data Augmentation Strategies:

A combination of the augmentation techniques was employed, so that the generalization abilities of the models could be increased, and that their ability to endure the variability in the data should be covered. This led to countering bias of classes and even simulation of the circumstances of the real world images.

Interpretability Techniques:

There were several ways in which a model's decisions could be visiually explained. Grad-CAM was chosen because it formed intuitive visual heatmaps highlighting image regions that affect our predictions, which is essential to verify whether the models are interested in the right leaf features.

Justification for Selected Approach:

We made the final decision following the following:

Accuracy: Fine-tuned models which were made using the Java Plum leaf disease dataset were more successful in their classification. The hybrid feature extraction models also could provide good performance by using persistent deep feature extraction with machine learning classifiers.

Efficiency: The two used pretrained weights that saves much time and computing power in the necessary training.

Better generalization: Both transfer learning and hybrid feature extraction models generalized well with pretrained features even with quite a small dataset.

Explainability: Grad-CAM was able to provide usable visual feedback with Grad-CAM being incorporated, this increased the trust in the transfer learning prediction but also the hybrid model prediction.

3.3 Project Plan

This project was designed in phases to encourage and sort out the development of a precise and interpretable system of classification of a the Java plum in a systematic manner. The first step was to conduct an extensive literature review in order to familiarize themselves with existing methods and find appropriate pretrained models. After that, preprocessing and augmentation mechanisms of data were deployed to improve the quality and diversity of the data. Subsequently, two transfer learning strategies were implemented to train models: a feature extraction strategy with pretrained CNN architectures, where the pretrained CNNs served fixed feature extractor roles and their outputs were sent to traditional machine learning classifiers; and a fine-tuning strategy, with the pretrained CNNs being partially unfrozen and retrained with classification layers on top of the leaf disease data. Grad-CAM was incorporated to assist in visual interpretability of model predictions. The last step was to document and present a thesis.

3.4 Task Allocation

Successful implementation of this project meant splitting up the work into clearly defined tasks and allocation of duties in order to complete them on time.

This table indicates the chronology of the main activities in each phase of the project from week 12 to week 48.

Table 3.4: Task allocation table

Tasks	Weeks																		
	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48
Data collection																			
Preprocess all the data																			
Model Development																			
Model Fine tuning																			
Documentation																			

3.5 Summary

In this chapter, we described the approach of identifying and classifying Java Plum leaf diseases with the help of deep learning and provided an organized look at how the study was designed and executed. The initial move was to collect high quality images of the Kaggle Healthy vs. Diseased Leaf Image Dataset such that the data set would be as heterogeneous as possible, where healthy and the various disease-suffering conditions were present in the dataset. The models were trained through extreme preprocessing of the images. These included scaling all the pictures to a fixed size of 224x224 pixels, bilinear filtering to eliminate noise, CLAHE to enhance contrast, sharpening, normalization of pixel values and data enhancement techniques such as rotation and flipping. Such steps ensured that the dataset was balanced, representative, and potentially generalized the models strongly.

Two primary modeling strategies were employed to maximize the performance of detection. The former used transfer learning on already trained convolutional neural networks, including ResNet50, VGG16, DenseNet121, MobileNetV2, EfficientNetB0, and EfficientNetV2B3 and EfficientNetB5. These models have been trained initially on frozen layers to retain generic image features and then further fine-tuned on the Java Plum data set to be useful in the disease classification task. The second methodology was hybrid approaches where the features retrieved by pretrained CNNs such as VGG16, ResNet50, MobileNetV2,EfficientNets and DenseNet121 were fed into traditional machine learning classifiers such as the random forest, SVM. The power of deep learning was combined with the power and interpretability of traditional ML methods in this method.

These two methods were strictly compared on accuracy, precision, recall, F1-score, and mean average precision (mAP) and the confusion matrices that gave detailed results about the performance on different classes. Grad-CAM was used to visualize the image regions that most contributed to model predictions, enhancing interpretability and giving confidence in the models decision-making. It is possible to select the most popular model and make an informed decision with the help of quantitative evaluation and qualitative visualization of its achieved results and, accordingly, effectively, accurately, and transparently detect Java Plum leaf diseases, which will then be a strong basis in practice by being applied in the agricultural process.

Chapter 4

Implementation and Results

This chapter presents a comparison and contrast of several ready-to-use deep learning models to detect Java Plum leaf disease with a strong focus on the significant performance observed with both strategies. The specific results are confirmed with the help of metrics, confusion matrices, ROC curves, and visualization of the final results through Grad-CAM.

4.1 Environment Setup

The studies for detecting Java plum leaf disease were carried out using both cloud-based platforms and local computing resources. Enough processing power was guaranteed by the configuration to train and assess deep learning and machine learning models.

Hardware environment

A PC with an Intel Core i5-13500 processor, an Nvidia RTX 3060 GPU, and 16GB of RAM was used for the experiment.

Software environment and tools

The local computer and the Kaggle platform with GPU acceleration were used for model testing and training. Python was used for the implementation, and libraries including TensorFlow, Keras, Scikit-learn and OpenCV were used.

Python: Main programming language, specifically has a rich library collection.

TensorFlow: Main library, used for training and increasing the model accuracy.

Scilit-learn: Mainly used for model evaluation in terms of feature selection, accuracy, F1-score, precision, and recall.

OpenCV: It is used in preprocessing work for image resizing, contrast stretching and denoising.

Jupyter Notebook: It provides a well-interactive environment for modelling and debugging

4.2 Comparative Analysis

This section presents a comprehensive evaluation of the proposed java plum leaf disease detection framework using a few pretrained ML and DL models. All the models were evaluated using a diverse set of metrics including accuracy, precision, recall, F1-score, training and testing loss, training time, inference time, ROC curves, confusion matrices, and Grad-CAM based interpretability and fine-tuned using a consistent architecture.

Table 4.2: Performance Summary of all Transfer learning Models before fine tuning

Model	Accuracy	Precision	Recall	f1-Score	AUC-Score
EfficientNetB0	87.2	87.2	87.2	87.2	93.2
ResNet50	92.6	92.8	92.6	92.6	96.8
MobileNetV2	88.3	89.3	88.3	88.3	97.0
DenseNet121	92.6	93.1	92.6	92.6	98.7
VGG16	84.0	84.1	84.0	84.0	93.4
EfficientNetV2B3	89.4	89.7	89.4	89.4	96.3
EfficientNetB5	93.6	93.7	93.6	93.6	97.5

While transfer learning significantly improved classification performance across models, the results revealed in Table 4.2 shows variation in accuracy and reliability. Models such as VGG16 (84%) and EfficientNetB0 (87.2%) underperformed, indicating difficulties in adapting to the dataset despite their proven scalability on larger benchmarks. In comparison, MobileNetV2 (88.3%) and EfficientNetV2B3 (89.4%) provided more stable outcomes but did not reach the level of accuracy required for dependable disease detection. Stronger performance was observed with DenseNet121 (92.6%) and ResNet50 (92.6%), both of which effectively balanced accuracy and efficiency, making them practical candidates for real-world applications. The best results, however, came from EfficientNetB5, which achieved 93.6% accuracy along with consistently high precision, recall, and AUC scores. This demonstrates the effectiveness of modern, compound-scaled architectures in capturing fine-grained disease features and delivering near state-of-the-art results within the transfer learning paradigm. By benchmarking diverse pretrained CNNs, this study highlights the trade-off between model complexity, computational efficiency, and predictive performance. While EfficientNetB5 achieved the highest accuracy, models like ResNet50 and DenseNet121 stand out as reliable alternatives for resource-constrained environments, offering a balanced pathway toward scalable agricultural disease detection systems.

Table 4.2.1: Performance Summary of all Transfer learning Models after fine tuning

Model	Accuracy	Precision	Recall	f1-Score	AUC-Score
EfficientNetB0	96.8	96.8	96.7	96.8	99.1
ResNet50	94.7	94.7	94.7	94.7	98.7
MobileNetV2	86.2	89.4	86.2	86.1	97.3

DenseNet121	95.7	95.8	95.7	95.7	99.4
VGG16	91.5	91.9	91.5	91.5	98.1
EfficientNetV2b3	95.7	95.7	95.7	95.7	98.9
EfficientNetB5	97.9	98.0	97.9	97.9	99.4

Table 4.2.1 shows, fine-tuning the pretrained models yielded a substantial performance boost compared to the initial transfer learning results with frozen layers. Models that previously struggled, such as VGG16 (84%) and EfficientNetB0 (87.2%), showed dramatic improvements after fine-tuning, reaching 91.5% and 96.8% accuracy, respectively. This highlights the importance of domain adaptation through fine-tuning, which enables models to better capture subtle disease-specific features in Java plum leaves. MobileNetV2 and EfficientNetV2B3, while moderate accuracies of 88.3% and 89.4% before fine-tuning, MobileNetV2 dropped to 86.2% likely due to its lightweight architecture while EfiicientNetV2B3 improved to 95.7%. ResNet50 and DenseNet121 both with 92.6% accuracy rose to 94.7% and 95.7%, demonstrating that even older architectures benefit significantly from targeted adaptation. The highest performance once again was achieved by EfficientNetB5 reaching 97.9% accuracy this time. Overall, fine-tuning not only corrected underperforming models but also pushed already strong candidates like EfficientNetB0 and EfficientNetV2B3 and EfficientNetB5 reached the highest accuracy of 97.9% and across all metrics. Compared to the frozen-layer stage, fine-tuning proved to be the decisive step in unlocking the full potential of these architectures, bridging the gap between state-of-the-art accuracy and practical robustness for agricultural disease detection.

Table 4.2.2: Performance Summary of Hybrid Feature Extraction Models

CNN Model	Classifier	Accuracy	Precision	Recall	F1 Score	AUC Score
DenseNet121	SVM	95.7	95.7	95.7	95.7	99.1
DenseNet121	AdaBoost	95.7	95.7	95.7	95.7	99.1
DenseNet121	Random Forest	92.6	92.4	92.6	92.5	98.9
DenseNet121	Decision Tree	80.9	80.8	81.1	80.8	81.1
ResNet50	Random Forest	93.6	93.5	93.8	93.6	97.5
ResNet50	SVM	91.5	91.3	91.6	91.4	97.3
ResNet50	AdaBoost	91.5	91.3	91.6	91.4	97.9
ResNet50	Decision Tree	81.9	81.8	81.6	81.7	81.6

VGG16	AdaBoost	92.6	92.4	92.6	92.5	96.4
VGG16	Random Forest	89.4	89.4	89.0	89.2	96.5
VGG16	SVM	91.5	91.3	91.6	91.4	96.7
VGG16	Decision Tree	75.5	75.3	75.1	75.2	75.1
MobileNetV2	Random Forest	91.5	91.4	91.4	91.4	97.4
MobileNetV2	SVM	91.5	91.4	91.4	91.4	97.2
MobileNetV2	AdaBoost	89.4	89.2	89.5	89.3	96.9
MobileNetV2	Decision Tree	80.9	80.6	80.6	80.6	80.6
EfficientNetB5	Random Forest	95.7	95.7	95.7	95.7	98.4
EfficientNetB5	SVM	94.7	94.6	94.7	94.6	97.7
EfficientNetB5	AdaBoost	94.7	94.6	94.7	94.6	98.3
EfficientNetB5	Decision Tree	77.7	77.7	77.9	77.6	77.9
EfficientNetB0	SVM	94.7	94.7	94.5	94.6	98.4
EfficientNetB0	AdaBoost	90.4	90.3	90.7	90.4	97.0
EfficientNetB0	Random Forest	87.2	87.1	87.1	87.1	96.1
EfficientNetB0	Decision Tree	72.3	72.1	71.6	71.7	71.6
EfficientNetV2 B3	AdaBoost	92.6	92.6	93.0	92.5	97.9
EfficientNetV2 B3	Random Forest	92.6	92.4	92.6	92.5	96.2
EfficientNetV2 B3	SVM	92.6	92.4	92.6	92.5	97.4
EfficientNetV2 B3	Decision Tree	84.0	84.3	84.7	84.0	84.7

The hybrid feature extraction approach of combining CNN feature extraction with traditional machine learning classifiers produced strong results as shown in table 4.2.2, though overall not surpassing the fully fine-tuned CNN models. Among the hybrids, DenseNet121 paired with SVM,DenseNet121 paired with AdaBoost and EfficientNetB5 paired with Random Forest achieved the best overall performance, reaching 95.7% accuracy and similar other metrics for all three with EfficientNetB5 just having a tiny bit edge on the AUC score. This was closely followed by EfficientNetB5 with SVM and EfficientNetB0 with SVM, both of which demonstrated competitive performance across all metrics. The other nontrivial competitors were ResNet50 + Random Forest with 93.6% accuracy and VGG16 + AdaBoost with 92.6%, which remained at the same level and provided the usefulness of the two CNNs as feature extractors as the component of a hybrid pipeline. The best configurations of EfficientNetV2B3 are also good and can be considered a viable option in the hybrid configuration. MobileNetV2 combinations, in their turn, despite their relatively strong power up to 91.5% accuracy, were often worse than the best EfficientNet and DenseNet combinations. In any models, the Decision Trees always lagged behind in generation and its accuracies never exceeded the low-80s scale and in some situations even lessened to the middle of the 70s. This shows that they are relatively weak compared to more complex methods of ensemble and SVMs. Overall, the results confirm that not all but some hybrid feature extraction can be good at generating good performance and especially with a stronger CNN backbone such as DenseNet121 and EfficientNetB5.

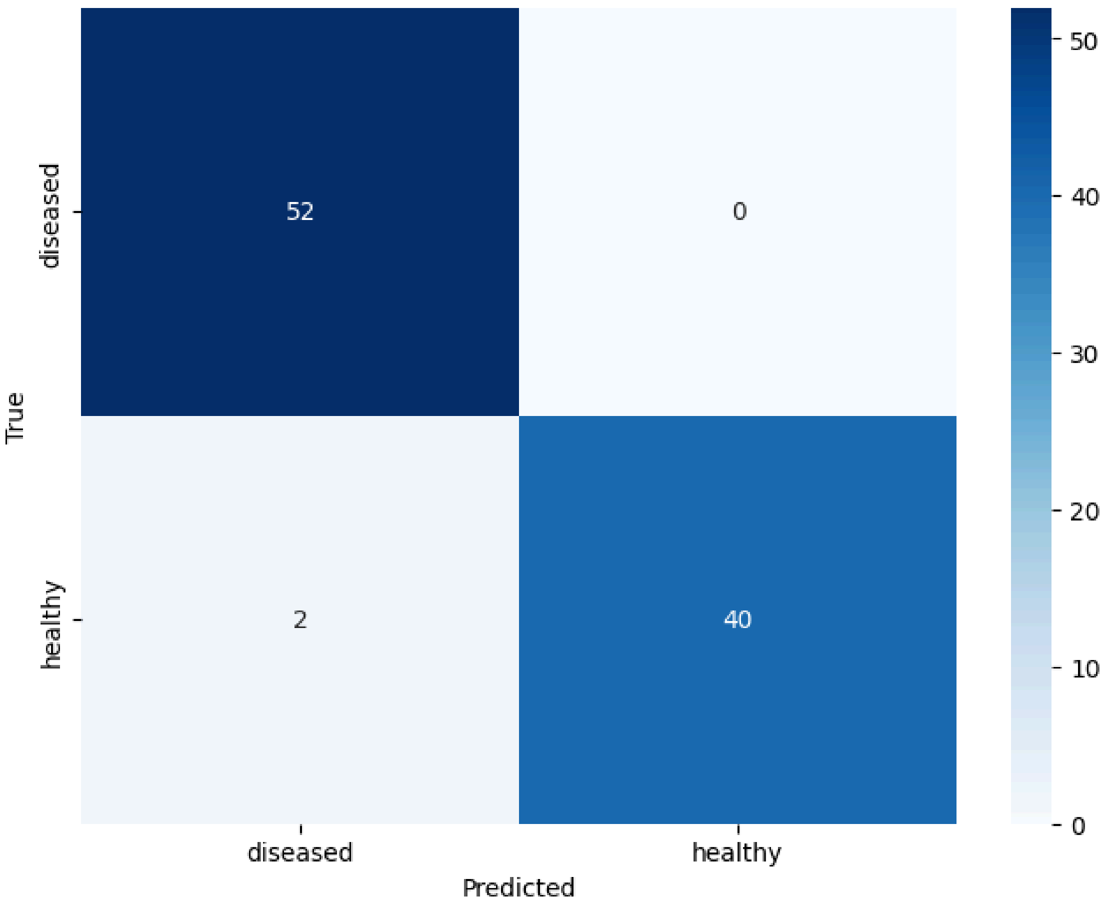
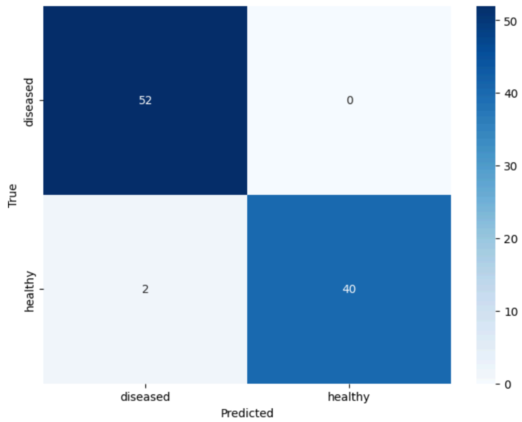
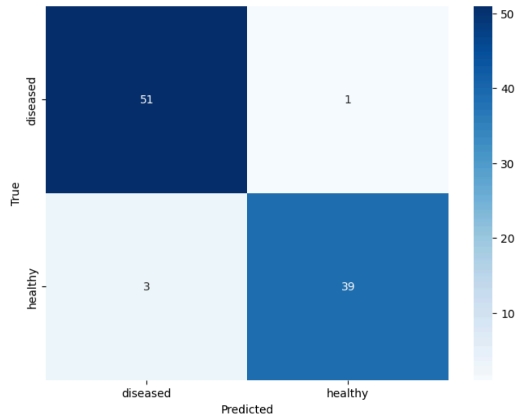


Figure 4.2: Confusion matrix of best performing model EfficientNetB5

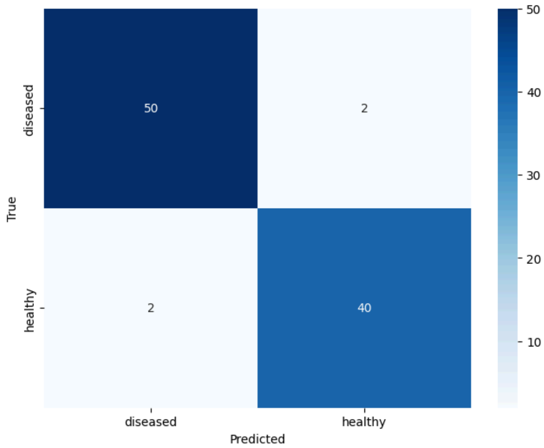
In figure 4,2 as it can be seen the confusion matrix shows the performance of EfficientNetB5. From the 52 samples of diseased class, all were correctly identified and of the 40 samples of class healthy, 40 were correctly identified, with only two misidentification. This is equivalent to high accuracy, precision, recall and F1-score. Results suggest the model would be very effective in the classification task, but it would be essential to test it on various data to guarantee the generalization.



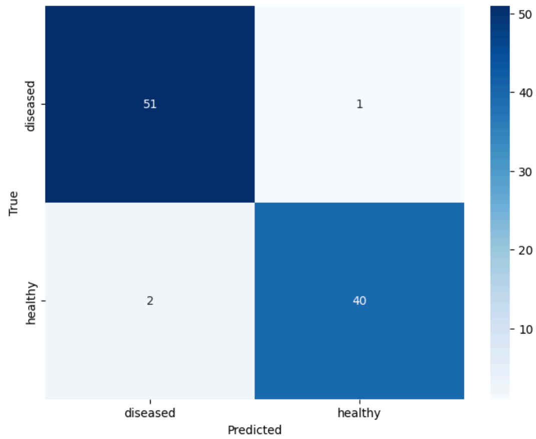
EfficientNetB5



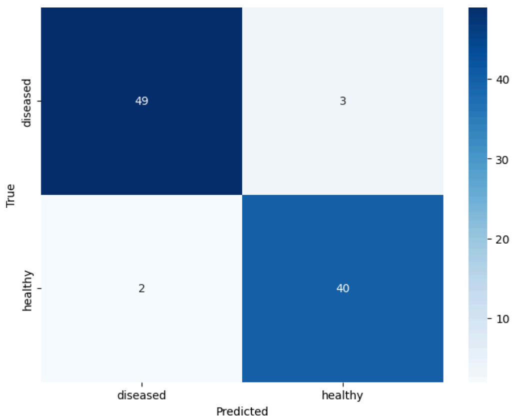
DenseNet121



EfficientNetV2B3

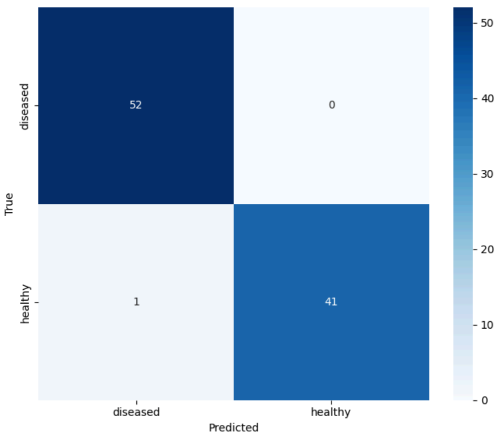


EfficientNetB0

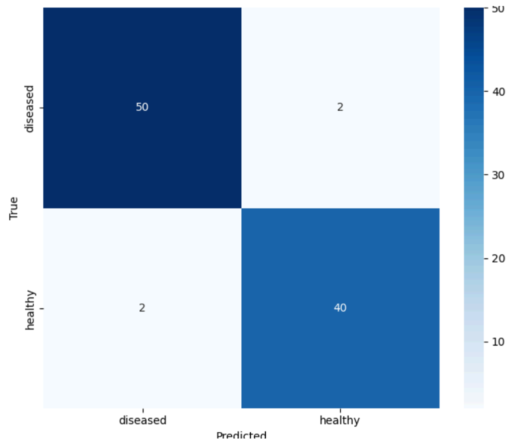


ResNet50

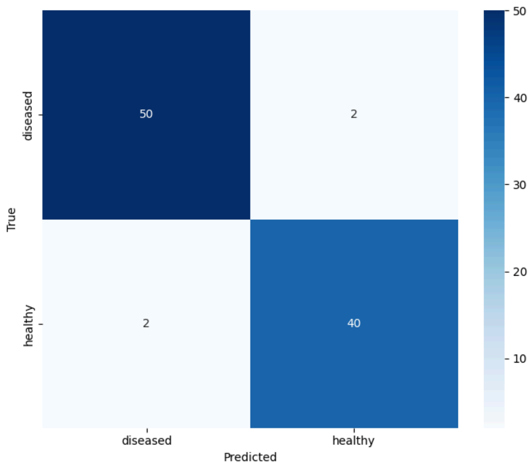
Figure 4.2.1: Confusion Matrix of top transfer learning fine tuned models



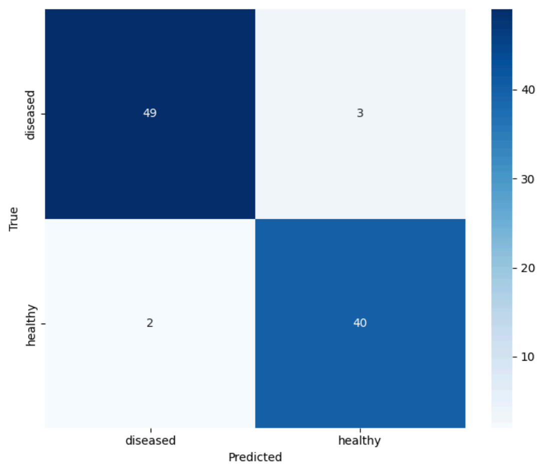
DenseNet121 + SVM



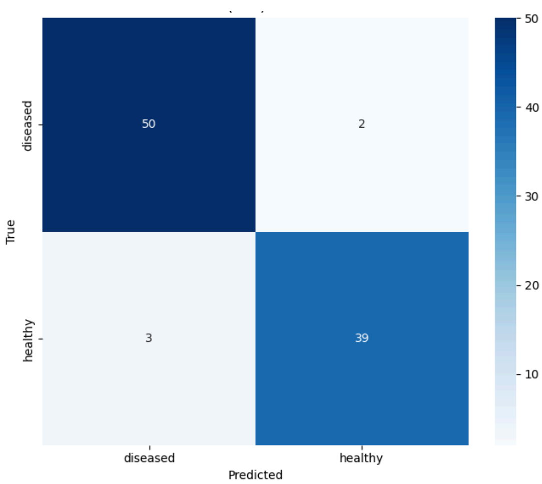
DenseNet121 + AdaBoost



EfficientNetB5 + Random Forest



EfficientNetB5 + SVM



EfficientNetB0 + SVM

Figure 4.2.2: Confusion Matrix of top five hybrid feature extraction models

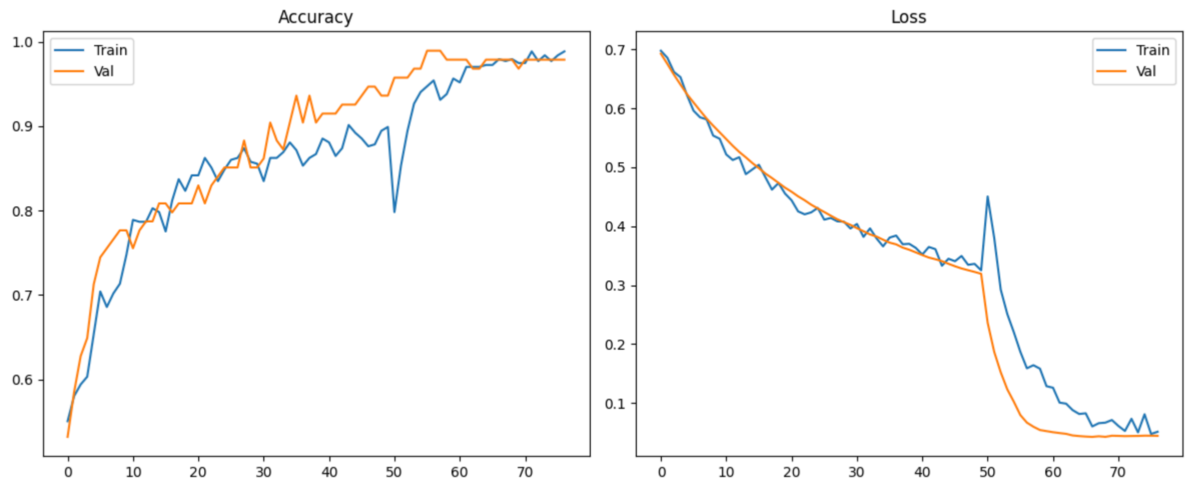
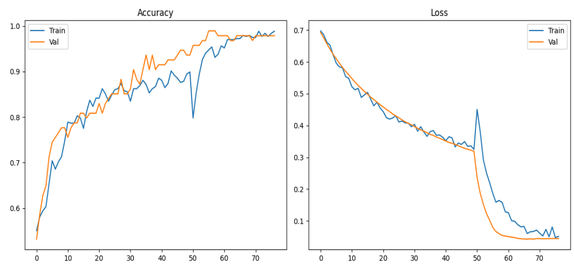
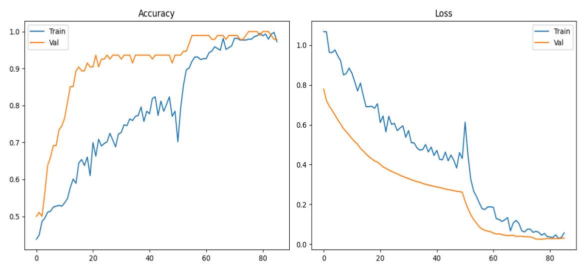


Figure: 4.2.3: Training and validation accuracy loss curve (EfficientNetB5)

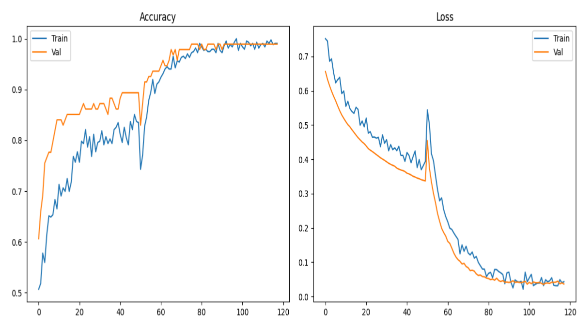
In Figure 4.2.3, the training and validation curves show a stable model which is well-fitted. In the first 10 epochs, training and validation both accuracy remained around 55–80%, with loss dropping steadily from 70%. After 40 epochs the validation accuracy went around 90% and after nearing the 50th epoch, a sharp improvement occurred, where validation accuracy rapidly increased to around 97%, while the corresponding losses dropped at close to zero. Importantly, the training and validation curves followed similar trends without significant divergence, indicating that the model did not suffer from overfitting. Similarly, the consistently high accuracy rules out underfitting. This behavior suggests that a change in learning dynamics, such as learning rate adjustment or layer unfreezing in transfer learning, enabled the model to achieve strong convergence.



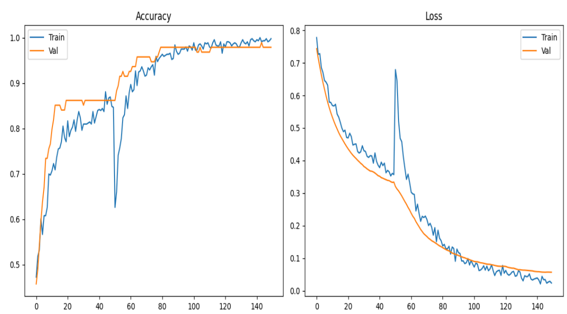
EfficientNetB5



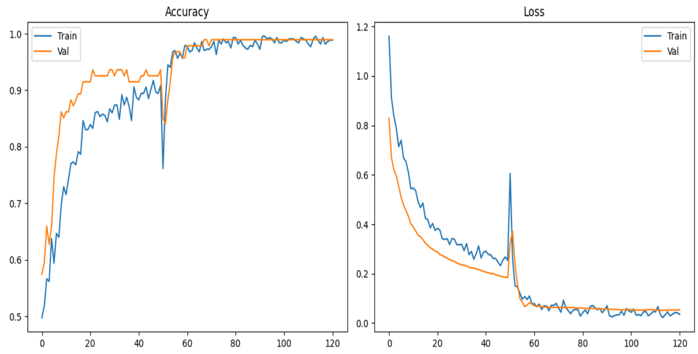
DenseNet121



EfficientNetV2B3



EfficientNetB0



ResNet50

Figure 4.2.4: Training and validation accuracy-loss curve of top transfer learning fine tuned models

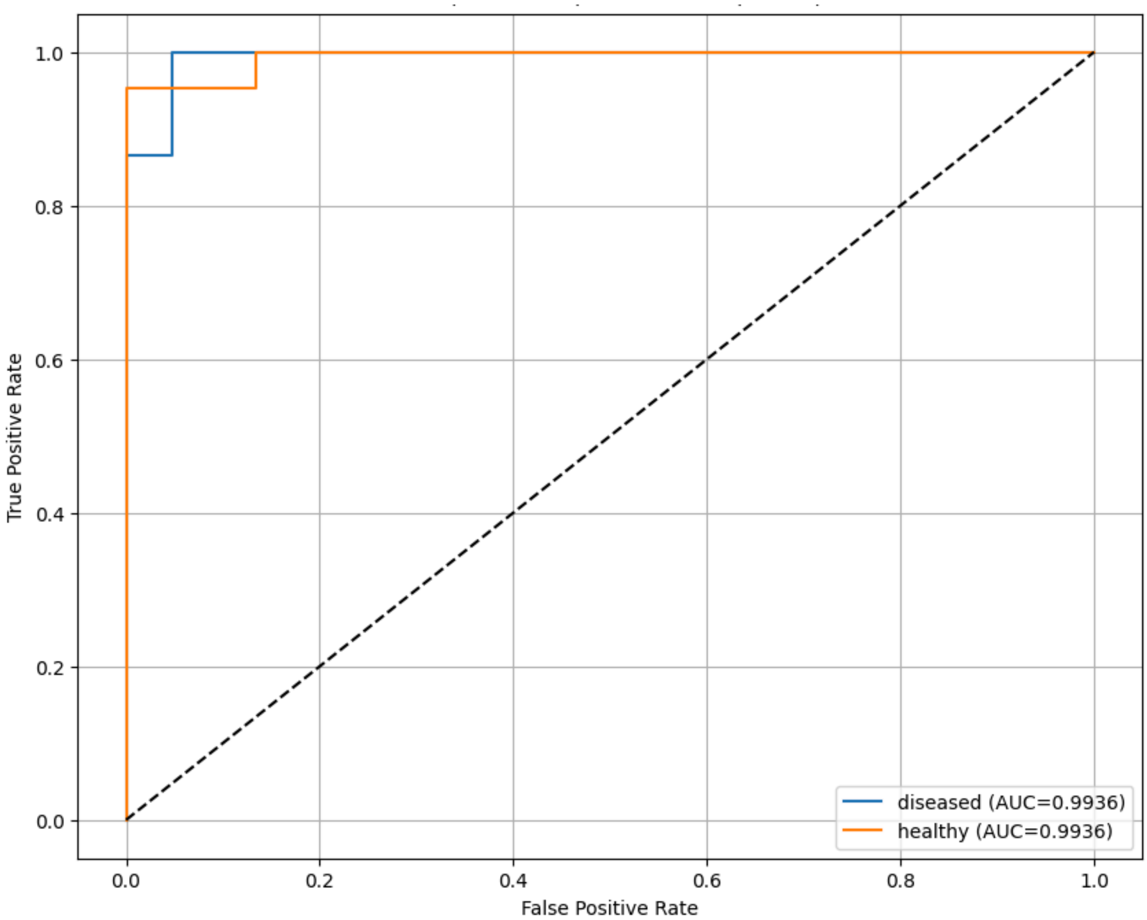
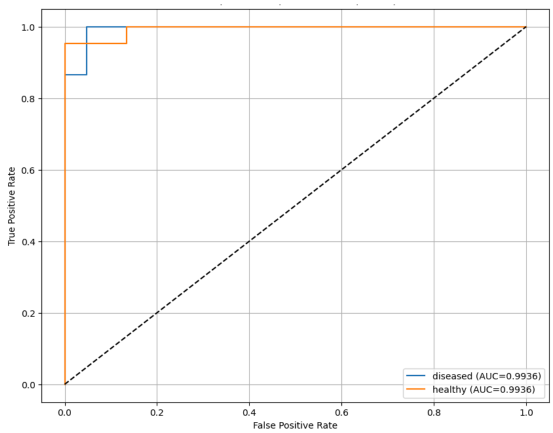
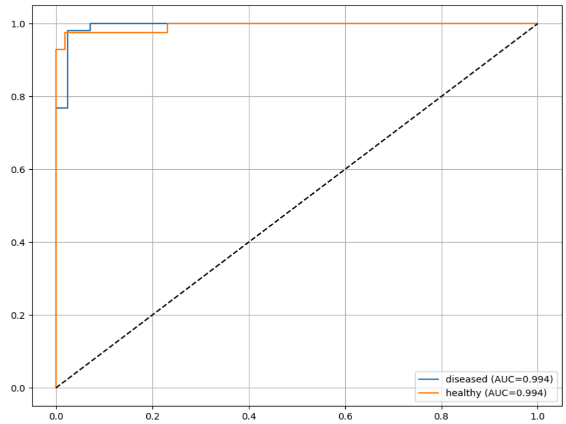


Figure 4.2.5: ROC curve of best performing model EfficientNetB5

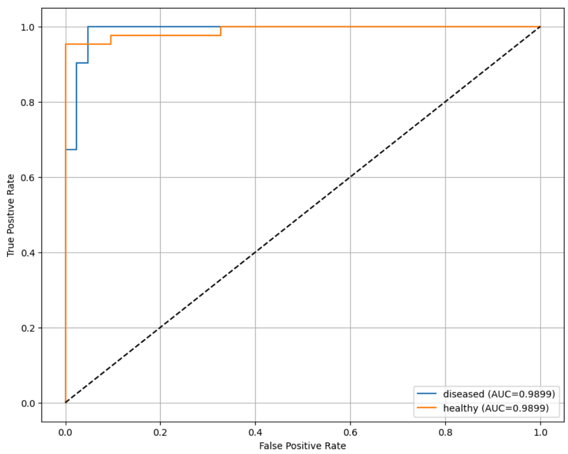
In figure 4.2.5 the ROC curves show that the model had an almost perfect classification performance. The classes diseased and healthy both achieved an AUC value of 0.9936, a very near value of 1.0, indicating great separability between the two classes. These curves are close to the top-left corner of the graph indicating a very high true positive rate and a very low false positive rate. These findings confirm that the model works very well to differentiate between the two classes.



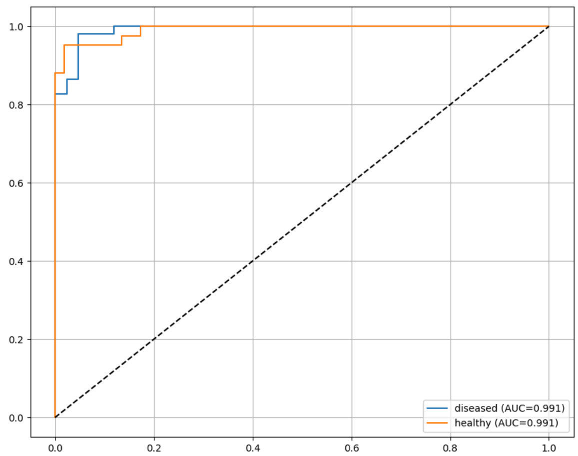
EfficientNetB5



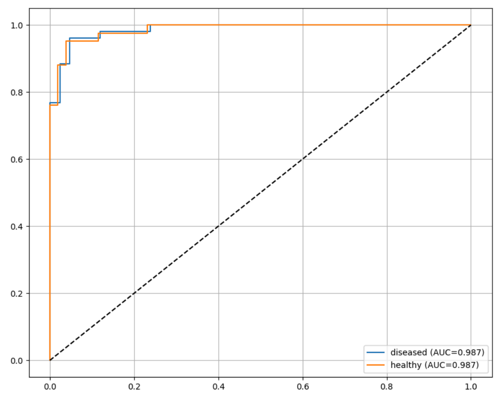
DenseNet121



EfficientNetV2B3

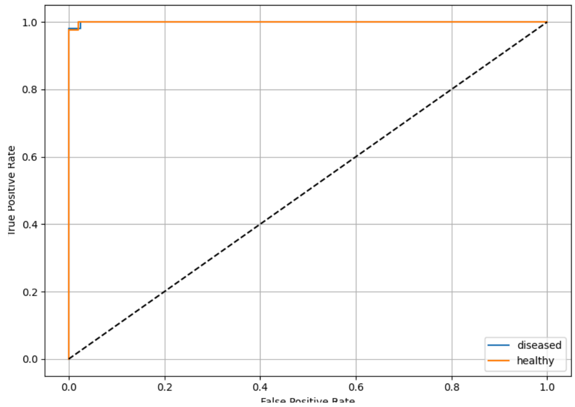


EfficientNetB0

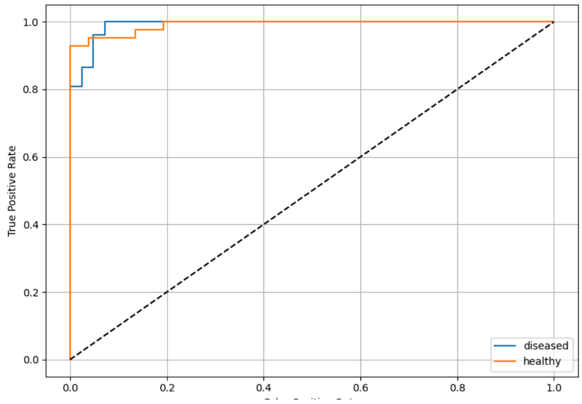


ResNet50

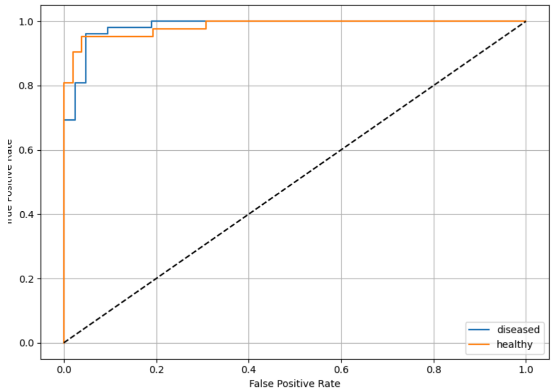
Figure 4.2.6: Roc curve of top transfer learning fine tuned models



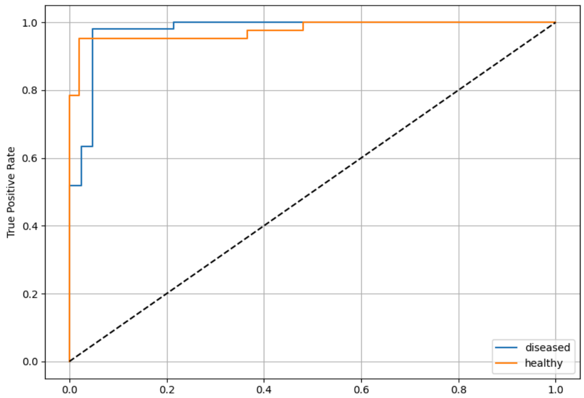
DenseNet121 + SVM



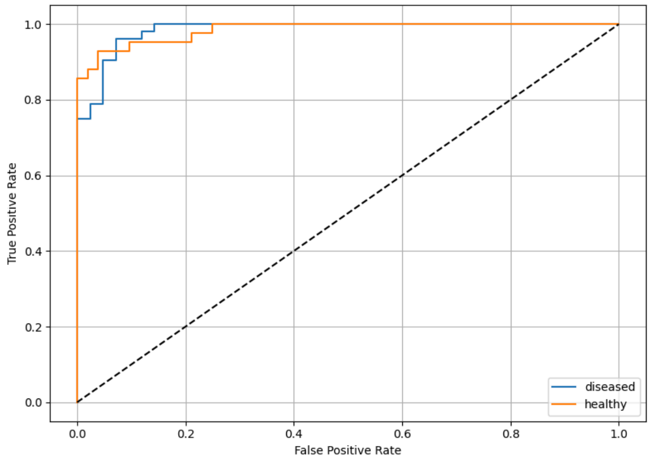
DenseNet121 + AdaBoost



EfficientNetB5 + Random Forest



EfficientNetB5 + SVM



EfficientNetB5 + SVM

Figure 4.2.7: ROC curve of top hybrid feature extraction models

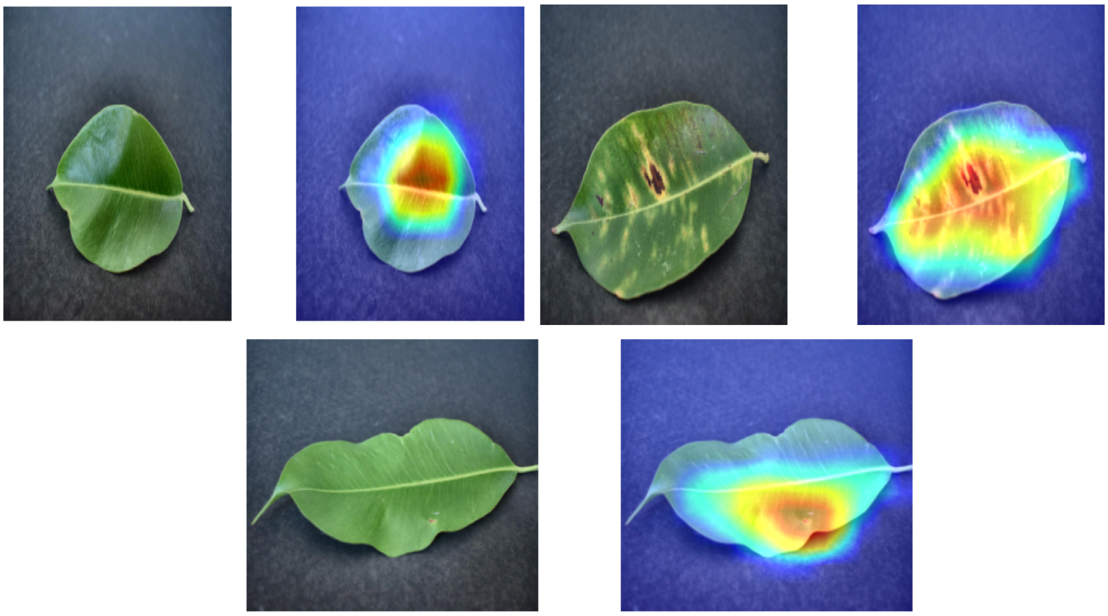


Figure 4.2.8: Grad-CAM of best performing model EfficientNetB5

The Grad-CAM effects in Figure 4.2.8 inform us of the portions of the leaf images that contributed the most to the models predictions. The emerging activation maps are focused on the central vein and related textures, which are significant indicators of the plants health. The very fact that the activation is a localized one in these biologically significant areas means that the model is not acting on some background noise, or irrelevant artifacts, but is probing meaningful patterns that relate to disease. The interpretability step is value addition on top of raw accuracy since it allows the researcher and the agricultural practitioner to understand that the model is making the correct predictions as the mind would prefer. Such transparency may be particularly important in the building of trust when working with AI-based disease detection systems, where misdirected or discriminating attention can lead to misclassification in a practical context.

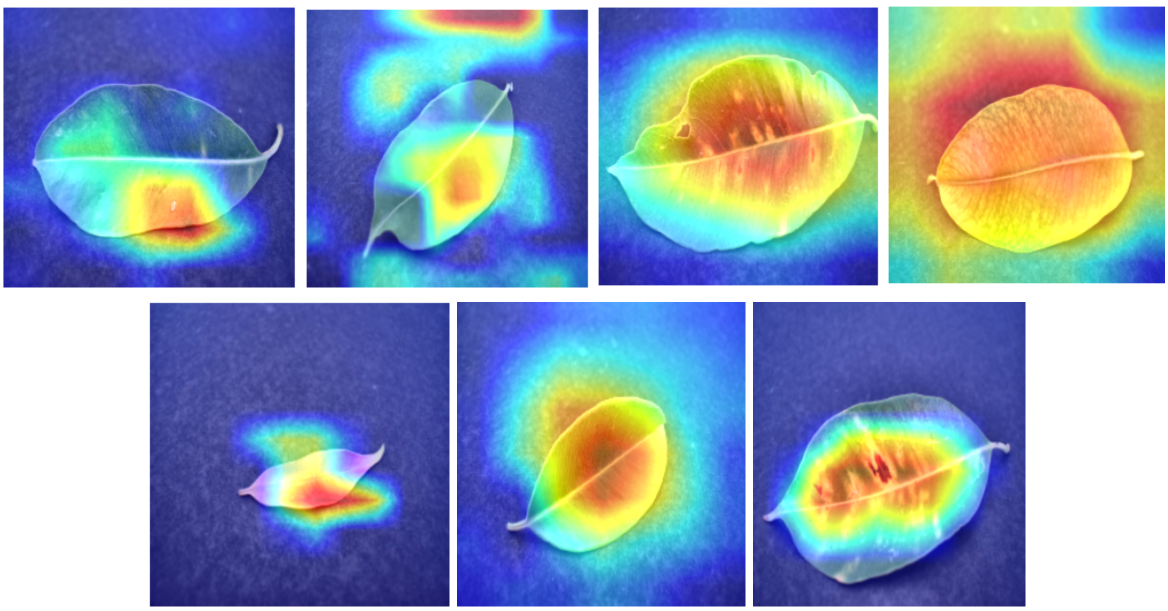


Figure 4.2.9: Grad-CAM visualization for other models

Table 4.2.3: Comparative analysis with other papers

Author	Used Algorithm	Accuracy
Bhowmik et al[1]	Vision Transformer(ViT)	97.5
Mehta et al[2]	FL-CNN	96
Vats et al[3]	FL-CNN	96
Our Research	EfficientNetB5	97.9

4.3 Results and Discussion

The performance of the different models were compared in terms of accuracy, precision, recall, F1-score and AUC-score. The three tables that summarize the results are Table 4.2, Table 4.2.1 and Table 4.2.2. A distinct performance difference was present between the pre-fine-tuning and fine-tuning stages. For example, EfficientNetB5 had an accuracy of 93.6% in the transfer learning phase but with further fine-tuning, the accuracy reached an optimal value of 97.9% with similar improvements in precision, recall, F1-score, and AUC. This shows the effectiveness of fine-tuning in mobilizing the full capability of deeper CNNs. DenseNet121 also showed good stability and reliability, going from 92.6% accuracy before fine-tuning to 95.7% after fine-tuning. A similar trend was seen in ResNet50 which improved from 92.6% to 94.7% and VGG16 from 84.0% to 91.5%. The lightweight EfficientNetB0 also improved quite a lot, from 87.2% before fine-tuning to 96.8% after fine-tuning. When compared to the hybrid CNN+ML approach the fine-tuned CNNs showed consistently better performance. For example, although DenseNet121+SVM and DenseNet121+AdaBoost achieved 95.7% accuracy with an AUC of 99.1%, and EfficientNetB5+Random Forest also achieved 95.7% accuracy, none of them could be better than the fully fine-tuned EfficientNetB5 (97.9%) or DenseNet121 (95.7%). Other combinations of hybrids like VGG16+AdaBoost (92.6%), ResNet50+Random Forest (93.6%) performed well but were below the fine-tuned CNNs.In the lowerend, less powerful pipelines like EfficientNetB0+Decision Tree with 72.3% accuracy, 71.6% and VGG16+Decision Tree with 75.5% accuracy gave visibility of the weaknesses of less powerful classifiers in the hybrid framework. In addition to raw accuracy, it used Grad-CAM to provide interpretability by visual heatmaps that described predictions made by the model. Such a balance of performance and explainability makes the model especially useful in agricultural tasks where trust is critical.

Discussion on Confusion Matrix:

The performance of the fine-tuned deep learning methodology proves the superiority of this method compared to conventional hybrid pipelines. From the 52 samples of diseased class, all were correctly identified and of the 40 samples of class healthy, 40 were correctly identified, with only two misidentification. This can be interpreted to mean very high accuracy, precision, recall and F1-score, which means the model is incredibly successful at identifying both healthy and diseased Java plum leaves.Conversely, the hybrid approach of CNNs to serve as fixed feature extractors with machine learning classifiers, despite its effectiveness, was not always as precise and robust. Though other methods like SVM were as competitive, they nonetheless lagged behind in the capabilities of the fine-tuned model to adjust the feature representations directly to the task-specific dataset. This performance disparity highlights the benefit of fine-tuning, in which pretrained convolutional layers are both re-used but also trained to capture domain-specific fine-scale patterns specific to leaf disease images.

Discussion on Training and Validation accuracy-loss curves:

As Figure 4.2.3 indicates, EfficientNetB5 was efficient and well-fitted. The accuracy remained low during the early epochs but after fine-tuning, it improved significantly and reached approximately 97.9% with losses approaching zero. The similar trends of training and validation curves indicate that the model has not overfitted nor underfitted, and powerful convergence results from the unfreezing of layers and changes in the learning rate.

Similar tendencies were observed with other fine-tuned models like DenseNet121, EfficientNetV2B3 and VGG16, demonstrating high performance and consistent validation scores.

Discussion on ROC curve and AOC values:

The ROC curves in Figure 4.2.5 also confirm the usefulness of the fine-tuned models. The AUC score of both classes is so high that it is almost as close to 1.0 as possible. It means that there is a very good separability between the two classes with the model again showing a very high true positive rate with a very low false positive rate. The distance of the curves to the top-left corner supports the strength of the classifier to reduce the misclassification and reach the maximum predictive reliability.

The fine-tuned transfer learning models significantly outperformed the hybrid pipelines where pretrained CNNs were applied only to extract features and then used by machine learning classifiers, in all evaluation metrics. Although hybrid approaches like CNN+SVM had quite decent performances, they lacked the flexibility and richness of fine-tuned nets, such as EfficientNetB5. The benefit is that fine-tuning has a greater ability to optimize deeper layers to achieve very specific disease-related features, resulting in better classification capacity and close-to-perfect ROC-AUC values.

Discussion on Interpreting with Grad-CAM:

As Figure 4.2.6 demonstrates, EfficientNetB5 Grad-CAM visualizations indicate that the model relies on features related to the disease when making its predictions; using the visualizations as evidence, the central vein and textures are some of the biologically relevant features of leaves. Even other unpopular models, like EfficientNetB3, VGG19 or MobileNetV2 can be used to do good Grad-CAM localization with small errors. In these models, they mostly focus on the symptom areas though occasionally a small background noise is given. The interpretability creates increased confidence in such models, since it shows that these projections are in accord with the knowledge of experts. This explainability is critical to real world agricultural applications, which can give practitioners the benefits of consistent and biology-based decisions and can also use smaller and efficient deep learning models that can be deployed to resource-limited systems.

Overall Observations:

The EfficientNetB5 model is far much better in plant disease detection with a high accuracy of up to 97.9% and AUC of 99.4%. At the transfer learning phase, before the fine-tuning, EfficientNeatB5 had a relatively high score of 93.6% accuracy score and AUC score of 97.5% which is why it requires fine-tuning to realize the full representational ability of the model. The model also significantly became better when the unfreezing layers and learning rate were set and it showed near perfect classification in the confusion matrix- it nearly ensured that all the samples would be classified correctly with a small number of cases of misclassification.

Validation and accuracy-loss curves indicate that the model fits and generalizes excellently since both the accuracy and loss values are optimally converging without traces of overfitting or underfitting. The strength of the model is also shown by the ROC curve, whose AUC of 97.5% suggests that the model better distinguishes between the healthy and diseased classes and reduces the number of false positives and negatives.

Grad-CAM visualizations of EfficientNetB5 show that its predictions depend on biologically significant regions of leaves such as the central veins and the texture, which compares well to expert knowledge of the domain and provides important interpretability and reliability.

Although feature-based algorithms with fixed CNN feature extractors and machine learning classifiers may be computationally efficient and effective in cases of limited resources, they typically do not match fine-tuned models in either accuracy or flexibility. Fine-tuning gives the network the ability to adapt to domain-specific nuances directly on the data, resulting in more expressive feature representations and better classification. This way, EfficientNetB5 using fine-tuning provides an optimal balance of high accuracy, robustness, and explainability and is thus most appropriate when it comes to implementation in real-world agricultural disease detection systems.

4.4 Summary

The detailed analysis and the results of the Java Plum leaf disease detection are demonstrated in chapter 4. This chapter begins with the description of the experimental environment configuration, which makes use of both cloud-based environments and local computing devices with an Intel Core i5-13500 processor, an Nvidia RTX 3060 graphics card, and 16GB of RAM. Python and libraries like Keras, TensorFlow, Scikit-learn, and OpenCV are essential software applications that enable image preprocessing, model training, and evaluation. The chapter provides a comparison between different pretrained deep learning frameworks, and hybrid CNN-plus-machine learning classifier frameworks. Performance was quantified using measures such as accuracy, precision, recall, F1-score, AUC, training and inference time, and also visual interpretability measures such as confusion matrices, ROC curve, and Grad-CAM explanations. The results of transfer learning using frozen pretrained layers showed mixed results where EfficientNetB0 and VGG16 showed the worst performance with EfficientNetB5 taking the first position with 93.6% accuracy. Fine-tuning improved the performance of all models and peaked with EfficientNetB5 with a 97.9% accuracy. VGG16 and DenseNet121 as well were meaningfully improved upon fine-tuning, which highlights the strength of domain-specific adaptation. The models with traditional ML classifier and CNN feature extraction produced high-performing, but generally lower-accuracy performances than fully fine-tuned CNNs validating the strength but relative limitation of hybrid solutions over end-to-end deep learning fine-tuning. Though some feature extraction models did get close to 95% accuracy which could be considered as good alternatives to some lower performing transfer learning models. Both training and validation curves demonstrated good generalized models without overfitting. Grad-CAM visualization revealed information about significant parts of the image that affect predictions, which is helpful in promoting model transparency and reliability to farmers. Finally, the findings support the importance of fine-tuning to unlock pretrained models potential and EfficientNetB5 became the most effective architecture. This chapter also finds that integrating effective transfer learning models with interpretability methodologies can provide effective, and understandable plant disease detection systems which can be applied to agriculture in real-life scenarios.

Chapter 5

Engineering Standards and Design Challenges

This chapter introduces compliance with engineering standards, societal and environmental effects, ethical issues, sustainability plans, project management and financial analysis, and mapping of complex engineering problems, knowledge profiles, as well as engineering activities. It emphasizes how the proposed Java plant leaf disease detection system is technically, ethically, and sustainably compliant as it solves farm problems.

5.1 Compliance with the Standards

To develop the disease detection system of java plum leaf, adherence to associated engineering requirements was needed to guarantee scalability, dependability, and effectiveness of the system. In this section we discuss the hardware, software and communication standard used in the project and other alternative methods and why the choices that were made.

5.1.1 Software Standards

TensorFlow and Keras were selected as the preferred frameworks to build the disease detection system of java plum leaf due to their scalability, robustness, and widespread use in Agricultural AI. The frameworks are well suited to large datasets, provide pre-trained models, and are integrated with other critical libraries, such as OpenCV and NumPy. Although they have a large computational cost, and the learning curve is steep, the community support and documentation are good reasons to suggest TensorFlow to the Agricultural applications.

Alternative: PyTorch

PyTorch was an option due to its flexibility and dynamic computation graph. Although it is much faster than other frameworks at prototyping and research, it does not have the fully-fledged deployment capabilities of TensorFlow nor as many ready-made models of agricultural AI applications.

5.1.2 Hardware Standards

Performing hardware selection was essential to well-organised training and real-time predictions. The selected hardware greatly influenced the model performance, deployment speed, and training rate, especially with large agricultural image datasets.

Name of selected standard: Nvidia RTX 3060.
This work was performed using Nvidia RTX 3060 12gb gpu to construct architecture. It is employed in the preprocessing and as well as in the training of the deep learning model.

5.1.3 Communication Standards

The Java Plum leaf disease detection architecture is designed to adhere to the current communication standards in order to be easily deployable and interoperable. Common formats are used to process image data: JPEG/PNG, whereas experimental data is saved in CSV and is compatible across all platforms. It builds upon the TensorFlow and Keras APIs, which adhere to well-documented communication protocols to support training, evaluation, and visualization, making them reproducible and interoperable with other workflows. Performance metrics and Grad-CAM heatmaps are examples of outputs that are in universally readable formats and are thus available to researchers and agricultural practitioners. The framework can incorporate web-based standards such as RESTful API via HTTP/ HTTPS in future to support real time communication between mobile or edge devices in order to provide field level monitoring of disease.

5.2 Impact on Society, Environment and Sustainability

Java Plum leaf disease detention system may produce huge societal and environmental impacts. Assisting farmers in trapping and treating diseases in their leaves promptly and responsively will make sure that they will lose less crop and yield better. It helps to directly support rural livelihoods, increase the food security of the population and contributes to the economic stability of farming populations by ensuring only the necessary application of pesticides is carried out. This will reduce the volume of the chemicals that will be flowing off, provides superior soil and water systems and provides a healthier ecosystem. In the meantime, it makes the cost of production to the farmers lower and the disease is more sustainable to manage. As to the long-term sustainability, the introduction of AI-based solutions into the agricultural sphere will promote the process of resources management into a more intelligent state and will assist in the shift towards the concept of precision farming. The framework seeks to build trust and usability, by making sure that technological innovations are environmental and farmer-friendly through the use of explainability techniques including the use of Grad-CAM in powerful detectors.

5.2.1 Impact on Life

Java plum leaf disease detection system is an important part of the life of the human being because it can provide a farmer with the means that can be used in the process of the disease detection at the initial stages. This lessens crop failures, secures a given income and increases output directly to the living standards in agricultural societies. This also results in the production of healthier food and a safer environment to both the farmers and their consumers since it reduces the application of harmful chemicals by using direct treatment in treating them. In the long run, this system leads to agricultural sustainability, which results in food security and elevated the overall well being.

5.2.2 Impact on Society & Environment

Java plum leaf disease detection has severe long-term impacts on the society, environment and sustainability. The system helps the farmers in the timely and accurate detection of the disease, and avoids the use of expert help and the reduction of losses of crops. It directly affects farmers positively on their agricultural production, livelihood and food security. The system in environmental terms enables precision agriculture, whereby plants affected by diseases are treated rather than excessive pesticides applied and covering large fields. This protects the quality of the soil and water, chemical and biodiversity use are maintained. The positive relationship exists between the implementation of ML and DL in the agricultural sector and the effective utilization of the resources as well as sustainability of the agricultural operations.

5.2.3 Ethical Aspects

This system was developed keeping in mind some ethical considerations to make the research and application responsible. Only publicly available datasets were involved without any ownership rights or breach of data privacy. This system is built to benefit farmers and communities with equitable access to technology without discrimination or bias. In addition, explainable AI techniques, including Grad-CAM can be used to bring transparency to the model decision-making, allowing the user to interpret and trust the predictions. IT helps in sustainable farming and minimizes unnecessary use of chemicals. This study also practices ethical responsibility to protect the environment.

5.2.4 Sustainability Plan

This study concentrates on technical and practical methods in order that consequently, it will be sustainable in the long run. The models are designed to be scaled technically, and can be combined with lightweight architectures, lightweight optimization strategies such as pruning and quantization to run on mobile- or edge-based devices. This renders it accessible to the farmers in resource limited settings. Practically, it is possible to update the system continuously by adding new datasets, retraining on new samples of leaves and response to new plant diseases. Such flexibility will render the structure suitable to agricultural uses over a long period. In addition, the approach supports environmental sustainability because it promotes accurate farming. It not only saves the needless pesticides by simplifying the process of diagnosis of diseases in earlier stages and by making this accurate, but also saves natural resources and assists in building up healthier ecosystems. Finally, AI methods like Grad-CAM are also explainable, thus visible and most dependable, which is essential to a sustained adoption by farmers and other agricultural families. Such a mixture of technical scalability, environmental responsibility, and user trust is what is behind the sustainable impact of the project.

5.3 Project Management and Financial Analysis

Since our project was experimental we used free resources like Kaggle so our costs were minimal. This segment gives the estimated cost of developing and deploying the java plum leaf disease detection system that is developed using a deep learning AI model. This includes the cost of software, and the extent to which we utilised the hardware, R&D deployment, and so on.

Initial estimated budget

The project budget is:

- **Hardware resources**: 0 BDT (we used a personal Desktop to experiment for this project)
- **Software resources**: Tools and Licensing: 5,000 BDT (use of cloud services and optional premium tools; primarily open-source platforms such as Kaggle used)
- **R&D:** 20,000 BDT (including model training, exploratory analysis and personal effort)
- **Deployment cost**: 10,000 BDT (hosting web, domain, SSL,, and basic security)
- **Operating cost:** 5000 BDT (Internet, electricity, minor cloud usage)

Estimated total budget: 40,000 BDT

Alternative Budget

- **Hardware**: 0 BDT (we used a personal Desktop)
- **Software**: 0 BDT (100% depend on open-source tools and free cloud tiers)
- **R&D and Testing:** 15,000 BDT (in hours of optimized work)
- **Operational cost**: 5,000 BDT (no cloud, this is the least case)
- **Deployment cost:** 5000 BDT (basic hosting with free-tier SSL)

Overall Minimal Budget: 25,000 BDT

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

This section represents a mapping of the complex problem-solving categories to the specific engineering problems. Each category is mapped and analysed accordingly, with a rationale provided for every selection. The mapping is presented in Table 5.1, and further explanation is offered for the rationale behind the mappings for the different Engineering Problems (EP1 to EP7).

EP1: Depth of Knowledge: It shows that knowledge of plant pathology and computer vision was highly integrated in the study. Taking the knowledge of leaf disease in Java Plum with the help of the high-performing deep learning feature like EfficientNet and DenseNet, the study points out how the ability to interdiscipline knowledge can enhance the detection of agricultural diseases.

EP3: Depth of Analysis: In this study, the rigorous evaluation pipeline is applied, with accuracy, F1-score, confusion matrix, ROC-AUC, and the interpretability tool such as Grad-CAM. Fine-tuned transfer learning models and feature extraction by the use of ML classifiers are a dual method that is used to guarantee a comprehensive comparative analysis.

EP4: Familiarity of Issues: Some of the well-known problems in agricultural AI, which are directly tackled in this study, include small data sets, overfitting, and model transparency. Through addressing these concerns with transfer learning and visual interpretability, it adds to known problems but provides workable solutions to them.

EP7: Interdependence: The article emphasizes how data quality, pretrained CNNs, machine learning classifiers and interpretability tools are mutually dependant. These factors together make sure that performance gains are not obtained to the expense of transparency or generalization.

Table 5.1: Mapping with Complex Engineering Problem.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✔		✔	✔			✔

Mapping with Knowledge Profile

We provide mapping with Knowledge Profile in this section. Subsection to put rationale for each mapping in Table 5.2

K1: Natural Science: The paper works with plant leafs so natural science knowledge is needed

K2: Mathematics: Our work is reliant on machine learning and deep learning algorithms so understanding of mathematical knowledge is needed.

K3: Engineering Fundamentals: The paper uses fundamental concepts of machine learning and optimization, such as transfer learning, feature extraction, fine-tuning, regularization, and learning rate scheduling, as the basis of creating the correct and generalizable disease detection models.

K4: Specialist Knowledge: The expertise in this field is shown using sophisticated architectures like EfficientNetB5, DenseNet121, and hybrid CNN+ML pipelines. The addition of Grad-CAM interpretability is yet another example of domain specific knowledge in explainable AI in agriculture.

K5: Engineering Design: The study will entail the development of the robust training plan, epochs before and after the fine-tuning, the use of the augmentation techniques and the development of the hybrid pipelines combining deep and machine learning classifiers to achieve higher accuracy and adaptability.

K6: Engineering Practice: The practical work can be seen in data preprocessing, augmentation, web deployment to make it usable, and evaluation of its performance with consideration of real-world agricultural problems, such that the solution is feasible not just in the context of academic experiments.

K7: Comprehension: The analysis demonstrates that the agricultural issue is comprehended by attributing technical means to the practical requirements including food security, trust of farmers, and sustainability. It also solves standard limitations such as small data sets and overfitting models.

K8: Research Literature: Large amounts of literature review, including CNN-based designs, ViTs, YOLO designs, federated learning strategies, and hybrid designs, are conducted, placing the study in the context of the larger discipline, and revealing gaps in Java Plum leaf disease detection.

Table 5.2: Mapping with knowledge Profile.

K1 Natural Science	K2 Mathematics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K7 Comprehension	K8 Research Literature
✔	✔	✔	✔	✔	✔	✔	✔

5.4.2 Engineering Activities

We provide mapping with engineering activities in this section. Subsection to put rationale for each mapping in Table 5.3.

Mapping with Complex Engineering Activities

EA1: Range of Resources: The study utilises state-of-the-art computing infrastructure, powerful GPUs, deep learn-ing frameworks (TensorFlow, Keras), and high-quality datasets. It integrates ethical guide-lines and leverages transfer learning and CNNs to achieve accurate java leaf disease detection, ensuring both technical precision and ethical integrity.

EA3: Innovation: The study innovates by using hybrid model architectures for java leaf disease detection. The methodology improves accuracy, efficiency, and scalability, contributing to the advancement of deep learning in agriculture and offering a reliable and effective solution.

EA4: Consequences for Society and Environment: This study enhances early disease detection of the java plum leaf, improving the agricultural sector. It also minimises environmental impact by optimising computational resources, while upholding data privacy, fairness, and transparency to maintain public trust.

Table 5.3: Mapping with Complex Engineering Activities.

EA1 Range of re-sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✔		✔	✔	

5.5 Summary

The Java plum leaf disease detection system was developed with software, hardware, and communication standards to ensure reliability, scalability, and sustainability. TensorFlow and Keras were chosen for model training; on the contrary, the Nvidia RTX 3060 GPU supported efficient processing. The system impacts farmers by delivering early disease detection, reducing significant crop loss, lessening pesticide use, and promoting sustainable agriculture. In terms of ethical considerations, including dataset transparency and explainable AI methods, were integrated to ensure fairness and trust. Financial analysis estimated project costs between 37,000–55,000 BDT, balancing open-source tools with deployment needs. The project addresses complex engineering problems through innovative deep learning models, multidisciplinary knowledge and ethical, environmentally responsible practices, ultimately enhancing agricultural sustainability, productivity, and societal well-being.

Chapter 6

Conclusion

This conclusion chapter emphasizes the successful design and testing of a deep learning architecture to detect Java Plum leaf disease. It identifies the limitations of the study which include dataset size and real-world variability and suggests future directions including scaling up the datasets, improving training practices, multimodal data integration, making it easier to understand, and implementing practical web-based solutions to implement them to agricultural practice.

6.1 Summary

This study aimed to develop an advanced deep learning-based framework for the detection and classification of Java Plum leaf diseases using image data. The project systematically evaluated multiple pretrained convolutional neural networks (CNNs), including ResNet50, VGG16, DenseNet121, MobileNetV2, EfficientNetB0, EfficientNetV2B3, and EfficientNetB5, across two primary modeling strategies: transfer learning with fine-tuning and hybrid feature extraction combined with machine learning classifiers such as Random Forest, SVM. The major contributions of this study include the implementation of a robust comparative analysis framework for both transfer learning and hybrid models, the application of comprehensive preprocessing and data augmentation pipelines to improve model generalization, and the integration of Grad-CAM visualizations to enhance interpretability. The experimental results showed that both modeling strategies had high performance, and the transfer learning model with EfficientNetB5 has the highest overall performance of 97.9%. Competitive performance was also observed with the hybrid feature extraction models in which the combination of deep feature representations with classical machine learning classifiers is effective. The paper also revealed in-depth information on overfitting trends, learning curves, confusion matrices, and Grad-CAM visualization, which allowed conducting a comprehensive analysis of the model performance. The overall analysis confirms the efficiency of pretrained CNNs with fine-tuning and hybrid models, and the significance of using deep feature extraction in conjunction with classical classifiers to implement strong, accurate and explainable plant disease detection. The results provide a concrete basis to implement in practice in the field of agriculture and help implement the Java Plum leaf diseases management in time and as effectively as possible.

6.2 Limitation

Even though there was a high performance in classification, the study has some limitations. First, it was a rather small and uneven dataset, which was available in Kaggle. Although data augmentation was used to enhance variability, they might have a limited ability to generalize to real-life scenarios such as changing leaf orientations, light, severity of disease, and environmental conditions. Strength and dependability would be improved by diversification and raising the number of representative samples.
Second, the paper utilized the standard training set ups and optimizers primarily without trying other optimization strategies. This lack of comparative analysis may prevent potential

improvement in the speed of convergence, stability.
Third, the research was restricted to image based-classification, and not to the other contextual or multi-modal characteristics like conditions of the environment, developments of disease pathology, or health elements in plants that could further enhance detectability and viability in the field.
Fourth, although Grad-CAM has been used to justify the decision by visualizing it, these justifications are not in detail and it may not be the full story of the decision making process in the model. Further techniques of explainability can introduce further information, and this would render the deployment more acceptable within the farming advisory systems.
Finally, certain pretrained models such as EfficientNet variants are computationally expensive, i.e. require a lot of hardware. This may limit their application in resource limited or real world field applications where efficiency must be sacrificed to accuracy. The setbacks will form the core of the framework development to realistic and scaled solutions to the management of Java Plum leaf disease.

6.3 Future Work

Based on the positive findings of this research, a number of future research directions can be proposed. First, a larger and more varied group of Java Plum leaf images representing a variety of disease classes, leaf orientation, lighting conditions and environmental conditions that would increase model generalization ability and robustness in real world agricultural environments.
Future studies might also address systematic hyperparameter optimization and other training methods to further enhance the performance of models.
Multimodal data integration is another potential path. There is the possibility of a more integrated view of the dynamics of the disease with the integration of leaf pictures and additional contextual data such as soil condition, weather conditions, maturity of the plants or the development of the disease over time, which could help to detect the disease better in a realistic agricultural setting.
The issue of better explainability is of significance. To understand the model decision-making process more deeply and make it more palatable to end-users, such as farmers and other specialists in the field, complementary interpretability methods like SHAP or more complex Grad-CAM visualizations to the current Grad-CAM can further yield better visualization as well.
Lastly, it might be possible to create a more advanced web-based implementation of the model that would enable farmers and other agricultural experts to remotely access the disease detection system by merely uploading leaf images. This would make the framework more accessible and usable allowing it to be implemented more widely to monitor real-world Java Plum leaf disease.

References

[1] Auvick Chandra Bhowmik, Md. Taimur Ahad, Yousuf Rayhan Emon, F. Ahmed, B. Song, and Y. Li, “A customised vision transformer for accurate detection and classification of Java Plum leaf disease,” Smart Agricultural Technology, vol. 8, pp. 100500–100500, Aug. 2024, doi: https://doi.org/10.1016/j.atech.2024.100500.

[2] S. Mehta, V. Kukreja, and S. Vats, “Innovative Approaches to Java Plum Leaf Disease Identification: Federated Learning meets Convolutional Neural Networks,” 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–6, Jul. 2023, doi: https://doi.org/10.1109/icccnt56998.2023.10307120.

[3] Satvik Vats, S. Mehta, V. Kukreja, and Manika Manwal, “Federated Learning CNN’s in Agriculture: Unveiling Disease Severity Analysis of Java Plum Leaf’s,” 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, pp. 1–6, Aug. 2023, doi: https://doi.org/10.1109/iccubea58933.2023.10392276.

[4] Y. Aravind Reddy and Adimoolam M, “A Framework System for Plant Leaf Disease Detection using K-Nearest Neighbours and comparison of its features with Naive Bayes Classification,” 2022 International Conference on Business Analytics for Technology and Security (ICBATS), Feb. 2022, doi: https://doi.org/10.1109/icbats54253.2022.9758924.

[5] A. S. M. Farhan Al Haque, R. Hafiz, Md. A. Hakim, and G. M. Rasiqul Islam, “A Computer Vision System for Guava Disease Detection and Recommend Curative Solution Using Deep Learning Approach,” 2019 22nd International Conference on Computer and Information Technology (ICCIT), pp. 1–6, Dec. 2019, doi: https://doi.org/10.1109/iccit48885.2019.9038598.

[6] S. P. S, S. D. R, B. S. C, and S. K. Kannaiah, “Automatic Plant Leaf Disease Detection using AlexNet and Image Processing Techniques,” 2024 International Conference on Cybernation and Computation (CYBERCOM), pp. 441–446, Nov. 2024, doi: https://doi.org/10.1109/cybercom63683.2024.10803225.

[7] C. Yao et al., “Two-Stage Detection Algorithm for Plum Leaf Disease and Severity Assessment Based on Deep Learning,” Agronomy, vol. 14, no. 7, pp. 1589–1589, Jul. 2024, doi: https://doi.org/10.3390/agronomy14071589.

[8] T. Righi, M. C. E. Meftah, A. Laouid, M. Al-Khalidi, and M. Kara, “An intelligent agriculture monitoring framework for leaf disease detection using YOLOv7,” STUDIES IN ENGINEERING AND EXACT SCIENCES, vol. 5, no. 2, p. e10498, Nov. 2024, doi: https://doi.org/10.54021/seesv5n2-516.

[9] G. Wen and M. Li, “An Improved YOLOv8 Algorithm for Leaf Disease Detection,” 2023 International Conference on Artificial Intelligence and Automation Control (AIAC), pp. 21–24, Nov. 2023, doi: https://doi.org/10.1109/aiac61660.2023.00013.

[10] H.-T. Thai and K.-H. Le, “MobileH-Transformer: Enabling real-time leaf disease detection using hybrid deep learning approach for smart agriculture,” Crop Protection, vol. 189, p. 107002, Mar.

2025, doi: https://doi.org/10.1016/j.cropro.2024.107002.

[11] Sherihan Aboelenin, Foriaa Ahmed Elbasheer, M. M. Eltoukhy, W. M. El-Hady, and K. M. Hosny, “A hybrid Framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer,” Complex & Intelligent Systems, vol. 11, no. 2, Jan. 2025, doi: https://doi.org/10.1007/s40747-024-01764-x.

[12] P. V et al., “A Channel Attention-Driven Hybrid CNN Framework for Paddy Leaf Disease Detection,” arXiv.org, 2024. https://arxiv.org/abs/2407.11753 (accessed May 23, 2025).

[13] S. Saleem, A. Hussain, N. Majeed, Z. Akhtar, and K. Siddique, “A Multi-Scale Feature Extraction and Fusion Deep Learning Method for Classification of Wheat Diseases,” Journal of Computer Science, vol. 21, no. 1, pp. 34–42, Jan. 2025, doi: https://doi.org/10.3844/jcssp.2025.34.42.

[14] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, “Deep feature based rice leaf disease identification using support vector machine,” Computers and Electronics in Agriculture, vol. 175, p. 105527, Aug. 2020, doi: https://doi.org/10.1016/j.compag.2020.105527.

[15] D. Banerjee, V. Kukreja, Dibyahash Bordoloi, and K. Joshi, “Hybrid CNN & Random Forest Model for Effective Turmeric Leaf Disease Diagnosis,” Dec. 2023, doi: https://doi.org/10.1109/smartgencon60755.2023.10442441.

[16] Hicham El Akhal, A. Yahya, N. Moussa, and A. El, “A novel approach for image-based olive leaf diseases classification using a deep hybrid model,” Ecological Informatics, vol. 77, pp. 102276–102276, Nov. 2023, doi: https://doi.org/10.1016/j.ecoinf.2023.102276.

[17] S. Sundhar, R. Sharma, P. Maheshwari, S. R. Kumar, and K. T. Sunil, “Enhancing Leaf Disease Classification Using GAT-GCN Hybrid Model,” arXiv.org, 2025. https://arxiv.org/abs/2504.04764

[18] A. A. Nasser and M. A. Akhloufi, “A Hybrid Deep Learning Architecture for Apple Foliar Disease Detection,” Computers, vol. 13, no. 5, pp. 116–116, May 2024, doi: https://doi.org/10.3390/computers13050116.

[19] T. Pavan, Chipinapi Keerthi Sadha, Pokuru Harshini, Visalakshi Annepu, Kalapraveen Bagadi, and V. Rami, “Plant Leaf Disease Classification using Transfer Learning using EfficientNetb5,” Dec. 2023, doi: https://doi.org/10.1109/nelex59773.2023.10421367.

[20] R. Kaur, M. Uppal, D. Gupta, Kazim Raza Talpur, A. Shah, and S. Saini, “Classification of Healthy and Diseased Plant Leaves Using Deep Learning Model,” 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), pp. 1–6, Nov. 2024, doi: https://doi.org/10.1109/icetas62372.2024.11119833

[21] AM, “Healthy vs. Diseased Leaf Image Dataset,” Kaggle.com, 2022. https://www.kaggle.com/datasets/amandam1/healthy-vs-diseased-leaf-image-dataset

213-15-4479

ORIGINALITY REPORT

17%	12%	12%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	arxiv.org Internet Source	1%
3	www.mdpi.com Internet Source	1%
4	Auvick Chandra Bhowmik, Md. Taimur Ahad, Yousuf Rayhan Emon, Faruk Ahmed, Bo Song, Yan Li. "A customised vision transformer for accurate detection and classification of Java Plum leaf disease", Smart Agricultural Technology, 2024 Publication	1%
5	Poonam Nandal, Mamta Dahiya, Meeta Singh, Arvind Dagur, Brijesh Kumar. "Progressive Computational Intelligence, Information Technology and Networking", CRC Press, 2025 Publication	<1%
6	ouci.dntb.gov.ua Internet Source	<1%
7	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
8	Submitted to Sardar Patel Institute of Technology Student Paper	<1%
9	journal.nsps.org.ng Internet Source	