A Light-weight CNN with integration of Explainable AI to detect Guava Leaf Disease

By
Kazi Sharfunnabi
212-15-4177

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

Dr. Md. Taimur Ahad
Associate Professor
Department of Computer Science and Engineering Daffodil International University

Co-Supervised by

Ms Faiza Feroz
Lecturer
Department of Computer Science and Engineering Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

May 14, 2024

APPROVAL

This Project titled **“A Light-weight CNN with integration of Explainable AI to detect Guava Leaf Disease”**, submitted by Kazi Sharfunnabi, ID No: **212-15-4177** 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 **14 May, 2025**.

BOARD OF EXAMINERS

Dr. S.M Aminul Haque **Chairman**
Professor and Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Sharmin Akter **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Ms. Syada Tasmia Alvi **Internal Examiner**
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

14.05.25

Dr. Md. Arshad Ali **External Examiner**
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science & Technology University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Md. Taimur Ahad**, **Associate Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Md Taimur Ahad

Dr. Md. Taimur Ahad
Associate Professor
Department of Computer Science and
EngineeringDaffodil International
University

Co-Supervised by:

Faiza Feroz 13.5.25

Ms Faiza Feroz
Lecturer
Department of Computer Science and
EngineeringDaffodil International
University

Submitted by:

Kazi Sharfunnabi 13.5.25

Kazi Sharfunnabi
Student ID: 212-15-4177
Department of Computer Science and
EngineeringDaffodil International
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 **Dr. Md. Taimur Ahad**, **Associate Professor**, 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

This thesis is focused on the implementation and evaluation of a deep learning system that was specifically trained to be used for detection of guava leaf diseases. The designed system is based on a lightweight CNN model, i.e., M-Net; it combines XAI techniques to enhance interpretability. The goal was to create a model that guarantees reliable and efficient performance and easy interpretability for easy identification of diseases and timely action by farmers. A dataset comprising the images of guava leaves was used for the research that comprised two locations located in Ashulia and Savar, Dhaka, each image being identified on the basis of one of the seven precise disease classes. To begin with various pretrained Convolutional Neural Network (CNN) models, such as VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2 were tested for the task of guava leaf diseases detection. While MobileNetV2 had showed remarkable performance, M-Net performed on high drag, with 99.04% accuracy – a value that makes it more appropriate for rapid disease identification on mobiles in resource-limited settings. To enhance the transparency of the system, SHAP and Grad-CAM techniques were implemented and gave farmers a clear idea of how the model decides on recommendations leading to trust. For the system, an optimized web application for mobile devices was created allowing farmers to upload leaf images within a short time and receive the results of disease classification instantly. This study shows that using deep learning with XAI has the great promise to improve decision-making in agriculture particularly where expert help is lacking. The project indicates a positive innovation on the use of AI technology in managing crops diseases in the field.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter discusses the Guava Leaf Disease Detection System and why deep learning is used for plant disease classification. It identifies key objectives, strategy, and relevance of the study which introduces the following parts of the thesis.

1.1 Introduction

The cultivated tropical fruit, guava (Psidium guajava), is touted for its excellent nutrition that includes high vaiant of vitamin C, dietary fibre and iradaptants. Guava is of important to the agricultural economies of many tropical and subtropical countries including India, China, Brazil and Mexico [6,8]. Although guava is commonly eaten raw, it is also used prepared into juices, jams, sauces and dried snacks. Its various uses enhance its economic value greatly earning farmers a lot of money as well as supporting local and international markets [4,8]. Although beneficial, the guava cultivation is becoming increasingly risky with exposures to plant diseases, especially those affecting the leaves to affect photosynthesis and reduce the yield of fruits [6]. Lack of identification or control of diseases, which impact guava leaves, such as anthracnose, rust, and bacterial blight results in significant yield losses [6,30]. Early detection of these diseases is important if they are slowed down from spreading and this is key to the viability of guava agriculture and the environment sustainability.

Beyond local markets, the economic value of guava extends beyond the imaginations of many because it is a source of livelihood for millions of smallholder farmers in developing nations. Guava's cultivation has emerged as a very important source of income for rural communities across many aspects, ranging from agricultural up to the processing and distribution. Guava is one of the most important export products in Asia, Africa and Latin America and is an essential component of international agricultural trade [8]. Increasing global interest in guava has underscored the need for commitment to high-quality crops and large yields. However, emerging disease is still a critical problem where the leaf diseases pose a significant threat to guava cultivation [6,30]. Location of these diseases at their early stages and timely treatment of them is required for the profitability of guava farming to be maintained and farmers to take advantage of the high value of this crop in the market [6,8].

The incorporation of Convolutional Neural Networks (CNNs) in deep learning has considerably improved plant disease detection process based on the automated image analysis and facilitating high accuracy. The efficiency of CNNs for plant disease classification has improved significantly since they are capable of pulling hierarchical features directly from raw images without any explicit feature engineering [6,14]. The lightweight CNN models, such as M-Net, have attracted great attention due to their superior efficiency and low computational requirement that makes them very well suited for the resource-limited setting prevalent in rural agriculture [6,7]. M-Net's reduction in design makes it a practical choice, as it performs reliably with little processing power, suitable for rural agricultural mobile and power-limited devices [6]. Its real-time operation makes this efficient model enable farmers to quickly spot diseases and respond accordingly to minimize crop losses and thereby improve yield and productivity in farms [6,7].

With a proven track record of success in many image classification tasks, especially in plant disease detection, well-established pretrained convolutional neural networks such as VGG19, Densenet201, InceptionV3, ResNet152V2, and MobileNetV2 have been shown to be the way to go. As these models have been trained on broad data-sets, they are able to accurately process unknown information and effectively identify leaf diseases [4,30]. In contrast to the greater resource requirements, the pretrained models retain the significant advantage to be adaptively trained on specific plant disease datasets, thereby improving the results in complex classification tasks [6]. Applying pre-trained models with lightweight models such as M-Net provide an extremely efficient solution on par with accuracy and computational needs for the detection of guava leaf diseases [4]. This synergistic approach allows the implementation of state-of-the-art models in real-life use cases, with the optimal system efficiency, even in cases of restricted resources [7]. By utilizing hybrid approach, the system is able to tackle varied disease detection problems and stay viable in limited-resource Agricultural systems [4,6].

Employing XAI methods is crucial to deep learning model transparency and under- standing. For plant disease detection applications, both XAI techniques such as GradCAM and SHAP help significantly understand model decision processes, thus establishing trust and confidence in the system [1]. For instance, GradCAM generates heatmaps that highlight the main aspects within a leaf image on which the model's classification decision will rely, providing the user with crucial visual hints as to why specific diseases are recognized. The importance of actionable information for making sound choices in agriculture for disease management renders such enhanced interpretability imperative [14]. By integrating XAI with CNN models, this work improves the ability of plant disease detection systems, as farmers can also count on precision as well insight into predictions, ultimately contributing to wise decision making and the accelerated introduction of AI tools into agricultural practice [1,14]. Explaining AI through its penetrance in disease detection models increases usability where farmers are empowered by timely and well understood AI-recommended actions.

The integration of AI in the detection of diseases has been made easy by the extensive use of mobile and web applications. Farmers can use these apps to click on the images of sick leaves and send it for immediate analysis by advanced machine learning algorithms thereby providing instant disease diagnosis [8,42]. The proliferation of the mobile device in rural areas has empowered access to state of the art disease detection tools that were accessible only to large-scale agricultural enterprises [42]. Combining mobile applications with CNN-based disease detection systems give farmers a hands-on tool for on-site identification of plant diseases, which is not a requirement for costly professional equipment [8]. It significantly minimizes the adoption barrier and equips farmers to act promptly with the help of mobile access and latest information to disease detection tools. Besides, the presence of such apps allows farmers to follow expert advice thus minimizing a possibility of outbreaks of diseases and minimizing the chances of large-scale damage to crops. By virtue of having AI-enabled mobile platforms, this research advances the usage of the most advanced plant diseases detection tools in farming making those resources more applicable for the rural farmers [8,42].

This thesis introduces M-Net, a simplified CNN model matched with Explainable AI approaches like GradCAM and SHAP to find guava leaf diseases. By combining a lightweight CNN with interpretability of XAI, the system can be both fast and interpretable. With the help of this dual way, the system can easily identify the disease as well as provide transparent explanations of the decisions made by the model. By building these tools into an easily usable web application, the system becomes more convenient for the farmers to use. The platform allows farmers to upload photographs of leaf afflictions, has real-time disease identification, and has an interpretable analysis of the CNN's decision-logic. This project seeks to bring to the farmers' disposal an up to date solution to the detection and management of guava leaf diseases prior to their spreading, thus helping the farmers take informed decisions to save their crop and increase efficiency in agriculture.

1.2 Motivation

Across the board, the essence of the study is to come up with some means of addressing the monumental challenges affecting the control of the diseases of plants facing farmers (particularly on guava which is highly susceptible to various leaf diseases compromising on productivity and quality). The traditional means of diagnosing plant diseases through visual inspection and expert consultation are often tedious and erroneous, particularly when expensive or inaccessible experts are not present. These shortcomings bring an urgency for a more effective, reliable, and scalable means of identifying plant diseases within agriculture.

New possibilities for the transformation of the disease detection process arise with the advancement in artificial intelligence and deep learning (such as Convolutional Neural Networks). Because of massive computation needs, most of the existing CNN models cannot be applied to real-time use in low-resource applications such as agricultural fields. The purpose of this study is to overcome this challenge by developing a lightweight CNN model (M-Net) with a very high accuracy and little computational requirement, which is very favorable for mobile devices or low-power implementations. The introduction of XAI techniques such as SHAP and Grad-CAM helps to expose this reasoning to the users. By creating transparency on the decision-making process, this approach gives farmers confidence to trust the system and encourages their crop management by being more informed about decisions regarding diseases. By embedding advanced technologies in a mobile/web system, this dissertation aims to provide farmers with a practical solution for crop protection, real-time, and scalable, while promoting sustainable agriculture for increased efficiency.

1.3 Objectives

This research aims to develop an effective CNN-based XAI method for recognizing guava leaf diseases, by using a lightweight CNN. The specific objectives are:

- Develop a lightweight CNN (M-Net) to be specialized for guava leaf disease detection, focusing on a compact, high-performance model that can be run under limited computational processes while being real-time optimized to low resource environments.
- Compare M-Net's results with those pre-trained CNN models that are popular (VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2) to measure its performance in the detection of guava leaf disease and choose the model that outshines most to apply transfer learning.
- M-Net with XAI features such as Grad-CAM and SHAP to enhance the interpretability and transparency, allowing users to understand the model's reasoning, and gain more faith in its predictions outcomes.
- Design an intuitive Mobile-Based-Platform for farmers to submit pictures of guava leaves for real-time disease detection and clear explanations with the promise that this resource can be used by less technologically developed farmers.
- Examine the capabilities of the system as it pertains to accuracy, processing time and interpretability, just to ensure that it is fit for practical purposes of

disease detection, provision of mobile solutions and support for user understanding.

- Develop a practical resource for farmers on early detection of guava leaf diseases, thus improving agricultural management, reducing massive yield loss to farmers, and helping them adopt better farming practices.

With these targets, this work is aimed at developing a user-friendly system for diagnosing guava-leaf problems towards an advocacy of sustainable agriculture.

1.4 Methodology

The present study uses a systematic approach to develop, develop and evaluate a deep learning-based system for identifying guava leaf disease. This process involves a sequence of steps, from data acquisition to constructing the model, designing and deploying a simple web application for users. With the aid of XAI techniques, the transparency and interpretability of the model are enhanced, simplifying its use for the farmers. The following represents a summary of the methodology we used:

- Data Collection and Preprocessing: The dataset was collected from two guava yards located in Ashulia, Savar, Dhaka. All images were brought to a uniform size to ensure that there is uniformity in the dataset. To increase the diversity of the dataset and facilitate the generalization capability of the model, augmentation techniques like flipping, rotating, and shearing were used. In this way, we avoided overfitting, and our results were uniform in the training and evaluation datasets.
- Model Design and Architecture Development: For addressing guava leaf disease detection, we propose the design of a lightweight CNN model, M-Net. The opportunity to optimize for both high accuracy and low computational overhead enabled the usage of the model in mobile devices and situations with limited computing power capabilities. In addition, we fine-tuned state-of-the-art pretrained models like VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2 on the guava leaf disease dataset to enhance the models' feature extractions' capabilities.
- Integration of Explainable AI (XAI) Techniques: However, since Explainable AI techniques such as Grad-CAM and SHAP were introduced to the model, its transparency gained momentum. Grad-CAM offers heatmaps that identify damaged parts on the leaf, while SHAP measures the important roles of individual features in the model's predictions. With these techniques, the system also becomes more transparent and its trustworthy to farmers who can make better decisions.
- Model Training and Evaluation: Hyperparameter tuning was used during model training on a preprocessed dataset to increase accuracy and decrease overall computational costs. The evaluation metrics were a classification

report, confusion matrix, and AUC ROC curve, that helped identify accurate decimal numbers, precision, recall, F1-score, and model's separation of disease categories.

- Mobile/Web Application Development: A leaf-location-capable mobile-versioned web application was developed to request leaf input from farmers, which would allow for instant classification of the disease. The web application uses the trained CNN model to classify diseases and the use of XAI tools such as Grad-CAM to show affected areas on the leaves thus making the results interpretable.
- Performance Testing and Comparative Analysis: For evaluating the efficiency of the model while working with new data, it was tested against another test dataset. Metrics evaluated during the evaluation included accuracy, precision, recall, F1-score, and AUC ROC curve. We evaluated trade off between M-Net and pretrained models to determine performance in reference to efficiency against accuracy to guide the selection of model that would optimally support resource-limited environments.

With this approach, the system becomes both efficient and effective in solving the practical problems of plant disease identification and assisting farmers in approaching the optimization of guava cultivation.

1.5 Project Outcome

The major achievement of this proposed system is that it designed the lightweight, efficient, easy to understand and interpret deep learning based model for guava leaf disease identification. M-Net is a lightweight CNN model that Explainable AI tools are used to provide instant disease classification, so that farmers are provided with a reason so as not to make the model a blackbox. Key outcomes include:

- Accurate Disease Detection System: The M-Net model provided an accurate detection and classification of different guava leaf diseases with high accuracy using the visual data. The model demonstrated the stability of practic usability by the evaluate based on metrics accuracy, precision, recall and F1-score. In addition, the model was designed with computational efficiency in mind, allowing it to run smoothly on resource constrained devices like mobile phones or low-end computers.
- Explainable AI Integration for Transparency: By applying Grad-CAM and SHAP methodologies, the system now looked visually friendly on how the system operates. With the use of Grad-CAM, the system displayed the heatmaps that clearly showed the regions of the guava leaf affected by a disease and this made it easy for the farmers to identify the infection zones. Such transparency enhances the confidence in the model and enables the farmers to be knowledgeable for a timely and swift action.

- Mobile-Optimized Web Application: A web application was developed which allows farmers to upload guava leaf pictures for immediate disease classification. By creating a user-friendly interface compatible with mobile devices, the system becomes easy to use for smartphone owners in remote areas where diagnosis of diseases is complicated. Immediate effects arising out of the classification process of diseases expedite action in the field.
- Real-World Applicability and Usability: With the project, it was achieved a practical approach towards identifying guava leaf illnesses using AI in tandem with an accessible system. Through better crop management and loss reduction on yield as well as increased productivity, the system becomes an affordable and effective tool for farmers to monitor agricultural operations for diseases.
- Comparative Performance Analysis: Compared with other pretrained models (VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2), M-Net performed better in terms of balancing between computation efficiency and accuracy particularly where resource is limited. This evidence reinforces the usefulness of lightweight models for practical agriculture environments.

To conclude, the Guava Leaf Disease Detection System holds considerable promise for the improvement of agronomy through early disease detection, improved decision-making and more effective crop management and yield conservation. The results contribute additional insights into the developing field of AI for agriculture by offering a predictable, transparent solution to assist sustainable farming.

1.6 Organization of the Report

This thesis is organized into six chapters, each focusing on different aspects of the research process, from identifying the problem to drawing conclusions. The structure is as follows:

- Chapter 1 - Introduction: The following chapter illustrates the primary research study focus, the significance of guava cultivation, the barriers presented by leaf diseases, and the use of artificial intelligence (AI) for disease diagnostics. Additionally, the chapter presents the objectives and methods of the study, describing the approach to design the disease detection system.
- Chapter 2 - Background and Literature Review: This chapter reviews earlier studies on plant disease detection, focused on studies around guava leaf diseases. It identifies popular techniques such as deep learning, Convolutional Neural Networks (CNNs), Explanation AI (XAI). Moreover, this chapter addresses outstanding issues that are present in the existing literature that this thesis aims to address.
- Chapter 3 - Research Methodology: This chapter is an account of the research methodology that was used in this research project. It presents a

comprehensive description of data collection and preprocessing, of designing the M-Net model, of incorporating XAI, and of M-Net web application development. In addition, this chapter touches upon training the model and the criteria for evaluation.

- Chapter 4 - Implementation and Results: This chapter provides the practical application of the disease detection model as well as development of the web application. It gives an overview of the experimental configuration, evaluates the model performance, and compares the results. Results from classification report, confusion matrix, and AUC roc curve analysis, are also shown.
- Chapter 5 - Discussion and Analysis: In this chapter, I examined the results of the outcomes that the system has, as well as where it can be improved. In this section, the results of the M-Net model are compared with pretrained CNN models, and the pragmatic applications to guava farmers in the real world are explored. The chapter also examines the role that XAI plays in enhancing of the system's user friendliness.
- Chapter 6 - Conclusion and Future Work: The final chapter summarizes the principal research findings and gives conclusions reached in the analysis. Moreover, the current chapter describes the scope of the present study as well as suggests directions for continued study, which include improving the model and extending its implications to other areas of agriculture.

In stepwise, each chapter describes the work that has been done so far, including the major results, the research methodology, and the general implications of the model.

Chapter 2

Background

This chapter gives a comprehensive review of the past studies and techniques used in disease detection in plants, with focus on the use of deep learning models in the agricultural sector. It reviews past guava leaf disease detection studies to describe the use of pre-trained CNN models and what is lacking in current research this thesis hopes to address.

2.1 Introduction

Technology has had a significant impact on agriculture, and plant disease detection is one of the areas with high priority and is critical to the protection of the world's food security. Techniques such as manual visual inspections by farmers and agricultural experts may be time consuming and rely too much on individual perception and in most cases lead to inaccuracies. This therefore means that there is growing need for better and more accurate tools which has led to the development of such technologies as machine learning and deep learning. Among such models Convolutional Neural Networks (CNNs) has proved to be a strong performer in image based classification, making these a good fit for plant disease detection. However, CNNs have been very successful, but their lack of being able to offer an explanation for their choices is still a significant challenge. Lack of transparency in these models often results in farmers and specialists doubts and misinterpretation of their results.

This chapter's discussion emphasizes the detection of guava leaf disease because of common conditions such as Rust, and Caterpillar cutting etc., leading to poor plant health as well as low yield in the amount of fruits harvested. Traditional means of diagnosing diseases do not give timely and precise information, especially to early detections. As a result, deep learning models, in particular convolutional neural networks (CNNs), are being broadly applied for plant disease classification. Still, CNNs are great in terms of accuracy but tend to fall short in terms of clarity, which means that users may not come to believe in CNNs predictions. In turn, explainable artificial intelligence (XAI) tools such as Grad-CAM or SHAP have been combined with CNNs and it becomes possible to see and understand the regions of an image which are responsible for the model choices. This instils a sense of confidence in the decisions made by the model and allows farmers to then determine the exact areas of the leaf affected by disease.

The chapter reviews existing studies of plant disease diagnosis, specifically relating to guava leaves and outlines the failures of existing models to highlight the need for a light and transparent model such as M-net. Furthermore, the chapter emphasizes the importance of creating affordable smartphone and web applications with the use of CNNs for detecting plant diseases, right on spot, to serve farmers in rural areas.

2.2 Literature Review

Significant current developments in deep learning, specifically with the application of CNNs, have largely improved plant disease detection. CNNs are suitable for disease detection at early stages because of their expertise in identifying leaf image patterns. This ability to detect diseases at an early stage is crucial in reducing crop damage. Many of the literature on plant disease detection focus on transfer learning, the use of streamlined convolutional neural networks, and techniques that make it more interpretable. This section reviews central studies which address these issues and sets the stage for the development of the M-Net model within the thesis, which is for interpretable realtime guava leaf disease detection.

Studies have shown that transfer learning can effectively solve the problem of sparsely labeled data in plant disease detection because prior learned models can be re-used. Sharma et al. [5] proposed a transfer learning-based solution for the classifying of rice leaf diseases through pretrained architectures – VGG16 and ResNet50, adapted for use on limited data. By addressing data scarcity, this approach demonstrated the utility of transfer learning to achieve high accuracy in agricultural situations. This strategy supports the research goals of this thesis because here we used transfer learning to improve models such as VGG19 and MobileNetV2 for guava leaf disease recognition. Pretrained models' use enables the system to retain the reliable generalization capabilities despite operating under a limited labeled data. Similarly, Vishnoi et al. [24] used CNNs for detecting apple leaf diseases and showed that CNNs can generalize well across different crops, with accuracy 96%. This is very important since the central focus of this thesis is to advance the M-net model to transfer learning from apple leaf diseases recognition to guava leaf disease identification.

Semantically, a major element that was addressed in previous studies was the development of lightweight CNN models suitable for real-time processing by mobile devices. Chang & Lai [12] reported a lightweight CNN architecture for potato leaf disease detection with 94% accuracy and a mobile optimization. Advancing from these principles, the M-Net model presented in this thesis is intended to offer quick, reliable disease detection for guava leaves, ideally for mobile and embedded systems in resource poor areas. At the same time, Tufekci et al. [20] developed a CNN system that can identify rice diseases at 94%

precision combined with mobile software for on-site detection and diagnosis. This research illustrates how it is possible for deep learning models to equip farmers with prompt and precise diagnostics for diseases, which forms an integral part of the web application in this thesis. Furthermore, Vedhamuru et al. [10] presented a lightweight deep learning model using cross-residual skip connection separable CNNs, which guaranteed low computational cost as well as high accuracy. This strategy reflects the design intention of M-Net, which aims to curtail computation needs in the deployment to mobile phones.

Although models, such as CNNs, made a great performance, the unexplainability of these models presents a significant drawback in particular in agriculture, an area where decision-making trust is paramount for the farmers. To address this, some XAI techniques such as SHAP and GradCAM have been tailored to work with CNNs, and this allows the farmers to understand the front-end of the model decision. With the CNNs, Gupta et al. [22] applied Particle Swarm Optimization (PSO) for hyperparameter tuning resulting in improved performance and reduced operation times. In continuation of this thesis, hyperparameter tuning is crucial, and the approach taken here corresponds to the one described. The work of Rodríguez-Lira et al. [3] reviewed the co-location of CNNs and XAI: using XAI with CNNs results in greater accuracy and computational speed than with traditional models. In this thesis, SHAP and GradCAM was used to enhance model transparency, enabling farmers to understand how the model reached its decision and which leaf portions are most important for disease classification.

Several studies have underscored the need for mobile-optimized software programs that enable real-time disease identification. Aishwarya et al. [46] proposed a mobile-centric framework that uses light CNNs for tomato plant disease detection. The framework in question, specifically designed for use in agriculture settings, yielded an accuracy rate of 94%, allowing farmers to get real-time disease classifications through smartphones. The findings of this work are integrated into the mobile-compatible web application used within this thesis, which further supports real-time guava leaf disease detection and interaction with farmers using mobile phones. Mustak Un Nobi et al. [7] suggested real-time guava leaf disease detection using MobileNet and showed greater accuracy and efficiency with fewer resources. The said work is aligned with goals of this thesis with which M-Net model is being suggested for real-time classification and mobile device-compatible performance.

The use of hybrid approaches based on Convolutional Neural Networks (CNN) and other optimisation techniques is an ongoing research trend for plant disease detection. The use of a hybrid approach based on CNN and Particle Swarm Optimization for hyperparameter optimisation has been suggested by Gupta et al. [22], which proved effective with greater accuracy with lower processing time, demonstrating the potential use of hybrid approaches for plant disease classification. This is also supported by the methodology adopted by this thesis in using optimisation techniques for systematic optimisation of CNN models for guava leaf disease detection with a view to attaining maximal accuracy and runtime performance for facilitating real-time applications. The study by Pandian et al. [33] explored deeper models of CNN for leaf disease detection from plants, emphasising an important consideration among them contributing to greater improvement in accuracy using model depth. This is one of the guiding principles for modelling the M-Net model for the present study in seeking an ideal trade-off between model depth and computational efficiency with a view to attaining high accuracy with low resource usage.

Table 2.1: Summary of Literature Reviewed.

Author (s)	**Methodology**	**Accuracy**	**Key Findings**
Sharma et al. [5]	Transfer learning-based model for rice leaf disease classification, fine-tuning pretrained CNN models.	High accuracy	Leveraged transfer learning with VGG16 and ResNet50 to overcome data scarcity, enabling high classification accuracy with smaller datasets.
Vishnoi et al. [24]	Applied deep learning (CNNs) for detecting apple leaf diseases.	96%	Demonstrated CNNs' ability to generalize across plant species, reinforcing their applicability to guava leaf disease detection.
Chang & Lai [12]	Lightweight CNN model optimized for mobile devices to detect potato leaf diseases.	94%	Focused on real-time disease classification with a model optimized for devices with limited computational resources.
Gupta et al. [22]	Hybrid model combining CNNs and Particle Swarm Optimization (PSO) for hyperparameter optimization.	High accuracy	Improved classification accuracy and processing speed by integrating PSO for hyperparameter tuning.
Tufekci et al. [20]	CNN-based system for rice disease detection optimized for mobile platforms.	94%	Integrated CNN with mobile devices for real-time disease classification, demonstrating practical use in field settings.
Vedhamuru et al. [10]	Lightweight deep learning model using cross-residual skip connection separable CNNs.	High accuracy	Achieved high accuracy while reducing computational complexity, suitable for real-time applications on mobile devices.
Simhadri et al. [28]	Review of deep learning trends for plant disease detection, focusing on CNNs for various crops.	N/A	Highlighted the ability of CNNs to fine-tune for smaller datasets and adapt to specific crops.

Rodríguez-Lira et al. [3]	Overview of machine learning techniques for plant disease identification, focusing on CNNs.	N/A	Demonstrated CNNs' superiority over traditional methods in terms of accuracy and computational efficiency.
Aishwarya et al. [46]	Mobile-based real-time disease detection system using lightweight CNNs.	94%	Enabled real-time disease classification on smartphones, optimized for resource-limited field environments.
Mekala et al. [27]	System combining plant leaf disease classification with actionable disease management recommendations.	N/A	Bridged disease detection and management, helping farmers make informed decisions about treatment.
Mustak Un Nobi et al. [7]	Real-time disease detection system for guava leaves using MobileNet CNN.	High accuracy	Optimized for mobile platforms, providing efficient real-time classification of guava leaf diseases.
Bala & Bansal [37]	Review of deep learning in plant pathology, focusing on CNNs for leaf disease detection.	N/A	Reinforced the reliability and accuracy of CNNs for classifying complex disease patterns in plant leaves.
Chang et al. [12]	Real-time potato leaf disease detection system optimized for mobile devices.	94%	Demonstrated the effectiveness of lightweight CNNs for disease detection on mobile devices.
Pandian et al. [33]	Deep CNN model with five layers for detecting plant leaf diseases.	N/A	Showed that deep architectures improve classification accuracy across various plant species.
Singh & Yogi [34]	Comparison of several deep learning models for plant leaf disease detection.	N/A	Concluded that CNNs provided superior speed and accuracy for real-time disease detection.

2.2.1 Similar Applications

The use of Convolutional Neural Network (CNN) models for plant disease identification is a rapidly expanding field of scholarly research. Several studies have used CNN-based approaches to enable quick detection of plant diseases through mobile and web platforms and provide farmers with easily available, accurate, and timely data. A major function of such a system is enhancing detection efficiency, especially for farmers who are based in rural areas where there is limited access to agricultural specialists.

Rashid et al. [6] suggested a deep learning model for rapid detection from guava leaf field images. The model is comparable with that of this thesis because it rapidly detected guava leaf disease from images with the help of CNNs. Foo et al. [8] suggested a mobile application for guava leaf disease detection based on a model with a CNN with a 96% accuracy rate. The mobile application is comparable with that of this thesis because farmers would utilize it for real-time checking. Tembhurne et al. [42] suggested a mobile/web detection software for a disease for a tomato based on images taken from fields using a model with CNNs with a 94% accuracy rate. The above has been achieved with a mobile disease detection system for guava leaves.

Mazumder et al. [15] utilized transfer learning techniques together with a lightweight convolutional neural network (CNN) architecture for leaf disease identification for multiple plant species, reaching a performance of 97%. They demonstrated, by fine-tuning pre-trained VGG and ResNet models, the viability of transfer learning for plant disease classification. This is relevant to the methodology used within this present research, where VGG19 and MobileNetV2 were also pre-trained and then fine-tuned for guava leaf disease classification. Zhu and Gao [17] presented the MC-ShuffleNetV2, which is a lightweight architecture for CNN, specifically for detection of maize disease, and achieved a performance of 96% and is therefore especially suited for low-power processing scenarios. This is aligned with lightweight M-Net architecture used within the current research, which is able to find a delicate balance of accuracy alongside computational costs, making it amenable for use within resource-limited scenarios.

Tyagi et al. [41] created a mobile app for rice leaf disease identification using a simple convolutional neural network (CNN) model, which has an accuracy rate of 94%. This system allows farmers to upload images and obtain real-time classification of diseases. The mobile aspect of disease detection is relevant to the web application designed within the thesis, which also allows for real-time guava leaf disease identification using a mobile-friendly interface. Ghosh et al. [14] used Grad-CAM, an explainability tool in artificial intelligence, to generate heatmaps of infected areas on sunflower leaves. The use of Grad-CAM for visual explanation is similar to what is used within this research, which makes use of Grad-CAM and SHAP for making explanations more interpretable and understandable.

Gajjar et al. [30] applied a real-time plant disease detection through convolutional neural networks (CNN) on embedded low-computation platforms with 93% detection accuracy. Focus on embedded platforms and real-time detection is consistent with the web-based application developed within this thesis, which provides real-time disease detection on mobile platforms for guava leaf diseases. Yulita et al. [44] introduced a DenseNet-based convolution neural network (CNN) model for tomato leaf disease detection with 96% accuracy and integrated explainable artificial intelligence (XAI) techniques for model interpretation. This work had a direct impact on implementing techniques such as Grad-CAM and SHAP using XAI within this thesis for model visual interpretation and maintaining user confidence.

An application for diagnosis of maize leaf disease with a 94% accuracy level has been created by Khan et al. [43]. This is made achievable through diagnosis with an application using mobile platform-image upload of leaves based on the mobile-first philosophy of this thesis. Mekala et al. [27] provided a hybrid approach of combining plant disease identification with disease treatment recommendation so that, once the disease is identified, farmers are able to derive useful insights. This is an improvement over previous work since recommendation for management is incorporated into the system for identification of guava leaf disease.

Table 2.2: Summary of Similar Applications.

Author (s)	Methodology	**Accuracy**	**Key Findings**
Rashid et al. [6]	Hybrid Deep Learning	Not mentioned	Developed a hybrid model for real-time detection of multiple guava leaf diseases.
Foo et al. [8]	Mobile App Development, CNN	96%	Created an Android app for guava leaf disease detection, providing real-time feedback to farmers.
Tembhurne et al. [42]	Deep Learning, Mobile/Web Application	94%	Developed a mobile/web application for tomato leaf disease detection with real-time disease classification.
Mazumder et al. [15]	Transfer Learning, Lightweight CNN	97%	Proposed a lightweight transfer learning-based CNN for disease detection across multiple plants.
Zhu & Gao [17]	ShuffleNetV2, Lightweight Deep Learning	96%	Developed MC-ShuffleNetV2 for maize disease recognition, balancing accuracy and computational efficiency.
Tyagi et al. [41]	Deep Learning, Mobile Application	94%	Created a mobile-based rice leaf disease detection system using a lightweight CNN.
Ghosh et al. [14]	Hybrid Deep Learning, Explainable AI	High accuracy	Used Grad-CAM and XAI to explain sunflower disease predictions, improving transparency.

Gajjar et al. [30]	CNN on Embedded Platform	93%	Developed a real-time plant disease detection system for embedded platforms with low power consumption.
Yulita et al. [44]	Dense CNN Model, Mobile Application	96%	Developed a tomato disease detection app using DenseNet-based CNNs, with integrated XAI for transparency.
Khan et al. [43]	Deep Learning, Mobile Application	94%	Implemented a mobile-based system for maize leaf disease detection, providing real-time feedback to farmers.

2.2.2 Related Research

Application of deep learning algorithms, especially Convolutional Neural Networks (CNNs), has been incorporated in plant disease identification and has brought about an increase in speed, accuracy, and efficiency in disease classification activities. These activities are important in enabling timely control of diseases in agriculture for loss reduction and productivity enhancement. Various research has contributed considerably in the field using CNNs, transfer learning, Explainable AI (XAI) methods, and mobile or web-based platforms. The methods adopted by the various research are the basis for research presented in this thesis and focus on disease detection in guavas using an efficient CNN model coupled with more explainable AI for increased transparency.

Demilie et al. [32] conducted a comparative study of methods used in plant disease diagnosis in an attempt to compare classical machine learning methods with the newer deep learning approaches, including Convolutional Neural Networks (CNNs). Their study reported accuracy and processing time increases associated with CNNs, especially in handling large sets of images. The study highlights the importance of deep learning in diagnosing diseases in plants and has relevance on methods applied within this thesis, in which CNNs are used to perform effective inspection of guava leaves and related diseases. Similarly, Pandian et al. [33] proposed a five-layer deep CNN for plant leaf disease classification. Their design achieved an impressive accuracy rate of 94% in disease identification of tomato plant leaves and thus proved that newer CNN configurations have improved detection capabilities. The study has been instrumental in developing the M-Net model employed in this thesis, placing high importance on accuracy and efficiency in disease detection in guava leaves.

Latif et al. [39] to accurately detect rice plant diseases at 96% accuracy. The authors demonstrated that adapting pretrained models like VGG or ResNet can be utilized to detect plant diseases. Transfer learning is also used in this thesis to train pretrained models (VGG19 and MobileNetV2) for guava leaf disease detection and increase efficiency of the model in resource poor environments. Ulutaş et al. [25] developed an aggregated CNN model for tomato leaf disease recognition that was 95% accurate by using several interconnected CNN architectures. Combining various CNN architectures, the ensemble model showed better accuracy and robustness, similar to the strategy used for this study optimizing the detection of guava leaf diseases.

Prakash & Geetha [13] introduced a deep attention convolutional network (DACN), tailored to the potato leaf disease detection to achieve 97% accuracy. The use of attention mechanisms incorporated into the model facilitated focusing on important pieces of the leaf image leading to increasing the classification accuracy. This work emphasizes the role of the attention mechanism on CNNs to accentuate a need for customer-specific CNN architectures as an attempt to improve disease detection in particular plants like the guava leaf. Ranjana et al. [26] utilized Mask R-CNN model to detect plant leaf disease with an accuracy of 94%. Mask R-CNN classifies diseases and segments the leaf, providing a better notion of the extent of disease occurring in the leaf. More work can be undertaken to explore this segmentation technique in more depth to allow for more detailed inspection of the locations of the guava leaf disease.

Atila et al. [36] proposed that, a light-weight deep learning model in the form of EfficientNet could be successfully used to classify plant leaf disease with 96% accuracy. Although very efficient, EfficientNet can provide strong classification with very few processing demands which makes it suitable for deployment over mobile or embedded devices. This method is in accordance with the goals of this research since the development of an effective lightweight CNN model (M-Net) process seeks to support real-time mobile-based disease diagnosis. Likewise, Vallabhajosyula et al. [29] constructed a deep ensemble neural network for plant leaf disease detection at 98% accuracy attained through the combination of various pretrained models.

Gong et al. [35] developed a sophisticated Faster R-CNN model for detecting apple leaf diseases, reaching a precision level of 97%. The model emphasized on accuracy in the detection of diseases and can be used as a useful tool in future studies to help in improving classification of guava leaf diseases whenever accuracy is of great importance. A DenseNet-based CNN model was developed by Yulita et al. [44] for tomato leaf disease detection, achieving 96% accuracy, while XAI methods were employed to explain the model’s decisions. This development has encouraged the incorporation of XAI approaches such as Grad-CAM and SHAP in this thesis to make the process of decision-making more transparent and interpretable.

Lastly, Khan et al. [43] proposed a mobile-enabled deep learning system for the detection of maize leaf diseases with commendable 94% accuracy. Farmers are able to use their mobile app to upload images of the leaves immediately and get real-time disease diagnosis, after which they will pay the mobile-first strategy used in this thesis which has a mobile-optimized web app for instant guava leaf disease classification.

Table 2.3: Summary of Related Research.

Author (s)	Methodology	Accuracy	Key Findings
Demilie et al. [32]	Comparative study of plant disease detection methods	Not mentioned	Compared CNNs with traditional methods, highlighting CNN's superiority in accuracy and speed.
Pandian et al. [33]	Deep CNN architecture with five layers	94%	Proposed a deep CNN model for plant leaf disease detection, achieving high accuracy.
Latif et al. [39]	Deep learning for rice disease detection	96%	Applied a CNN model for detecting rice leaf diseases, achieving high classification accuracy.
Ulutaş et al. [25]	Ensemble CNN model for tomato leaf diseases	95%	Used ensemble CNN models to improve disease detection in tomato plants.
Prakash & Geetha [13]	Deep Attention CNN (DACN) for potato disease detection	97%	Proposed a CNN model with attention mechanisms for detecting potato leaf diseases.
Ranjana et al. [26]	Mask R-CNN for plant disease detection	94%	Used Mask R-CNN for disease detection and segmentation in plant leaves.
Atila et al. [36]	EfficientNet model for leaf disease classification	96%	Demonstrated EfficientNet's ability to classify plant diseases efficiently with high accuracy.
Vallabhajosyula et al. [29]	Transfer learning-based deep ensemble model	98%	Applied transfer learning and ensemble models for plant leaf disease detection.
Gong et al. [35]	Improved Faster R-CNN for apple leaf diseases	97%	Developed an improved Faster R-CNN model for precise apple leaf disease detection.

2.3 Gap Analysis

Although significant advancements in AI-enabled plant disease detection with CNNs have been accomplished, much still remains to be achieved in order to enhance the accuracy, efficiency, and environmental friendliness of these systems for real-world, and/or resource-constrained environments. The next section proposes key zones that need further study and development in the process of enhancing plant disease detection systems:

- Limited Disease Detection for Specific Crops: Most studies have focused on staple crops such as tomatoes, rice, and maize with minimal provisioning attention on economically important fruit crops such as guava. Anything that highlights the study of guava leaf diseases is essential because it can bring about models, which can be applied to fruit trees hence making AI-based systems helpful in many agricultural disciplines.

- Lack of Real-Time Disease Severity Assessment: Although some models can detect diseases most cannot perform instant assessments of severity of diseases. Integration of early disease detection coupled with real-time scoring has the potential of allowing farmers to prioritize their interventions which will improve the efficiency of precision agriculture. With the aim of closing this gap, farmers would be in a better position to make more informed decisions informed by choices that they could directly tailor to respond to the most severe outbreaks initially.

- Limited Integration of Hybrid Models: Currently, models often only utilize CNNs without paying attention to other AI tools (e.g., transfer learning), optimization algorithms (e.g., Particle Swarm Optimization), or ensemble methods. The application of hybrid techniques would enhance performance, increasing the precision of detection as well as the versatility of the model in varying field conditions and plant types. This area provides the possibility to construct more robust models that can accurately predict illness in different environmental conditions.

- Low-Resource and Low-Cost Deployment Solutions: Despite highly advanced AI systems being designed, most of the systems for disease detection capitalize on powerful data centres, which are out of reach for smallholder farmers in rural or resource-poor areas. One of the major challenges ahead involves enhancing affordable and easily accessible options to the farmers, whereby they now use tools for disease detection without the burden of costly infrastructure. Filling this gap will pave the way for easy adoption of AI tools by farmers.

- Explainable AI (XAI) for Trust and Adoption: The CNNs are often harshly rubbed about it possible lack of transparency, and this may undermine farmers' confidence in believing in the aired program's results. The application of such techniques as Grad-CAM and SHAP makes it possible to visualize model decision-making that might help farmers feel confident in AI systems. We can add explainable AI approaches to applications of detecting plant disease to increase transparency and trust which are important approaches towards greater adoption of AI by the agricultural field.

- Limited Dataset Availability for Guava Leaf Disease: Development of AI models is often hindered by difficulties in the availability of enough labeled datasets. While large amount of dataset are available for certain crops datasets pointing to guava leaf diseases are scanty. The absence of high-quality annotated images limits the model's ability to generalize well with regards to various guava diseases. The generation of exhaustive, marked datasets for guava diseases will result in more efficient models and increased speed in the development of reliable detection technology.

- Focus on Single Disease Classification: There is a tendency for research to focus on determining a single disease in crops, yet in the real world, crops can be affected by various diseases at once. The need is to develop classification models that were able to identify more than one disease on one single image. By developing models that can identify multiple diseases for a single shot, farmers would be able to get a fuller diagnosis, improving crop management and health as they deal with a host of matters in one stride.
- Need for Field-Ready, Integrated Solutions: Even though AI systems have promise in a controlled environment, few practical tools had been created for the field. The real-world performance of such models is affected by factors such as lighting, image clarity as well as differences in the field conditions. Also, a lot of the solutions do not provide timely updates / practical recommendations, for instance, for determining the severity of diseases or suggesting proper treatments. Solutions needed are those that can provide immediate disease diagnoses and actionable recommendations to farmers while in the field to make the process more interactive and accessible.

Overall, the analysis of gap in AI-based plant disease detection research identifies some of the most essential flaws of current AI-based plant disease detection research, that relate to guava leaf diseases and resource-limited settings. Mitigation of such limitations will reinforce AI systems to ensure they become more usable and deployable on an agricultural scale. This thematic area is a prerequisite towards developing an efficient, transparent and inclusive disease detection system for farmers to effectively manage their crops, and consequently enhance agricultural productivity and sustainability.

Table 2.4: Summary of Gaps.

Authors(s)	**Gap**
Rashid et al. [6]; Foo et al. [8]	Limited Disease Detection for Specific Crops: Most studies focus on widely grown crops like rice and tomatoes, leaving a gap in AI-driven disease detection models for specialized crops like guava.
Vishnoi et al. [24]; Gupta et al. [22]	Lack of Real-Time Disease Severity Assessment: Many existing models identify diseases but fail to assess the severity or extent of infection, which is essential for effective crop management.

Atila et al. [36]; Tyagi et al. [41]	Limited Integration of Hybrid Models: Many studies focus on single AI techniques (e.g., CNNs) but do not explore hybrid models that combine multiple techniques, which could improve detection accuracy and adaptability.
Gajjar et al. [30]; Mustak Un Nobi et al. [7]	Low-Resource and Low-Cost Deployment Solutions: Many models rely on high-performance computing, which is not feasible in rural, low-resource settings. A gap exists in optimizing solutions for mobile and embedded platforms.
Ghosh et al. [14]; Ranjana et al. [26]	Explainable AI (XAI) for Trust and Adoption: AI models often lack transparency, and there is a need for explainable AI methods (like Grad-CAM and SHAP) to improve trust and adoption among farmers.
Yulita et al. [44]; Tembhurne et al. [42]	Limited Dataset Availability for Guava Leaf Disease: There is a shortage of labeled datasets for guava leaf diseases, which limits the ability to train accurate and reliable models for this specific crop.
Pandian et al. [33]; Singh & Yogi [34]	Focus on Single Disease Classification: Many studies focus on detecting only one disease, whereas real-world applications require models that can classify multiple diseases simultaneously.
Vallabhajosyula et al. [29]; Tufekci et al. [20]	Need for Field-Ready, Integrated Solutions: While many systems are tested in controlled environments, there is a gap in creating field-ready solutions that provide real-time feedback and disease management recommendations.

2.4 Summary

This chapter describes the state of AI-based detection of plant diseases in the current context as well as the usage of Convolutional Neural Networks (CNNs) and other deep learning strategies. The reviewed literature illustrates significant advances in developing disease detection systems: various crops, tomatoes, and rice are represented, suggesting revolutionary results of AI in agriculture. While the progress is remarkable, there are issues in the continuing search, namely, the lack of attention to some crops, such as guava, a limited ability to assess the disease severity in real-time, and occasional use of hybrid AI models.

Furthermore, a huge deficiency of cheap and resource-efficient solutions exists which could support the widespread deployment of technologies in underprivileged regions. Another major challenge is the lack of available labeled datasets for both guava and the limitations of existing models in the conduct of multi-disease classification, or providing practical real-time information to the farmers. Filling in these gaps may result in significant improvement of the accuracy, efficiency, and appropriateness of plant disease detection systems.

One of the most important future work areas pertains to the development of domain-specific models of overlooked crops, modifications for utilization on portable or low power platforms, integration of XAI to enhance model interpretability, and creation of rich datasets for better generalization on real-world applications. This thesis aims to enhance these areas by developing an effective, real-time, and comprehensible system for guava leaf disease detection, which will cover the hitherto unknown weaknesses in prior studies. This chapter lays down the foundation, which prepares the stage for the subsequent chapters, which will delineate the methodology, model construction, and the methodology used in addressing these issues in the detection of guava leaf disease.

Chapter 3

Research Methodology

This chapter presents the methodology used in developing the Guava Leaf Disease Detection System, including data collection, specification of data sets, selection of the model, training methodology, and setting of metrics. Moreover, the chapter addresses the creation of a personalized M-Net model and the incorporation of Explainable AI (XAI) approaches to make the understanding and clarity of the model predictions better.

3.1 Methodology/Requirement Analysis & Design Specification

3.1.1 Overview

This chapter describes the methodology used in developing a lightweight Convolutional Neural Network (CNN) for real-time guava leaf disease detection. The objective was to develop a model that is extremely accurate, lightweight and, user-friendly, particularly to be deployed on smartphones and in resource-constrained situations. This technique involves the integration of transfer learning techniques with widely used pretrained models VGG19, MobileNetV2, and ResNet152V2 and the development of a lightweight custom model, M-Net. After preprocessing and augmenting the dataset, seven types of guava leaf diseases were identified for the model, increasing the imagery from 1050 to 8400 photos. The M-Net model proved noteworthy at 99.04% accuracy which surpassed the pre-trained models for effectiveness and speed.

To improve transparency, the system was augmented with the Explainable AI (XAI) techniques SHAP and Grad-CAM, from which the users get to see how the model is arriving at its decisions. Such a process enables the farmers to determine the most influential portions of the leaf to the model's decisions, and can give the farmers confidence in the advice given. The model was incorporated into a web application that can be used on mobile devices that enable an immediate upload of leaf images for a swift discovery of diseases using cheap hardware. These applications, with their mobile optimization, can be utilized productively in resource-limited rural areas. This chapter describes the mechanisms of preparing the dataset, developing the model, training it with hyperparameter tuning and implementation of XAI methods focused on real-time abilities of the system, high accuracy and mobile friendliness.

3.1.2 Proposed Methodology

The procedure of developing a light CNN model for detecting diseases of guava leaves is divided into several stages with an objective of achieving high accuracy, efficiency, and compatibility with mobile devices and limited resources. The approach is based on transfer learning, developing a custom lightweight CNN architecture (M-Net), tuning hyperparameters, and applying refined training plan, such as learning rate scheduling, early stopping mechanism, and checkpointing, to increase efficiency and performance. A mobile-friendly web application is also created to facilitate real-time classification of the diseases of the leaves. The below describes the methodology's each stage in detail:

1. Dataset Collection

For the present study, pictures of guava leaves infected with various diseases were taken from two guava nurseries at Ashulia, Savar, Dhaka. The dataset includes seven classes:

- Caterpillars
- Cutting Weevil
- Die Back
- Healthy
- Mealybug Pests
- Red Rust
- Yellow Spot

The initial dataset consisted of 1050 images (150 images per class). To increase the size of the training dataset, we used data augmentation techniques such as rotation, sheering and split to create 8400 images (1200 per class). Such data augmentation application was a cornerstone for improving model robustness, preventing overfitting, and ensuring proper generalized performance in datasets that have yet to be seen.

2. Preprocessing and Augmentation

When the dataset was increased, several preprocessing steps were taken as follows:

- Image Resizing: All the photos were standardized in size for there to be a uniformity of the input data.
- Normalization: The pixel values of all images were scaled to a range between 0 and 1, making the training process more efficient.
- Data Augmentation: Rotational, shearing, and splitting transformations were employed to increase the diversity of the dataset, simulate a wide set of environmental conditions, and increase the ability of the model to work with different orientations and distortions of the leaves.

By adding these preprocessing steps, the dataset increased so it became more diverse and larger, the content of which made it easier to establish a solid base for the training of the models.

3. Transfer Learning Using Pretrained CNN Models

In the primary phase of our methodology, we had used transfer learning with five pre-trained CNN models. VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2. This approach is that of using models trained on massive data (e.g., ImageNet) on a new task with limited training datasets. In this phase, the following process was adopted:

- Model Selection: These five pre-trained models were selected for their excellent results on image classification problems.
- Fine-Tuning: The output layer for each pretrained network was changed with the addition of a layer of relevance for identifying the seven types of guava leaf diseases. The Guava dataset was augmented and used to fine-tune the pretrained models to recognize the pattern of guava leaf diseases.
- Evaluation: Benefits of the models were assessed by calculating their accuracy, precision, recall and F1 score after its training. Despite their remarkable performance, there was the need to build a more efficient and compact CNN that could be deployed on mobile devices.

4. Development of M-Net (Lightweight CNN)

The subsequent stage was focused on developing a customized lightweight CNN model (M-Net) that was particularly designed to work perfectly in real-time on embedded and mobile platforms. M-Net was especially designed to provide increased accuracy with fewer parameters than conventional CNN architecture. This stage was focused on:

- Custom Architecture: To better keep up the efficiency, the model M-Net used the low number of layers and parameters without sacrificing high performance features. It was extremely crucial for ensuring that the model could carry out real-time disease detection on low power demanding mobile devices. That is, without consuming high power.
- Hyperparameter Tuning: Parametric tuning of the important parameters like learning rate, batch size and number of epochs was performed to optimize model performance. Grid search and random search algorithms guided the selection of these parameters, which assist in determining the optimum settings.
- Learning Rate Scheduler: A learning rate scheduler was implemented, which automatically changed the learning rate over time, speeding up convergence and preventing over fitting.
- Early Stopping: Early stopping was implemented to minimize the risk of overfitting, and to calculate time in an optimum manner. In training, the process was halted when there was no more improvement seen on the validation set.
- Checkpointing: Checkpointing mechanism was used to save the best model while training. This ensured that the best version of the model could be used for evaluation and deployment.

- After training, M-Net achieved an accuracy of 99.04%, surpassing the pretrained models in both accuracy and computational efficiency.

5. Explainable AI (XAI) Integration

While the web application does not display XAI results, the integration of Explainable AI (XAI) techniques like SHAP and Grad-CAM was applied to enhance the model's interpretability. These techniques enable us to:

- SHAP: SHAP values explain the contribution of each feature (pixel) to the final classification. While not directly part of the web app, SHAP was used to gain insights into which features were most important for disease classification.
- Grad-CAM: Grad-CAM generates visual heatmaps to highlight regions in the image that most influenced the model's prediction. This technique was useful during model evaluation to understand the decision-making process of the CNN and ensure that the model is making the correct diagnoses based on relevant features of the leaf images.

These techniques were integrated into the model development phase but were not part of the web application's user interface, as the focus of the app is on providing simple, real-time classification results to farmers.

6. Web Application Development

The final stage of the methodology involved developing a web application for real-time guava leaf disease detection. The web app allows farmers to upload images of guava leaves, which are then classified by the trained M-Net model. The following steps were involved in the development of the web application:

- Frontend Development: A user-friendly frontend was designed to allow easy image upload. Farmers can simply take a photo of the leaf using a smartphone camera and upload it to the application.
- Backend Integration: The backend infrastructure that supported the web application had the M-Net model integrated. When uploaded, the backend deals with the image and makes use of the trained model to give the pertinent disease label to the leaf.
- Mobile Optimization: The platform was intended to work well on mobile devices, which makes it accessible to farmers who do not have high-end hardware. This makes the web application viable in the rural areas in which the mobile devices are the leading digital tool.
- The web application provides instant disease classification to ensure that farmers can access updated information about the status of their guava crops as soon as possible. As the model is lightweight, the web application runs seamlessly across less powerful hardware.

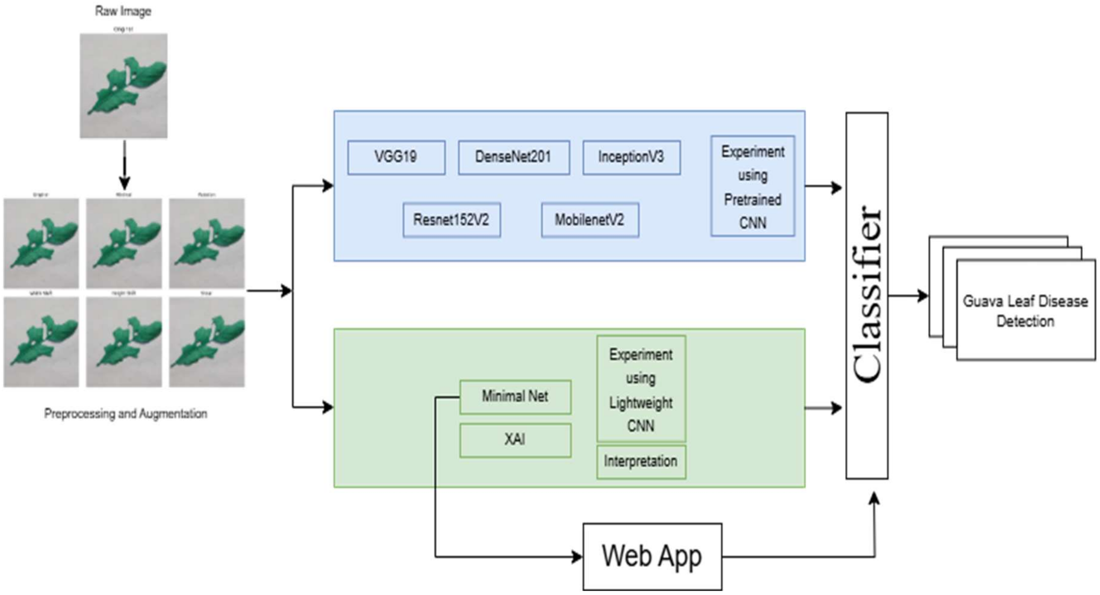


Figure 3.1: Proposed Methodology

3.1.3 Functional and Nonfunctional Requirements

In developing the Guava Leaf Disease Detection System, functional and nonfunctional requirements had to be set in order to ensure the system works efficiently. If functional requirements outline the system's services and operations, nonfunctional requirements reveal its performance aspects that involve such attributes as speed, adaptability and protection against threats.

Functional Requirements

1. Image Upload and Classification
    - Users should have the ability to upload images of leaves of guava to the system for identification of diseases. Ultimately, the objective is to have the system analyze uploaded images and identify the exact disease in the input. This capability is a key determinant in relation to how users interact with the system.
2. Real-Time Disease Detection
    - Images that are uploaded to the system need to be classified as soon as possible, processing the data in seconds. Rapid processing lets farmers respond quickly to problems affecting the crops, retaining healthy management.
3. Disease Class Prediction
    - The system should be able to specify one of seven or recognized disease classifications. Caterpillars, Cutting Weevil, Die Back, Healthy (Note: Die Back, Healthy, Mealybug Pests, Red Rust, and Yellow Spot are not names of vegetable). Knowing the exact disease tells farmers the actions to take to control it.
4. User-Friendly Interface
    - An easy to use interface should be incorporated, so that farmers who have minimal experience in technology are able to easily upload images and understand the results. By incorporating simple and navigable layouts, the tool is able to become more accessible to and user friendly for all farmers.
5. Mobile Compatibility
    - Mobile optimization should be prioritized to enable farmers visit it using hand phones which are common among rural areas. This makes it easier to the users who may be constrained in terms of technology.
6. Model Updates and Performance Monitoring
    - The web application should allow for the model to be updated on occasion, which enhances performance and ensures the accuracy since new disease data arise.

Nonfunctional Requirements

1. Performance
    - The system must interpret and classify images in 3-5 seconds to provide instant findings to facilitate timely and productive farming decisions.

2. Scalability
    - The system needs to be resilient enough to cope with load increases when adoption expands, and data volumes grow. This ensures outstanding performance even with the increased user loads.
3. Availability and Uptime
    - The system should support 24 hour operation to enable farmers use it when they wish without regard to their location or time zone.
4. Usability
    - The system should feature an intuitive design, which will allow even farmers with no special technical skill to upload images and comprehend results easily.
5. Security and Privacy
    - The system must secure both the security and privacy of all user data, particularly from farmers, as well as assure that only selected people can access the information.
6. Maintainability
    - It is necessary that the system should be easily maintained and updated, which would allow adding extra disease categories and further model improvements.
7. Resource Efficiency
    - The system must run effectively on low-cost, low-power consumptions devices like smartphones and embedded systems to be available in rural areas with limited resources and connectivity.

These requirements ensure that the Guava Leaf Disease Detection System runs productively and highly reliable and also readily available, especially in regions, where resources are scarce. They guide the design and improvement, ensuring that the system provides reliable, fast, easily accessible detection of diseases, leading to the improvement of more effective crop management and increased agricultural output.

3.1.4 Context Diagram

A Context Diagram is somewhat like an overall view of the structure, the extent, and the modes of the communication of a system. Context diagram of the Guava leaf-disease detection system clearly depicts the relationship between the farmer, the web application, and the backend server (where M-Net model is hosted). The diagram illustrates communication connections between the users, the system, emphasizing that the method guarantees that detection of diseases is simple and within the reach of the farmers especially where advanced technology is unavailable.

Components of the Context Diagram

1. Farmer (User):
    - Role: The farmer is the primary user who interacts with the system.
    - Actions:

        - Upload images of guava leaves via mobile or desktop.
        - Receive real-time disease classification results.
    - Goal: Diagnose guava plant health and take appropriate action based on detected diseases.
2. Web Application:
    - Role: Acts as the user interface between the farmer and the backend system.
    - Actions:
        - Accept image input from the farmer.
        - Display disease classification results.
    - Goal: Provide an easy-to-use platform for farmers to upload images and instantly receive disease classification results.
3. Backend Server (Model):
    - Role: Hosts the trained M-Net model for processing leaf images.
    - Actions:
        - Receive the image from the web application.
        - Process the image with the M-Net model.
        - Return disease classification result to the web application.
    - Goal: Accurately classify the disease based on the image and send results back to the user interface.
4. Trained M-Net Model:
    - Role: The core of the system, the CNN model, classifies guava leaf diseases.
    - Actions:
        - Process the input image and classify the disease (e.g., Caterpillars, Healthy).
        - Return disease labels based on the image.
    - Goal: Provide accurate disease classification results to support decision-making.
5. External Data Sources:
    - Role: Provides data for model updates and retraining.
    - Actions: Supply additional image datasets or disease reference data.
    - Goal: Ensure the model stays updated with the latest information for improved accuracy.

Interactions in the Context Diagram

- Farmer to Web Application: The farmer uploads a guava leaf image using the web interface.
- Web Application to Backend Server: The image is sent to the backend server, which hosts the trained M-Net model.
- Backend Server to M-Net Model: The backend server sends the image to the M-Net model for classification.
- M-Net Model to Backend Server: The model returns the disease classification result to the backend server.

- Backend Server to Web Application: The backend server sends the classification result back to the web application, which displays it to the farmer.

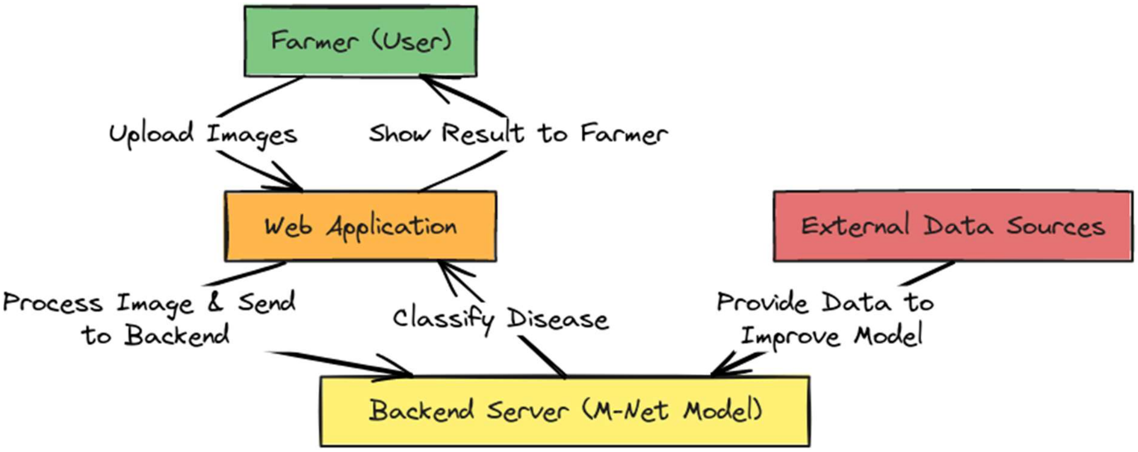


Figure 3.2: Context Diagram

The Context Diagram showcases the system's high-level interactions:

- Farmer (User): Provides input (guava leaf images) and receives real-time disease classifications.
- Web Application: Facilitates the user’s interaction with the backend, allowing the upload of leaf images and display of results.
- Backend Server: Handles the classification task, interacting with the M-Net model to process images and return results.
- M-Net Model: The central component, responsible for accurately classifying the disease based on the uploaded image.
- External Data Sources: Continuously supply new data for updates and retraining, ensuring the model remains accurate over time.

3.1.5 Data Flow Diagram Level 1

Data Flow Diagrams (DFDs) provide a visual representation of the flow of data within a system; where data flows, and what data is passed, is made explicit. This section describes Level 1 DFD for the Guava Leaf Disease Detection System, showing the flow of data in key elements of the system and storage areas.

1. Components of the DFD Level 1:

- External Entity: Farmer (User)
    - Role: The farmer becomes the external interface whereby he uploads images of guava leaves and retrieves the outcome of disease classification.
    - Input: Web application pictures of guava leaves uploaded.
    - Output: The user uses the results of the disease classification.

- Process 1: Web Application
    - Role: The web application serves as the driver's seat for the farmers to interact with the system's backend. It handles uploading of images and reports to the farmer the results of the classification of the disease.
    - Input: Leaf image submitted by the farmer.
    - Output: The backend system processes the uploaded image, provides the disease classification result and sends it back to the farmer.
    - Sub-processes:
        - Image Upload: Processes the image transmitted by the farmer..
        - Result Display: Serves the farmer to know his/her disease classification result.
- Process 2: Backend Server (M-Net Model)
    - Role: The M-net model on backend server uploads the image and does disease classification.
    - Input: Data from an uploaded image from the web application.
    - Output: Outcome of the disease classification (such as "Healthy", "Red Rust," etc.) that has been detected.
    - Sub-processes:
        - Image Preprocessing: When adapting the image dimension, normalizing its pixel values in accordance with the standards and after that undertaking other pre-processing where necessary before being fed into the model.
        - Disease Classification: The M-Net model then processes the image data and makes the disease classification.
- Data Store 1: Image Data Store
    - Role: Uploaded leaf images are stored as they wait to be processed through the backend server in this data store.
    - Input: Images posted online from the web application by the user.
    - Output: The uploaded images of the leaves are passed along to the backend server for classification.
- Data Store 2: Disease Classification Model (M-Net)
    - Role: This data store is in charge of the pre-trained M-Net model for classifying diseases.
    - Input: Decorations that are processed and analyzed by the server that exists in the backend.
    - Output: The classification of the disease is retrieved by the web application from the backend.
- External Entity: External Data Sources
    - Role: Data retrieved from the outside sources yield new datasets that are critical for model M- net improvement and its further periodic retraining.
    - Input: An altered model that has enhanced its abilities in recognizing plant diseases. An altered model that has enhanced its abilities in recognizing plant diseases..

2. Data Flow Description:

- Image Upload (Farmer to Web Application): Farmer uses the web application to upload a photo of a guava leaf. While the process continues, it is stored in Data Store 1.
- Data Transfer (Web Application to Backend Server): The image is copied from the web application to the backend server for further disease classification processing.
- Image Processing and Classification (Backend Server to M-Net Model): After reaching the backend server, the image is preprocessed including resizing and normalization before it is forwarded to M-Net model for disease classification.
- Disease Classification Result (M-Net Model to Backend Server): The M-Net model takes the image, determines its disease status and returns the classification result (such as “Caterpillars”、“Healthy” etc.) to its backend server.
- Result Display (Backend Server to Web Application): The backend server passes the classification result back to the web application which latter is displayed to the farmer. When the result of the classification is received, the backend server sends it to the web application that ultimately presents this on the farmer.
- External Data (External Data Sources to M-Net Model): Model retraining of the M-Net model is facilitated by the introduction of new data sets and model modifications from external data sources. The model will still be able to maintain high levels of accuracy in diagnosing plant diseases with these updated considerations.

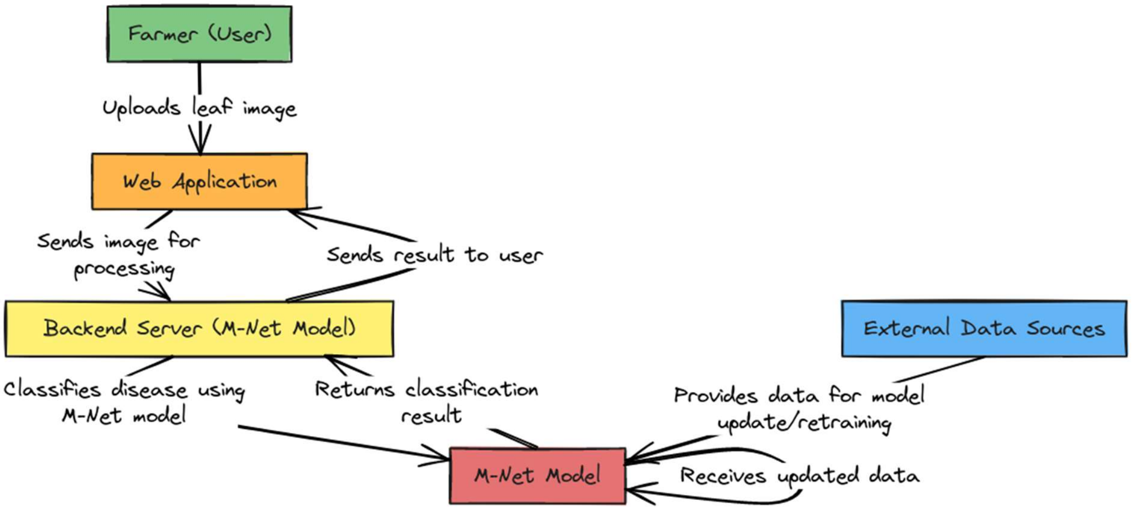


Figure 3.3: DFD Level 1 Diagram

- Farmer: Farmers interact with the system as they upload images that are processed in real-time to produce disease classification results.
- Web Application: This serves as an opening through which farmers upload leaf images and get the classification results.

- Backend Server (M-Net Model): Runs the model, analyzes images and sends classification result to web application.
- External Data Sources: Submits new raw datasets and presents model improvements so as to maintain system's accuracy high on a continuous basis.

At a high level, Level 1 Data Flow Diagram depicts the Guava Leaf Disease Detection System's workflow, which describes how it processes farmer input, creates images, outputs the classifications for the disease and uses outside data for continuous model updates.

3.1.6 UI Design

UI design of the Guava Leaf Disease Detection System is a determining factor in the system being accessible, user-friendly as well as user-friendly to farmers, especially with little technical knowledge. The goal is to create an application which is intuitive and user friendly for farmers of various levels of technical knowledge. We will outline the main elements of the UI design explaining how every page fulfills the particular demands of people who use the system.

1. Initial UI / Landing Page
When the users come to the web application, the landing page is the first interface they interact with. It should be easily understandable in order to explain the purpose of the system and provide an instruction manual step-by-step for how to operate it.
Header:

- The concept of a motto, "Guava Leaf Disease Detection," is illustrated with a leaf icon that encapsulates the high agricultural emphasis of the app. It is essential for creating a friendly environment and rapidly defining the main goal of the app.

Main Action Button:

- A dominant, readily visible "Browse files" button constitutes the centerpiece of the page. Users are led to start classification by pressing the conspicuous "Browse files" button where they can upload an appropriate image.

Instructions:

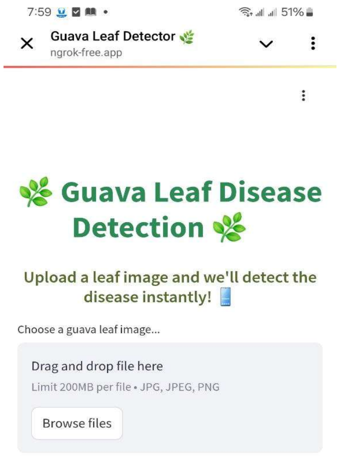
- Plain-how-to instructions guide users through the process, such as "Just upload a leaf image, and we'll recognize the disease immediately". This makes the system easy for farmers who have little experience with technology as it requires brief instructions.

Animation:

- There is a play element in the form of an animation with a running figure at the top right corner while the app is loading showing it is working on your request. Such an animation is important because it provides some comfort that the application is listening to the user's input.

Mobile-First Design:

- The landing page focuses on mobile responsiveness in order to align the layout for mobile devices smoothly. This practice is necessary because many farmers use smartphones, which restrict them to devices without desktop features.



🌿 Guava Leaf Disease Detection 🌿

Upload a leaf image and we'll detect the disease instantly! 📱

Choose a guava leaf image...

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Figure 3.4: Landing Page

2. Image Upload Screen

When one chooses “Browse files,” the user gets to the upload screen where he/she can choose a picture of the guava leaf for analysis..

File Upload Functionality:

- Users can upload an image in the app by either browsing their file system or choosing an image from already saved photos. This ability allows users that have their files stored in different capacities, and uploads images with ease for anyone.

File Constraints:

- The system has a maximum file size of 200MB for images and those formats which it accepts are JPG, JPEG, and PNG. This information is shown prominently in the user interface to avoid compatibility problems with image files.

Preview of Uploaded Image:

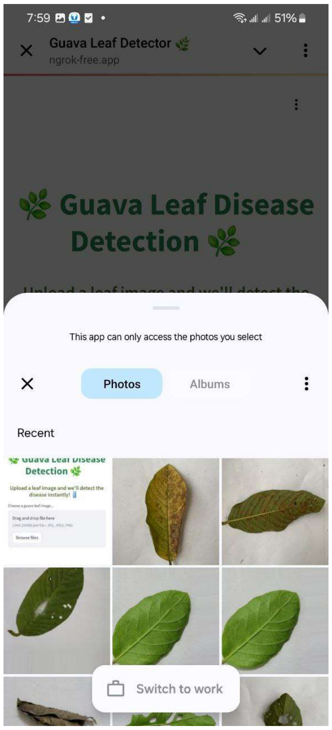
- Once the image is selected, a preview is displayed to allow the user to verify that the correct image has been uploaded. This feature provides a sense of security and transparency for the user, reducing errors in the upload process.

Clear Instructions:

- Short, friendly instructions guide users on how to upload images clearly and avoid common mistakes, such as ensuring the leaf is clearly visible and properly centered in the image.

Mobile Compatibility:

- The design adapts seamlessly to mobile devices, with large, touch-friendly buttons that are easy for farmers to tap, even if they are not familiar with advanced tech interfaces.



This app can only access the photos you select

Figure 3.5: Image Upload Screen

3. Image Processing (Prediction Phase)

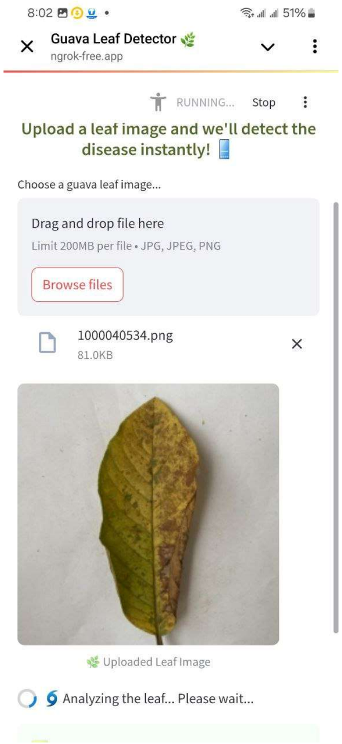
After the image is successfully uploaded, the system begins analyzing the leaf to detect the disease. This phase must be communicated clearly to the user, ensuring they understand the system is actively working.

Processing Indicators:

- A loading animation or spinning icon (such as a circular progress indicator) appears to show the user that the image is being processed.
- Additionally, a message like "Analyzing the leaf... Please wait" appears, indicating the system is in the prediction phase. This message reassures the farmer that the system is working on classifying the disease.

Real-Time Feedback:

- The web application provides real-time feedback during the analysis, which keeps the user informed about the status of their request. A smooth transition between steps enhances the user experience, ensuring they are not left wondering about the progress.



Upload a leaf image and we'll detect the disease instantly!

Choose a guava leaf image...

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

Uploaded Leaf Image

Analyzing the leaf... Please wait...

Figure 3.6: Image Processing Page

4. Disease Classification Result Screen

Once the image has been processed, the system displays the disease classification result. This is the final output the user is waiting for, so it must be presented clearly and accurately.

Disease Name Display:

- The disease name (e.g., Yellow Spot, Healthy) is displayed prominently on the result screen, often with a green check mark to indicate a successful classification.

Image Display:

- The uploaded image remains visible, showing the leaf along with the detected disease to provide context. This visual representation reinforces the classification and ensures the user understands which leaf was classified.

Disease Description:

- Optionally, a brief description of the disease can be shown, including symptoms, possible causes, and suggested treatment or prevention methods. This feature can add a lot of value to the app, and its inclusion in subsequent updates might just propel the app's total value into the stratosphere.

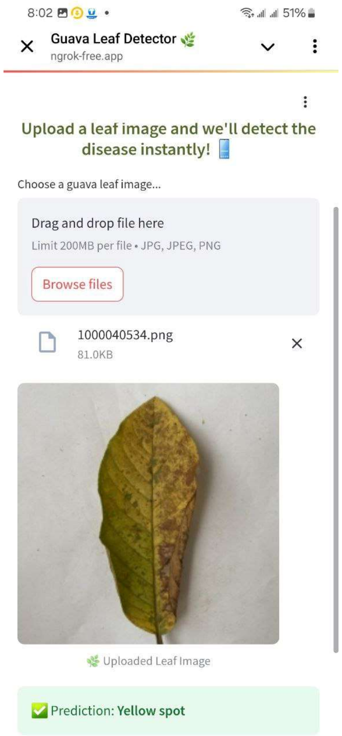
Actionable Next Steps:

- The system may contain such features as "Upload Another Image" and "Get Treatment Suggestions" to assist farmers get rid of the diseases identified.

Mobile-Friendly Layout:

- The design makes ensures that all results can be visible easily on small mobile screens and buttons are easily tappable without zooming in anywhere. Also, the design is consistent and simple on all devices.

The interface of the Guava Leaf Disease Detection System is designed emphasizing simplicity, ease of use, and mobile compatibility. This design makes it easy for farmers whether they are technical or not to maneuver through the system, upload pictures, and receive timely information on different diseases. Efficient instructions, easy to use interface, and easy navigation allows the app to be relevant among the farmers in regions where there is minimal introduction to digital concepts.



Guava Leaf Detector 🌿
ngrok-free.app

Upload a leaf image and we'll detect the disease instantly! 📱

Choose a guava leaf image...

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

1000040534.png
81.0KB

🌿 Uploaded Leaf Image

✅ Prediction: **Yellow spot**

Figure 3.7: Web App Result Page

With its emphasis on real time disease detection and point and view a user interface, the system enables farmers to trust a tool they may rely on for making informed crop decision.

3.2 Detailed Methodology and Design

The Guava Leaf Disease Detection System was developed to provide farmers with a real-time, mobile-friendly application for detecting issues with their guava plants. From the two guava yards in Ashulia, Savar, Dhaka, the dataset consisted of photographs of seven different types of diseases: Caterpillars, Cutting Weevil, Die Back, Healthy, Mealybug Pests, Red Rust and Yellow Spot. The initial data consisting of 1050 images was expanded to 8400 after using data augmentation tools including rotation, shearing and splitting. The dataset was divided into training, validation and testing sets following its resizing, normalization and augmentation.

Using transfer learning, five CNN-pre-trained architectures, were applied to develop a robust model viz., VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2. Despite their good performance, these models needed heavy computational requirements. Thus, we proposed M-Net – a lightweight CNN approach that does not compromise accuracy but enhances computational efficiency. M-Net exhibited accuracy of 99.04% thereby outperforming the pretrained models and was hence easily deployable on mobile devices.

XAI techniques like SHAP and Grad-CAM were embedded in the model to increase its transparency in prediction, although not evident in the web app. Developed with Flask, the application was customized for a mobile-friendly version allowing farmers to upload leaf images and get immediate leaf disease classification using the M-Net model. Improvement in training efficiency and outcomes resulted from including the implementation of learning rate scheduling, early stopping, and checkpointing methods while developing the model. The system was designed to run on a cloud server to facilitate scalability where the farmers could get access from any part. The user testing suggested that the app functioned well on different devices, especially with emphasis on compatibility with mobile access. Other additional features may include the provision of disease management tips and offline use to make the system more usable.

3.2.1 Dataset

3.2.1.1 Data Collection

Guava Leaf Disease Detection System dataset was collected from two guava orchards located in Ashulia, Savar, Dhaka, to capture images depicting various forms of diseases and normal leaves in real-life agricultural environments. The first data collection yielded 1050 images, which were evenly split among the seven categories of diseases, each receiving 150 images. The use of high-resolution smart phones to capture images was implemented to ensure the needed quality and focus to allow for reliable detection of diseases. The devices that were used for the purpose of capturing images included:

Table 3.1: Device Specifications

Device	Resolution	Camera Specifications
Samsung Galaxy S21+	3024x4032	150 dpi, 24-bit color depth, f/1.8 aperture
iPhone 11	3024x4032	72 dpi, 24-bit color depth, f/1.8 aperture

These smartphones were selected due to their high-quality cameras, which allowed for clear and detailed images of the guava leaves, capturing even subtle symptoms of the diseases. High-resolution 3024x4032-pixel images gave the clear images for analysis, and the f/1.8 aperture gave superior image quality in a wide range of lighting, as expected in agricultural settings. All images were carefully filtered and verified post-collection, by an experienced professional who ensured that each image was correctly identified as one of the disease types or a healthy plant. It was very important to ensure every image being sorted correctly for without such validation step, integrity of the dataset would be compromised.

3.2.1.2 Dataset Descrption

The training dataset for the Guava Leaf Disease Detection System consists of six classes representing various diseases and the normal condition of guava leaves. The classes are:

- Caterpillars
- Cutting Weevil
- Die Back
- Healthy
- Mealybug Pests
- Red Rust
- Yellow Spot

Caterpillars

Cutting Weevil

Die Back

Healthy

Mealybug Pests

Red Rust

Yellow Spot

Figure 3.8: Images of Seven Classes

Initially, the dataset covered 1050 images, i.e. 150 images per class and equal number of images across all disease categories. The original set of images was critical in building the first model and guiding the system to be able to understand differences of guava leaf disease. All images were thoroughly scrutinized by an expert who examined the dataset manually to ensure each image was correctly classified for each of the respective disease class. Manual validation stopped allowing the incorrectly labeled images to influence the model which was trained with high quality, correctly labeled data. In the table below, a representation of the original distribution of images per class is as follows:

Table 3.2: Image frequency of classes

Class	Number of Images
Caterpillars	150
Cutting Weevil	150
Die Back	150
Healthy	150
Mealybug Pests	150
Red Rust	150
Yellow Spot	150

By holding an even distribution, we ensured that the model was trained using equal representation from all disease classes, hence avoiding any form of bias. Through manual classification of images, the expert significantly improved the accuracy of the dataset, which ensured necessary labeling that directly empowered the model to identify and classify guava leaf diseases.

3.2.2 Data Preprocessing

Data preprocessing is the requirement of converting the dataset into the format which is required for training machine learning model. The main purpose of data preprocessing is to convert the raw image data into the format which can be used for the Guava Leaf Disease Detection System. Following this procedure, the dataset gets processed standardizing, enriching and diversifying it, which ultimately contributes to the improved performance output of the model.

1. Image Resizing

Due to the fact that not all collected images were obtained in the same manner, such differences may negatively affect the performance of the model. To provide uniform input dimensions for neural network, each image was resized in the same way into the standard size of 224X224 pixels. Through a standardization of the images' size, the model was better equipped at processing data, with a better feature extraction and quick training time.

2. Normalization

A consistent input range for inputs used in machine learning is important for effective training convergence of any model. Each pixel value of the images was normalized between 0 and 1 by normalizing them through their division by 255. The data was rescaled from a 0-255 scale to a floating-point spectrum between 0 and 1, by normalising the pixel values. Normalization maintains that important featues will not overwhelm features of lesser magnitudes during learning, making it easier for the model to equally work on the data.

3. Data Splitting

To allow the model to train, validate and test successfully, the dataset was separated into three distinct parts; training, validation, and test sets:

- Training Set (70%): Most of the images 70% or 5880 were scoped for the training of the model. With the help of this part, the model was able to draw important characteristics for each category of diseases.
- Validation Set (20%): 20% of the images (1680 images) was used for validation while the model is being trained. The validation set was used to tune hyperparameters and monitor the model's performance on data that it had not yet seen.
- Test Set (10%): The final 10% of images (840) were left over for final evaluation purposes, providing an independent evaluation of the model's potential to perform well with unseen data.

The data splitting ensured that the model was trained on a sufficient amount of data, while still allowing for reliable evaluation during and after the training process.

4. Label Encoding

Since the dataset consisted of images belonging to different disease classes, it was necessary to convert the categorical labels (disease classes) into a format that could be used by the model. Label encoding was applied to convert the disease names (e.g., "Caterpillars", "Healthy", etc.) into numeric values. A unique integer was used for each disease class; this simplified the neural network training process to a degree.

3.2.3 Model Selection

Model selection process plays an important role in the success of machine learning project development. In order to design the Guava leaf disease detection system, we needed a model that was efficient at processing image data and able to correctly classify guava leaf diseases. CNNs are most appropriate for image classification as they are able to automatically process visual inputs, and they continue to record a good performance in image recognition competitions.

To achieve high performance and fast inference, we started out with transfer learning using pretrained CNN models. We chose these models because of their experience in training on huge datasets and were able to generalize to new applications such as disease detection in guava leaves. The main purpose of using pretrained models was to reuse the features they have already learned from large data sets, and to save time to allow the model to converge on disease detection in guava leaves.

After the pretrained models were assessed, a hyper-efficient CNN model known as M-Net was developed to further improve the system's efficiency in a real-time mobile environment. M-Net's design is aimed at implementing efficient processing as well as accurate results, that makes the implementation on devices of limited hardware resources possible, including smartphones.

3.2.3.1 Pretrained CNN Models Employed

1. VGG19

As one of the most widely employed CNN models, VGG19 finds a lot of application in images' classification. VGG19 has been proposed by the Visual Geometry Group based at the University of Oxford and consists of 19 layers, 16 of them are convolution layers using 3x3 filters. Despite its proficiency in detecting fine details in images, VGG19 consumes too many parameters to be used in mobile devices without a few optimizations. The first layers of the model start with a 64-channel convolutional filter and it is continued with a max-pooling, before other layers have increased sizes of their filters (128, 256, 512). Notably, even under high computational requirements, VGG19 becomes a powerful model for feature extraction, when set with pre-computed weights from large-scale datasets such as ImageNet.

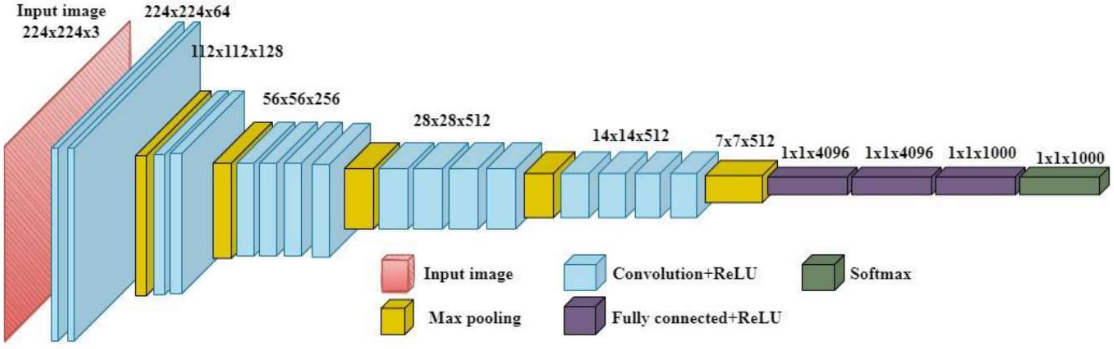


Figure 3.9: Architecture of VGG19 (Online)

2. DenseNet201

DenseNet201 utilizes a rather novel architecture in which every layer has access to all of its predecessors in a feed-forward way thus enabling better reuse of features. The dense connections increase gradient flow and allow more effective parameter usage. The DenseNet201 mode trained using 201 layers performs superior results in assigning images to classes while significantly fewer number of parameters are used than is the with classical CNN modes. While applying batch normalization and ReLU activations helps to create stable training, the complexity of the model imposes high computational costs, and thus obligatory optimization to use on devices with limited resources.

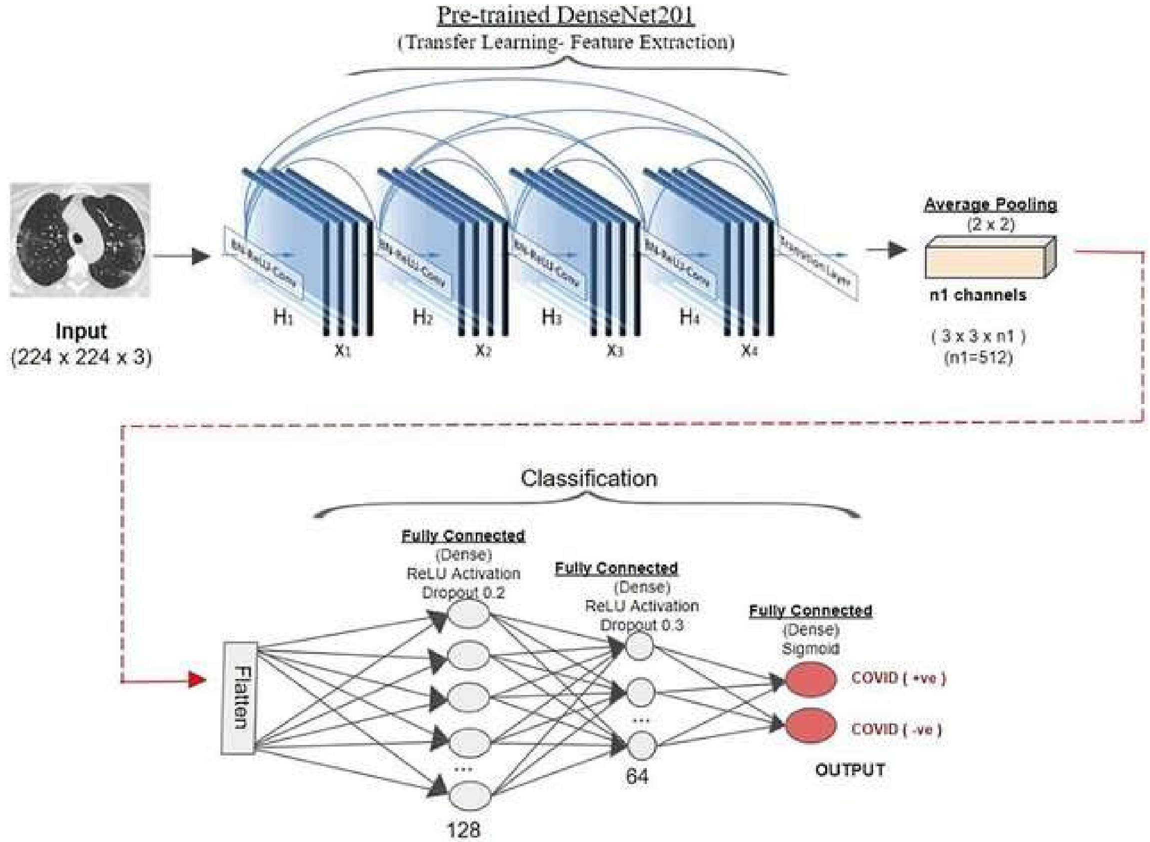


Figure 3.10: Architecture of DenseNet201 (Online)

3. InceptionV3

The InceptionV3 has a deep CNN architecture that combines various convolutional layers through use of filters of the sizes 1x1, 3x3, and 5x5 to improve multi scale feature extraction. Using factorized and asymmetric convolutions, it outperforms previous Inception models with improved performance and efficiency. Efficiency is a key factor especially with fewer parameters, where inceptionv3 outperform other models such as Vgg19, while maintaining strong performance for classification. However, the additional model size may not be implementable on mobile devices in practice, specifically in real-time situations, without further refinement.

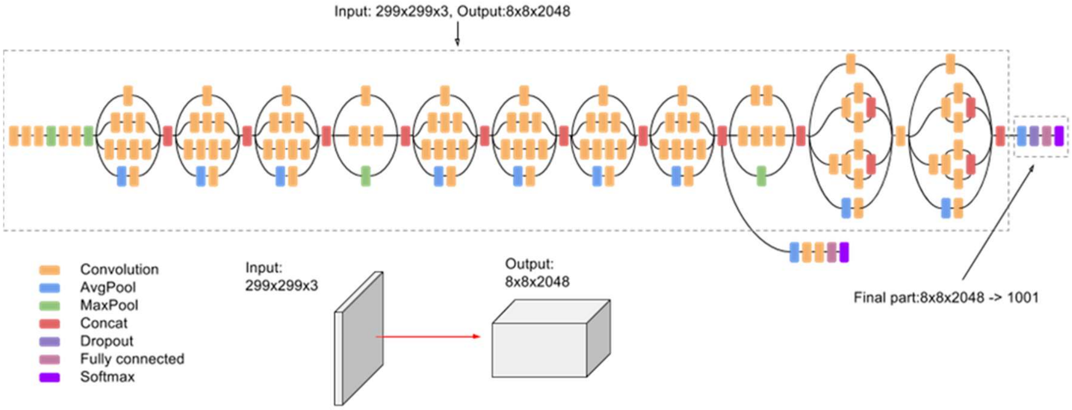


Figure 3.11: Architecture of InceptionV3 (Online)

4. ResNet152V2

As an amendment to ResNet, the ResNet152V2 avoids the vanishing gradient problem by introducing skip connections that help to maintain gradient propagation effectively over complicated multilayered configurations. With its 152 layers, ResNet152V2 is well-armed to learn complex features and be high performing on a challenging task. To increase training stability, and to counter overfitting, the model includes batch normalization, and ReLU activations. The outstanding performance of ResNet152V2 is not exempt from high amounts of parameters in the system that require enormous computational resources and may require more optimisations for real time use in mobile applications.

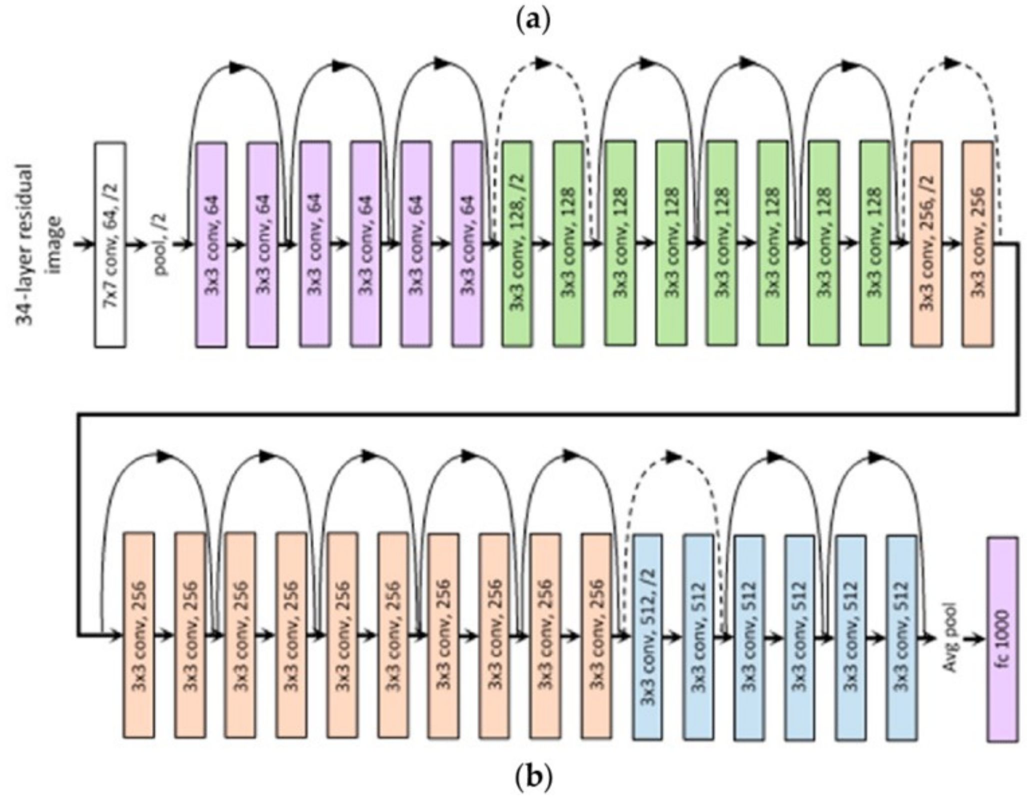


Figure 3.12: Architecture of ResNet152V2 (Online)

5. MobileNetV2

MobileNetV2 is a lightweight CNN architecture that is constructed for convenient implementation on mobile devices and embedded systems, with an emphasis on resource executive utilization of such settings. Using depthwise separable convolutions, which distributes the convolution task into different steps, the mobile net v2 achieves great savings in computational complexity. The addition of inverted residuals and linear bottleneck layers allows MobileNetV2 to improve efficiency at the expense of minimal loss in accuracy. The sleek architecture of MobileNetV2 enables it to outshine traditional real-time mobile applications even if it is inferior on accuracy versus deeper networks as applied to complex problems. Despite this, MobileNetV2 does remain an effective option for mobile disease-detection systems, producing a good trade-off between efficiency and effectiveness.

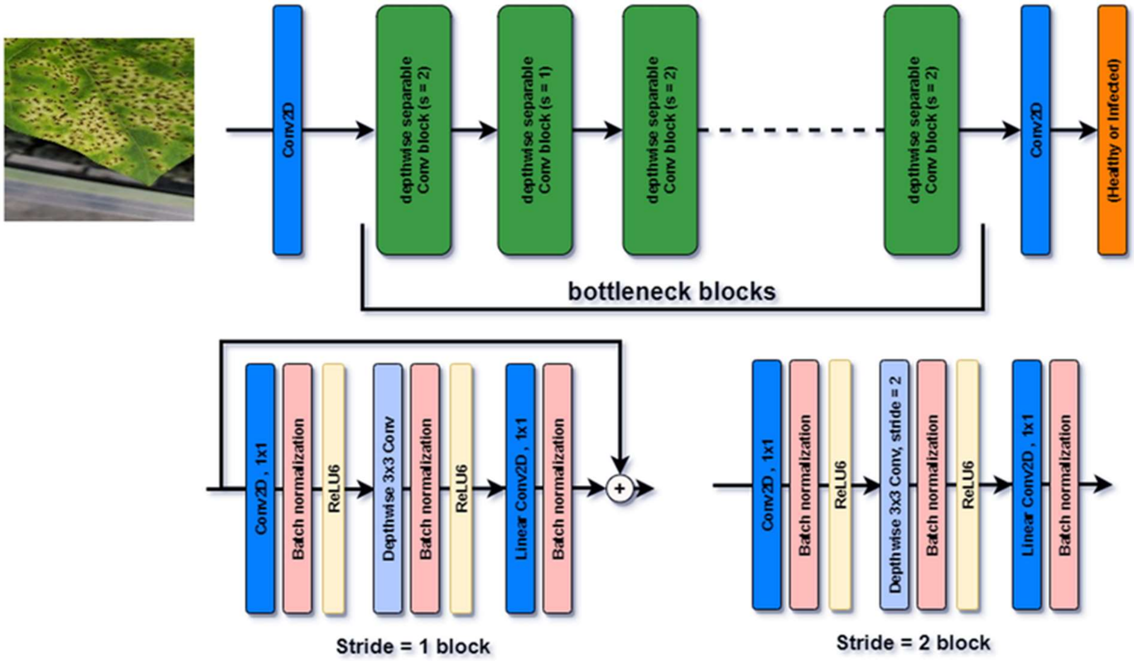
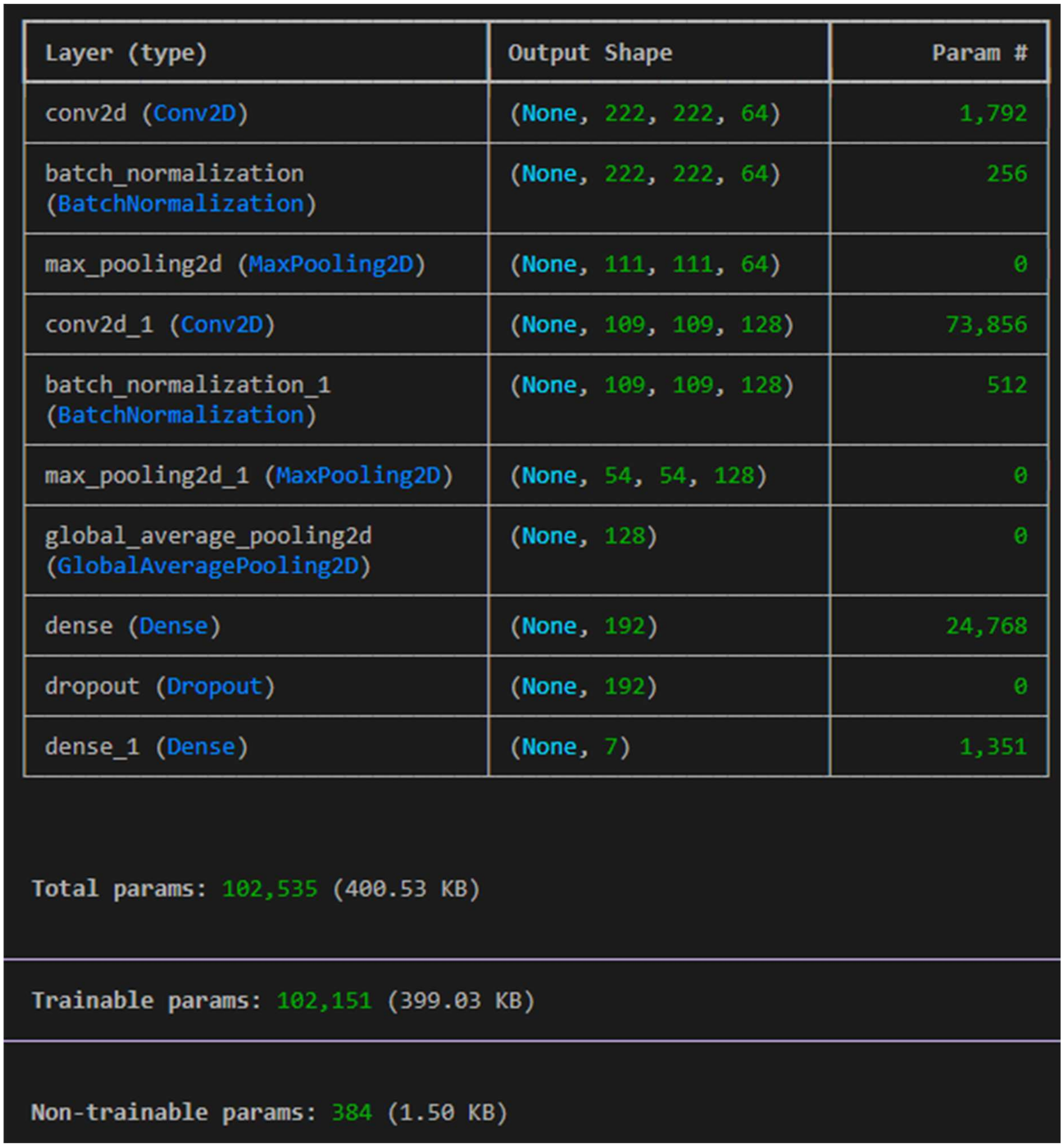


Figure 3.13: Architecture of MobileNetV2 (Online)

3.2.3.2 Proposed M-net Model

M-Net is a CNN that was specially designed to improve real-time disease detection with minimal computational needs. In order to attain accurate guava leaf disease classification and also take into consideration for the network to be run efficiently on all mobile devices including smartphones as well as low resources platforms, M-Net was developed. The efficiency of M-Net makes it perfectly suitable for farmers to perform real-time disease monitoring and diagnosis themselves on their smartphones or handheld devices.

The first image below demonstrates the lightweight nature of the M-Net model by illustrating its trainable parameters and model complexity. In comparison to heavier models like ResNet152V2 or VGG19, M-Net's 102,535 trainable parameters (as shown in the image) represent a much smaller network, making it more computationally efficient and faster in terms of both training and inference.



```
┃ Layer (type)                     ┃ Output Shape               ┃          Param # ┃
┃ conv2d (Conv2D)                  │ (None, 222, 222, 64)       │            1,792 │
│ batch_normalization              │ (None, 222, 222, 64)       │              256 │
│ (BatchNormalization)             │                            │                  │
│ max_pooling2d (MaxPooling2D)     │ (None, 111, 111, 64)       │                0 │
│ conv2d_1 (Conv2D)                │ (None, 109, 109, 128)      │           73,856 │
│ batch_normalization_1            │ (None, 109, 109, 128)      │              512 │
│ (BatchNormalization)             │                            │                  │
│ max_pooling2d_1 (MaxPooling2D)   │ (None, 54, 54, 128)        │                0 │
│ global_average_pooling2d         │ (None, 128)                │                0 │
│ (GlobalAveragePooling2D)         │                            │                  │
│ dense (Dense)                    │ (None, 192)                │           24,768 │
│ dropout (Dropout)                │ (None, 192)                │                0 │
│ dense_1 (Dense)                  │ (None, 7)                  │            1,351 │

Total params: 102,535 (400.53 KB)

Trainable params: 102,151 (399.03 KB)

Non-trainable params: 384 (1.50 KB)
```

Figure 3.14: M-Net Model Summary

Key Components of the M-Net Architecture

The architecture of M-Net consists of several key layers, each designed to handle the different stages of image processing and disease classification efficiently. The goal was to build a network that is not only accurate but also lightweight and capable of running on resource-constrained devices such as smartphones. Here's a breakdown of the architecture components used in the M-Net model:

Convolutional Layers (Conv2D):

- The initial layers of the M-Net model use 2D convolutional layers (Conv2D) to perform feature extraction. These layers use filters on the input image to ascertain features similar to that of edges, textures, shapes from the image. The use of small filter sizes (3x3) makes it possible for the model to capture the fine print of the guava leaves effectively which is essential in detecting disease symptoms.

- The first Conv2D layer performs the input image with 64 filters and gives (None, 222, 222, 64)output shape, and the second layer also uses 128 filters and provides (None, 109, 109, 128) output shapes.

Batch Normalization:

- Immediately after each instance of convolution, batch normalization is performed. Standardizing the statistics of the convolutional layer outputs makes this technique stabilize training. This technique reduces fluctuations in the input distributions to increase the speed of training by increasing the speed and reliability of the model.

MaxPooling2D:

- Since this step emphasizes the salient information extracted by the network the image is spatially down-sized. By performing max pooling, the model reduces the complexity of the data, making it more manageable while maintaining essential features for classification. For example, after the first convolutional layer, the image is downsampled to (None, 111, 111, 64) and then to (None, 54, 54, 128) after additional convolutional and pooling operations.

Global Average Pooling2D:

- Instead of using fully connected layers after the convolutional layers (as in traditional CNNs), M-Net uses Global Average Pooling (GAP). This layer computes the average of the entire feature map and reduces it to a single vector. The output shape of this layer is (None, 128). GAP helps reduce the model size and prevents overfitting by performing a global feature extraction instead of relying on dense connections.

Dense Layers:

- After the feature extraction and pooling layers, M-Net has a fully connected (dense) layer with 192 units. This layer processes the features and prepares the data for the final classification. This dense layer enables the model to classify images based on the features extracted during the earlier convolutional and pooling stages.

Dropout:

- To prevent overfitting, a Dropout layer is included after the dense layer. During training, dropout randomly sets a fraction of the input units to 0, forcing the network to learn robust features that are less likely to overfit to the training data. By acting like this, the model is no longer so reliant on particular neurons which allows the model to be more generalizable to new data.

Final Dense Layer (Output Layer):

- The top layer is a dense layer of 7 units, each unit representing one of the 7 disease classes (including the healthy leaf). Each neuron of the output layer represents one disease class, and model uses softmax function to find the greatest probability class. By using this output layer, the model can produce a class prediction result for each of the input images, whereby each of the captured images undergoes the determination to be a picture of a specific disease or a healthy leaf.

The image below gives a diagrammatic representation of the proposed M-Net model structure. Looking at the diagram, it can be seen how data progresses from the input image, to a series of convolutional layers, followed by batch normalization, max-pooling, then global average pooling, with the ending being dense layer. Using depthwise separable convolutions and global average pooling, M-Net manages to be computationally efficient having a lower parameter count than traditional CNNs.

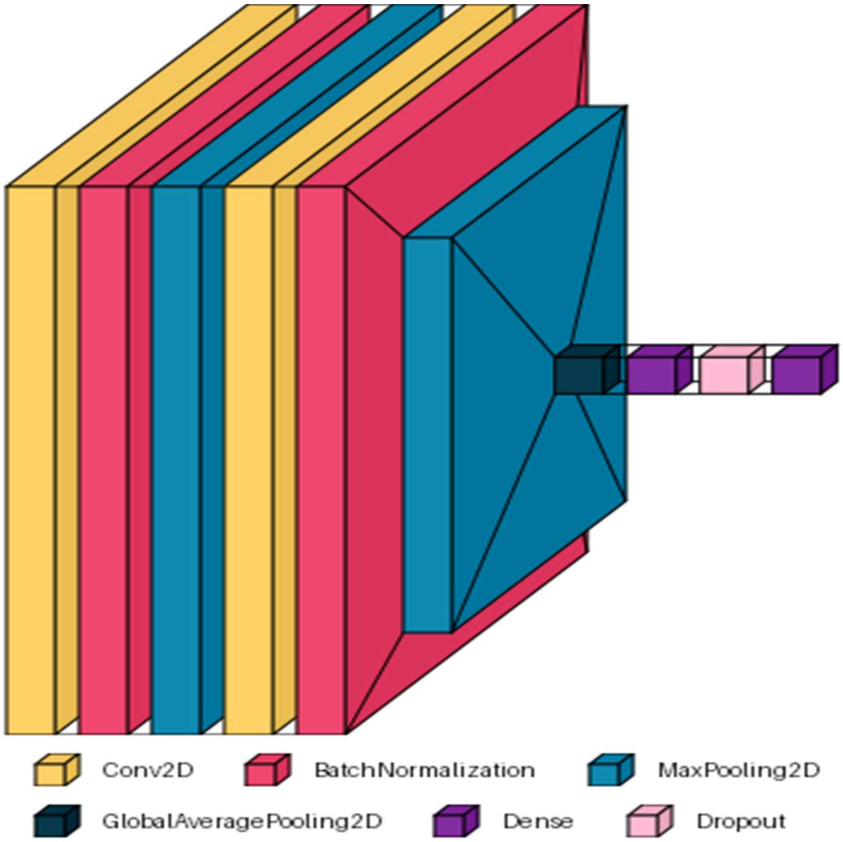


Figure 3.15: Architecture of Proposed M-Net

3.2.4 Explainable AI (XAI)

Explainable Artificial Intelligence (XAI) XAI is an important factor toward developing trust and transparency in machine learning models when they are applied in real practical lives. Famous for the accuracy that it has in detecting diseases, M-Net is a deep learning model that uses transparent explanations about its predictions to earn trust from the farmers and the end users. In agricultural settings, the value of model predictions means that the reliability of the decision-making process of the system is essential. In order to improve the interpretability and transparency of the model, we introduced two respected XAI methods into the M-Net model: SHAP (SHapley Additive exPlanations) and Grad-CAM (Gradient-weighted Class Activation Mapping). By applying such approaches, the model's reasoning becomes visible for users and they are able to understand why in some images there are certain diseases detected.

1. SHAP (SHapley Additive exPlanations):
SHAP (SHapley Additive exPlanations) is one of the popular methods that gives each feature (e.g. a pixel, in the case of a picture) a measure of how it contributes to the model's prediction. Through SHAP, users will be able to develop a quantitative perspective of contribution of each pixel in the image of a guava leaf towards the model prediction.

- How SHAP Works:
  - Shapley value theory (adopted from game theory) is underlying SHAP values, giving each feature (each pixel or region in an image) a score based on the feature's influence on the final decision.
  - For each prediction, SHAP computes how the presence or absence of each pixel influences the final output. This results in a global and local explanation:
    - Global explanation shows which features are generally the most influential across all predictions.
    - Local explanation shows the impact of specific features for individual predictions.
- SHAP in M-Net:
  - For images fed through the M-Net model, SHAP values were provided to these images, pointing out which form of the leaf (e.g., color variations, lesions, or geometric patterns) was critical in classifying them into a disease class (e.g., "Yellow Spot", or "Healthy").
  - Using this approach, not only were the accurate classifications ascertained, but also the precise areas of the leaf that contributed most to the decision were revealed. It, therefore, improves the transparency of predictions that can be yielded by the M-Net model.

2. Grad-CAM (Gradient-weighted Class Activation Mapping):
Grad-CAM is commonly used as XAI model that produces heat maps in order to clarify why certain parts of the image became so significant in the model's decision. While SHAP provides a numerical attribution to feature importance, Grad-CAM outputs a visual map, which makes it easier to interpret how the model generates its predictions.

- How Grad-CAM Works:
  - Grad-CAM works by computing the gradients, of gradient in relation to the predicted class with regard to the convolutional layers of the network. The gradients are used as weights for the feature maps, and when these weighted maps are combined then the resultant heatmap is generated.
  - Putting a heatmap on the top of the original image shows which area contributed most to the model's prediction. Regions in the heatmap which show a redder color emphasize areas in the image, which were

most relevant for establishing the prediction.

- Grad-CAM in M-Net:
    - In the M-Net model, heatmaps for the images of the guava leaf were created using the Grad-CAM method, highlighting those areas that were most relevant to a particular disease in the image. As an example, once the model recognizes that Red Rust is present on a leaf, Grad-CAM will generate a heatmap, that emphasizes upon the areas most affected by the disease.
    - These heatmaps assisted the users in visually explaining the model's reasoning and identifying where exactly on the leaf the disease symptoms were demonstrated. This is a very beneficial feature for farmers since they can directly examine the affected parts of their leaves.

3.3 Project Plan

Systematic approach is used in the development of the Guava Leaf Disease Detection System with emphasis on key phases for easy implementation. This part outlines the major stages, tasks and timelines, which were used to develop the whole project.

1. Initial Research and Literature Review (1 Month)
   Objective: Get an understanding of current methods of plant disease detection with special emphasis on classification of guava leaf disease.
   Tasks: Analyze work on academic research for deep learning, methods of convolutional neural networks (CNN), transfer learning, and XAI approaches.
   Outcome: Determinedly aware that lightweight models and XAI were essential with regard to increasing interpretability in the system.
2. Data Collection and Preparation (2 Months)
   Objective: Gather and prepped a series of pictures for training and testing.
   Tasks: Collect images from Ashulia, cross-verify validation with an expert, resize, normalization, augmentation, and have the data distributed.
   Outcome: We generated a data-set of 8400 images, provide 70% for training, 20% for validation, and 10% for testing.
3. Model Development and Selection (3 Months)
   Objective: Develop the model of the disease detection algorithm with CNNs and determine which architectural choice will be the most effective one.
   Tasks: Start from prevalidated CNN models and look where M-Net stands for real time. Change hyperparameter values and add SHAP and Grad-CAM for the better understanding of the model.
   Outcome: Developed M-Net for 99.04% accuracy purposes and used XAI techniques for interpretability.
4. Model Evaluation (1 Month)
   Objective: To ensure accuracy and generalization, evaluate the model on the test set

Tasks: To evaluate and fine-tune the model, use metrics like accuracy, precision, recall, and AUC-ROC.
Outcome: Records a final 99.04% accuracy after comprehensive evaluation steps.

5. Web Application Development (2 Months)
   Objective: Develop a mobile responsive web application that should have the ability to detect diseases in real time.
   Tasks: Use Streamlit for application development, integrate the model flawlessly, and ensure mobile responsiveness.
   Outcome: Web application could be successfully deployed making it possible for farmers to upload images and receive instant disease classifications.
6. Testing and Deployment (1 Month)
   Objective: Ensure the application works across devices and deploy it for real-world use.
   Tasks: Perform device testing over an Android and iOS system, gather comments from farmers, and deploy the application on a cloud server for nationwide use.
   Outcome: Released the web application that is easy to use on mobile devices and instant identification of leaf diseases.
7. Future Work and System Improvement (Ongoing)
   Objective: Plan future improvements and enhancements.
   Tasks: Expand the set of identified diseases, include recommendations for treating diseases, include offline possibility, and continue improvements to a model.
   Outcome: The application will evolve to become a fully-fledged solution for plant diseases management.

This Project Plan ensures a structured and timely development of Guava Leaf Disease Detection System that provides the farmers with a convenient and efficient mechanism for controlling of their crops. Following this systematic approach, the project has created a light-weight, reliable model and introduced an intuitive web application.

3.4 Task Allocation

I am the sole contributor to the Guava Leaf Disease Detection System, and I made sure all the processes of the project were followed from research to data collection, model creation, and deployment, and final report preparation. This section summarizes the main contributions I made in the development of the Guava Leaf Disease Detection System:

1. Initial Research and Literature Review
   - Examined the existing methods of plant disease detection based on machine learning and deep learning.

   - Studied Convolutional Neural Networks (CNN) for image classifying , and identified appropriate Explainable AI (XAI)methods for increasing model transparency.
   - Analyzed current systems & identified the most suitable methodology for developing the Guava Leaf Disease Detection System..
2. Data Collection and Validation
   - Acquired 1050 images of the leaves of guava collected from Ashulia, Savar, Dhaka, representing seven classes of the disease..
   - Reduced the dataset to only include images showing clear and detailed views of the leaf to ensure high quality images possessed..
   - Worked with an expert to review/verify the images ensuring that correct disease identification was made.
3. Data Preprocessing
   - Normalized and resized the images to a typical size of 224x224 pixels for CNN model uniformism.
   - Implemented data augmentation so as to ensure augmentation of the dataset and increasing its size.
   - The dataset were split into the training (70%), validation (20%, respectively), and test (10%) data sets.
   - Applied label encoding to encode disease classifications as text to numeric values that can be used in training.
4. Model Development and Training
   - Embedded transfer learning atop multiple CNN models pre-trained VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2 learning from the feature experience acquired training on large-scale ImageNet.
   - Designed a concise CNN paradigm for mobile usage, M-Net, which could straightaway diagnose diseases.
   - Tweaked and tuned hyperparameters in order to increase the general performance of trained models.
   - To enhance transparency and interpretability, applying Explainable AI (XAI) methods such as SHAP and Grad-CAM.
5. Model Evaluation
   - Evaluated the accuracy, precision, recall, F1 score, and AUC-ROC curves of effective model measurements.
   - Recommended M-Net as the better model based on its accuracy of 99.04% to deploy based on it..
6. Web Application Development
   - Used Streamlit in developing the web application that allows farmers to upload images of guavas and get immediate diagnosis results for the diseases.
   - Made the application mobile responsive with priority in order to make it easy to use by farmers on smartphones.
   - For fast inference and efficient image processing, integrated the trained model into the backend.

7. Testing and Deployment
    - Conducted device test to ensure that the web application could run flawlessly both on Android and iOS machinery.
    - Hosted the web application on the cloud server to ensure farmers across the globe have prompt access to it remotely through the internet.
8. Documentation and Final Report
    - Developed the full thesis report, with extensive discussion regarding the methodology, model development, experimental results, and concluding remarks.
    - Traced each of the steps of the project, from collecting data, preparing it for modelling, to including XAI and deploying the final system..
    - Prepared the final presentation to submit for academic evaluation.

As the sole contributor, I implemented all the parts of the Guava Leaf Disease Detection System without anyone's help. I have proved my abilities to design, develop, and deploy a machine learning-based solution for real-time disease identification in agro environments by finishing the project.

3.5 Summary

The Guava Leaf Disease Detection System strives to assist farmers by coming up with an efficient and in time solution for identifying cases of diseases associated with Guava crop. The system combines CNNs with XAI approaches in making both accurate and interpretable identification of diseases. The team pulled a reliable dataset from the guava farms in Ashulia, Savar, and Dhaka before moving on to preprocess the images in the form of resizing, normalisation and augmentation. Several of these pre-trained models, VGG19, DenseNet201, InceptionV3, ResNet152V2, and MobileNetV2, were evaluated but the custom M-Net model ultimately proved superior in terms of accuracy, combining 99.04% accuracy with real-time usability on mobiles.

To increase transparency, we have integrated XAI methods, including SHAP and Grad CAM, which allow the farmers to recognize the manner in which predictions are made. A web application that was mobile compatible was designed so that farmers could send haystack images and receive a quick identification of the disease. The application was made available through a cloud server, meaning that farmers could use it from any device connected to the internet. In the end, the system provides accurate outcomes, rational decision-making and knowing, and users-friendly mobile functions – the farmers will get efficient crop management skills and AI's importance in their sustainable farming.

Chapter 4

Implementation and Results

The results of the performed experiments are presented in this chapter, which include the result of the pretrained CNN models as well as the performance of the customized M-Net model. The chapter touches on a variety of evaluation metrics, such as accuracy, precision, and recall, and measures how XAI visualization outputs such as SHAP and GradCAM perform. The analysis of the results provides an insight into the performance of the model and where it may be improved.

4.1 Environment Setup

Every experiment in this study depended on Python 3.10 and the TensorFlow and Keras deep learning frameworks. Research was mostly done in Google Colab Pro ecosystem using its GPU acceleration based on NVIDIA Tesla T4 connected GPUs, which optimized the training and evaluation of deep learning models. The models were implemented using a combination of essential machine learning and deep learning libraries, including NumPy, Pandas, Matplotlib, scikit-learn, OpenCV, and imbalanced-learn (for general dataset handling). This project adheres to internationally recognized engineering standards to ensure compatibility, robustness, data privacy, and practical feasibility across both software and hardware components. Table 4.1 summarizes the key software and libraries used in the experimental pipeline:

Table 4.1: Software and Libraries Used

Tool/Library	Purpose
Python 3.10	Core programming language
TensorFlow 2.x	Deep learning framework
Keras	High-level neural network API
NumPy	Numerical operations and array manipulation
Pandas	Data handling and manipulation
Matplotlib, Seaborn	Visualization and plotting
OpenCV	Image preprocessing and manipulation
imbalanced-learn	Data preprocessing and general handling
SHAP, Grad-CAM	Explainable AI tools
Streamlit	Web interface for initial model deployment

Ngrok	Secure tunneling for local server access

The dataset, collected from two guava yards in Ashulia, Savar, Dhaka, was preprocessed by resizing all images to a uniform dimension of 224×224 pixels, followed by normalization and data augmentation techniques such as rotation, flipping, and shearing to enhance model generalization. The dataset was manually balanced, so there was no need for techniques like SMOTE. It was split into training, validation, and test sets using a 70:20:10 ratio, respectively. Table 4.2 illustrates the training parame

Table 4.2: Model Training Parameters

Parameter	Value
Input Image Size	224 × 224 × 3
Batch Size	32
Epochs	50
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Crossentropy
Metrics	Accuracy, Precision, Recall, F1-Score, AUC-ROC

To optimize model training and prevent overfitting, callback functions such as EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint were employed. Additionally, a learning rate scheduler was used to dynamically adjust the learning rate during training, helping the model converge more effectively. These callbacks monitored performance metrics and automatically adjusted learning rates or stopped training based on validation loss trends to avoid overfitting. After training, the best-performing model was selected for explainability analysis using Grad-CAM and SHAP to visualize the regions of the image that contributed to the model's decision. The final model was exported in .h5 format for deployment. Testing was performed using the Streamlit-based web application, where the model saved as a .h5 file was used to test real-time predictions.

4.2 Testing and Evaluation

The Testing and Evaluation phase of the Guava Leaf Disease Detection System aimed at quantifying the performance of the trained model using different measures of evaluation and visualization tools. The goal was to find out whether the model was capable of performing well in classifying guava leaf disease and whether the model was capable of dealing with unseen data in a robust manner. The central scoring metrics include Accuracy, Precision, Recall, F1-score, Confusion Matrix, Loss and Accuracy Curves, and AUC-ROC Curve.

1. Accuracy

Accuracy represents the percentage of correctly classified images, including both diseased and healthy leaves, out of the total images. While it gives a general sense of model performance, accuracy can be misleading when the classes are imbalanced.

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

2. Recall (Sensitivity)

Recall, or sensitivity, measures the proportion of actual positive cases (such as diseased leaves) that the model correctly identifies. A high recall value indicates the model is effective at detecting most disease cases, which is crucial for preventing unnoticed spread in real-world settings.

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

3. Precision

Precision assesses how many of the predicted positive cases (such as diseased leaves) were correctly identified. High precision means the model avoids falsely identifying healthy leaves as diseased, reducing unnecessary actions from farmers or field workers.

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

4. F1-Score

The F1-score combines both precision and recall into one metric, providing a balanced evaluation. It's especially useful in situations like this, where both false positives and false negatives need to be considered, particularly in multiclass leaf disease detection tasks.

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

5. Confusion Matrix

The Confusion Matrix offers a detailed view of the model's performance, showing the true positives, false positives, true negatives, and false negatives for each disease class. It helps visualize misclassifications and highlights areas for improvement. The matrix reveals which classes are being misclassified and to what extent, providing valuable insights for refining the model or gathering additional training data.

6. Loss and Accuracy Curves

The Loss and Accuracy curves were plotted to track the model's performance during training and validation over time. These curves help monitor:

- Overfitting: When the model performs well on training data but poorly on validation data.
- Underfitting: When the model struggles to perform well on both training and validation data.

The Loss curve tracks the error in predictions, while the Accuracy curve measures the model's success rate in making correct classifications.

7. AUC-ROC Curve

The AUC-ROC curve (Area Under the Receiver Operating Characteristic Curve) is a tool for evaluating model performance, especially in multi-class classification problems. It plots the True Positive Rate (Recall) against the False Positive Rate across different threshold values.

- AUC (Area Under the Curve): A higher AUC signifies better performance. An AUC of 1 indicates perfect performance, while an AUC of 0.5 is as good as random guessing.

The AUC-ROC curve was used to evaluate the model's ability to distinguish between diseased and healthy leaves, showing how the model balances true positives and false positives at different thresholds.

4.3 Results and Discussion

In this section, we will analyze experimental results achieved with both transfer learning CNN models (experimental CNN and semantic CNN) and custom M-net model. Firstly, we will review how the pretrained models of the convolutional neural network (CNN) such as VGG19, DenseNet201, InceptionV3, ResNet152V2 and MobileNetV2 perform in distinguishing between guava leaf diseases, in terms of their accuracy and overall efficiency. The performance of these pretrained models was evaluated to measure their generalization performance over the dataset considering the important metrics such as accuracy, precision, recall, and F1-score. We will then discuss the findings of custom M-net model, which was finetuned with a change in hyperparameters and customised for the use on resource-constrained devices. We will do a comparison between M-net and the performance of pretrained models with a focus on its strengths in terms of accuracy and computational efficiency. In addition, we will examine the effects of XAI techniques like Grad-CAM and SHAP showing how they improved the model's interpretability and reliability. We will interpret the results to compare the benefits and weaknesses of the transfer learning models with the M-net, and evaluate its practical feasibility to solve the problem of guava leaf disease identification.

4.3.1 Results of Pretrained CNNs

This study evaluated the pretrained convolutional neural network model performance on the Guava Leaf Disease Detection System for leaf disease classification. Five models were tested: VGG19, DenseNet201, InceptionV3, MobileNetV2, and ResNet152V2. Results on the accuracy for each model on the dataset are shown in Table 4.1.

Table 4.3: Accuracies of Pre-trained CNNs

Model	Accuracy
VGG19	85.97%
DenseNet201	95.14%
InceptionV3	95.20%
MobileNetV2	98.91%
ResNet152V2	98.10%

The data clearly shows that the highest accuracy was achieved by MobileNetV2 in 98.91% among the pretrained models that were: InceptionV3 in 95.20% and DenseNet201 in 95.14%. Even though VGG19 performed worst by attaining an accuracy of only 85.97%, it left many traditional machine learning models in the dust. The surprising performance of MobileNetV2 model, in spite of its low complexity, indicates its applicability in resource constrained scenarios, including smartphones. The findings showcase the performance of pretrained models for compression of training time without compromising reliability.

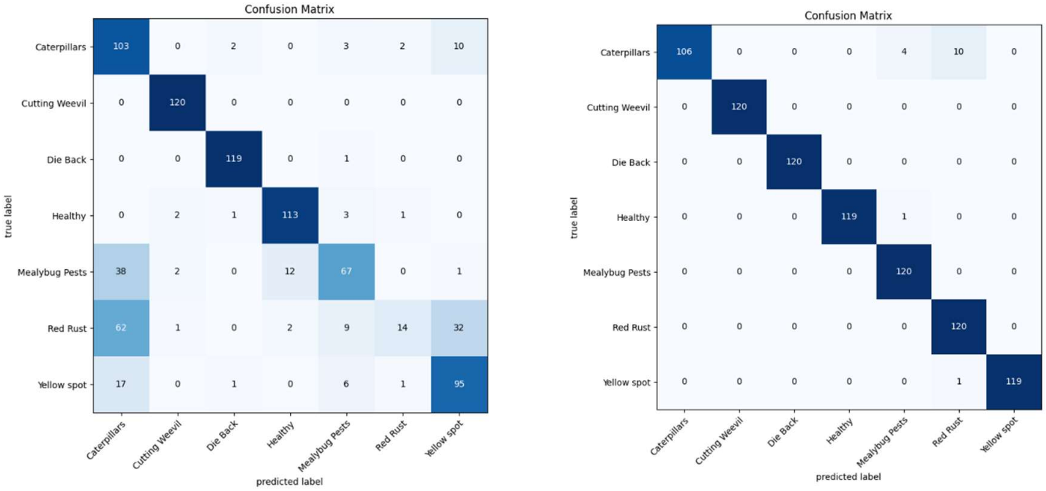
The classification report illustrated in Table 4.6 is an important analysis on the performance of the pretrained CNN models on the Guava Leaf Disease Detection dataset, giving breakdowns of results with disease classes, according to precision, recall, and F1-score. The evaluation shows that MobileNetV2 outperformed the other models for all metrics of overall accuracy, precision, recall, and F1-score. For several classes, such as Cutting Weevil, Die Back, Healthy and Yellow Spot, its precision, recall and F1-Score jumped to the perfect 100%. These results reveal that MobileNetV2 delivered better accuracy, with fewer misidentifications for these diseases. This success has been due to the lightweight nature of the MobileNetV2, and hence, it can deliver quality results using little computational resources, thus making it the model of choice for real-world mobile deployment. Results of these show that ResNet152V2 and InceptionV3 were equally successful, having high values of precision and recall for most part of the classes taken. Both models had very low error rates, with excellent results for Die Back and Healthy, both models producing nearly error-free results. However, MobileNetV2 dramatically outstripped these models in some classes, including Mealybug Pests and Yellow Spot, where MobileNetV2 provided perfect recall and precision. For most disease classes, the F1-scores of ResNet152V2 and InceptionV3 were both above 0.95 which translated to assertive reliability of the model in distinguishing guava leaf diseases. However, obviously, the VGG19 model underperformed relative to all the other pretrained models in our study. The model had substantially lower precision for Caterpillars (0.47) and Mealybug Pests (0.75) compared to other classes. The fact that Caterpillars had a low F1-score of 0.61 means that the model was not consistent in categorizing this class, probably since the catapillars architecture was less complex than those in architectures used models such as MobileNetV2 and ResNet152V2. However, VGG19 was successful in, strong recall rates for numerous classes particularly Healthy and Die Back where the vast majority of cases were correctly identified. From the data, it is clear that VGG19 had specific issues whereas MobileNetV2, ResNet152V2, and InceptionV3 performed exceptionally well in their detection and classification of the guava leaf diseases spectrum. These have been pre-trained models used as trustable footings for detection of disease symptoms.

Table 4.4: Classification report of the Transfer Learning models

Model	**Class**	**Precision**	**Recall**	**F1-score**
VGG19	Caterpillars	0.47	0.86	0.61
	Cutting Weevil	0.96	1.00	0.98
	Die Back	0.97	0.99	0.98
	Healthy	0.89	0.94	0.91
	Mealybug Pests	0.75	0.56	0.64
	Red Rust	0.78	0.12	0.20
	Yellow spot	0.69	0.79	0.74
Dense Net201	Caterpillars	1.00	0.88	0.94
	Cutting Weevil	1.00	1.00	1.00
	Die Back	1.00	1.00	1.00
	Healthy	1.00	0.99	1.00
	Mealybug Pests	0.96	1.00	0.98
	Red Rust	0.92	1.00	0.96
	Yellow spot	1.00	0.99	1.00
Incepti onV3	Caterpillars	0.97	0.90	0.94
	Cutting Weevil	1.00	1.00	1.00
	Die Back	1.00	1.00	1.00
	Healthy	0.94	0.99	0.96
	Mealybug Pests	0.96	0.98	0.97
	Red Rust	0.96	0.93	0.95
	Yellow spot	0.96	0.97	0.97
Mobile NetV2	Caterpillars	1.00	0.97	0.99
	Cutting Weevil	1.00	1.00	1.00
	Die Back	1.00	1.00	1.00

	Healthy	1.00	1.00	1.00
	Mealybug Pests	1.00	1.00	1.00
	Red Rust	0.98	1.00	0.99
	Yellow spot	1.00	1.00	1.00
ResNet 152V2	Caterpillars	0.94	0.97	0.95
	Cutting Weevil	1.00	0.99	1.00
	Die Back	1.00	1.00	1.00
	Healthy	0.99	1.00	1.00
	Mealybug Pests	0.98	0.98	0.98
	Red Rust	0.97	0.94	0.96
	Yellow spot	1.00	1.00	1.00

The confusion matrices for the five pretrained CNN models, VGG19, DenseNet201, InceptionV3, MobileNetV2, and ResNet152V2, present useful data on how these models perform in classifying various guava leaf disease. Among the five models, MobileNetV2 had the highest accuracy with the least errors, especially for classification of Yellow Spot, Die Back, and Healthy leaf conditions. MobileNetV2 is praised for its capacity to perform disease classification tasks efficiently, with efficient computation, even in minor misclassifications such as those in Caterpillars and Red Rust. ResNet152V2 offered competitive performance with a few minor misclassifications observed in Red Rust and Mealybug Pests. DenseNet201 and InceptionV3 both demonstrated excellent results where the errors were relatively minimal at most disease classes, though were slightly difficult to differentiate Red Rust and Mealybug Pests. In contrast, VGG19 performed the worst with Caterpillars and Mealybug Pests it delivered more false positives and lower F1-scores. In conclusion, all of the pretrained models did produce an impressive end result; however, MobileNetV2 was clearly the best performing model for detecting guava leaf disease as it offers the strongest performance to calculation trade-off to be particularly well suited for a mobile application.



Confusion Matrix

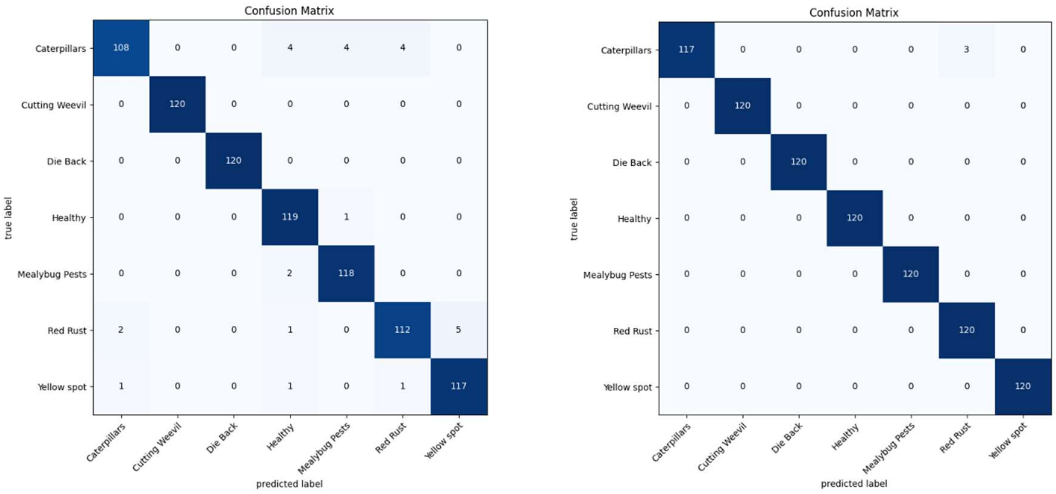
true label \ predicted label	Caterpillars	Cutting Weevil	Die Back	Healthy	Mealybug Pests	Red Rust	Yellow spot
Caterpillars	103	0	2	0	3	2	10
Cutting Weevil	0	120	0	0	0	0	0
Die Back	0	0	119	0	1	0	0
Healthy	0	2	1	113	3	1	0
Mealybug Pests	38	2	0	12	67	0	1
Red Rust	62	1	0	2	9	14	32
Yellow spot	17	0	1	0	6	1	95

VGG19

Confusion Matrix

true label \ predicted label	Caterpillars	Cutting Weevil	Die Back	Healthy	Mealybug Pests	Red Rust	Yellow spot
Caterpillars	106	0	0	0	4	10	0
Cutting Weevil	0	120	0	0	0	0	0
Die Back	0	0	120	0	0	0	0
Healthy	0	0	0	119	1	0	0
Mealybug Pests	0	0	0	0	120	0	0
Red Rust	0	0	0	0	0	120	0
Yellow spot	0	0	0	0	0	1	119

DenseNet201



Confusion Matrix

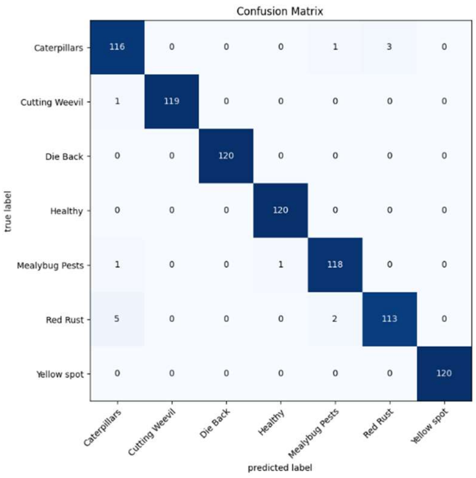
true label \ predicted label	Caterpillars	Cutting Weevil	Die Back	Healthy	Mealybug Pests	Red Rust	Yellow spot
Caterpillars	108	0	0	4	4	4	0
Cutting Weevil	0	120	0	0	0	0	0
Die Back	0	0	120	0	0	0	0
Healthy	0	0	0	119	1	0	0
Mealybug Pests	0	0	0	2	118	0	0
Red Rust	2	0	0	1	0	112	5
Yellow spot	1	0	0	1	0	1	117

InceptionV3

Confusion Matrix

true label \ predicted label	Caterpillars	Cutting Weevil	Die Back	Healthy	Mealybug Pests	Red Rust	Yellow spot
Caterpillars	117	0	0	0	0	3	0
Cutting Weevil	0	120	0	0	0	0	0
Die Back	0	0	120	0	0	0	0
Healthy	0	0	0	120	0	0	0
Mealybug Pests	0	0	0	0	120	0	0
Red Rust	0	0	0	0	0	120	0
Yellow spot	0	0	0	0	0	0	120

MobileNetV2



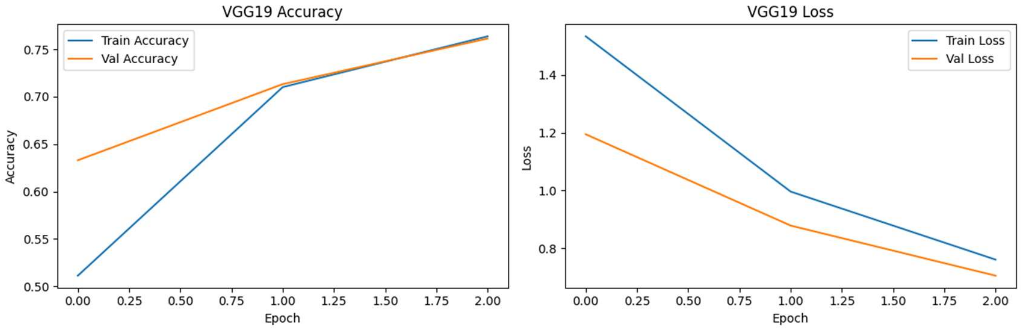
Confusion Matrix

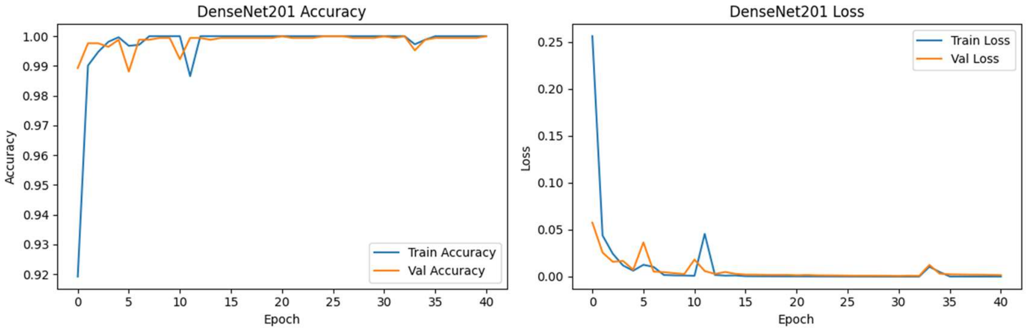
true label \ predicted label	Caterpillars	Cutting Weevil	Die Back	Healthy	Mealybug Pests	Red Rust	Yellow spot
Caterpillars	116	0	0	0	1	3	0
Cutting Weevil	1	119	0	0	0	0	0
Die Back	0	0	120	0	0	0	0
Healthy	0	0	0	120	0	0	0
Mealybug Pests	1	0	0	1	118	0	0
Red Rust	5	0	0	0	2	113	0
Yellow spot	0	0	0	0	0	0	120

ResNet152V2

Figure 4.1: Confusion Matrix of the Pre-trained CNNs

Graphs that point us to relevant information about a CNN model's training behavior are the ones that illustrate accuracy and loss for pretrained models, like VGG19, DenseNet201, InceptionV3, MobileNetV2 and ResNet152V2. Glaring from the curves for MobileNetV2 and ResNet152V2 is outstanding performance as both models quickly converge to ideal accuracy and maintain low loss implying efficient learning and good generalization. MobileNetV2 had perfect accuracy on both the training and validation sets and had fast loss decrease, which soon leveled off, implying that it worked fine and did not overfit. DenseNet201 retained its high accuracy, which was achieved after few training epochs at 99%, and the loss curve indicated fast convergence. InceptionV3 also showed gradual progression, seeding out at an accuracy higher than 97%, with closely positioned training and validation loss indicating good generalization. However, VGG19 showed relatively poor results with an accuracy floating at 75% after two epochs, as well as a validation loss, which was always higher than the training loss, a sign of overfitting. The data suggests that: MobileNetV2 and ResNet152V2 are outperforming DenseNet201 and InceptionV3 with high accuracy and computational efficiency however; DenseNet201 and InceptionV3 managed to produce good results despite a reduced generalization.





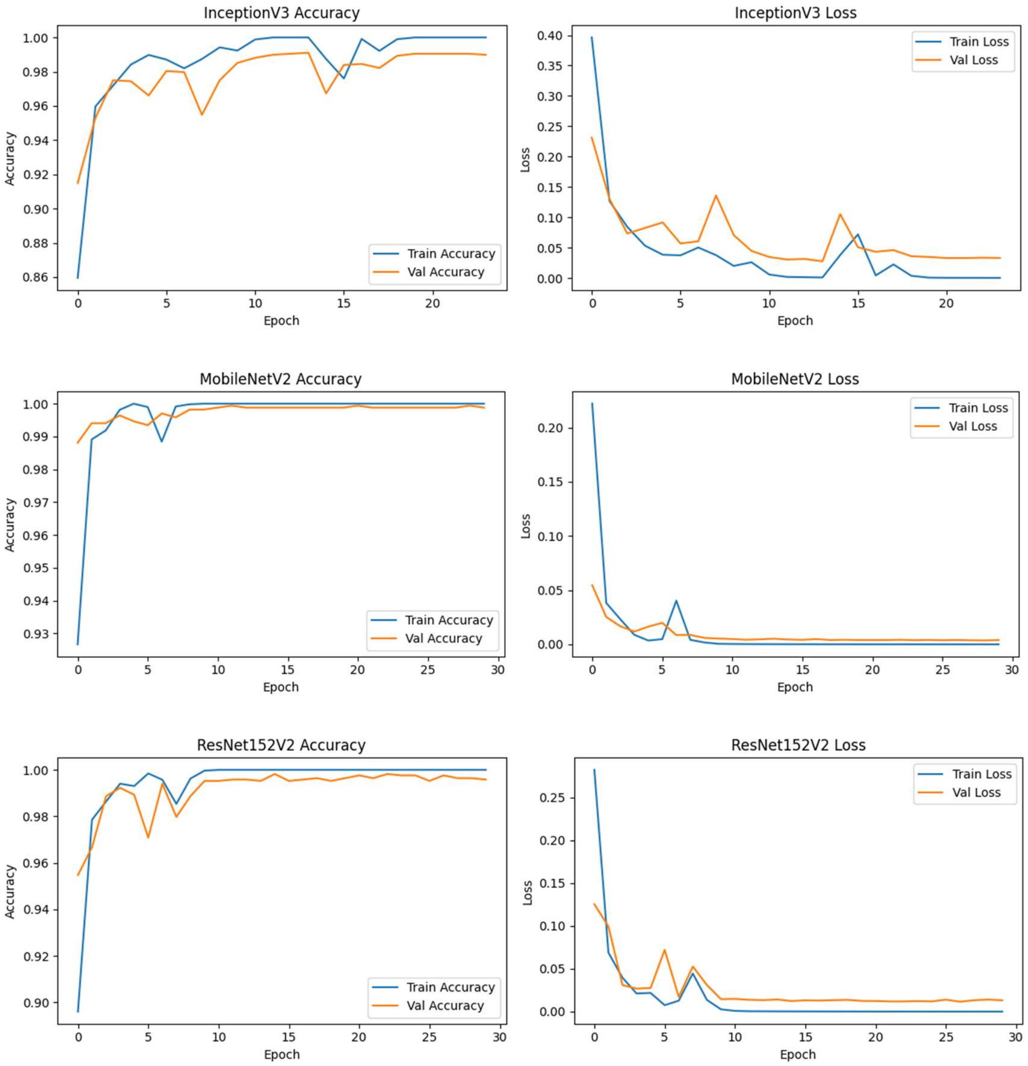
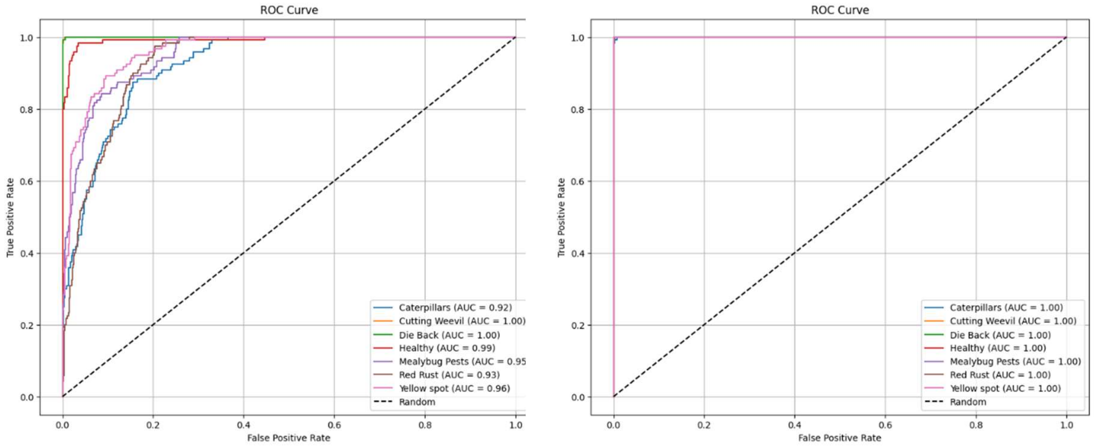


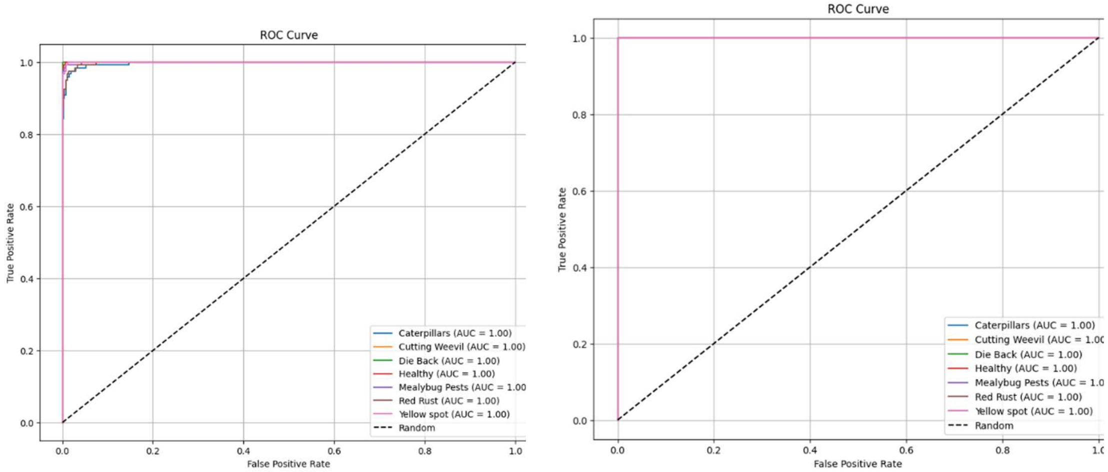
Figure 4.2: Accuracy and Loss Curves of the Pre-trained CNNs

The high accuracy AUC ROC curves for pretrained CNN models – VGG19, DenseNet201, InceptionV3, MobileNetV2 and ResNet152V2 – suggest that these models are very good at discriminating between guava leaf disease classes. MobileNetV2 and ResNet152V2 performed exceptionally well, scoring an AUC score of 1.00 in various classes, which meant almost perfect classification. These models showed strong separation between the genuine positive and false positive cases, especially with such classes as Yellow Spot, Healthy and Cutting Weevil. DenseNet201 performed also very well, with close-to 1.00 AUC values for many classes, which portrays a very ideal classification, while still InceptionV3 retained extraordinary AUCs for most classes, albeit with slight throttles in certain diseases like Red Rust and Mealybug Pests. While VGG19 was still competitive, it showed slightly decreased AUC values in problematic classes such as Caterpillars and Mealybug Pests, indicating that, compared to some other models, VGG19 had more challenges classifying these diseases correctly. Overall, MobileNetV2 and

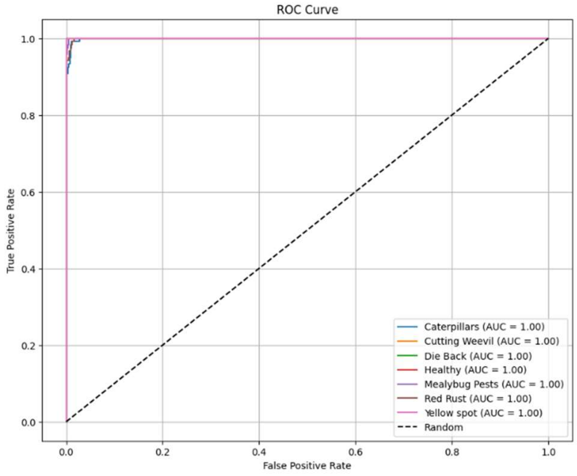
ResNet152V2 produced the most accurate and reliable predictions on guava leaf disease detection, as it appears from the AUC ROC curves.



VGG19 DenseNet201



InceptionV3 MobileNetV2



ResNet152V2

Figure 4.3: AUC ROC Curves of the Pre-trained CNNs

4.3.2 Results of Customized M-Net

At an outstanding accuracy level of 99.04%, M-Net outperformed all the pretrained CNN models that were employed, including MobileNetV2 and ResNet152V2, the top models that had been identified in the transfer learning category. The phenomenal level of accuracy exhibited by the model accentuates the superiority of the model in the diagnosis of the seven types of guava leaf diseases. The slender shape of M-Net proved to be decisive for it and made it a perfect option for real-time use, due to its ability to function well on mobile devices or systems without high hardware requirements. The light architecture of M-Net also enabled it to run effectively on limited power devices, and this made it a good platform for practical applications in agriculture. The equilibrium between high accuracy and effective computation enables M-Net to provide reliable disease detection in real-time applications while keeping performance both speedy and farmer-friendly when used outdoors.

The classification report for the Customized is outstanding in all disease categories. The model achieved excellent results in precision, recall, and F1-score for essentially all classes of diseases. For Caterpillars the 0.98 precision, 0.97 recall, and F1-score of 0.97 is a metric that shows off M-Net's impressive capability to correctly classify this class. All diseases were correctly classified including Cutting Weevil, Die Back, Mealybug Pests, Healthy, Red Rust, and Yellow Spot, which gave F1-scores of 1.00 in many cases, which shows that the model can precisely identify each disease. M-Net showed an exceptionally accurate Yellow Spot-, Healthy-, and Die Back-detection ability characterized by perfect 1.00 precision and recall rates in all three instances, highlighting its high competency in difficult disease determination. The classification report combined explains the greater accuracy and effectiveness of using M-Net for detecting the guava leaf diseases and delivers outstanding results for an online application.

Table 4.5: Classification report of Customized M-Net

Model	Class	Precision	Recall	F1-score
M-Net	Caterpillars	0.98	0.97	0.97
	Cutting Weevil	1.00	0.98	0.99
	Die Back	1.00	1.00	1.00
	Healthy	0.98	1.00	0.99
	Mealybug Pests	1.00	0.99	1.00
	Red Rust	0.97	0.99	0.98
	Yellow spot	1.00	1.00	1.00

The confusion matrix of Customized M-Net model can be shown to offer powerful evidence of the heavy capacity of its ability to classify guava leaf diseases. The model recorded low error rates since most of the predicted labels matched the actual ones. In the case of Caterpillars, the model identified 115 cases correctly and incorrectly classified 4 images as Yellow Spot, with no misclassification occurring for other diseases. Similar to the above, in Cutting Weevil handling, the model generated 2 errors on a total of 119 instances, which demonstrated its superior performance. Die Back, Healthy, and Yellow Spot had flawless performance, since the model classified all instances correctly. The model demonstrated great accuracy since only one case was misclassified in the pest Mealybug and two cases for Red Rust. The confusion matrix revealed that the M-Net proved to be very accurate in identifying distinct disease types with few errors, which made it suitable for practical use in identifying guava leaf diseases.

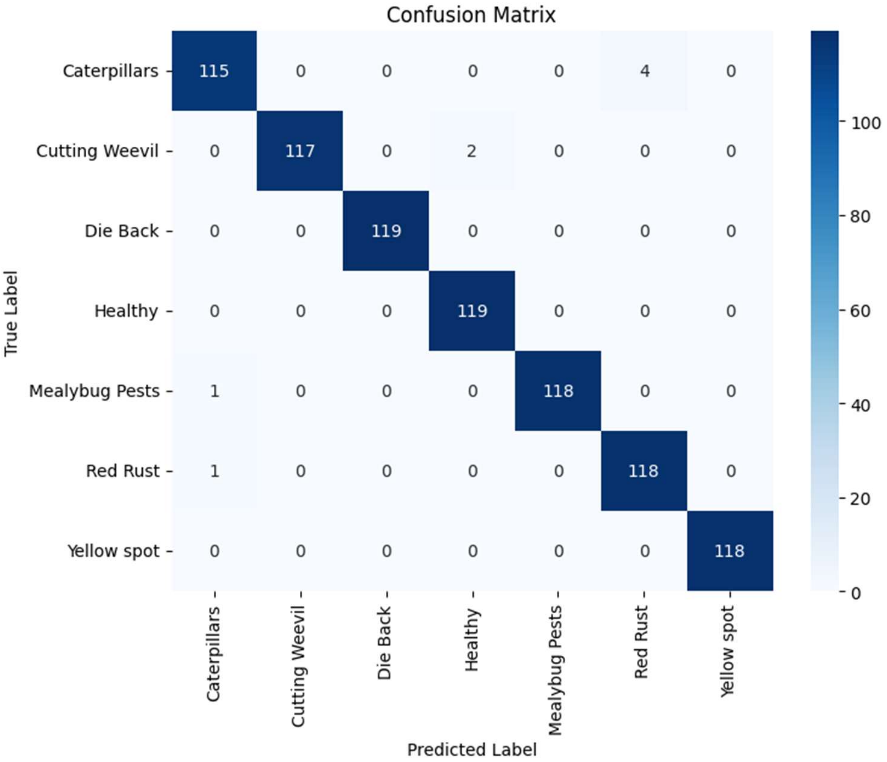


Figure 4.4: Confusion Matrix of the Customized M-Net

The Loss and Accuracy Curve of the Customized M-Net model indicates remarkable performance during training. Training accuracy increased gradually, from approximately 60% to almost 100%, with the ultimate proof that the model was able to train correctly to distinguish between guava leaf diseases. Similar to the training accuracy, the validation accuracy also rose in a progressive fashion to nearly 1.00 that gives evidence that the model has generalized well to unseen samples. At the beginning of training, a significant decrease in the loss curve was evident with evidence of rapid learning and rapid fall in error. Several epochs later, both training and validation losses remained at a low level, which prominently showed that the M-

Net provided high accuracy with effective ability not to let it overfit. The stability that we notice in both curves indicates that the model has adapted to the data successfully, i.e., it promises accuracy as well as robustness when applied to practical applications.

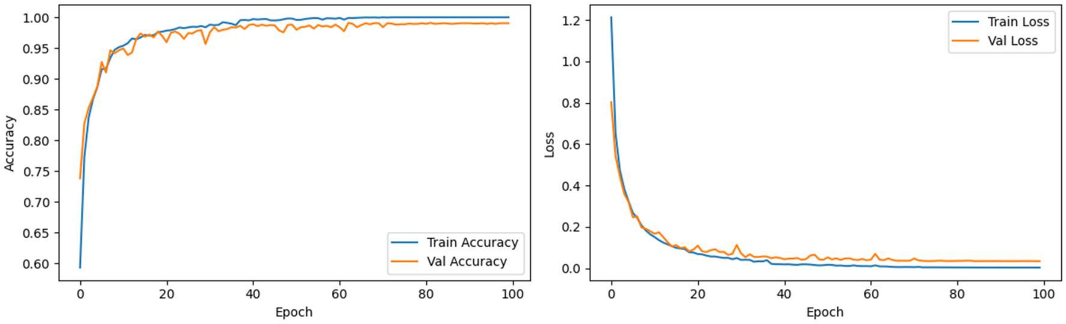


Figure 4.5: Accuracy and Loss Curves of the Customized M-Net

The exceptional AUC ROC curve on the Customized M-Net model covers all the disease classes. Every class has a True Positive Rate (TPR) that is extremely close to 1.0, and this underscores the ability of the model to find relevant instances even if a few instances are identified that are not relevant. All classes demonstrate AUC values that are magnificently close to 1.0, which shows nearly-perfect performance throughout. The AUC for Caterpillars is 0.98, while for Cutting Weevil, Die Back, Healthy, Mealybug Pests, Red Rust, and Yellow Spot, AUC values are 0.99 or 1.00 as a result of which the model can apparently perform very easily for the classification of all these diseases. Such extra evidence shows the strength and dependability of M-Net model because it always correctly classifies guava leaf diseases.

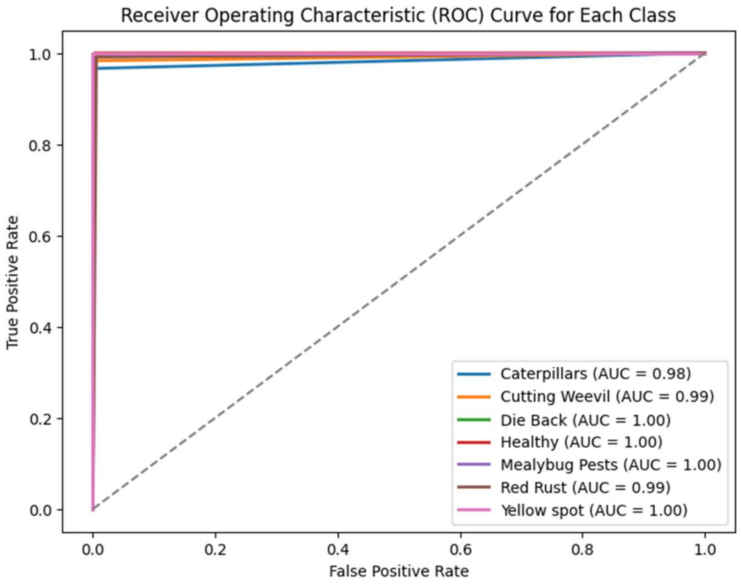


Figure 4.6: AUC ROC Curve of the Customized M-Net

4.3.3 Results of XAI

4.3.3.1 SHAP Output

The SHAP output from the Customized M-Net model provides a clear sketch of the leaf regions that contribute greatly towards the model's classification. The left panel shows the live guava leaf, while on the right there is the SHAP explanation overlay to demonstrate the model's reasoning. We present the SHAP values using a color gradient, where blue corresponds to negative contribution and red corresponds to positive contribution. The intensity of the color indicates how much of the model classification outcome is impacted by different regions of the leaf. In this figure, areas showing symptoms of the disease are marked with increased positive SHAP values denoting the important impact of those regions onto the model classification result. By viewing the SHAP output (graphically), users have locus on which portions of the image significantly contributed to the model's detection of diseases, thereby increasing model transparency and gaining confidence from the user. This image perfectly shows us how it can be understood how the M-Net model works if combined with the XAI methods like SHAP.

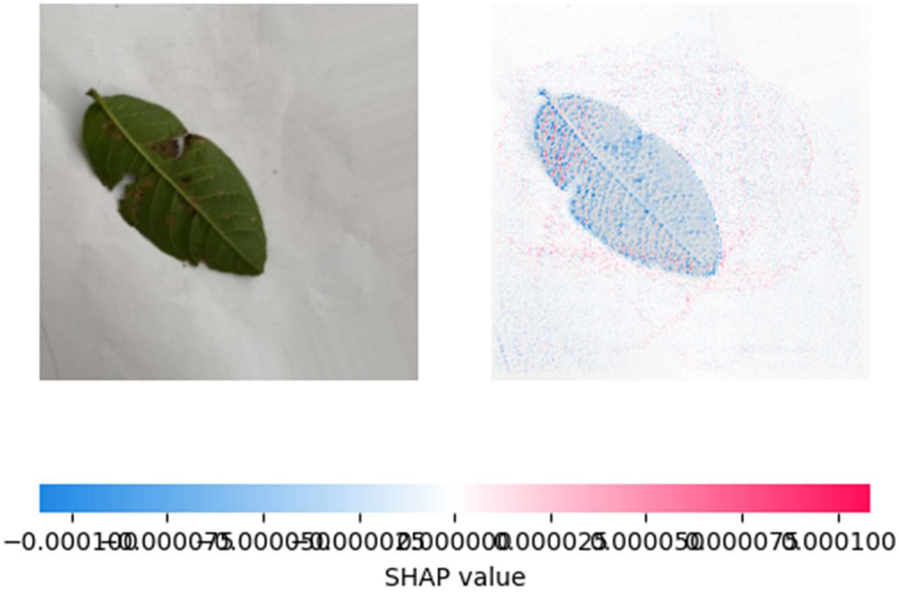


Figure 4.7: SHAP output on a selected image

4.3.3.2 GradCAM Output

The Grad-CAM output shown here captures the specific parts of the guava leaf that account for the decision of the model for a single image. LEFT side shows the original leaf image and on the right side is displayed the Grad-CAM visualisation overlaid. The colors in the Grad-CAM overlay are indicative of the particular areas of the image the model focused on most when classifying it. The brighter and more intense ones, usually dotted in yellow and red hues, indicate the places exerting most influence over the model's classification. Areas of such marks often correspond with areas of the leaf that have been struck by disease, which suggests insight in how the model has come to its classification. Seeing these regions lends added transparency to the model and this has significant value in applications that demand high reliability, for example, plant disease diagnosis.

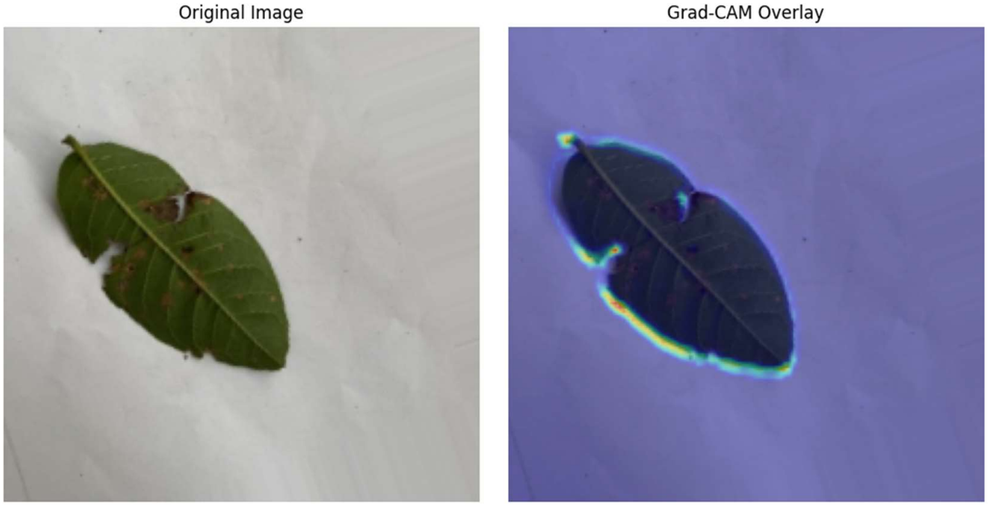


Figure 4.8: GradCAM output on a single image

The GradCAM results of a set of five images signify what the model-customized M-Net focuses on concerning the images when classifying. For each image within the batch of images, GradCAM generates a heatmap that highlights which parts of the picture the model considers most important for classification. In its support, one can find original example scenes ("Caterpillars_1.png," "Caterpillars_10.png," etc.), shown on the right along with their corresponding GradCAM heatmaps.

The model correctly identifies and highlights image parts displaying signs of disease such as apparent caterpillar damage and matches closely to the disease signs model assigned labels. For example, on "Caterpillars_1000.png," GradCAM focuses on those regions where the leaf-damage is easily visible to the eye. The use of GradCAM offers perfectly translucent understanding of how the model reaches its decision, which makes it crucial for getting a clue of its behaviour in a real-world setting.

When GradCAM is used for a set of five images, the model demonstrates that it can reliably emphasize key features relevant to every class, such as “Caterpillars”, “Healthy”, etc. This further verification increases confidence in the M-Net's accuracy and reliability for real time identification of disease.


Image: Caterpillars_1.png
Original
Grad-CAM


Image: Caterpillars_10.png
Original
Grad-CAM


Image: Caterpillars_100.png
Original
Grad-CAM

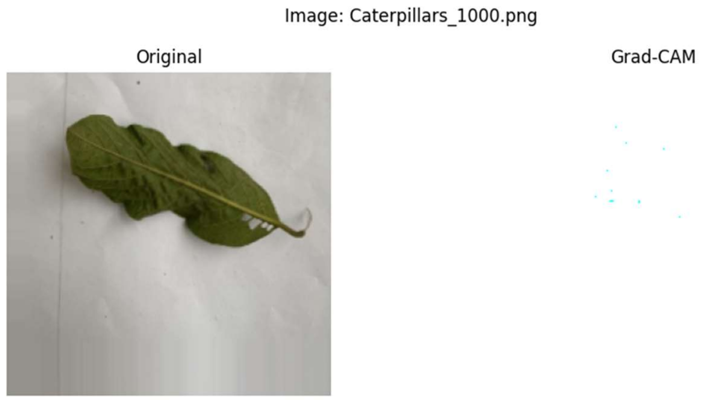


Image: Caterpillars_1000.png
Original
Grad-CAM

Figure 4.9: GradCAM output on a batch of images

4.4 Summary

The experimental process described in this chapter began with the acquisition, preparation, and improvement of the guava leaf dataset using several methods of pre-processing and augmentation. Subsequently, we evaluated five pre-trained CNN architectures; VGG19, DenseNet201, InceptionV3, MobileNetV2 and ResNet152V2, and the MobileNetV2 and ResNet152V2 models had the highest accuracy as compared to others. To the contrary, the customized lightweight M-Net model showed much higher performance with accuracy of 99.04%. Because of its well designed architecture, M-Net is also very suitable for the real time deployment scenario. In addition, integration of SHAP and GradCAM, both Explainable AI (XAI) methods, increased the model's performance and transparency of decision making. From the evaluative metrics such as the confusion matrices, the classification reports, and the AUC ROC curves, all indicated that M-Net was excellent in all classes. M-net's high accuracy and lightweight nature of the machine makes it a favorable choice for practical guava leaf disease detection resulting in reliable and understandable outputs within an actual practical environment.

Chapter 5

Engineering Standards and DesignChallenges

This chapter explores the engineering standards that were used in the development of the Guava Leaf Disease Detection System and the problems encountered during the design and application thereof. This chapter explains how the engineering principles are embedded in the system so that it can optimize its performance, reliability, and users' experience and discusses the broad social, ethical, and environmental implications of the project.

5.1 Compliance with the Standards

Throughout the development of the Guava Leaf Disease Detection System, strict adherence to relevant engineering standards was maintained to ensure that the system was both effective and reliable. The project complied with software quality standards such as those outlined in ISO/IEC 25010:2011, focusing on usability, performance efficiency, and portability. Tools like Google Colab, Kaggle, and Lightning AI were utilized to run experiments, ensuring scalable and accessible platforms that supported data processing and model training while meeting best practices for reproducible research. For image resizing, Microsoft PowerToy was used, which allowed for efficient batch resizing of images, ensuring consistency across the dataset. By using widely recognized, open-source technologies and high-performance platforms, the project ensured that both software development and model deployment adhered to recognized engineering standards, promoting transparency, efficiency, and scalability throughout the project.

5.1.1 Software Standards

The development of the Guava Leaf Disease Detection System adhered to established software engineering standards, ensuring the system's effectiveness, reliability, and maintainability. The project followed ISO/IEC 25010:2011 standards, focusing on key software quality attributes such as usability, performance efficiency, and maintainability. These standards ensured that the web application was not only intuitive and user-friendly, but also capable of providing real-time disease classification with high accuracy.

For model development, Google Colab and Kaggle were used to train and experiment with the model, benefiting from cloud computing resources. Lightning AI was utilized to optimize deep learning training, improving both performance and efficiency. The M-Net model was saved as an .h5 file, and Streamlit was chosen to create an interactive, lightweight web application. Streamlit seamlessly integrated with Python, allowing users to upload images and receive instant disease classifications in the browser. Ngrok was employed to expose the local development server to the internet, providing a secure, stable tunnel for easy deployment without the need for complex server setups. This approach ensured rapid development while optimizing the app for mobile use, offering scalability and ease of future updates.

5.1.2 Hardware Standards

The Guava Leaf Disease Detection System was designed for accessibility and efficiency, particularly for farmers in rural areas with limited access to high-end hardware. The system's hardware was chosen to meet user needs while ensuring performance, scalability, and compatibility.

Image Capture Devices: High-resolution smartphones were used for data collection due to their affordability, accessibility, and ability to capture detailed images for accurate disease classification. The smartphones used with their specifications are already given in Table 3.1.

These devices' high-resolution cameras ensured the capture of clear images, essential for detecting disease symptoms. Their f/1.8 aperture provided good performance under varying lighting conditions, which is crucial for agricultural environments.

Computational Resources: For training the models, Google Colab and Kaggle were used, providing cloud-based GPU environments that reduced the need for expensive local hardware. Lightning AI helped distribute the training tasks efficiently. The lightweight M-Net model was deployed, ensuring smooth performance on low-cost smartphones by integrating the trained model as an .h5 file and using Streamlit for real-time classification.

Mobile Device Compatibility: The web application is optimized for mobile devices, making it compatible with a range of Android and iOS smartphones. The app’s lightweight design ensures that it works well on devices with standard specifications, allowing farmers to use it without requiring high-end hardware.

Connectivity Standards: The system relies on Wi-Fi or mobile data for cloud communication. For areas with unstable connections, Ngrok was used to securely expose the app for testing, ensuring accessibility even with limited connectivity.

The hardware standards were selected to ensure the system is accessible and efficient while maintaining performance, enabling farmers in low-resource environments to access disease detection tools easily.

5.1.3 Communication Standards

The Guava Leaf Disease Detection System was designed to ensure seamless communication between various system components, such as image uploads, model inference, and result presentation. Modern communication standards were adopted to maintain efficiency, security, and scalability throughout the system.

Data Transmission Protocols: Communication between the client (user's device) and the backend server (hosting the disease classification model) was facilitated using HTTP/HTTPS protocols. HTTPS was prioritized for secure data transmission, ensuring that the images of guava leaves uploaded by users were encrypted during transfer. TLS/SSL encryption was implemented to guarantee that data was secure and protected from interception or tampering.

Ngrok for Hosting: For real-time image classification, Ngrok was used to expose the local server to the internet. This allowed for the web application to be accessed remotely using unique URLs, with minimal server configuration. Ngrok employs a token-based authentication system, ensuring that the communication tunnel is secure. This setup made it easier to test and share the application without deploying it on a dedicated server.

Mobile Optimized Communication: Given that the application is optimized for mobile devices, the communication was designed to be lightweight and efficient. Mobile-optimized protocols minimized data exchange during image uploads, which is important in areas with limited internet bandwidth. The system ensures low latency, providing real-time predictions and reducing waiting times for farmers. Communication between the mobile app and the cloud server is managed via a REST API-like architecture using HTTP requests, ensuring smooth interaction for mobile users.

Real-time Predictions and Feedback: The system supports asynchronous communication for real-time image classification. When a farmer uploads a leaf image, the system processes the image in the background, returning the classification result immediately. This quick feedback enables farmers to diagnose diseases and take timely action. The mobile-optimized interface ensures minimal delays and instant feedback, which is critical for effective disease management.

Scalability and Future Proofing: The system is built with scalability in mind, using cloud platforms and secure protocols to handle future growth. As the user base expands, the system can support an increasing number of users and requests. Additionally, it is designed to accommodate future updates such as adding new disease categories, disease management recommendations, and offline functionality without compromising performance or security.

5.2 Impact on Society, Environment and Sustainability

The Guava Leaf Disease Detection System significantly improves the lives of farmers, especially in rural areas. By providing a real-time, accessible tool for disease detection, it enables farmers to identify and address leaf diseases early, preventing significant crop losses and improving overall yield. This directly contributes to better food security and financial stability for farming families. The mobile device-based solution allows farmers to take immediate action, even in remote areas, without requiring expert intervention. Consequently, the system enhances agricultural productivity and livelihoods, empowering farmers to increase their income while reducing financial risks. By offering timely solutions, it plays a crucial role in supporting the well-being of individuals and communities that rely on agriculture for sustenance.

5.2.1 Impact on Life

The Guava Leaf Disease Detection System has a profound impact on the lives of farmers, particularly those in rural areas. By providing a real-time, accessible tool for disease detection, the system helps farmers identify and address leaf diseases early, preventing significant crop loss and improving overall yield. This directly leads to better food security and financial stability for farming families. The ability to identify diseases on mobile devices ensures that farmers can take immediate action, even in remote areas, without requiring expert intervention. As a result, the system not only improves agricultural productivity but also enhances livelihoods, empowering farmers to increase their income and reduce financial risk. By providing timely solutions, the system plays a critical role in supporting the well-being of individuals and communities that depend on agriculture for their sustenance.

5.2.2 Impact on Society & Environment

The Guava Leaf Disease Detection System also has a significant impact on society and the environment. By enabling early disease detection, the system helps farmers reduce their reliance on pesticides and fertilizers, which can have harmful effects on both the environment and human health. The system promotes sustainable agricultural practices by encouraging targeted treatments that minimize chemical usage and reduce the risk of soil degradation and water contamination. This not only helps in protecting local ecosystems but also contributes to the broader goal of environmental sustainability in agriculture. Furthermore, the system's mobile-first design ensures accessibility for smallholder farmers, empowering them to adopt modern farming techniques regardless of their technological or financial constraints. This accessibility promotes digital inclusion and reduces disparities in technology adoption, helping to bridge the gap between farmers in rural areas and those in more developed regions.

5.2.3 Ethical Aspects

Ethical considerations were integral to the design and deployment of the Guava Leaf Disease Detection System. The project adhered to data privacy standards, ensuring that any data collected from farmers, such as images of their guava leaves, was handled with informed consent and stored securely. As the system processes visual data, it is essential to ensure that the information is used responsibly and for the intended purpose of disease detection. Furthermore, the system was designed to be fair and unbiased, ensuring that the disease detection model performs consistently across various conditions, including different lighting, leaf orientations, and geographical regions. This prevents the system from favoring specific areas or conditions and ensures that all farmers benefit equally from the technology. Additionally, the integration of Explainable AI (XAI) methods, such as SHAP and Grad-CAM, fosters trust by providing transparent insights into the model's decision-making process. This interpretability is crucial in building user confidence and promoting accountability in the use of AI-driven tools.

5.2.4 Sustainability Plan

The Guava Leaf Disease Detection System has been designed with long-term sustainability in mind, ensuring its ongoing effectiveness and relevance. The sustainability of the system focuses on environmental, economic, and social sustainability, all critical for its impact on agricultural communities.

Environmental Sustainability: The system helps reduce the overuse of harmful chemical pesticides and fertilizers by enabling early disease detection, allowing farmers to apply treatments only when necessary. This minimizes the environmental impact, preserving soil health and water quality. The lightweight nature of the M-Net model ensures low energy consumption, reducing the carbon footprint of the technology, aligning with global sustainability goals.

Economic Sustainability: The system is affordable and scalable, utilizing open-source technologies and cloud platforms for cost-effective model training. The lightweight M-Net model runs on low-cost smartphones, making it accessible to smallholder farmers in rural areas. By improving crop yields and increasing income, the system supports economic growth and financial stability for farmers, ensuring its long-term viability.

Social Sustainability: The system contributes to digital inclusion by providing AI-driven technology to farmers with basic smartphones. The real-time disease detection helps farmers make proactive decisions, reducing crop losses and improving food security. Its user-friendly design ensures accessibility for farmers of all technical backgrounds and financial statuses, promoting social equity in rural communities.

Future Proofing: The system is designed for future updates, allowing it to incorporate new disease categories and function offline in areas with limited internet access. Future iterations can include automated disease management recommendations, further enhancing its value for farmers.

By integrating sustainability into its design and development, the system ensures continued impact and viability, contributing to sustainable agriculture, global food security, and economic empowerment in rural areas.

5.3 Project Management and Financial Analysis

The financial analysis for the Guava Leaf Disease Detection System focuses on the budget required for development, deployment, and its future sustainability. The project was designed to be cost-efficient, ensuring it could be developed and scaled without significant upfront investments. Below is a breakdown of the budget and revenue model, including alternate budget options for future scaling, all presented in Bangladeshi Taka (BDT).

1. Budget Required

- A. Initial Development Costs

Table 5.1: Initial Development Costs

Category	**Cost (BDT)**	**Rationale**
Data Collection	৳6,000	Costs for smartphones, data storage, and expert validation.
Software Development	৳5,000	Costs for using platforms like Google Colab, Kaggle, and Lightning AI.
Web Application Development	৳2,000	Streamlit (free) and hosting via Ngrok.
Cloud Services & Hosting	৳5,000	Costs for hosting the backend (AWS/Heroku).
Miscellaneous Costs	৳3,500	Costs for documentation and operational overhead.
Total Initial Development Costs	৳21,500	Total required for initial setup.

- B. Ongoing Operational Costs

Table 5.2: Ongoing Operational Costs

Category	**Cost (BDT)**	**Rationale**
Model Maintenance	৳10,000/year	Costs for model updates and additional disease classifications.
Web Hosting	৳5,000/year	Annual server hosting costs.
User Support & Documentation	৳5,000/year	Online support, FAQs, and user manuals.
Miscellaneous Updates	৳8,000/year	Software updates and bug fixes.
Total Ongoing Operational Costs	৳28,000/year	Total annual costs for operation.

- C. Alternate Budget and Rationales

Table 5.3: Alternate Budget and Rationales

Category	Cost (BDT)	Rationale
Increased Data Collection	৳10,000	To support data from more regions and diseases.
Additional Cloud Infrastructure	৳10,500	Scaling the cloud services to handle larger datasets and user load.
Partnerships and Marketing	৳20,500	Partnering with agricultural organizations and marketing.
Total Alternate Budget	৳41,000	Increased budget for future scaling and expansion.

- Rationale for Alternate Budget:
    - Increased Data Collection: Additional resources for data collection in new regions.
    - Increased Cloud Infrastructure: Scaling to handle more users and data.
    - Partnerships and Marketing: Engaging with agricultural organizations to increase reach.

2. Revenue Model

Several revenue models are proposed to ensure the system's long-term sustainability:

- A. Freemium Model
    - Basic Free Access: Core disease classification features remain free.
    - Premium Features: Subscription services (disease management

    recommendations, expanded data analysis) could range from ৳500 to ৳2,000/year.
    - Projected Revenue: With 1,000 premium subscribers at ৳1,000/year, the system could generate ৳1,000,000/year.
- B. Partnerships with Agricultural Organizations
    - Strategic partnerships with agricultural bodies could fund the system development.
    - Projected Revenue: ৳100,000 to ৳500,000 per partnership, depending on the scale.
- C. Data Analytics for Agricultural Agencies
    - Selling aggregated data insights (disease trends, pest outbreaks) to agencies or research institutions.
    - Projected Revenue: ৳205,000 to ৳575,000/year from data insights.

3. Cost-Benefit Analysis
The cost-benefit analysis suggests that the system will be highly cost-effective for farmers, delivering significant returns through increased crop yields, reduced pesticide use, and improved financial stability. The freemium model offers a scalable revenue stream to support future development and features, ensuring the system's sustainability. The financial projections indicate that the system will become financially viable within the first one to two years of deployment, with opportunities for growth as the user base expands and new features are added.

5.4 Complex Engineering Problem

The development of the Guava Leaf Disease Detection System involved solving several complex engineering problems that required deep technical expertise and creative solutions. The most challenging aspect of the project was ensuring that the system could efficiently handle the intricate task of disease detection in guava leaves, which required the integration of advanced machine learning algorithms, image processing techniques, and a user-friendly interface for farmers. Additionally, the system needed to be lightweight and capable of running on low-resource devices like smartphones to ensure broad accessibility in rural regions where access to high-end hardware may be limited. Balancing accuracy with computational efficiency while ensuring real-time disease detection was a critical aspect that posed several technical challenges. The engineering problem was further compounded by the need for explainable AI (XAI) to ensure that the decisions made by the model were interpretable by users, particularly farmers who have limited technical backgrounds.

5.4.1 Complex Problem Solving

The project required advanced knowledge across various domains, including deep learning (CNNs), transfer learning, data augmentation, and XAI techniques like SHAP and Grad-CAM. The development process was driven by the need for interpretability in AI models, ensuring the results could be understood by farmers.

Table 5.4: Mapping with complex problem solving.

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	EP 7 Interdependence
✓		✓	✓	✓	✓	✓

Mapping with Knowledge Profile for EP1

Table 5.5: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 EngineeringDesign	K6 EngineeringPractice	K8 Research Literature
✓	✓	✓	✓	✓

5.4.1.1 Justification for EP Attributes Mapping

EP1 – Depth of Knowledge: The system demanded expertise in multiple areas, from CNN architectures like VGG19 and MobileNetV2 to model optimization techniques like hyperparameter tuning and transfer learning.

EP3 – Depth of Analysis: A comprehensive experimental pipeline was implemented, with various pretrained models compared against the custom M-Net model. The final model achieved 99.04% accuracy, demonstrating the effectiveness of the optimized CNN architecture.

EP4 – Familiarity of Issues: The project addressed issues like data imbalance and noisy images, which were mitigated by data augmentation and careful model training. XAI techniques were crucial in addressing challenges related to model transparency.

EP5 – Extent of Applicable Codes: A range of pretrained CNN models was utilized, with TensorFlow and Keras forming the foundation for model training and deployment. Streamlit was used for rapid development of the user-facing application.

EP6 – Stakeholder Involvement: Agricultural experts were involved in data validation, ensuring accurate disease classification. This involvement was key in aligning the system with real-world agricultural needs.

EP7 – Interdependence: The system's components, from data collection to real-time predictions, were highly interdependent. For instance, the model's accuracy depended on the quality of the dataset, and the app's functionality relied on the model's performance for accurate disease detection.

5.4.1.2 Justification for Knowledge Profile Mapping (linked to EP1)

K3 - Engineering Fundamentals: The project utilized core machine learning principles such as classification, optimization, data augmentation, and evaluation metrics. These fundamentals were critical for the effective implementation of CNN models in plant disease detection.

K4 - Specialist Knowledge: Specialized knowledge in CNNs, transfer learning, and hyperparameter optimization was required for model development. XAI techniques like SHAP and Grad-CAM were integral to ensuring transparency and interpretability of the model.

K5 - Engineering Design: The design of the system involved selecting appropriate CNN models (VGG19, MobileNetV2) and optimizing them for the task of guava leaf disease classification. The design also included the integration of XAI tools for better model explainability.

K6 - Engineering Practice: The practical implementation involved training the model, integrating it with a user-friendly web application using Streamlit, and ensuring the system was accessible to farmers. The design of the web app also focused on ensuring mobile optimization and real-time disease detection.

K8 - Research Literature: The project built on existing research in deep learning for plant disease detection, particularly the use of CNNs and XAI methods in agricultural contexts. This literature guided the choice of models and approaches used in the project.

5.4.2 Engineering Activities

The engineering activities involved in the development of the Guava Leaf Disease Detection System were structured to tackle the complex engineering challenges identified in the project. These activities were aligned with key engineering principles to ensure the project's success, particularly in terms of technical performance, usability, and sustainability. The following outlines the mapping of the engineering activities with their justifications.

Table 5.6: Mapping with complex engineering activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓	✓	✓	✓	✓

5.4.2.1 Justification for Engineering Activities Mapping

EA1 – Range of Resources: A variety of tools and resources were used to facilitate the development and deployment of the system. High-resolution smartphones, such as the Samsung Galaxy S21+ and iPhone 11, were used to capture clear images of guava leaves for dataset creation. Python, Keras, and TensorFlow provided the necessary platforms for data preprocessing, model training, and optimization. Furthermore, XAI techniques like Grad-CAM and SHAP were used to enhance model transparency, while Streamlit enabled the creation of an interactive web application for real-time disease detection. GPU-based cloud platforms like Google Colab and Kaggle helped speed up model training and experimentation.

EA2 – Level of Interaction: The project incorporated multi-level interactions between the model, the web interface, and the user (farmer). The backend model (M-Net) interacted with the frontend, where farmers could easily upload leaf images for disease classification. XAI methods like Grad-CAM and SHAP provided visual explanations of the model's decision-making process, promoting interpretability and trust. Real-time interaction between the user and the system facilitated efficient decision-making by offering immediate feedback on disease detection.

EA3 – Innovation: This project incorporated several innovative approaches, particularly the development of a lightweight CNN model (M-Net) designed to run efficiently on mobile devices. This made disease detection accessible in low-resource settings, where computational power is limited. The integration of XAI methods further improved the system's transparency, making it more reliable for farmers. The mobile-optimized web application created using Streamlit provided an innovative, low-cost solution that could be accessed through smartphones, ensuring usability in rural areas.

EA4 – Consequences for Society and Environment: The system was designed with a focus on minimizing environmental impact. By enabling early disease detection, it promotes targeted pesticide use, reducing chemical overuse and minimizing environmental pollution. The system enhances crop yield and food security, which has a direct positive impact on rural communities. Additionally, by decreasing the reliance on chemical treatments, the system helps preserve soil health and water quality, contributing to sustainable agricultural practices.

EA5 – Familiarity: The engineering activities involved in this project were grounded in well-established techniques in machine learning, agricultural technology, and mobile application development. CNNs and transfer learning are widely used in plant disease detection, and XAI methods like Grad-CAM and SHAP are commonly applied to improve model interpretability. Streamlit, a popular framework for rapid web development, was used to ensure the system's accessibility and interactivity, making the technology easy for farmers to adopt.

5.5 Summary

This chapter summarized the design and engineering for the Guava Leaf Disease Detection System highlighting the key challenges faced and strategies deployed to achieve this. Designed to help farmers located in distant areas with limited contact with professionals in identifying diseases, the system was developed to offer effective and timely detection of diseases. By employing state-of-the-art machine learning techniques, such as Convolutional Neural Networks (CNNs), and XAI tools like Grad-CAM and SHAP, the system allows accurate disease classification, while keeping transparency and building trust of users. The optimized light weight model allows real time disease detection in inexpensive smart phones facilitating its dissemination in poor settings.

The project not only valued technical excellent; it also sought to address the problems of society and environment. Inby its focus on eco-friendly practices and use of less pesticides, the solution increases environmental sustainability while improving farmers livelihood. With increased crop output and reduced deficiency, the system offers a viable and easy to implement strategy for improving food security and increasing agricultural yield. Finally, the Guava Leaf Disease Detection System serves as a model of AI in agriculture, yielding long-term benefits for farmers throughout Bangladesh and similar areas with a continuous development of technology and with a caring user interface design.

Chapter 6

Conclusion

This chapter presents the key results of the research and the thesis' important contributions. The chapter presents the study's limitations and directions for further research that can improve the accuracy, generalizability, and real-world application of the model, by constructing a larger dataset, making the system more explainable, and deploying it in mobile apps for broader scale.

6.1 Summary

This thesis presents the M-Net model, a deep learning approach for guava leaf disease detection, with an emphasis on image classification. The dataset includes seven types of diseases and has been extracted from two guava yards at Ashulia and Savar, Dhaka, consisting of seven disease categories. Initially, we tried five pre-trained CNN models (VGG19, DenseNet201, InceptionV3, MobileNetV2, and ResNet152V2) using transfer learning approaches. While the outcomes procured by the pre-trained models were encouraging, the M-Net model was designed to be streamlined, light form in a strategic manner. The M-Net model showed outstanding performance reaching 99.04% outperforming the pre-trained models and reflecting the efficiency of the model in resource scarce environments.

To enhance the system further, Explainable AI such as SHAP & Grad-CAM were integrated into the M-Net model. Through the implementation of these techniques, the system grew more transparent, and the users were able to understand how each of the predictions was made. SHAP explained which features played the greatest roles in the predictions, while Grad-CAM gave an example of the areas of the leaf that were influential. Testing the model was done using metrics such as confusion matrices, classification reports, and AUC ROC curves, by which the M-Net was proven to be more effective than other methods in all the tested evaluation criteria. Real time classification implementation has been demonstrated through the development of a mobile-optimised web app that will enable farmers to submit images for immediate identification of disease. With a web application having been designed to function on minimalistic devices, accessibility is enhanced on a considerable scale for farmers in rural areas with limited technological abilities. Finally, the thesis represents the value of deep learning and XAI integration in addressing real-world agricultural problematics and contributing to an effective, scalable platform that guarantees understanding of guava disease. This system has a great potential to fuel improvement in precision agriculture to maximize yields and encourage sustainable farming practices.

6.2 Limitation

Admittedly, despite the impressive results of the Guava Leaf Disease Detection System, some limitations have to be accepted. To begin with, the training dataset contained exclusively guava leaf diseases as seen in two yards of Ashulia, Savar, Dhaka that may not have reflected the whole spectrum of guava leaf diseases in distinct regions. Such limitation can affect the model's generalization capability when evaluated using a different dataset of guava leaves or in different areas. Moreover, variation in lighting, leaf age, or small changes in the way the disease looks could affect how well the model works. Another limitation is that the model has been again customized for guava leaf diseases diagnosis and there are scarcely any records of its performance on other crops or pathogens. Additional experiments are needed to compare the performance of the M-Net model on datasets other than the guava leaves and its actual training set. A threat exists that the model will be unable to differentiate between diseases that present comparable visual symptoms, such as Caterpillars and Mealybug Pests, which may lead to errors. Even though data augmentation and transfer learning are applied, the amount of increase in the model's performance is likely to be higher if more diverse samples are introduced to the training set. Although SHAP and Grad-CAM were used in an effort to increase model transparency, their outputs may not always reflect expert views, and Grad-CAM heat maps could highlight uninterested areas, undermining user confidence. If these XAI approaches are implemented in the web application, it will improve the farmers' real-time insight into how decisions are made with the model. Performance of the mobile application is affected by hardware capabilities, network conditions, as well as the quality of images uploaded to the app by its users. Lack of optimal picture of the image can have adverse outcomes where the model may be unable to diagnose a disease, particularly those with mild symptoms. In addition, the model's performance may vary with the user's device's processing capabilities.

Overall, the system is a constructive solution, and enhancements are made by expanding the dataset, perfecting the model capabilities to diagnose diseases, and including advanced XAI methods to enhance understanding of model outputs. The future focus on these improvements will promote the model's flexibility, clarity, and practical uses that go beyond the laboratory.

6.3 Future Work

Despite the effectiveness of the Guava Leaf Disease Detection System, there is a high potential for making a lot of improvements and extensions of it. So, in order to improve generalization, the model should be trained on a wider range of a more diverse dataset, which would include images with various geographic and species backgrounds. This is likely to make the model function in many different agriculture environments. It is possible that if image synthesis and domain adaptation are introduced as components of the process for data augmentation, the accuracy will improve particularly when it comes to instances in which visual differentiation of disease symptoms is difficult.By integrating XAI algorithms such as SHAP and Grad-CAM into the web platform, farmers would be able to instantly have a feel for the reasoning behind the system's recommendations in real time. Enhancing the compatibility capabilities of the model with various devices especially the end smartphones and providing offline options would serve the farmers irrespective of their location's internet availability. Exploration of high-level model structures such as transformer models or ensemble based methods, could increase both accuracy and speed. In addition, including disease progression tracking or predictive models that take into account environmental parameters would help farmers to be more proactive when trying to prevent diseases.

Ultimately, however, despite the ability of the current system to answer the detection of diseases effectively, further inquiry does have the possibility to enhance the model's accuracy, usability and functionality, therefore maintaining its effectiveness when applied to agriculture.

References

[1] Rallapalli, S., & Saleem Durai, M. A. (2021). A contemporary approach for disease identification in rice leaf. *International Journal of System Assurance Engineering and Management*, 1-11.

[2] Simhadri, C. G., Kondaveeti, H. K., Vatsavayi, V. K., Mitra, A., & Ananthachari, P. (2024). Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies and techniques. *Information Processing in Agriculture*.

[3] Rodríguez-Lira, D. C., Córdova-Esparza, D. M., Álvarez-Alvarado, J. M., Terven, J., Romero-González, J. A., & Rodríguez-Reséndiz, J. (2024). Trends in Machine and Deep Learning Techniques for Plant Disease Identification: A Systematic Review. *Agriculture*, *14*(12), 2188.

[4] Akter, S., Sumon, R. I., Ali, H., & Kim, H. C. (2024). Utilizing Convolutional Neural Networks for the Effective Classification of Rice Leaf Diseases Through a Deep Learning Approach. *Electronics*, *13*(20), 4095.

[5] Sharma, M., & Kumar, C. J. (2022). Improving rice disease diagnosis using ensemble transfer learning techniques. *International Journal on Artificial Intelligence Tools*, *31*(08), 2250040.

[6] Rashid, J., Khan, I., Ali, G., Alturise, F., & Alkhalifah, T. (2023). Real-Time Multiple Guava Leaf Disease Detection from a Single Leaf Using Hybrid Deep Learning Technique. *Computers, Materials & Continua*, *74*(1).

[7] Mustak Un Nobi, M., Rifat, M., Mridha, M. F., Alfarhood, S., Safran, M., & Che, D. (2023). GLD-DET: Guava leaf disease detection in real-time using lightweight deep learning approach based on MobileNet. *Agronomy*, *13*(9), 2240.

[8] Foo, C. F., Tay, K. G., Al-Qershi, O., Huong, A., Chew, C. C., & Alzaeemi, S. A. (2024). Android-Based App Guava Leaf Diseases Identification using Convolution Neural Network. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 73-88.

[9] Rashid, J., Khan, I., Abbasi, I. A., Saeed, M. R., Saddique, M., & Abbas, M. (2023). A hybrid deep learning approach to classify the plant leaf species. *Comput Mater Contin*, *76*, 3897-3920.

[10] Vedhamuru, N., Malmathanraj, R., & Palanisamy, P. (2024). Lightweight deep and cross residual skip connection separable CNN for plant leaf diseases classification. *Journal of Electronic Imaging*, *33*(3), 033035-033035.

[11] Bala, M., & Bansal, S. (2024). Unveiling the Power of Deep Learning in Plant Pathology: A Review on Leaf Disease Detection. *ECS Journal of Solid State Science and Technology*.

[12] Chang, C. Y., & Lai, C. C. (2024). Potato leaf disease detection based on a lightweight deep learning model. *Machine Learning and Knowledge Extraction*, *6*(4), 2321-2335.

[13] Prakash, K., & Geetha, B. G. (2024). Potato Plant Leaf Disease Detection Distinctive Deep Attention Convoluted Network (DACN) Mechanism. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, *48*(4), 1567-1593.

[14] Ghosh, P., Mondal, A. K., Chatterjee, S., Masud, M., Meshref, H., & Bairagi, A. K. (2023). Recognition of sunflower diseases using hybrid deep learning and its explainability with AI. *Mathematics*, *11*(10), 2241.

[15] Mazumder, M. K. A., Mridha, M. F., Alfarhood, S., Safran, M., Abdullah-Al-Jubair, M., & Che, D. (2024). A robust and light-weight transfer learning-based architecture for

accurate detection of leaf diseases across multiple plants using less amount of images. *Frontiers in Plant Science*, *14*, 1321877.

[16]Tasfe, M., Nivrito, A. K. M., Al Machot, F., Ullah, M., & Ullah, H. (2024). Deep Learning Based Models for Paddy Disease Identification and Classification: A Systematic Survey. *IEEE Access*.

[17]Zhu, S., & Gao, H. (2024). MC-ShuffleNetV2: A lightweight model for maize disease recognition. *Egyptian Informatics Journal*, *27*, 100503.

[18]Deepalakshmi, P., Lavanya, K., & Srinivasu, P. N. (2021). Plant leaf disease detection using CNN algorithm. *International Journal of Information System Modeling and Design (IJISMD)*, *12*(1), 1-21.

[19]Tugrul, B., Elfatimi, E., & Eryigit, R. (2022). Convolutional neural networks in detection of plant leaf diseases: A review. *Agriculture*, *12*(8), 1192.

[20]Tüfekci, G., Kayabaşi, A., & Ulusoy, İ. (2022, May). A Comparative Analysis of Revealing Temporal Patterns for Driver Drowsiness Detection. In *2022 30th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

[21][Aabidi, M. H., EL Makrani, A., Jabir, B., & Zaimi, I. (2023). A Model Proposal for Enhancing Leaf Disease Detection Using Convolutional Neural Networks (CNN): Case Study. *International Journal of Online & Biomedical Engineering*, *19*(12).

[22]Gupta, A., Gupta, D., Husain, M., Ahmed, M. N., Ali, A., & Badoni, P. (2023). A PSO-CNN-based approach for Enhancing Precision in Plant Leaf Disease Detection and Classification. *Informatica*, *47*(9).

[23]Nandhini, S. K. T. S., Suganya, R., Nandhana, K., Varsha, S., Deivalakshmi, S., & Thangavel, S. K. (2021). Automatic detection of leaf disease using CNN algorithm. In *Machine Learning for Predictive Analysis: Proceedings of ICTIS 2020* (pp. 237-244). Springer Singapore.

[24]Vishnoi, V. K., Kumar, K., Kumar, B., Mohan, S., & Khan, A. A. (2022). Detection of apple plant diseases using leaf images through convolutional neural network. *IEEE Access*, *11*, 6594-6609.

[25]Ulutaş, H., & Aslantaş, V. (2023). Design of efficient methods for the detection of tomato leaf disease utilizing proposed ensemble CNN model. *Electronics*, *12*(4), 827.

[26]Ranjana, P., Reddy, J. P. K., Manoj, J. B., & Sathvika, K. (2022). Plant leaf disease detection using mask R-CNN. In *Ambient Communications and Computer Systems: Proceedings of RACCCS 2021* (pp. 303-314). Singapore: Springer Nature Singapore.

[27]Mekala, K. R., Modala, V. S. R. S., Devalla, V., & Kompalli, A. B. (2022). Leaf Disease Detection and Remedy Recommendation Using CNN Algorithm. *International Journal of Online & Biomedical Engineering*, *18*(7).

[28]Sharma, V., Tripathi, A. K., Daga, P., Nidhi, M., & Mittal, H. (2024). ClGanNet: A novel method for maize leaf disease identification using ClGan and deep CNN. *Signal Processing: Image Communication*, *120*, 117074.

[29]Vallabhajosyula, S., Sistla, V., & Kolli, V. K. K. (2022). Transfer learning-based deep ensemble neural network for plant leaf disease detection. *Journal of Plant Diseases and Protection*, *129*(3), 545-558.

[30]Gajjar, R., Gajjar, N., Thakor, V. J., Patel, N. P., & Ruparelia, S. (2022). Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. *The Visual Computer*, 1-16.

[31]Mithu, M. A., Momo, S. I., Hasan, M. M., Rahman, K. M., & Sattar, A. (2022). Pumpkin leaf disease detection: Convenience of cnn over traditional machine learning in terms of image classification. In *Smart Systems: Innovations in Computing: Proceedings of SSIC 2021* (pp. 347-357). Springer Singapore.

[32]Demilie, W. B. (2024). Plant disease detection and classification techniques: a comparative study of the performances. *Journal of Big Data*, *11*(1), 5.

[33]Pandian, J. A., Kanchanadevi, K., Kumar, V. D., Jasińska, E., Goňo, R., Leonowicz, Z., & Jasiński, M. (2022). A five convolutional layer deep convolutional neural network for plant leaf disease detection. *Electronics*, *11*(8), 1266.

[34]Singh, G., & Yogi, K. K. (2023). Performance evaluation of plant leaf disease detection using deep learning models. *Archives of Phytopathology and Plant Protection*, *56*(3), 209-233.

[35]Gong, X., & Zhang, S. (2023). A high-precision detection method of apple leaf diseases using improved faster R-CNN. *Agriculture*, *13*(2), 240.

[36]Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, *61*, 101182.

[37]Nagamani, H. S., & Sarojadevi, H. (2022). Tomato leaf disease detection using deep learning techniques. *International Journal of Advanced Computer Science and Applications*, *13*(1).

[38]Kurmi, Y., Saxena, P., Kirar, B. S., Gangwar, S., Chaurasia, V., & Goel, A. (2022). Deep CNN model for crops' diseases detection using leaf images. *Multidimensional Systems and Signal Processing*, *33*(3), 981-1000.

[39]Latif, G., Abdelhamid, S. E., Mallouhy, R. E., Alghazo, J., & Kazimi, Z. A. (2022). Deep learning utilization in agriculture: Detection of rice plant diseases using an improved CNN model. *Plants*, *11*(17), 2230.

[40]Patil, M. A., & Manohar, M. (2023). Plant Leaf Disease Classification Using Optimal Tuned Hybrid LSTM-CNN Model. *SN Computer Science*, *4*(6), 710.

[41]Tyagi, S., Reddy, S. R. N., Anand, R., & Sabharwal, A. (2024). Enhancing rice crop health: a light weighted CNN-based disease detection system with mobile application integration. *Multimedia Tools and Applications*, *83*(16), 48799-48829.

[42]Tembhurne, J. V., Gajbhiye, S. M., Gannarpwar, V. R., Khandait, H. R., Goydani, P. R., & Diwan, T. (2023). Plant disease detection using deep learning based Mobile application. *Multimedia Tools and Applications*, *82*(18), 27365-27390.

[43]Khan, F., Zafar, N., Tahir, M. N., Aqib, M., Waheed, H., & Haroon, Z. (2023). A mobile-based system for maize plant leaf disease detection and classification using deep learning. *Frontiers in Plant Science*, *14*, 1079366.

[44]Yulita, I. N., Amri, N. A., & Hidayat, A. (2023). Mobile application for tomato plant leaf disease detection using a dense convolutional network architecture. *Computation*, *11*(2), 20.

[45]Mehedi, M. H. K., Nawer, N., Ahmed, S., Khan, M. S. I., Hasib, K. M., Mridha, M. F., ... & Nguyen, T. T. (2024). PLD-Det: plant leaf disease detection in real time using an end-to-end neural network approach based on improved YOLOv7. *Neural Computing and Applications*, *36*(34), 21885-21898.

[46]Aishwarya, N., Praveena, N. G., Priyanka, S., & Pramod, J. (2023). Smart farming for detection and identification of tomato plant diseases using light weight deep neural network. *Multimedia Tools and Applications*, *82*(12), 18799-18810.

[47]Pal, C., Chatterji, S., & Pratihar, S. (2024). An Offline Biotic Stress Recognition Tool for Rice Plants Through Domain Shift. *SN Computer Science*, *5*(5), 478.

turnitin **Similarity Report ID:** oid:3117:458453478

● 17% Overall Similarity

Top sources found in the following databases:

- 13% Internet database
- 14% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	**dspace.daffodilvarsity.edu.bd:8080** Internet	3%
2	**mdpi.com** Internet	1%
3	**arxiv.org** Internet	<1%
4	**link.springer.com** Internet	<1%
5	**frontiersin.org** Internet	<1%
6	**Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shuk...** Publication	<1%
7	**propulsiontechjournal.com** Internet	<1%
8	**ijsr.net** Internet	<1%