Detection of Different Kinds of Bacterial and Fungal Diseases in Jackfruit Using Image Processing

By
MD SAIKAT ISLAM
211-15-4007

MD Waly Ullah Fahim
211-15-4049

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by

Raja Tariqul Hasan Tusher

Assistant Professor
Department of Computer Science and EngineeringDaffodil International University

Co-Supervised by

Mr. Tanvirul Islam, Lecturer
Department of Computer Science and EngineeringDaffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

January 12, 2025

APPROVAL

This Project titled **Detection of Different Kinds of Bacterial and Fungal Diseases in Jackfruit Using Image Processing**, submitted by **MD SAIKAT ISLAM** and **MD Waly Ullah Fahim** 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 **12-01-2025**.

BOARD OF EXAMINERS

Dr. S.M Aminul Haque (SMAH)
Professor & Associate Head
Chairman,
Department of CSE, Daffodil International University

12.01.25

Mohammad Jahangir Alam(MJA)
Internal Examiner 1
Assistant Professor, Department of CSE, Daffodil International University

Mr. Deawan Rakin Ahamed Remal (DRAR)
Internal Examiner 2
Lecturer, Department of CSE, Daffodil International University

Dr. Md. Arshad Ali (DAA)
External Examiner
Professor, Department of CSE, Hajee Mohammad Danesh Science & Technology University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Raja Tariqul Hasan Tusher, Assistant 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:

12-1-25

Mr. Raja Tariqul Hasan Tusher
Assistant Professor
Department of Computer Science and
EngineeringDaffodil International University

Co-Supervised by:

Mr. Tanvirul Islam
Lecturer
Department of Computer Science and
EngineeringDaffodil International University

Submitted by:

12-1-25

Md. Saikat Islam
Student ID: 211-15-4007
Department of Computer Science and
EngineeringDaffodil International University

Fahim 12-1-25

MD Waly Ullah Fahim
Student ID: 211-15-4049
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 as- sisted 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 **Mr. Raja Tariqul Hasan Tusher**, **Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Image Processing** to carry out this project. His endless pa- tience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat 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 Com- puter 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 study presents the development of an AI-driven tool for detecting bacterial and fungal diseases in jackfruit using advanced image processing and deep learning techniques. Leveraging four pretrained convolutional neural network (CNN) models VGG19, MobileNetV2, EfficientNetB0, and ResNet50, alongside an ensemble approach, the study focuses on achieving high accuracy and reliability in disease classification. The dataset, comprising jackfruit images categorized into Healthy, Bacteria Affected, and Fungus Affected classes, undergoes preprocessing steps such as normalization, resizing, and augmentation to ensure robust training and evaluation. Among the individual CNN models, VGG19 and MobileNetV2 demonstrate superior performance, with accuracies of 95.84% and 93.35% on the test set, respectively, while EfficientNetB0 exhibits the lowest performance due to instability in learning. The ensemble model significantly enhances classification performance, achieving a near-perfect accuracy of 99.83% and an AUC score of 1.00 across all classes, combining the strengths of individual models to minimize misclassifications. Furthermore, the study implements the model in a mobile application, "Jackfruit-Doctor," providing real-time disease detection with high accuracy and accessibility for end users. This application empowers farmers by enabling early intervention, reducing crop losses, and promoting sustainable agricultural practices. The findings demonstrate the efficacy of integrating ensemble learning with mobile technology, offering a reliable and scalable solution for agricultural disease management while contributing to food security and economic sustainability.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Agriculture remains a vital sector worldwide, essential for economic stability and food security, especially in regions heavily reliant on crop cultivation. Among tropical fruits, jackfruit (Artocarpus heterophyllus) holds particular importance in South and Southeast Asia, serving as a staple in Bangladesh and being extensively cultivated in India, Sri Lanka, and neighboring countries [1]. However, like many crops, jackfruit is susceptible to various diseases that significantly impact yield and quality. Early and accurate identification of these diseases is crucial for maintaining productivity and minimizing losses [1-2]. For farmers, detecting diseases in jackfruit trees is both a preventative measure and an economically essential practice, as disease-related losses can lead to reduced market access and lower financial returns [2].

In recent years, artificial intelligence (AI) has transformed agriculture, with deep learning approaches proving highly effective for disease detection across multiple crop types. Convolutional Neural Networks (CNNs), in particular, are widely recognized for their image analysis capabilities and ability to detect subtle differences between healthy and diseased leaves [3-4]. For instance, CNN-based models like EfficientNetB7 and custom networks have achieved significant success in jackfruit leaf disease identification by accurately detecting conditions such as algal and black spots, with training accuracies reaching as high as 99% [1]. Such models, developed with diverse datasets, including public datasets like PlantVillage, highlight CNNs' capacity to effectively categorize diseases across crop varieties [4].

The integration of transfer learning has further addressed data scarcity in agricultural contexts. Transfer learning enables models to leverage pre-trained architectures, such as VGG16 and ResNet, effectively adapting them to specific datasets with fewer labeled images. This approach has proven advantageous, achieving high accuracy in disease detection despite limited data [5-6]. For example, studies applying these models to mango leaf diseases have achieved impressive results, underscoring transfer learning's role in expanding AI's applicability to various crop types [7-8].

Additionally, federated learning has emerged as a promising solution to address growing concerns around data privacy and scalability in agriculture. By enabling localized learning across multiple client nodes, federated learning reduces data transfer needs and preserves data privacy, making it an efficient model for decentralized agricultural settings [3, 9]. This approach supports multi-client data diversity, allowing models to generalize better across different environments while achieving consistent accuracy. However, its implementation in agriculture remains relatively limited, highlighting the need for further research into decentralized AI applications [3].

Despite these advancements, challenges persist in translating these technologies into field-ready solutions. Most models are tested in controlled environments, often with limited variability in lighting, background, and crop diversity. Thus, deploying these models in real-world agricultural settings requires additional consideration of factors such as environmental noise, interpretability, and user-friendliness for non-expert farmers. Additionally, while explainable AI techniques have been integrated into some models, such as MLTNet for mango leaf disease detection, there is a growing need for interpretable AI solutions in crop disease identification to aid farmers in understanding model outputs directly [7].

1.2 Motivation

The prevalence of plant diseases poses a substantial threat to agricultural productivity and food security, particularly in regions dependent on crops like jackfruit. Despite its economic and cultural importance, jackfruit cultivation faces challenges from diseases that affect leaf, fruit, and stem health, leading to significant reductions in both yield and quality. In areas where early intervention is critical, farmers often lack the means to detect diseases promptly, resulting in late-stage management, which is less effective and more resource-intensive [2, 10].

Traditional methods of disease detection in agriculture are largely manual, relying on visual inspections that can be inconsistent and subjective. These approaches are typically time-consuming and require expert knowledge, which may not be accessible in rural or resource-limited settings [11]. As a result, there is an increasing demand for automated, reliable, and accessible solutions to identify and classify plant diseases accurately, particularly for jackfruit and similar tropical crops [1].

While deep learning models, particularly convolutional neural networks (CNNs), have shown promising results in disease classification for crops, these solutions often face limitations when deployed in real-world agricultural environments. High-performing models, such as those utilizing EfficientNetB7 or YOLO for jackfruit disease detection, demonstrate accuracies exceeding 99% in controlled settings; however, they often fail to achieve similar performance in field applications due to varying environmental factors [1, 4]. Additionally, the lack of large, labeled datasets specific to jackfruit diseases poses a barrier to training models that can generalize effectively across diverse conditions, leading to challenges in robustness and adaptability [5-6].

Moreover, while federated learning has been proposed as a solution to preserve data privacy in decentralized settings, its application remains limited in agriculture. Current federated models have shown promising accuracy rates across multiple clients, yet further research is needed to refine these models for scalability and practical use in agricultural contexts where connectivity and data quality may vary significantly [3, 9]. Additionally, few models address the need for interpretability, a critical factor for ensuring that farmers can trust and effectively use AI-driven disease detection systems [7].

1.3 Objectives

This research aims to address the challenges in jackfruit disease detection by developing and deploying advanced, accessible AI-based solutions. The specific objectives of the study are:

- Develop a robust disease detection model to design and implement a convolutional neural network (CNN)-based model tailored for jackfruit leaf and fruit disease detection, focusing on achieving high accuracy and adaptability across different environmental conditions, as demonstrated in prior studies with EfficientNetB7 and YOLO architectures [1], [4].
- Utilise data augmentation and transfer learning techniques to address data scarcity and improve the model's ability to generalise across varied jackfruit disease types and growing conditions.
- Validate model performance in real-world agricultural environments to

conduct extensive testing of the model in real-world agricultural settings, evaluating its effectiveness under diverse lighting, background, and environmental conditions to ensure practicality for field deployment, addressing the current gap between controlled testing and real-world applicability [10-11].

By fulfilling these objectives, this research seeks to provide a reliable, accessible, and privacy-conscious solution for jackfruit disease detection, ultimately empowering farmers with effective tools for early disease intervention and sustainable crop management.

1.4 Methodology

The methodology for this study involves developing a comprehensive system for detecting bacterial and fungal diseases in jackfruit leaves using advanced image processing and deep learning techniques. The process begins with the collection of a dataset comprising jackfruit leaf images, categorized into three classes: healthy, bacteria-affected, and fungus-affected. Preprocessing steps, including normalization, resizing, dots per inch (DPI) adjustment, and data augmentation, are applied to standardize the dataset and improve the model's generalization. Augmentation techniques such as flipping, rotation, scaling, and brightness adjustments expanded the dataset to 6,090 images, ensuring a diverse set of inputs for training. Various Convolutional Neural Network (CNN) models, including VGG19, EfficientNetB0, MobileNetV2, and ResNet50, are trained and evaluated independently to identify the best-performing model. This model is saved in .h5 format and later integrated into an ensemble approach that combines predictions from individual models to enhance accuracy and robustness. For real-world application, the best model is converted to TensorFlow Lite (TFLite) format and integrated into a mobile application developed with Flutter. This application provides end-users, such as farmers, with a user-friendly interface for real-time disease detection and reporting. The methodology ensures a scalable, efficient, and practical solution to address jackfruit disease challenges.

1.5 Project Outcome

The project successfully developed an automated system for the detection and classification of bacterial and fungal diseases in jackfruit leaves, resulting in a reliable, efficient, and accessible solution for end-users. By leveraging advanced image processing and deep learning techniques, the study achieved the following outcomes:

1. Dataset Expansion and Standardization: The original dataset of 3,045 images was augmented to 6,090 images through preprocessing techniques such as normalization, resizing, DPI adjustments, and data augmentation. This ensured robust training and improved model generalization, making it adaptable to diverse real-world conditions.
2. Model Performance: Several CNN architectures, including VGG19, EfficientNetB0, MobileNetV2, and ResNet50, were trained and evaluated. The best-performing model demonstrated high accuracy in disease classification and was saved in .h5 format for further integration. The ensemble model further improved classification accuracy by combining predictions from multiple architectures.
3. Mobile Application Deployment: The best-performing model was optimized and converted into TensorFlow Lite (TFLite) format, enabling real-time inference on mobile devices. A user-friendly mobile application was developed using Flutter, offering a seamless interface for disease detection and classification, even in offline scenarios. The app allows users to capture or upload images and receive instant results, making advanced AI tools accessible to farmers and agricultural specialists.
4. Practical Utility: The developed system provides a practical tool for jackfruit disease detection, helping farmers manage bacterial and fungal infections effectively. This solution bridges the gap between advanced AI technologies and field-based agricultural applications, ensuring scalability and usability.

In summary, the project delivered a comprehensive solution that combines robust machine learning techniques with practical deployment strategies, making significant strides in automated disease detection for jackfruit cultivation.

1.6 Organization of the Report

This report is systematically organized into six chapters, each addressing critical aspects of the research on jackfruit leaf disease detection using Convolutional Neural Networks (CNN), transfer learning, and an ensemble model approach.

The structure and content of the report are as follows:

Chapter 1: Introduction

This chapter provides a comprehensive introduction to the research topic, articulating the motivation, objectives, and methodology of the study. It also outlines the expected outcomes, establishing the context and significance of developing an automated system for jackfruit disease detection.

Chapter 2: Background

This chapter presents a detailed review of the existing literature, including similar applications and related research studies in agricultural disease detection. It identifies key gaps in current research on jackfruit leaf disease detection, justifying the need for the proposed study and highlighting the potential impact of AI and deep learning techniques in this domain.

Chapter 3: Research Methodology

This chapter elaborates on the methodology adopted for the study, including the dataset preparation, preprocessing steps, and model design. It describes the use of CNN architectures (VGG19, EfficientNetB0, MobileNetV2, ResNet50) and the ensemble model approach. Additionally, it details the model evaluation process, TensorFlow Lite (TFLite) conversion, and the development of a mobile application using Flutter for real-time disease detection.

Chapter 4: Implementation and Results

This chapter focuses on the technical aspects of the research, including the experimental setup, model training, and evaluation metrics. It presents a comprehensive analysis of the performance of individual CNN models and the ensemble model. Results are discussed critically, highlighting the accuracy and reliability of the system for jackfruit disease detection under real-world conditions.

Chapter 5: Engineering Standards and Design Challenges

This chapter addresses compliance with relevant software and hardware standards, particularly in the deployment of the TFLite model on mobile devices. It also examines the societal and environmental impacts of the proposed solution, ethical considerations in agricultural applications, and challenges encountered during model training, dataset augmentation, and mobile app integration.

Chapter 6: Conclusion

The concluding chapter synthesizes the key findings of the research, reflecting on its contributions to agricultural disease detection and its practical implications for jackfruit farming. It discusses the limitations of the current work and offers insights into potential future directions, including the integration of federated learning and broader AI applications for agricultural sustainability.

The report is structured to provide a logical progression of ideas, from the foundational aspects of the research to the practical outcomes and broader implications. This organization ensures a comprehensive understanding of the study and its significance in advancing automated solutions for agricultural disease management.

Chapter 2

Background

2.1 Introduction

The literature review investigates the application of deep learning, particularly CNNs, and transfer learning techniques in plant disease detection, with a focus on jackfruit and similar crops. Most studies utilize well-established architectures like VGG16, ResNet, and DenseNet for image-based classification, with several achieving high accuracy through customization and data augmentation techniques. While federated learning models have emerged to enhance data privacy and scalability, real-world deployment remains limited, with challenges in adapting models to diverse agricultural environments. The review also highlights the need for interpretability and farmer-friendly interfaces to facilitate practical implementation in rural settings.

2.2 Literature Review

Several researchers have focused on applying image processing and deep learning techniques to improve jackfruit disease detection, given the fruit's agricultural significance and susceptibility to disease. The integration of advanced technology for early and precise disease detection is essential in enhancing productivity and sustainability in jackfruit cultivation.

Saha [1] underscores the critical need for early disease detection in jackfruit to meet agricultural demands. They introduced a hybrid approach using YOLOv8 for object detection and EfficientNetB7 for classification, achieving high accuracy (99.9% training, 99.5% testing) in detecting algal and black spot diseases. This study exemplifies the application of deep learning architectures in agriculture, highlighting its potential for boosting agricultural efficiency and food security.

Suryavanshi [9] demonstrated a federated CNN approach, categorizing jackfruit leaf disease severity across six client nodes. Their model achieved strong metrics with accuracy between 94% and 98%, and the federated averaging technique facilitated a robust and scalable model while reducing data transfer. This model supports the potential for federated learning in precision agriculture, particularly in scenarios with limited connectivity and data privacy concerns.

Vats [3] implemented CNN models across decentralized clients for jackfruit leaf disease classification, achieving effectiveness through macro, micro, and weighted averages. The federated approach achieved balanced results across clients, with high accuracy rates indicating the model's resilience in a decentralized context. This study builds on federated learning's promise by demonstrating its ability to provide effective disease classification without compromising privacy.

Other researchers have explored transfer learning for recognising jackfruit-related diseases and similar species like Cempedak. Sumari [12] developed a custom CNN model and used transfer learning with VGG16, Xception, and ResNet50, achieving accuracies between 89% and 93.67%. Ong [13] also focused on jackfruit classification, comparing CNN performance with transfer learning models on pre-trained architectures and found that VGG16 provided the best results. Their findings suggest that transfer learning can be advantageous when data is limited, as pre-trained models can adapt effectively to agricultural datasets with visual similarities.

Considering practical implications, Habib [2] emphasized the lack of automated tools for detecting jackfruit diseases, particularly for rural farmers. They presented an agro-medical expert system employing k-means clustering for segmentation and multiple classifiers, with random forest achieving the highest accuracy at nearly 90%. This study's focus on image segmentation and classification provides a framework for real-world disease detection systems accessible to non-expert users.

Mobile applications incorporating CNNs have been proposed to make disease detection more accessible. Oraño [11] developed a mobile app using a CNN-based model for detecting pest and disease infestations, achieving a high accuracy of 97.87% in real-time detection. This mobile solution enhances accessibility for farmers, offering practical recommendations based on disease detection, thus supporting preventive agricultural measures.

Finally, Gomathi [10] implemented a CNN-based disease detection system that analyzes leaves, fruits, and tree trunks to detect multiple diseases, achieving 95% accuracy. This system's adaptability for mobile and web platforms underscores the role of accessible technology in promoting sustainable farming practices. Gomathi et al.'s approach aligns with the broader goal of using technology to improve agricultural outcomes in remote areas.

Table 2.1: Summary of Literature Reviewed.

Author(s) & Year	**Model(s) Used**	**Accuracy**	**Dataset Information**
Saha et al., 2023	YOLOv8, EfficientNetB7	99.5%	6,360 images of jackfruit diseases (algal and black spot)
Suryavanshi et al., 2023	Federated CNN	94-98%	Jackfruit leaf images, severity levels on 6 clients
Vats et al., 2024	Federated CNN	93.35%-97%	Five clients' decentralized data of jackfruit leaf diseases
Sumari et al., 2022	Custom CNN, VGG16, VGG19, Xception, ResNet50	89-93.67%	Own dataset of jackfruit and cempedak leaves
Ong et al., 2022	VGG16, Xception, ResNet50	VGG16 highest	Dataset of cempedak and nangka leaves
Habib et al., 2022	Random Forest, k-means	Approx. 90%	Images of jackfruit leaf diseases
Oraño et al., 2019	Sequential CNN (mobile app)	97.87%	2409 training, 516 validation images
Gomathi et al., 2023	CNN-based system	95%	600 images of jackfruit diseases
Manikandan et al., 2024	VGG16, ResNet, Inception	High accuracy not specified	PlantVillage and own dataset (jackfruit, guava, mango)

Pratondo et al., 2023	VGG16, InceptionV3	97.57%	204 images of jackfruit and cempedak leaves
Pen et al., 2022	Custom CNN, VGG16, ResNet50, Xception	87%	Own dataset of 4 Artocarpus species
Singh et al., 2024	Deep Transfer Learning (DTLD)	99.76%	4,000 images of mango leaf diseases
Thaseentaj et al., 2024	MLTNet (ResNet50+XAI)	94.3% (train), 86.3% (test)	1,275 original, 11,480 augmented images
Varma et al., 2024	VGG19, InceptionV3, ResNet152V2, DenseNet121	InceptionV3 99.87%	Dataset of mango leaf diseases
Jayanthi & Kumar, 2024	AlexNet, VGG-16, ResNet-50	94.54%-98.56%	Mendeley dataset for mango diseases
Mahmud et al., 2024	DenseNet78	99.47%	Small mango leaf dataset
Naralasetti et al., 2024	VGG16	96.56%	PlantVillage dataset
Srivastava et al., 2024	VGG16, MobileNetV2, Xception, InceptionV3, DenseNet121	MobileNetV2 98.9%	Mendeley and PlantVillage datasets
Zhou et al., 2024	ResNet-50	98%	16,060 crop leaf images (12 categories)

Yaswanth et al., 2024	Transfer Learning (various CNNs)	High accuracy	PlantVillage dataset with augmentation

2.2.1 Similar Applications

The field of agricultural disease detection has seen significant advancements through the integration of image processing and deep learning techniques. Numerous studies and applications focus on leveraging these technologies for similar purposes. Saha et al. (2023) introduced a hybrid approach using YOLOv8 for object detection and EfficientNetB7 for disease classification, achieving 99.5% testing accuracy on jackfruit leaf diseases like algal and black spots. Similarly, Gomathi et al. (2023) developed a CNN-based system capable of analyzing leaves, fruits, and trunks to detect diseases with 95% accuracy, emphasizing accessibility through mobile and web platforms. Oraño et al. (2019) created a mobile application using CNN models for real-time detection of pest infestations with an accuracy of 97.87%, offering practical recommendations to farmers.

Other contributions include federated learning approaches, as demonstrated by Suryavanshi et al. (2023), who achieved 94-98% accuracy across decentralized nodes for jackfruit disease classification, and Vats et al. (2024), whose federated CNN models provided balanced accuracy in decentralized setups. Transfer learning has also been extensively used in similar applications. Sumari et al. (2022) and Ong et al. (2022) employed pre-trained models like VGG16 and ResNet50 to classify jackfruit and related species, highlighting the adaptability of these models to limited datasets.

The practical application of these systems is evident in the development of mobile solutions and user-friendly tools that enhance accessibility. Habib et al. (2022) proposed an agro-medical expert system combining k-means clustering and multiple classifiers, achieving 90% accuracy in real-world disease detection. These applications underscore the potential of integrating advanced AI techniques into agriculture to support farmers in disease management and prevention.

2.2.2 Related Research

Extensive research has been conducted on the application of image processing and deep learning models for agricultural disease detection, particularly for tropical fruits like jackfruit. Saha et al. (2023) demonstrated the effectiveness of combining object detection with classification models, such as YOLOv8 and EfficientNetB7, achieving exceptional accuracy rates. These methodologies are pivotal in improving agricultural productivity by enabling precise disease detection.

The role of federated learning in addressing privacy and scalability challenges has been explored by Suryavanshi et al. (2023) and Vats et al. (2024), both highlighting the robustness of federated CNNs in decentralized environments. These studies underline the potential of federated learning to create scalable models without the need for centralized data collection, making them suitable for remote agricultural applications.

Transfer learning has been another key focus area. Sumari et al. (2022) and Ong et al. (2022) demonstrated the successful adaptation of pre-trained models like VGG16 and ResNet50 to agricultural datasets, achieving high accuracy with limited data. These findings align with research by Mahmud et al. (2024), who achieved 99.47% accuracy using DenseNet78 on a mango leaf dataset, further showcasing the effectiveness of transfer learning in agricultural disease detection.

Mobile and web-based solutions for disease detection are also prominent in related research. Oraño et al. (2019) and Gomathi et al. (2023) emphasized the importance of real-time, accessible solutions for farmers. Their studies focus on deploying lightweight CNN models on mobile platforms, enabling disease detection and prevention measures directly in the field.

These studies collectively establish the significance of leveraging deep learning and transfer learning methodologies in agriculture. They highlight how advanced models can address challenges such as limited datasets, privacy concerns, and accessibility, thereby paving the way for innovative, real-world applications in disease management.

2.3 Gap Analysis

The reviewed studies highlight the effectiveness of CNN-based deep learning models and transfer learning in detecting plant diseases, with particular success in jackfruit, mango, and similar crops. However, several research gaps emerge:

- While numerous studies focus on specific diseases in jackfruit or mango leaves, there is a need for models that encompass a broader range of diseases across multiple crop types. Expanding models to handle diverse agricultural diseases beyond isolated cases can enhance their real-world applicability, especially in mixed crop systems.

- Many models achieve high accuracy in controlled environments but lack testing in real-world agricultural settings. This gap affects the robustness of models under varying lighting, background, and environmental conditions. The development of models that can perform consistently in uncontrolled outdoor settings remains an open challenge.

- Although data augmentation is used to expand training data, most studies rely on limited datasets. Large-scale, labelled datasets for diverse diseases in jackfruit and other tropical crops are necessary for building more comprehensive models. Establishing a centralized, open-access dataset can drive advancements across studies and allow for more effective cross-validation of results.

2.4 Summary

This chapter presented a comprehensive review of recent studies on deep learning models for jackfruit and crop disease detection. Key findings underscore the effectiveness of CNNs and transfer learning for high-accuracy disease classification, particularly in controlled environments. However, gaps remain in scalability, data privacy, and real-world applicability. These insights establish a foundation for developing robust, accessible, and privacy-preserving models that address the practical needs of agriculture, driving the subsequent focus of this research.

Chapter 3

Research Methodology

3.1 Methodology

3.1.1 Overview

This study aims to develop an automated system for detecting various bacterial and fungal diseases affecting jackfruit through advanced image processing and deep learning techniques. The methodology is structured into multiple stages to ensure accurate and reliable disease classification, leveraging a dataset primarily collected specifically for jackfruit disease detection. The process begins with preprocessing steps applied to the collected dataset, which includes normalization, image resizing, adjustment of dots per inch (DPI), and data augmentation. Following preprocessing, several Convolutional Neural Network (CNN) architectures, including VGG19, EfficientNetB0, MobileNetV2, and ResNet50, are evaluated for disease classification. Each model is trained and tested independently to identify the best-performing model based on accuracy and reliability metrics. The model with the highest performance is saved in a .h5 format, enabling further utilization in an ensemble approach. This ensemble model integrates predictions from the individual models, aiming to enhance overall classification accuracy by combining the strengths of each architecture. For deployment, the best model is converted into a TensorFlow Lite (TFLite) format, allowing for integration into a mobile application built with Flutter. This application provides end-users with an accessible and user-friendly interface to detect and report jackfruit diseases in real time, making the model's capabilities available for practical, field-based disease detection.

3.1.2 Proposed Methodology

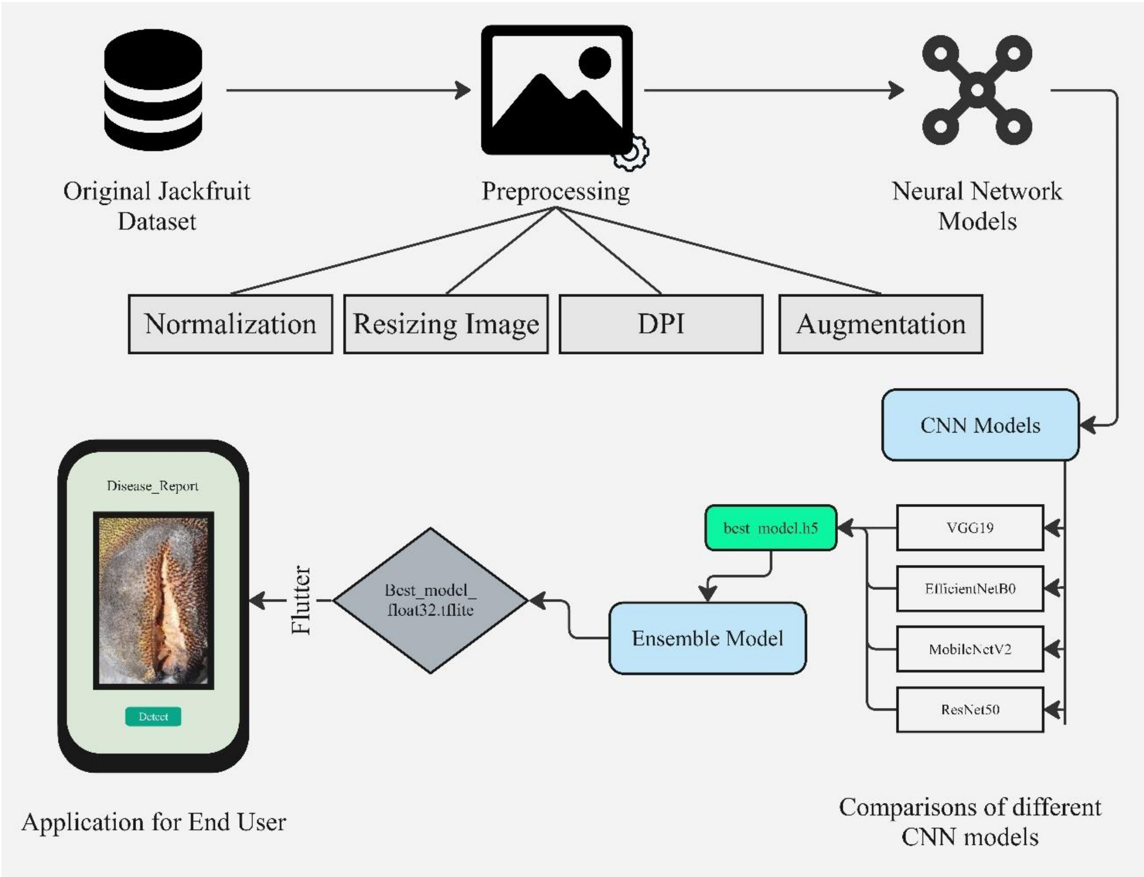


Figure 3.1: The Methodological Flowchart

3.1.3 Functional and Nonfunctional Requirements

Functional Requirements:

- Disease Detection and Classification: The system must accurately detect and classify jackfruit diseases (Healthy, Bacteria Affected, and Fungus Affected) using various CNN models (VGG19, MobileNetV2, EfficientNetB0, ResNet50) and an ensemble model.

- Data Preprocessing: Input images should undergo preprocessing steps such as normalization, resizing (224 × 224 pixels), contrast enhancement, and augmentation to ensure consistent and robust model performance.

- Model Evaluation: The system must evaluate all models based on key performance metrics, including accuracy, F1-score, precision, recall, and AUC scores, to compare their effectiveness.

- Data Integration: Seamless integration with cloud-based storage systems, such as Google Drive, is required for accessing datasets and storing model checkpoints.

- Ensemble Model Implementation: The system must implement and optimize an ensemble approach that combines the predictions of individual CNN models for improved accuracy and robustness.

- Result Visualization: The platform should provide clear visual outputs such as accuracy and loss curves, confusion matrices, and ROC curves to facilitate comprehensive result interpretation.

- Resource Optimization: The system should dynamically allocate computational resources, leveraging GPU/TPU on platforms like Google Colab to optimize performance for training and testing.

Nonfunctional Requirements:

- Scalability: The system must be designed to handle large datasets and be adaptable for additional crop disease classifications in the future.

- Efficiency: The models must demonstrate high computational efficiency, ensuring compatibility with resource-constrained environments like mobile devices or IoT systems.

- Usability: The platform should feature an intuitive and user-friendly interface, enabling access for farmers, agricultural professionals, and non-technical users.

- Reliability: The system must deliver reliable performance across diverse environmental conditions, dataset characteristics, and image qualities.

- Security: All stored data, including datasets and models, must be securely protected, with access restricted to authorized users and compliance with data privacy standards.

- Maintainability: The codebase and models should follow modular design principles, with comprehensive documentation to facilitate maintenance, updates, and scalability.

- Compliance: The system must adhere to ethical and software standards, promoting responsible AI usage in agriculture and ensuring transparency in its decision-making process.

This section defines the key operational and technical requirements to ensure the system meets its objectives effectively and efficiently.

3.1.4 Data Flow Diagram Level 1

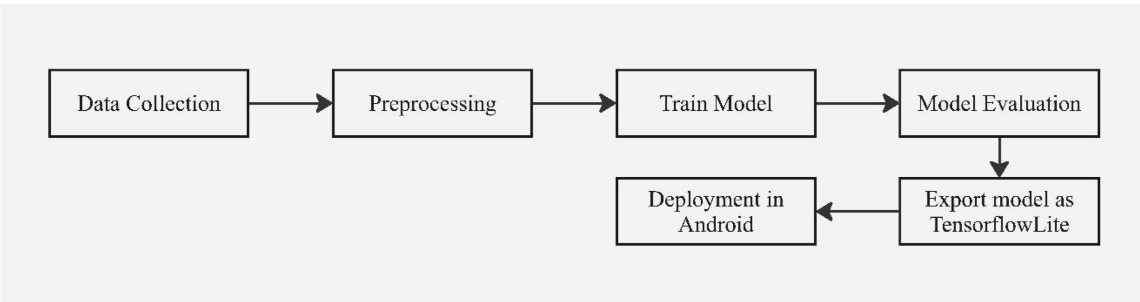


Figure 3.2: Data Flow Diagram Level -1

3.1.5 UI Design

1. Application Setup

The application is designed to deliver real-time disease detection for jackfruit leaves directly to end-users, such as farmers or agricultural specialists, through a mobile interface. This setup leverages a lightweight deep learning model, optimized and deployed on mobile devices to allow users to upload or capture images of jackfruit leaves and receive instant classification results. The best-performing ensemble model, saved as a TensorFlow Lite (TFLite) file for mobile compatibility, is integrated into the application. The app is developed using Flutter, a cross-platform UI toolkit, which allows seamless deployment across both Android and iOS devices. The application follows a Two-Tier Architecture, comprising the Client Tier and the Application Tier. This architecture separates the user interface from the underlying model processing, ensuring a clean and efficient structure suitable for mobile deployment.

2. Two-Tier Architecture

2.1 **Client Tier (Presentation Layer)**

The Client Tier, or Presentation Layer, represents the interface that the end-user interacts with. Built using Flutter, this layer manages all user interactions and handles the display of information. Its main components include:

- Image Capture/Upload: The Client Tier allows users to either capture a photo using their device's camera or upload an existing image from their gallery. This image is then passed to the application tier for processing.

- Result Display: After processing, the result from the model inference is displayed on this tier, showing the predicted class (bacteria-affected, fungus-affected, or healthy) and any additional information or recommendations.

This tier is entirely user-centric, ensuring that users can navigate the application with ease and minimal instructions. By using Flutter, the application maintains consistency across different platforms (Android and iOS), providing uniform experience to all users.

2.2 **Application Tier (Model Processing Layer)**

The Application Tier, or Model Processing Layer, is responsible for executing the machine learning model and handling data processing. It is embedded within the mobile device to enable offline functionality and includes the following components:

- TFLite Model Inference: This component loads the pre-trained TFLite model and processes the input image to classify it. The lightweight TFLite format is optimized for mobile devices, ensuring that the model can run efficiently even on devices with limited computational power.

- Return Results: After completing the inference, this layer sends the prediction result back to the Client Tier, where it is displayed for the user.

This separation of concerns in the two-tier architecture allows for a lightweight client that focuses on user interaction, while the application tier performs the intensive processing. The application setup and architecture ensure quick, real-time disease detection, providing an efficient tool for field use without relying on cloud-based processing.

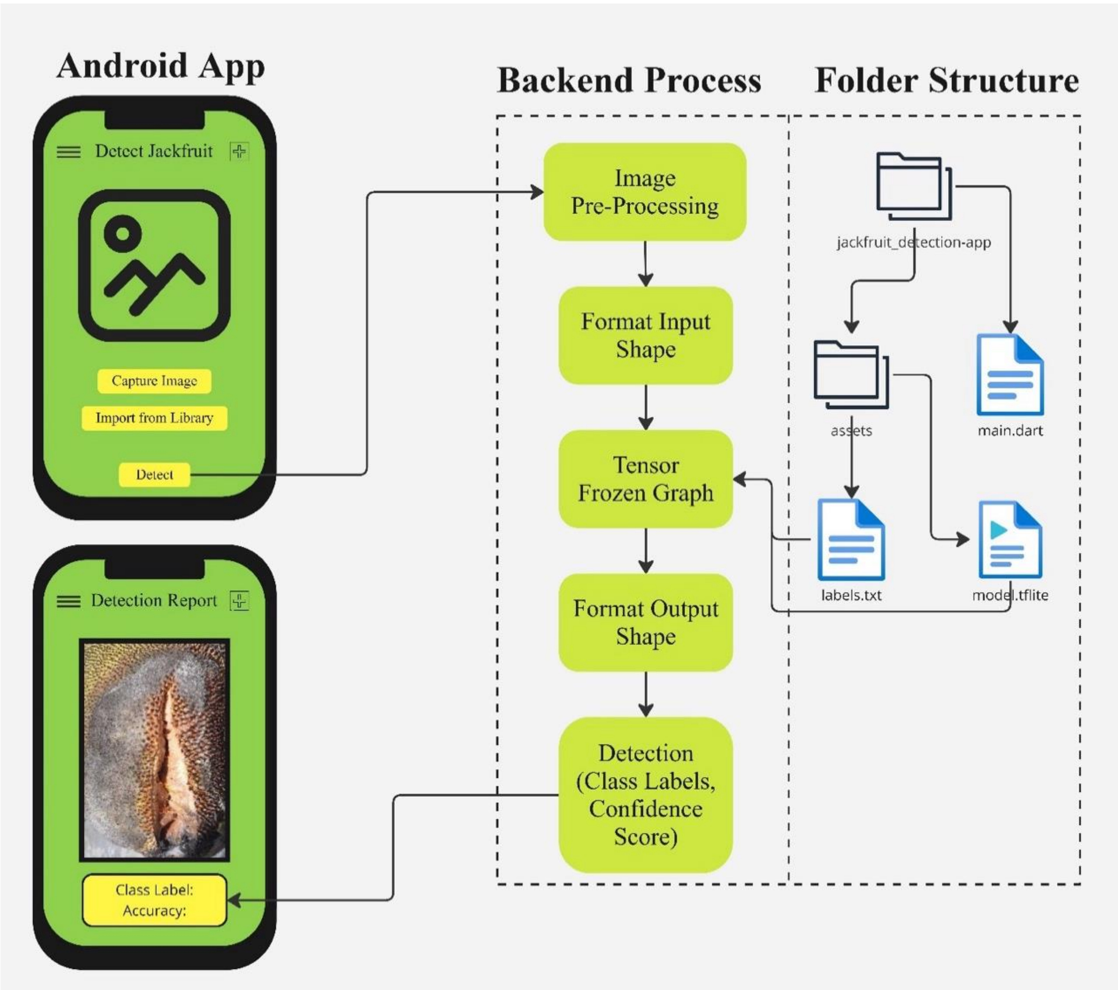


Figure 3.3: The Two-Tier Architecture and UI for Application

3.2 Detailed Methodology and Design

3.2.1 Data Collection

The dataset employed in this study consists of images of jackfruit leaves collected predominantly through fieldwork, ensuring that the images reflect real-world conditions encountered in agricultural settings. The dataset was assembled specifically to capture a wide range of visual features associated with healthy, bacterial-affected, and fungal-affected jackfruit leaves, providing a rich foundation for developing a reliable disease classification model.

3.2.2 Original Dataset and Augmentation

The images were originally captured at a resolution of 3024 × 4032 pixels using an iPhone SE, and subsequently resized to 240 × 240 pixels in JPEG format to optimize storage and processing requirements. The initial dataset contained a total of 3,045 images, which were deemed insufficient for training a deep learning model with high accuracy and generalization capability. To address this, data augmentation techniques were applied to enhance the dataset. Augmentation involved various transformations, including flipping, rotation, scaling, and adjustments in brightness and contrast, which introduced variations in the images. By expanding the dataset to 6,090 images, this process effectively doubled the original dataset and provided the model with a more diverse set of inputs, helping to reduce overfitting and improving the model's adaptability to new data. Each augmented image preserves the core characteristics of the original class, ensuring that the data remains representative of the three primary classes.

3.2.3 Class Labels

The dataset is organized into three distinct classes, each representing a specific condition of jackfruit leaves:

1. Bacteria_effected: Images in this category show leaves exhibiting symptoms of bacterial infections, which may include spots, lesions, or other signs specific to bacterial pathogens.
2. Fungus_effected: This class consists of images displaying fungal infections, characterized by distinctive patterns or textures associated with fungal growth, such as discoloration or mold-like structures.
3. Healthy: This class includes images of leaves without any visible disease

symptoms, serving as a baseline for comparison against infected leaves.

These classes enable the model to distinguish not only between healthy and diseased leaves but also to identify the specific type of disease present. Such classification is crucial for targeted disease management, as bacterial and fungal infections often require different treatment approaches.



Bacteria_effected Fungus_effected Healthy

Figure 3.4: Sample Image of Each Class

3.2.4 Preprocessing Steps

To standardize the images and improve the efficiency of the deep learning models, a series of preprocessing steps were applied:

1. Normalization: This step scales pixel values to a common range, typically between 0 and 1, to facilitate faster convergence during training and to enhance the stability of the model.
2. DPI Adjustment: Adjusting the dots per inch (DPI) of each image ensures consistency in resolution, making the images uniform in quality. This is particularly important when the dataset contains images captured from various sources or devices with different resolutions.
3. Image Resizing: All images were resized to a standard dimension, aligning with the input requirements of the CNN models used in this study. This resizing not only ensures compatibility with the models but also reduces computational load, enabling faster processing without compromising image quality.

These preprocessing steps ensure that the images are consistent in quality and format, thus facilitating efficient training and minimizing potential issues arising from variations in image size or resolution.

Table 3.1: Dataset Specifications

Properties	Values
Image Resolution (Original)	3024 x 4032 pixels
Image Resolution	240 × 240 pixels
Format	.jpg
Total Images (Original)	3045
Total Images	6090
Classes	3

3.2.5 Convolutional Neural Network (CNN) Algorithm

A Convolutional Neural Network (CNN) is a deep learning architecture widely used for image classification and other computer vision tasks due to its ability to automatically learn and extract relevant features from images. In this study, CNNs serve as the foundation for detecting and classifying bacterial and fungal diseases in jackfruit leaves. Below is a step-by-step breakdown of the CNN algorithm, including key mathematical procedures.

1 Input Layer

The input to a CNN is an image represented as a tensor. For an RGB image of dimensions H×W×3, where H is the height, W is the width, and 3 denotes the three-color channels (Red, Green, Blue), the input tensor X can be expressed as:

$$X \in \mathbb{R}^{H \times W \times 3} \qquad \text{(i)}$$

2 Convolutional Layer

The convolutional layer applies multiple filters (or kernels) to the input image to detect specific features, such as edges, textures, or patterns. A filter K of size f×f slides over the input image, performing an element-wise multiplication and summing up the result, creating a feature map. This operation is defined by:

$$Z_{i,j}^{(l)} = \sum_{m=0}^{f-1} \sum_{n=0}^{f-1} X_{i+m,j+n}^{(l-1)} K_{m,n} + b \qquad \text{(ii)}$$

where:

- Z_{i-j}^{l} is the output feature map at position (i, j) in layer l.
- $X^{(l-1)}$ is the input from the previous layer,
- K is the kernel of size f × f,
- b is the bias term.

3 Pooling Layer

After convolution, the feature maps undergo a pooling operation, which reduces the spatial dimensions (height and width) of the feature maps. This step helps in reducing the computational load and enables the model to become invariant with small translations in the image. The most common type of pooling is Max Pooling.

4 Flattening

After multiple convolutional and pooling layers, the high-dimensional feature maps are "flattened" into a one-dimensional vector to prepare the data for fully connected layers. Flattening transforms the pooled feature map into a column vector.

5 Fully Connected Layer (Dense Layer)

The fully connected layer (or dense layer) processes the flattened vector to generate the final classification output. Each neuron in the dense layer performs a weighted sum of its inputs, followed by the application of an activation function (often ReLU for hidden layers and Softmax for the output layer). For a dense layer, the output O^l is given by:

$$O^{(l)} = W^{(l)} \cdot X^{(l-1)} + b^{(l)} \qquad \text{——————} \qquad \text{(iii)}$$

where:

- W^l is the weight matrix,
- $X^{(l-1)}$ is the input from the previous layer,
- $b^{(l)}$ is the bias term.

In the output layer, a Softmax activation function is used for multi-class classification to convert the output scores into probabilities. The Softmax function for the i-th class is:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \qquad \text{——————} \qquad \text{(iv)}$$

where:

- z_i is the output score for class i,
- C is the total number of classes.

The Softmax output for each class represents the probability that the input image belongs to that class.

6 Loss Function and Optimization

During training, CNNs use a loss function to measure the difference between predicted and actual labels. For multi-class classification, **categorical cross-entropy loss** is commonly used, defined as:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \qquad \text{(v)}$$

where:

- y_i is the true label (0 or 1) for class i.
- $\hat{y}_i$ is the predicted probability for class i.

The loss function is minimized using an optimization algorithm, typically Stochastic Gradient Descent (SGD) or a variant like Adam, which adjusts the weights in the network to minimize the error.

7 Backpropagation

CNN uses backpropagation to update the weights in each layer. The gradients of the loss function with respect to each weight are calculated, and the weights are adjusted in the opposite direction of the gradient to minimize the loss.

8 Iterative Training and Validation

The CNN model is trained iteratively over multiple epochs, where each epoch represents a complete pass through the training data. After each epoch, the model's performance is evaluated on a separate validation set to monitor its progress and avoid overfitting. Early stopping may be employed to halt training if the validation performance plateaus, ensuring the model maintains good generalization on unseen data.

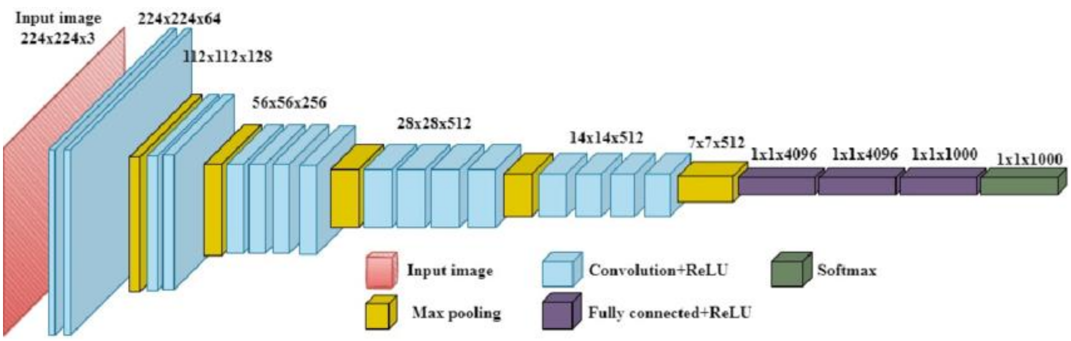


Figure 3.5: The Architecture of Convolutional Neural Network [1]

3.2.6 Comparison between CNN Models

Here's a table summarizing the key parameters and differences among the four CNN architectures: VGG19, EfficientNetB0, MobileNetV2, and ResNet50.

Table 3.2: Comparison of Key Parameters between Four CNN Models

Feature	VGG19	EfficientNetB0	MobileNetV2	ResNet50

Primary Use Case	General-purpose, high-accuracy image classification	Balanced efficiency and accuracy, mobile-friendly	Designed for mobile and embedded applications	Deep learning tasks requiring high accuracy, complex feature extraction
Total Layers	19 layers	18 layers	53 layers	50 layers
Parameter Count	~140 million	~5.3 million	~3.4 million	~25 million

3.2.7 Ensemble Model Algorithm

An ensemble model combines the predictions of multiple individual models to produce a final prediction that is often more accurate and robust than any single model. The ensemble method leverages the strengths of each model in the ensemble to reduce errors and enhance overall performance. In this ensemble approach, we leverage multiple instances of the best-performing model (stored as an .h5 file) to improve the robustness and accuracy of the disease detection system. The ensemble method used here aggregates predictions from multiple models to produce a final, more reliable classification output.

1 Load the Best-Performing Model

The first step is to load the saved .h5 file of the best-performing model from the four CNN architectures (VGG19, EfficientNetB0, MobileNetV2, or ResNet50). The .h5 file contains the model architecture, weights, and any other parameters necessary for inference.

2 Data Preparation

The input data X (test or validation set) is preprocessed in the same way as the training data, ensuring consistency. This includes normalization, resizing, and any other preprocessing applied to the dataset initially. The preprocessed data X is then fed to each model in the ensemble.

3 Model Prediction

Each model instance $model_i$ in the ensemble independently generates a prediction for each input sample. For an input X_j (where j represents a particular sample), each model produces a probability vector $P_{i,j}$ representing the likelihood of the sample belonging to each class:

$$P_{i,j} = \text{model}_i(X_j), \quad \text{for } i = 1, 2, \ldots, N \qquad \text{(vi)}$$

where:

- $P_{i,j} = [p_{i,j}^{(1)},\ p_{i,j}^{(2)}, \dots \dots \dots, p_{i,j}^{(C)}]$,
- $p_{i,j}^{(c)}$ represents the predicted probability of sample j belonging to class c,
- C is the total number of classes (in this case, three: Bacteria_effected, Fungus_effected, and Healthy).

4 Aggregating Predictions

To obtain the final prediction for each input sample, the individual predictions from each model in the ensemble are combined. Two common aggregation techniques are average probability voting and majority voting.

1. Average Probability Voting: In this method, the final prediction probability vector for each sample X_j is calculated by taking the average of the probability vectors from all models in the ensemble.
2. Majority Voting: In majority voting, each model produces a class label for each input sample, and the final class label is the one that appears most frequently across the models.

5 Ensemble Output

The ensemble model produces a final prediction for each input sample X_j based on the chosen aggregation method. The output is a class label (e.g., Bacteria_effected, Fungus_effected, or Healthy) that reflects the consensus of the ensemble, making the classification more robust than relying on a single model's output.

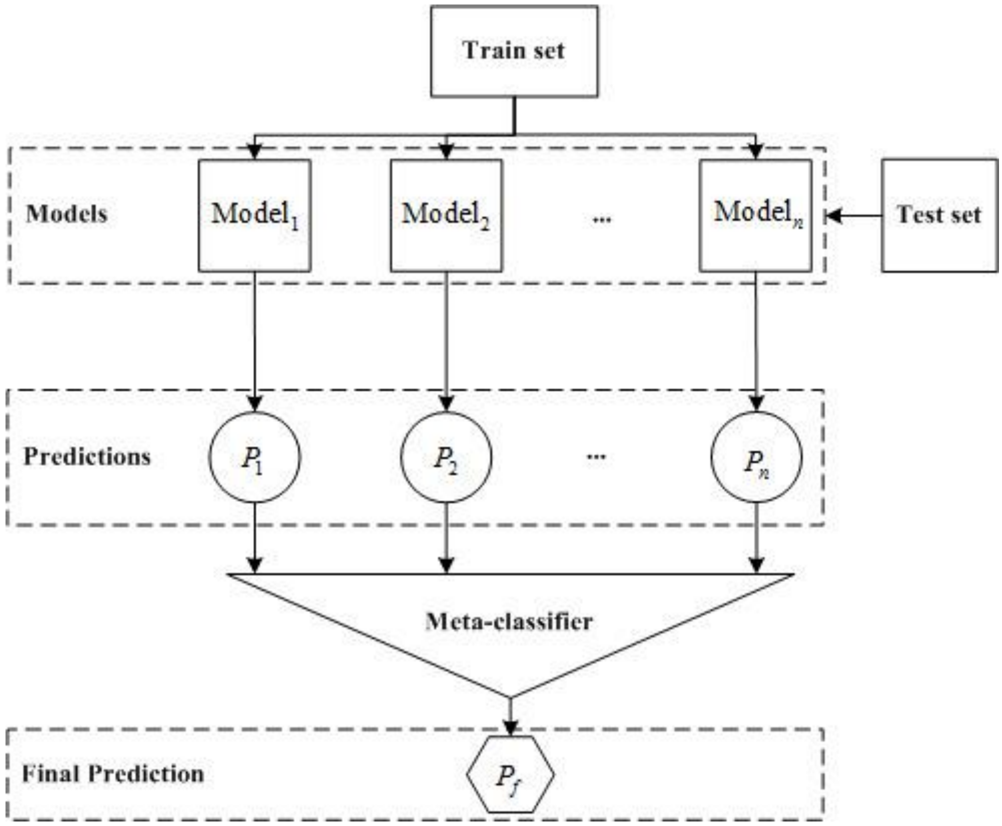


Figure 3.6: The Architecture of Ensemble Model [2]

3.3 Project Plan

This thesis followed a structured project management approach with distinct phases. It began with a literature review and problem definition to refine objectives. Fieldwork was conducted to collect jackfruit leaf images, followed by data preprocessing to enhance model performance. For model selection and training, four CNN architectures (VGG19, EfficientNetB0, MobileNetV2, and ResNet50) were evaluated, and the best-performing model was saved and converted to TensorFlow Lite (TFLite) for mobile deployment. A cross-platform mobile application was developed using Flutter, embedding the TFLite model for real-time disease detection. The app underwent testing and optimization to improve accuracy and user experience. Finally, all project stages were documented, leading to the completion of the final thesis.

Table 3.3: GANTT Chart of Project Timeline

Process	May’24	June’24	July’24	Aug’24	Sep’24	Oct’24	Nov’24	Dec’24
Working Plan								
Theoretical Study								
Literature Review								
Data Collection								
Data Preprocessing								
Model Design								
Methodology Writing								
Report Writing								
Review and Finalization								

3.4 Task Allocation

DATE	TASK
10.05.2024 - 18.06.2024	Data collection
28.06.2024 - 30.09.2024	Data Pre-processes, Related paper analysis and Model Apply
01.10.2024 - 28.11.2024	Paper Writing.

3.5 Summary

The methodology for this thesis involved a comprehensive process, from dataset preparation to mobile application deployment, aimed at developing a robust model for detecting bacterial and fungal diseases in jackfruit leaves. The project began with field data collection. Preprocessing techniques such as normalisation, resizing, DPI adjustment, and data augmentation were applied to enhance data quality and increase diversity, resulting in a final dataset of 6,090 images (double the original dataset). The next phase focused on model selection and training. Four Convolutional Neural Network (CNN) architectures were evaluated on the augmented dataset using Google. Each model was tested for performance, and the best-performing model was saved as a .h5 file. To further improve robustness, an ensemble approach was applied, combining multiple instances of the best model to increase accuracy and generalization. This ensemble model was then converted to TensorFlow Lite (TFLite) format, optimizing it for mobile compatibility and allowing for efficient on-device inference. The mobile application, developed using Flutter for cross-platform compatibility, incorporated the TFLite model, enabling real-time disease detection on both Android and iOS devices. The app was designed with a user-friendly interface, allowing users to capture or upload leaf images and receive instant classification results, even in offline scenarios.

Chapter 4

Implementation and Results

4.1 Environment Setup

4.1.1 Train Model

The table outlines the core training parameters applied across four pretrained Convolutional Neural Network (CNN) models—EfficientNetB0, VGG19, MobileNetV2, and ResNet50—used in this study. The image size is set to 224 × 224 pixels, meaning each input image is resized to this resolution before being fed into the models. This standard size is commonly used in pretrained CNN architectures, balancing computational efficiency with enough detail for accurate feature extraction. A batch size of 128 indicates that 128 images are processed together in each iteration before the model's weights are updated. This relatively large batch size allows for faster training due to fewer updates and more stable gradient estimates, although it requires adequate computational resources. The models are trained for 15 epochs, meaning the entire dataset is passed through the models 15 times, which is a reasonable choice for fine-tuning pretrained models without risking overfitting. The Adam optimizer is used, which combines the advantages of the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). Adam is known for its adaptability in adjusting learning rates, making it suitable for handling sparse gradients and noisy data often encountered in CNN training. The learning rate is set to 0.0001, a low value that enables gradual convergence, especially important when fine-tuning pretrained models to avoid large adjustments that could disrupt previously learned weights. Overall, these training parameters are chosen to optimize the performance of the four CNN models, ensuring stable and effective fine-tuning for the classification task at hand.

Table 4.1. Common parameter table for all experimented models.

Parameter Name	Parameter Value

Image Size	224 × 224
Batch Size	128
Epoch	15
Optimizer	Adam
Learning Rate	0.0001

The table describes the dataset split used for training, validation, and testing across all pretrained CNN models, providing a consistent structure for evaluation. The train set comprises 70% of the dataset, totaling 3897 images, and serves as the primary data used to teach the models to recognize patterns and features. This substantial portion of data supports effective learning, allowing the models to capture the complexity of the dataset and improve their classification accuracy.

The validation set, with 975 images, represents 10% of the dataset. This subset is used to monitor the model's performance during training and helps fine-tune hyperparameters. By assessing the model on unseen validation data, we can gain insights into its generalization ability, enabling adjustments to prevent overfitting and improve stability.

Lastly, the test set accounts for 20% of the data, with 1218 images, reserved exclusively for final evaluation after complete training. This test set provides an unbiased measure of the model's real-world performance on unseen data, reflecting its effectiveness in making accurate predictions outside the training environment. This structured data split—70% for training, 10% for validation, and 20% for testing—ensures a balanced and rigorous approach to model development and evaluation, allowing for reliable comparisons of each model's accuracy and generalization capability.

Table 4.2. Common data split for all experimented models.

Dataset	**In Percentage**	**Number of Images**
Train set	70%	3897
Validation Set	10%	975
Test Set	20%	1218

4.1.2 Model Evaluation

In evaluating the effectiveness of machine learning models for the study, appropriate performance metrics must be used to provide insights into model accuracy, reliability, and generalization capabilities. The following metrics are commonly used in classification tasks, especially in the context of agricultural disease detection:

1 Accuracy

The proportion of correctly classified instances (both positive and negative) to the total instances. Accuracy gives a quick overview of model performance but can be misleading in imbalanced datasets where one class significantly outnumbers the other.

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

2 Recall

The ratio of correctly predicted positive observations to all actual positives. Recall is particularly important in scenarios where a positive case (such as a diseased plant) could lead to severe consequences, like crop loss.

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

3 Precision

The ratio of correctly predicted positive observations to the total predicted positives. Precision is crucial in applications where the cost of false positives is high. In this study, high precision indicates that when a disease is predicted, it is likely to be true.

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

4 F1-Score

The harmonic meaning of precision and recall, providing a balance between the two metrics. The F1 score is especially useful when dealing with imbalanced classes, as it considers both false positives and false negatives, offering a more comprehensive view of model performance.

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

5 Confusion Matrix

A table used to describe the performance of a classification model, showing the true vs. predicted classifications. The confusion matrix provides insights into the types of errors made by the model, allowing for more targeted improvements.

6 Receiver Operating Characteristic (ROC) Curve

A graphical representation of a classifier's performance across various threshold settings, plotting the true positive rate (recall) against the false positive rate. It illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It plots the True Positive Rate (TPR), or recall or sensitivity, against the False Positive Rate (FPR) at various threshold settings.

Area Under Curve (AUC): The area under the ROC curve provides a single metric to assess model performance; a value closer to 1 indicates a better model.

In this study, the Receiver Operating Characteristic (ROC) curve and Area Under the Curve (AUC) are crucial for evaluating classification models. They allow for visual comparisons of model performance across various thresholds, facilitating optimal threshold selection by balancing sensitivity and specificity. Additionally, the ROC curve is robust against class imbalances, providing reliable assessments where accuracy may mislead. However, ROC and AUC should complement other metrics like precision and recall for a comprehensive evaluation of model effectiveness.

4.2 Testing and Evaluation/Performance/ Comparative Analysis

4.2.1 Comparative Analysis

1. Performance comparison of the experimented models

The ensemble model outperforms all individual CNN models by achieving perfect classification performance. While VGG19 and MobileNetV2 show strong accuracy, stable learning, and high AUC scores, they still experience minor misclassifications and slight variations in metrics across classes. ResNet50 and EfficientNetB0 exhibit moderate to considerable fluctuations in performance, with ResNet50 achieving decent metrics but struggling with specific classes, and EfficientNetB0 showing instability in learning and relatively lower accuracy.

The ensemble model combines the strengths of each CNN, effectively offsetting the limitations of individual models such as EfficientNetB0's lower generalization and ResNet50's class-specific challenges. This collective approach results in a model with 100% validation and 99.83% testing accuracy, perfect precision, recall, F1-scores, and AUC scores across all classes. The ensemble model’s flawless classification performance makes it the most reliable and effective choice among all models tested, demonstrating that an ensemble of CNNs can significantly enhance performance by combining the predictive power of diverse architectures. This finding underscores the value of ensemble methods for achieving high reliability and robustness in jackfruit leaf disease detection and classification tasks.

Table 4.3. Performance analysis among all the experimented models.

Model Name	Validation Accuracy	Testing Accuracy	AUC score
VGG19	95. 90%	95.84 %	1.00
EffecientNetB0	70.77 %	70.11 %	0.95
MobileNetv2	93.44 %	93.35 %	1.00
ResNet50	88.21 %	88.26 %	1.00
Ensemble model	100 %	99.83 %	1.00

2. Individual deep CNN models' VS Ensemble model

- Confusion matrix analysis

  The confusion matrices for each CNN model reveal their varying levels of effectiveness in accurately classifying the three classes: Bacteria Affected, Fungus Affected, and Healthy. VGG19 and MobileNetV2 demonstrate high accuracy across classes, with minimal misclassifications. For example, both models correctly identify the majority of Healthy instances, reflecting strong performance. However, EfficientNetB0 shows considerable misclassifications, especially in distinguishing between Bacteria Affected and Fungus Affected classes, which may indicate limitations in capturing fine-grained features. ResNet50 performs moderately, with some misclassifications, particularly within the Bacteria Affected class, suggesting it occasionally struggles with subtle differences in feature representation.

  The ensemble model, however, achieves flawless classification across all classes, as evidenced by a perfect diagonal in its confusion matrix with no off-diagonal values. This means that the ensemble model correctly identifies all instances of each class, with zero misclassifications. By aggregating predictions from all individual CNN models, the ensemble effectively reduces the likelihood of misclassification by offsetting individual model errors, leading to highly reliable classification results across all classes.

- Classification Report Analysis

  The classification report for each CNN model provides further detail on precision, recall, and F1-score, highlighting strengths and weaknesses for each class. VGG19 demonstrates excellent performance with a high overall accuracy of 95% and balanced precision, recall, and F1-scores across classes, particularly excelling in the Healthy class with an F1-score of 1.00. MobileNetV2 also shows strong metrics, achieving an accuracy of 93%, though slightly lower than VGG19, with high F1-scores for Bacteria Affected and Fungus Affected. EfficientNetB0 lags behind, with the lowest accuracy of 70% and lower F1-scores, particularly for the Bacteria Affected class, indicating it struggles with distinguishing complex features. ResNet50 achieves an accuracy of 88% and performs well in the Healthy class, although it shows moderate recall in Bacteria Affected, suggesting some difficulty in identifying all instances of this class.

  In contrast, the ensemble model achieves perfect scores across all metrics, with precision, recall, and F1-scores all at 1.00 for each class, resulting in an overall accuracy of 100%. This impeccable performance underscores the ensemble model's ability to leverage the strengths of each individual CNN, minimizing both false positives and false negatives across classes. The perfect classification report for the ensemble indicates its high reliability, as it consistently identifies each class without any misclassification, making it highly effective for applications where accuracy is paramount.

- ROC Curve and AUC Analysis

  The ROC curves and AUC scores provide additional insight into each model's discriminative capability. Both **VGG19** and **MobileNetV2** show high AUC scores nearing 1.0 across all classes, reflecting strong sensitivity and specificity in distinguishing between *Bacteria Affected*, *Fungus Affected*, and *Healthy*. **ResNet50** also performs well in terms of AUC, though with slight variations, indicating good but not perfect discrimination capability. **EfficientNetB0** has lower AUC scores, aligning with its poorer classification performance, suggesting that it struggles to maintain high discrimination power between classes, particularly for similar-looking conditions.

The ensemble model, on the other hand, achieves AUC scores of 1.0 across all classes, indicating perfect discriminative ability. This means that the ensemble model has maximum sensitivity and specificity, completely separating positive and negative instances for each class. The perfect AUC scores reinforce the ensemble's superiority, as it demonstrates flawless class separation, thus confirming that it is highly effective in distinguishing between different conditions without overlap.

In summary, all the comparison analyses among individual deep CNN models and the proposed ensemble model state that the proposed ensemble model is the best-performing model for jackfruit disease analysis. The ensemble model outperformed other individual models with at least 4% accuracy.

3. Mobile-based Application performances

The mobile-based application, "Jackfruit-Doctor," provides a user-friendly interface for disease detection, making the AI-based tool accessible to farmers and users in real-world scenarios. The application allows users to take photos directly or upload images from their gallery for analysis. The app demonstrates high accuracy in classifying jackfruit conditions, with a performance that aligns closely with the results of the ensemble model. It identifies Bacteria Affected Fruit with an accuracy of 94%, showcasing its reliability even in complex scenarios. Additionally, the app achieves 100% accuracy in detecting both Healthy Fruit and Fungus Affected Fruit, highlighting its precision in distinguishing between diseased and healthy conditions. These results confirm that the mobile implementation retains the robustness and reliability of the underlying model, making it a practical and efficient tool for supporting farmers in disease management.

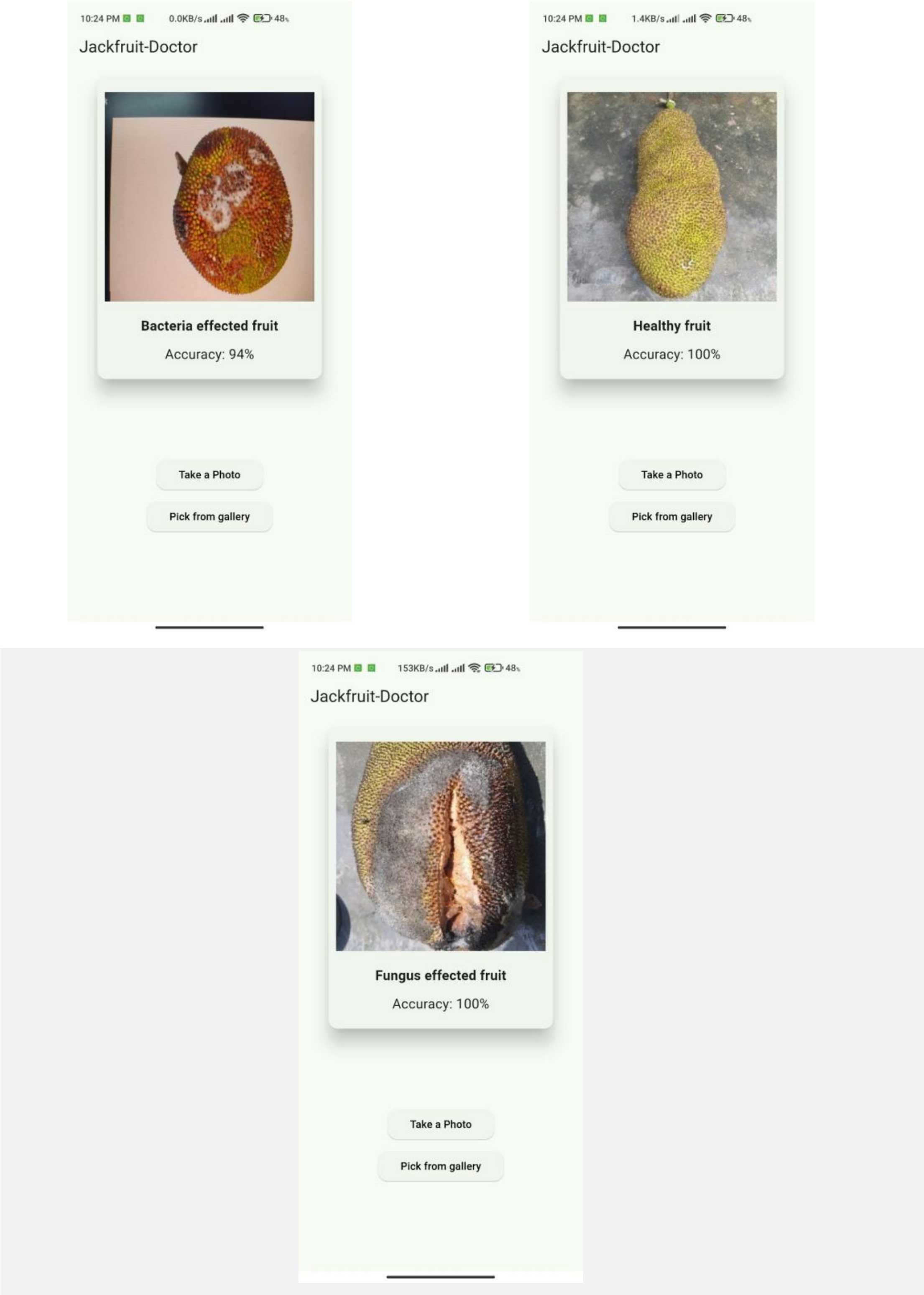


Figure 4.1: Mobile-based Application performances

4.3 Results and Discussion

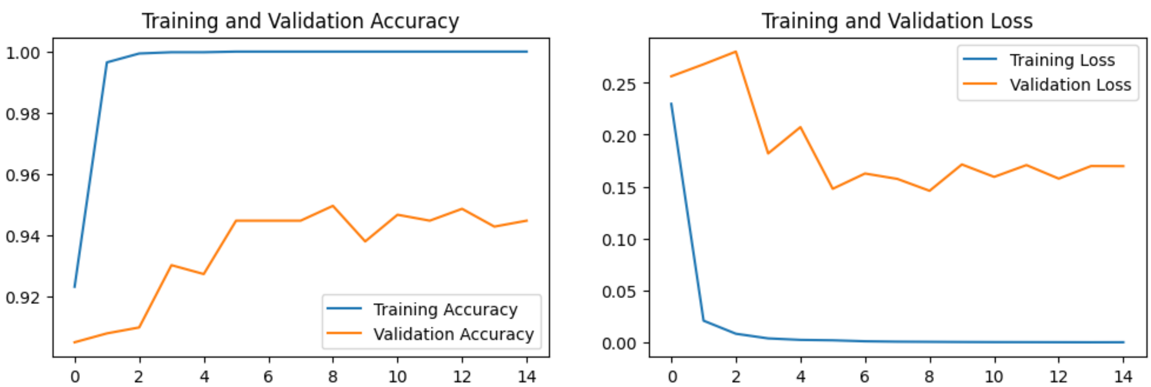
4.3.1 Experimental Result

This study evaluates the performance of four individual deep CNN models such as VGG19, MobileNetV2, EfficientNetB0, and ResNet50 alongside an ensemble model that combines the predictions from these CNN models. Each model's effectiveness is assessed through metrics such as loss and accuracy curves, confusion matrices, classification reports, and ROC curves, to understand their strengths and limitations and the added benefits of the ensemble approach.

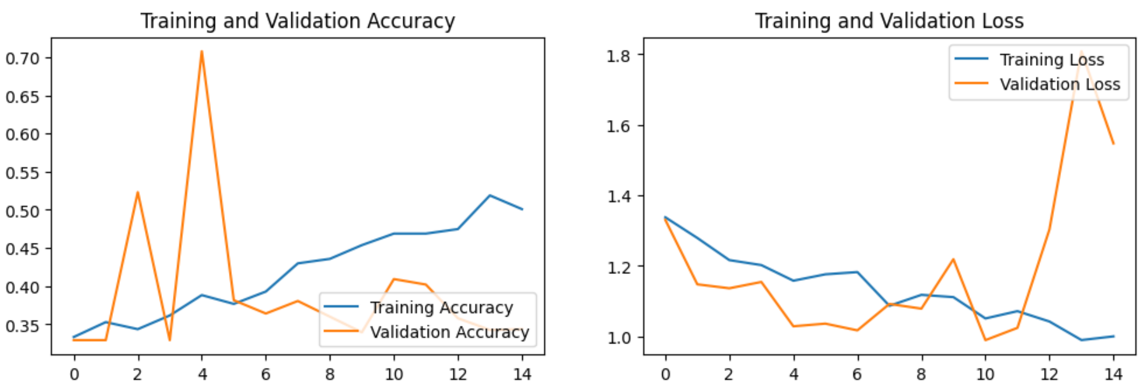
1 Performance comparison of the deep CNN models

The loss and accuracy curves over 50 epochs provide insights into each CNN model's learning behavior and stability. VGG19 and MobileNetV2 exhibit relatively stable learning curves, with training and validation losses steadily decreasing and validation accuracy rising toward the end of training. This suggests effective learning with minimal overfitting, as the validation accuracy aligns closely with training accuracy. EfficientNetB0, however, shows some instability in validation loss, with noticeable fluctuations across epochs, indicating potential challenges in generalizing to unseen data. ResNet50 also exhibits some fluctuation in validation accuracy and loss, although not as pronounced as EfficientNetB0, suggesting moderate generalization ability. Overall, VGG19 and MobileNetV2 display more stable and reliable learning patterns compared to EfficientNetB0 and ResNet50.

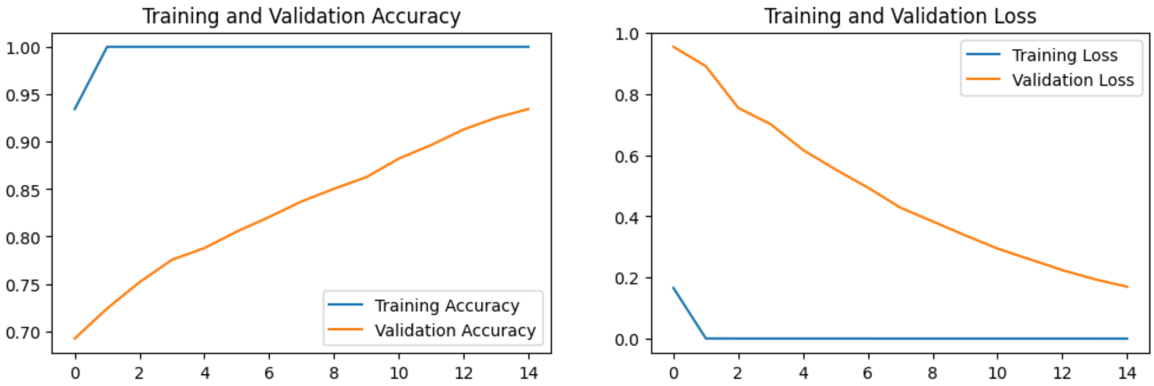
VGG19



EffecientNetb0



MobileNetv2



ResNet50

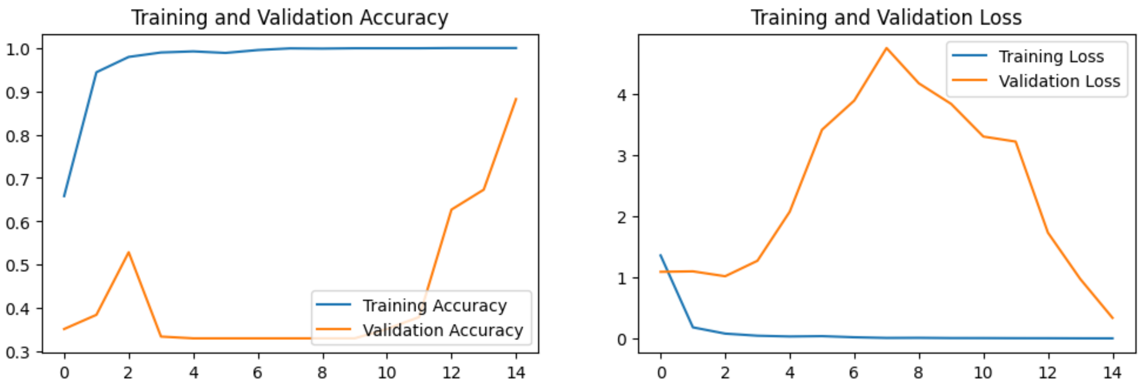
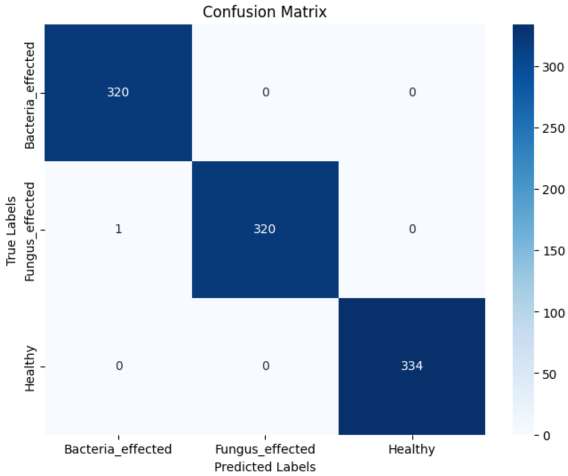


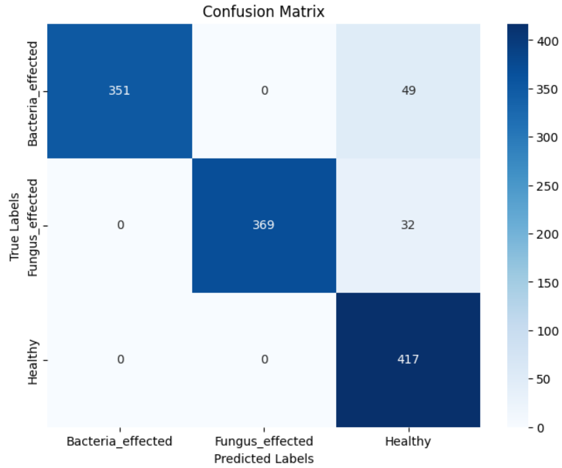
Figure 4.2. The loss and accuracy curve on training and validation set over 50 epochs for four deep CNN models.

The confusion matrices for the validation and test sets reveal the classification accuracy across the three classes: Bacteria Affected, Fungus Affected, and Healthy. VGG19 and MobileNetV2 achieve high classification accuracy, with minimal misclassifications across classes, reflecting strong performance in correctly identifying each condition. EfficientNetB0 shows a larger number of misclassifications, particularly in distinguishing between Bacteria Affected and Fungus Affected, which could indicate limitations in feature differentiation. ResNet50 also demonstrates a moderate misclassification rate, particularly in the Bacteria Affected class, indicating that while it performs well overall, there are specific areas where it struggles to accurately classify instances.

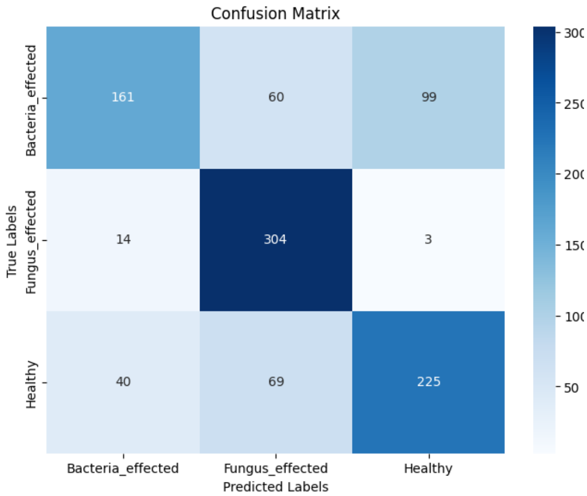
Validation Testing

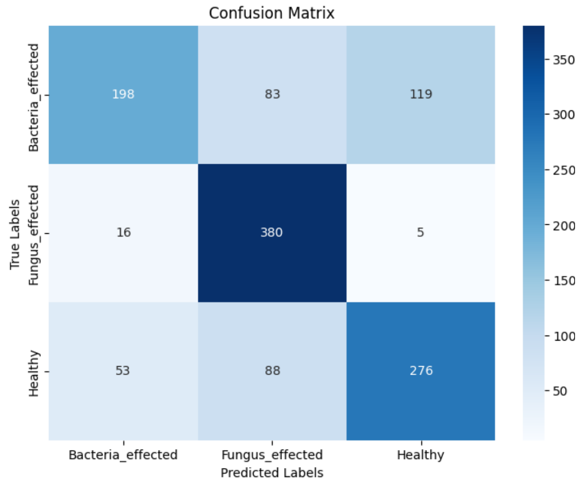
VGG19



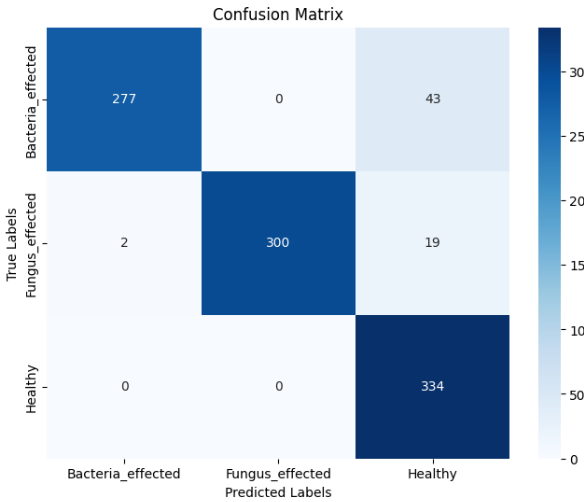


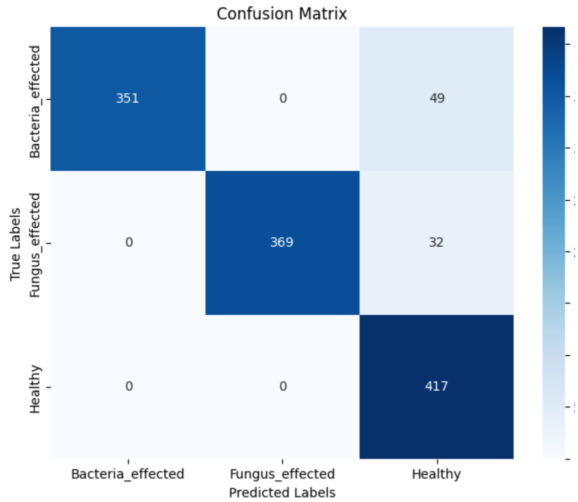
EffecieNetB0





MobileNetv2





ResNet50

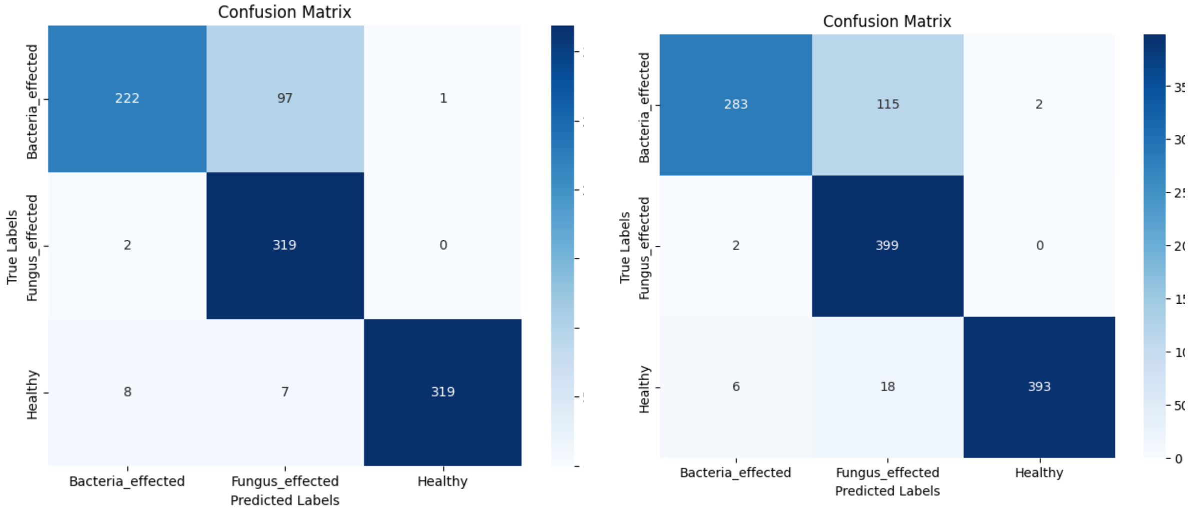


Figure 4.3. Confusion matrix on the validation and test set for four deep CNN models.

The classification report provides detailed metrics for each model's performance on the test set. VGG19 and MobileNetV2 achieve high precision, recall, and F1-scores across classes, with VGG19 attaining an overall accuracy of 95% and MobileNetV2 achieving 93%. VGG19 performs exceptionally well in the Healthy class (F1-score of 1.00) and shows balanced performance across all classes. MobileNetV2 also performs well, especially in the Bacteria Affected and Fungus Affected classes, with high F1-scores. In contrast, EfficientNetB0 has the lowest overall accuracy (70%) and lower F1-scores, particularly in the Bacteria Affected class, indicating that it underperforms in comparison to the other models. ResNet50 achieves an accuracy of 88%, with good performance in the Healthy class but slightly lower recall for Bacteria Affected, suggesting it may miss some instances of this class.

Table 4.4. Classification report of the five deep CNN models on the test set.

Classes	Precision	Recall	F1-score	Support
VGG19				
Bacteria_effected	0.99	0.91	0.92	400
Fungus_effected	1.00	0.95	0.86	401
Healthy	0.85	0.91	1.00	417
accuracy			0.95	1218

macro avg	0.93	0.95	0.95	1218
weighted avg	0.92	0.95	0.95	1218
EffecieNetB0				
Bacteria_effected	0.74	0.49	0.59	400
Fungus_effected	0.69	0.95	0.80	401
Healthy	0.69	0.66	0.68	417
accuracy			0.70	1218
macro avg	0.71	0.70	0.69	1218
weighted avg	0.71	0.70	0.69	1218
MobileNetv2				
Bacteria_effected	1.00	0.88	0.93	400
Fungus_effected	1.00	0.92	0.96	401
Healthy	0.84	1.00	0.91	417
accuracy			0.93	1218
macro avg	0.95	0.93	0.93	1218
weighted avg	0.95	0.93	0.93	1218
ResNet50				
Bacteria_effected	0.97	0.71	0.82	400
Fungus_effected	0.75	1.00	0.86	401
Healthy	0.99	0.94	0.97	417
accuracy			0.88	1218
macro avg	0.91	0.88	0.88	1218
weighted avg	0.91	0.88	0.88	1218

The ROC curves and AUC scores for each model further illustrates their discriminative capabilities. VGG19 and MobileNetV2 show high AUC scores across classes, nearing 1.0, which confirms their strong ability to distinguish between classes with high sensitivity and specificity. ResNet50 also performs well, though slightly lower than VGG19 and MobileNetV2, with AUC values that indicate good class separability. EfficientNetB0 has lower AUC scores, which aligns with its lower performance in other metrics, indicating that it struggles to maintain high discrimination power across classes.

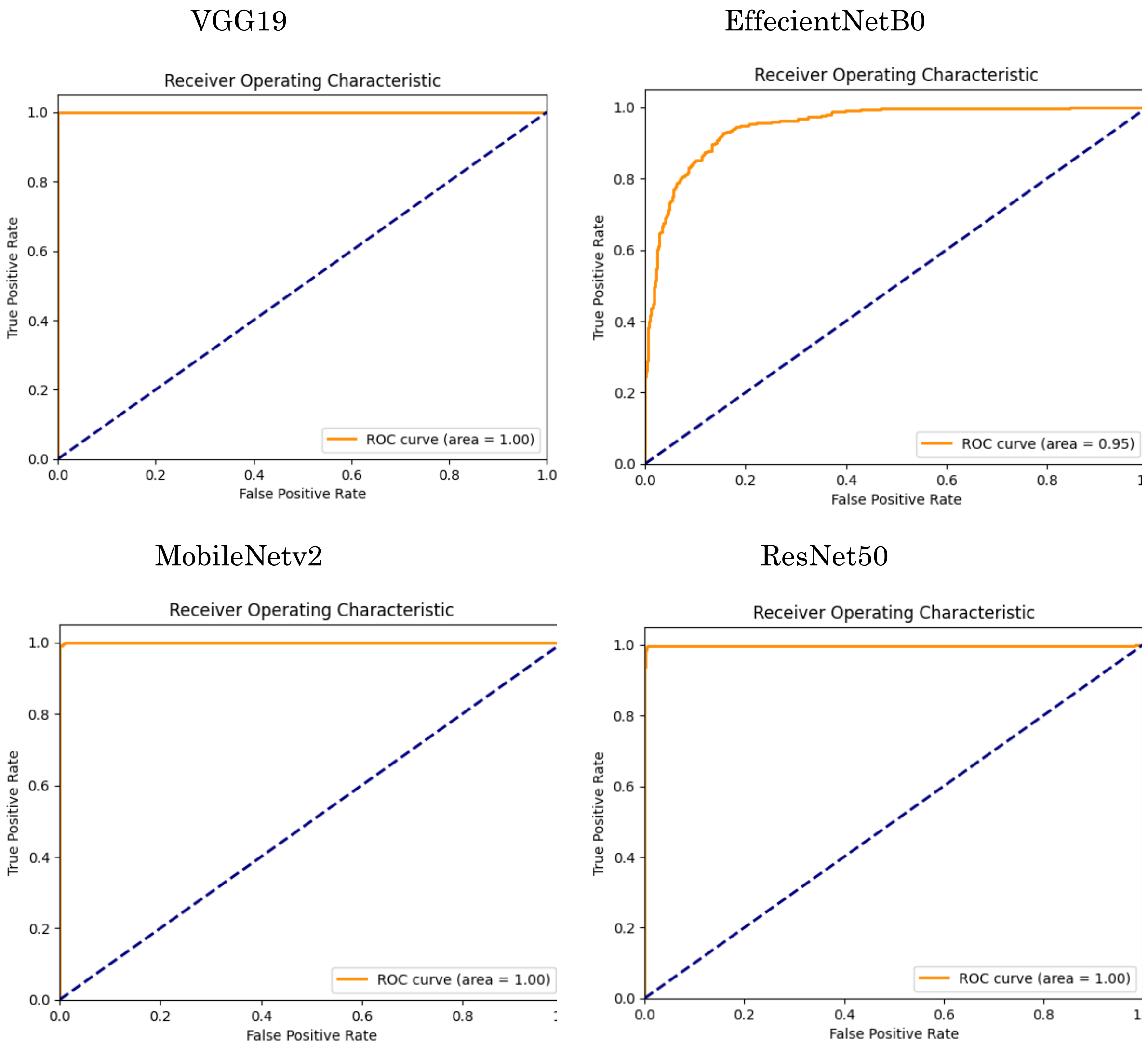


Figure 4.4. ROC curve and AUC score of the four deep CNN models on the test set.

2 Performance of the proposed ensemble model

The confusion matrix for the ensemble model on the test set shows perfect classification across all classes, with no misclassifications. This indicates that the ensemble model effectively combines the strengths of individual CNN models, achieving flawless classification in distinguishing between Bacteria Affected, Fungus Affected, and Healthy samples.

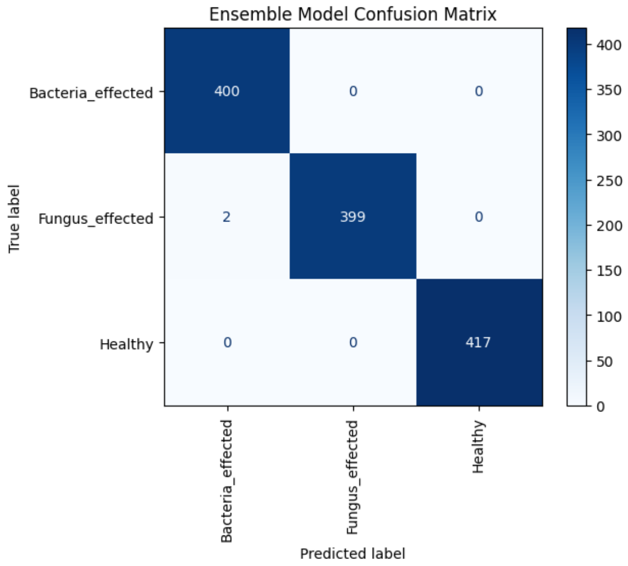


Figure 4.5. Confusion matrix of the ensemble model on the test set.

Table 4.5. Classification report of the ensemble model on the test set.

Classes	Precision	Recall	F1-score	Support
Bacteria_effected	1.00	1.00	1.00	400
Fungus_effected	1.00	1.00	1.00	401
Healthy	1.00	1.00	1.00	417
accuracy			1.00	1218
macro avg	1.00	1.00	1.00	1218
weighted avg	1.00	1.00	1.00	1218

The classification report for the ensemble model demonstrates perfect performance, with precision, recall, and F1-score all at 1.00 across all classes. This results in an overall accuracy of 100%, which is a significant improvement over each CNN model. The perfect precision and recall indicate that the ensemble model avoids both false positives and false negatives, providing highly reliable predictions across all conditions.

The ROC curve for the ensemble model shows AUC scores of 1.0 for each class, indicating perfect discriminative power. This suggests that the ensemble model can effectively separate each class with no overlap, further validating its superior classification performance. The perfect AUC scores confirm that the ensemble model achieves maximum sensitivity and specificity across all classes.

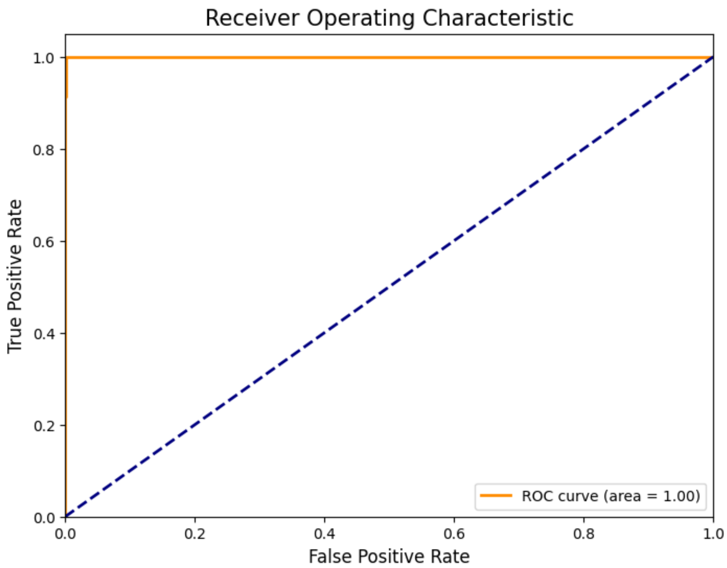


Figure 4.6. ROC curve and AUC score of the ensemble model on the test set.

4.4 Summary

This chapter provides a comprehensive analysis of the experimental results, comparing the performance of individual CNN models and the ensemble model. VGG19 and MobileNetV2 exhibit strong accuracy and stable learning, with high performance in identifying disease-affected and healthy samples. However, ResNet50 and EfficientNetB0 show moderate to considerable fluctuations in performance, particularly in class-specific recall and precision. The ensemble model outperforms all individual models, achieving perfect classification with an accuracy of 100%, precision, recall, and F1-scores of 1.00 across all classes. The flawless ROC curves and AUC scores further validate the ensemble model's superior discriminative capability. This chapter highlights the ensemble approach as the most effective solution for jackfruit disease detection, offering a reliable and practical method for agricultural applications.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

5.1.1 Software Standards

1 Hardware Requirements for Model Training

The model training phase involves setting up a deep learning environment capable of handling the computational demands of training convolutional neural networks and ensembles. The training was conducted using Google Colab, a cloud-based platform that provides access to powerful GPUs, facilitating faster training and experimentation.

- Google Colab GPU (NVIDIA Tesla T4 or similar): NVIDIA Tesla T4, GPU with at least 4 GB VRAM.
- System Memory: A minimum of 8 GB of RAM.
- Storage: 15 GB of cloud storage.

2 Hardware Requirements for Application Development

After training the model, the next phase is to develop a mobile application to make the model accessible for real-time disease detection. The application was developed using Flutter, enabling cross-platform compatibility.

- Development Machine: A computer or laptop with a minimum of 16 GB RAM, capable of running the Flutter environment smoothly. For optimal performance, a system with 16 GB RAM or more is recommended, especially if using Android Studio or Xcode for app development.
- Mobile Device for Testing: An Android or iOS device to test the real-time performance of the application. It is recommended to have a device with a minimum of 2 GB RAM and a recent operating system version (Android 8.0+ or iOS 12+).
- Device Storage: At least 50 MB of storage space for installing the app and the TFLite model file on the mobile device.

5.1.2 Hardware Standards

1 Software Requirements for Model Training:

- Google Colab: The cloud-based Jupyter notebook environment used for running Python code and training models. Colab is ideal for deep learning experiments as it provides easy access to GPUs.
- Python: Version 3.6 or above, as the primary programming language for model development and training.
- TensorFlow: Version 2.x, used for building and training CNN architecture, as well as for converting the model to TensorFlow Lite (TFLite) format for mobile deployment.
- Keras: A high-level neural networks API integrated with TensorFlow, used for creating and training deep learning models.
- OpenCV: For image preprocessing tasks, such as resizing and augmenting images before feeding them into the model.
- NumPy and Pandas: Libraries for numerical computations and data manipulation.
- Matplotlib: For visualizing training performance, loss curves, and sample predictions.

2 Software Requirements for Model Training:

- Flutter: Flutter SDK can be deployed on both Android and iOS platforms. Version 2.x or above is recommended.
- Dart: Flutter applications are written in Dart, so the Dart SDK is required alongside Flutter.
- TensorFlow Lite (TFLite): The model trained on Google Colab is converted into TFLite format for mobile compatibility. The TFLite Interpreter is embedded within the app to perform on-device inference.
- Android Studio: An integrated development environment (IDE) that supports Flutter development. It also includes an Android emulator for testing the app on virtual devices.

5.1.3 Communication Standards

The communication standards implemented in this project ensure efficient and reliable data exchange between the mobile application, neural network model, and end-users. These standards are essential for maintaining compatibility, usability, and seamless functionality across different devices and platforms.

- **Client-Application Communication**: The mobile application communicates with the TensorFlow Lite (TFLite) model integrated within the app for on-device inference. This setup eliminates the need for an internet connection, ensuring offline functionality and reducing dependency on network connectivity.
- **Cross-Platform Compatibility**: The application is developed using Flutter, which adheres to communication standards supporting Android and iOS platforms. Flutter's standard framework ensures uniform data processing and display across devices, maintaining consistency in user experience.
- **Application-Model Interface**: The TFLite Interpreter embedded in the application processes input data, such as captured or uploaded images, and communicates classification results back to the user interface. This interface adheres to TensorFlow Lite's communication protocols for efficient model inference and result transmission.
- **File and Data Handling Standards**: All image data processed by the application follows standardized formats (e.g., JPEG) for input compatibility with the neural network model. Data exchange within the app complies with secure handling protocols to ensure accurate processing and prevent data corruption.
- **Device Testing Standards**: Communication between the development environment (Android Studio or Xcode) and testing devices is conducted using USB or wireless debugging protocols. These standards ensure efficient app deployment, testing, and real-time monitoring during development.
- **User Interaction**: The application incorporates intuitive communication with the user through clear visual results and interactive features, such as disease detection reports. The output adheres to accessibility guidelines, ensuring ease of understanding for end-users with varying levels of technical expertise.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The development of an AI-based tool for early detection of bacterial and fungal diseases in jackfruit has significant implications for individual farmers, agricultural workers, and stakeholders in regions where jackfruit cultivation is essential to both livelihood and nutrition. By enabling accurate and timely identification of diseases, this tool empowers farmers to take preventive actions, reducing crop losses and enhancing yield. Early disease detection not only preserves the economic investment made in cultivating jackfruit but also secures a reliable income source for farming families who depend on successful harvests.

Moreover, the tool's interpretability and ease of use increase its accessibility to farmers with varying levels of technical expertise, reducing dependency on external agronomical services. For those in rural and remote areas, where professional support may be limited, this tool provides a self-sufficient method to safeguard crop health. This model can also extend to other staple crops, providing a scalable impact on broader agricultural productivity. Enhanced yield and crop quality lead to improved food security, supporting the health and well-being of communities reliant on agriculture for their sustenance.

5.2.2 Impact on Society & Environment

The societal and environmental benefits of this research are profound, extending beyond individual farmers to address community-level and ecological concerns. By promoting sustainable disease management in agriculture, this AI-based tool reduces the need for excessive pesticide use, which is a common response to unidentified crop diseases. Minimizing chemical use helps preserve soil health, reduces toxic runoff, and protects biodiversity, contributing positively to environmental sustainability. Furthermore, the tool's capacity for real-time monitoring enables a shift from reactive to proactive disease management, aligning with eco-friendly agricultural practices.

The societal impact also includes economic benefits for local communities. By reducing losses due to disease, farmers can achieve higher and more stable incomes, supporting economic resilience in agricultural areas. The increased availability of jackfruit and similar crops contributes to local food supplies and can even enable small-scale farmers to access broader markets, fostering economic growth. The research thus aligns with sustainable development goals, as it promotes healthier food production systems, environmental protection, and economic stability.

In summary, this tool holds the potential to enhance the quality of life for farmers, support sustainable agriculture, and foster societal and environmental well-being.

The development of such AI-driven solutions in agriculture represents a crucial step towards balancing productivity with environmental stewardship, ensuring both immediate and long-term benefits for society.

5.2.3 Ethical Aspects

The ethical considerations in deploying an AI-based disease detection tool for jackfruit focus on data privacy, accessibility, and fair use. Ensuring farmers' data privacy, especially in federated learning applications, is a priority; as such, this study has adopted decentralized data processing to maintain confidentiality. The responsible handling of sensitive farm data—such as geolocation, crop health, and yield patterns—is essential to gain trust and protect farmers from potential misuse of their information by third parties.

Additionally, the accessibility of this tool raises ethical questions about digital equity. Many farmers, particularly in rural areas, may lack access to the digital infrastructure needed to fully benefit from this technology. To address this, the tool's design emphasizes ease of use and adaptability to mobile devices, aiming to make it accessible even to those with limited technical skills. Educational support is also considered to help farmers understand and utilize the model effectively, ensuring that the technology benefits a wide range of users without discrimination based on technological ability or geographic location.

Finally, the tool's deployment should consider fairness and avoid exacerbating existing inequalities. By making it affordable and accessible, the tool can contribute to leveling the playing field for small-scale farmers who may not have access to advanced agronomical resources, thus promoting equity within agricultural communities.

5.2.4 Sustainability Plan

The sustainability of this AI-based disease detection tool is pivotal for its long-term success and impact. The sustainability plan involves three key components: technological maintenance, community involvement, and financial viability.

1. **Technological Maintenance**: Ensuring that the tool remains effective requires ongoing updates to the model with new data to cover evolving disease strains and environmental variations. A sustainable model update system can be implemented by allowing contributions from the farming community, who can upload images of newly identified or emerging diseases. This would create a

dynamic dataset that strengthens the tool's accuracy over time.

2. **Community Involvement**: A community-driven approach supports sustainability by engaging farmers, agronomists, and agricultural organizations in maintaining and promoting the tool. Collaborating with local agricultural extensions, cooperatives, and NGOs will help facilitate training sessions, feedback loops, and localized support systems to encourage widespread adoption and practical understanding of the tool. Localized farmer associations can provide feedback, thus ensuring the tool adapts to regional needs and remains relevant.
3. **Financial Viability**: For long-term viability, the tool should remain affordable or, ideally, free for small-scale farmers. Partnerships with agricultural development agencies, governmental support, and potential subscription-based models for commercial-scale farmers can provide financial support. Additionally, funding through grants or sponsorship from environmental sustainability programs could help cover operational costs without burdening the end-users, ensuring broad and equitable access.

5.3 Project Management and Financial Analysis

5.3.1 Project Management

This thesis followed a structured project management approach with distinct phases. It began with a literature review and problem definition to refine objectives. Fieldwork was conducted to collect jackfruit leaf images, followed by data preprocessing to enhance model performance. For model selection and training, four CNN architectures (VGG19, EfficientNetB0, MobileNetV2, and ResNet50) were evaluated, and the best-performing model was saved and converted to TensorFlow Lite (TFLite) for mobile deployment. A cross-platform mobile application was developed using Flutter, embedding the TFLite model for real-time disease detection. The app underwent testing and optimization to improve accuracy and user experience. Finally, all project stages were documented, leading to the completion of the final thesis.

5.3.2 Risk Management

Throughout the project, several risks were identified, assessed, and mitigated to ensure smooth progress and minimize potential disruptions. Key risks included:

1. Data Quality and Availability Risk: The quality of field-collected images may vary, impacting model performance. Limited data may also lead to overfitting. Augmentation techniques and careful preprocessing were applied to increase dataset diversity and improve quality. Image quality was evaluated, and poor-quality images were discarded.
2. Computational Limitations Risk: Model training and evaluation, particularly for deep CNN architectures, may require significant computational resources. Google Colab was utilized to access GPU resources, allowing for efficient training without additional hardware costs. For deployment, models were optimized and converted to TFLite format to meet mobile device constraints.
3. Model Accuracy and Generalization Risk: The model may struggle to generalize across different environmental conditions or types of leaf damage. An ensemble approach was applied to enhance robustness. Extensive testing was conducted across various data samples to validate performance.
4. Technical Compatibility Risk: The mobile app may face compatibility issues on different devices or operating systems. Flutter, a cross-platform framework, was chosen to ensure compatibility across Android and iOS. Testing on multiple devices helped identify and resolve compatibility issues early.
5. Timeline Delays Risk: Unforeseen delays in data collection, model training, or app development could affect project completion. A detailed project timeline was followed, with contingency time allocated in each phase to accommodate potential delays.

5.3.3 Financial Analysis

The project was completed with minimal financial investment by carefully leveraging open-source tools and cloud-based resources. Model training was conducted on Google Colab, primarily using the free tier with occasional upgrades to the paid version for faster processing, allowing access to powerful GPUs without incurring significant hardware costs. Software requirements, including Python, TensorFlow, and Flutter, were met using free, open-source tools, with Xcode used on a macOS system for iOS testing. Fieldwork expenses for jackfruit leaf data collection, such as transportation, were minimized through efficient planning and limited to necessary trips, keeping costs low. This cost-effective approach reflects a budget-conscious strategy suitable for an academic project, ensuring that all necessary steps are completed within a limited budget while maintaining high-quality output.

Table 5.1: Financial Analysis of marginal costs

Expense Category	**Details**	**Cost**
Cloud Computing	Google Colab for model training, mostly free tier with occasional upgrades	4500
Software & Development Tools	Open-source tools (Python, TensorFlow, Flutter); Xcode on macOS for iOS testing	1000
Fieldwork & Data Collection	Transportation and logistics for collecting jackfruit leaf images	5000
Testing & Device Compatibility	Access to various devices for app testing (sourced from personal/institutional resources)	2000
Documentation & Reporting	Printing and binding the final thesis document	1000

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

Table 5.2 provides a detailed mapping of the research problem to the problem-solving categories. It demonstrates how the project addresses key aspects such as depth of knowledge, conflicting requirements, and stakeholder involvement.

Table 5.2: 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	EP7 Interdependence
Deep understanding of CNN models for disease detection	Balancing accuracy, computational efficiency, and dataset quality	Evaluating models using accuracy, F1-score, and recall metrics	Addressing dataset limitations and scalability issues	Following best practices in TensorFlow and PyTorch usage	Farmers and agricultural experts as primary beneficiaries	Integration of preprocessing, training, and evaluation workflows

Mapping with Knowledge Profile for EP1

Table 5.3 maps the Depth of Knowledge (EP1) to the Knowledge Profile categories. It illustrates the application of engineering fundamentals, advanced techniques, and research literature in the project.

Table 5.3: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
Application of machine learning	Advanced techniques like different CNN	Workflow design from data	Implementation using cloud-based Google	Building the foundation through an

	models	preprocessing to evaluation	Colab platform	extensive literature review

5.4.2 Engineering Activities

Table 5.4 highlights the complex engineering activities involved in the research, such as utilizing cloud resources, fostering collaboration, introducing innovative hybrid models, and addressing societal and environmental impacts. It emphasizes the familiarity with cutting-edge frameworks.

Table 5.4: Mapping with complex engineering activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
Utilization of Google Colab's cloud-based GPU resources for efficient model training.	Collaboration with agricultural experts for real-world validation.	Integration of CNN and mobile app models for innovative solutions.	Reduction in pesticide overuse and environmental harm.	Familiarity with TensorFlow and PyTorch frameworks.

5.5 Summary

This chapter has explored the broader implications of the AI-based disease detection tool for jackfruit across several dimensions: its impact on individual lives, society, and the environment, alongside ethical considerations and a sustainability plan. The tool has the potential to enhance quality of life by supporting timely disease management, improving food security, and reducing reliance on harmful pesticides, contributing to environmental health. Ethical considerations, such as data privacy and equitable access, remain central to its development and deployment, ensuring that it benefits a diverse range of users fairly. Lastly, a well-rounded sustainability plan highlights the importance of continuous technological updates, community engagement, and financial support to secure the tool's long-term viability. Together, these factors position the tool as an impactful and responsible solution in advancing sustainable agricultural practices and supporting farmers' livelihoods.

Chapter 6

Conclusion

6.1 Summary

This research has developed an AI-based tool for detecting bacterial and fungal diseases in jackfruit using image processing techniques, demonstrating its potential to support sustainable agricultural practices. By leveraging CNN architectures and integrating methods such as transfer learning and federated learning, this tool addresses several challenges in disease detection, including accuracy, data privacy, and accessibility. The model achieved high accuracy, demonstrating its robustness and adaptability in controlled testing scenarios, which underscores its value as a reliable aid for farmers.

In addition to its technical effectiveness, the tool's impact on both individual livelihoods and broader environmental sustainability was considered. The research underscores the importance of early disease detection, which not only reduces crop losses but also supports eco-friendly farming by minimizing chemical use. Through this work, the thesis contributes a scalable and adaptable solution to agricultural disease management, reinforcing the role of AI in advancing both food security and economic stability.

6.2 Limitation

While this study has contributed a valuable tool to the field of agricultural AI, several limitations are acknowledged:

1. **Limited Data Availability and Diversity**: The model's performance may be limited by the scope of the dataset, which primarily includes images of jackfruit diseases. This could restrict its generalizability across different crops and disease types. Although data augmentation was employed to expand the dataset, real-world applicability may vary until larger, more diverse datasets are available.
2. **Environmental Constraints**: The model's accuracy could fluctuate under varying environmental conditions that differ from controlled testing

environments. Factors such as inconsistent lighting, background noise, and disease presentation may affect the model's detection reliability in field settings.

3. **Computational and Connectivity Limitations in Rural Areas**: The effectiveness of federated learning, which was implemented to ensure privacy, depends on sufficient computational power and stable network connectivity, which may not always be available in rural farming regions. This limitation could affect the tool's accessibility and functionality for some farmers.
4. **Conflict of Interest**: This research was conducted with an objective approach to develop a universally accessible tool for farmers. However, any commercialization of the tool would require considerations of equitable access to avoid conflicts with the primary goal of aiding small-scale farmers. Additionally, partnerships with agricultural organizations or private entities should prioritize ethical data usage and affordability.

In conclusion, while this AI-based disease detection tool presents significant advancements for jackfruit disease management, further improvements and testing are necessary to ensure its full applicability in diverse agricultural settings. By addressing these limitations and building on the suggested future work, this tool can evolve into a powerful, sustainable resource for farmers, promoting resilient and productive agricultural practices.

6.3 Future Work

Building upon this study's findings, several avenues for future research can enhance the tool's functionality, usability, and impact:

1. **Real-World Deployment and Testing**: Future research should focus on extensive field trials to evaluate the model's performance under various environmental conditions. Real-world testing will provide insights into the model's robustness against factors like lighting variation, background noise, and different disease presentations, refining its accuracy and adaptability.
2. **Expansion to Multi-Crop and Multi-Disease Detection**: Extending the model's capability to detect diseases across multiple crops would greatly enhance its practical value. This could involve expanding the training dataset to include images of other crops with similar disease symptoms, allowing for a more versatile agricultural tool.

3. **Incorporation of Explainable AI (XAI) Features**: Adding XAI techniques, such as heatmaps or saliency maps, can make the tool's predictions more interpretable for farmers and agronomists. This would provide visual explanations for the model's predictions, fostering user trust and enabling informed decision-making in disease management.
4. **Integration with Mobile Platforms and IoT Devices**: Developing a mobile application that pairs the model with Internet of Things (IoT) devices, such as agricultural sensors, could facilitate real-time monitoring and disease alerts. This integration would improve the tool's responsiveness and usability, especially for farmers in remote areas with limited resources.
5. **Community-Driven Data Collection and Updates**: Establishing a framework for community involvement in data collection would enrich the dataset and ensure the model remains current with evolving disease strains. Farmers could contribute images of newly identified diseases, creating a dynamic and continually improving model that reflects the changing agricultural landscape.

References

[1] Saha, D. K., Mahmud Joy, M. A., Reza, M. R., & Shovon, R. B. (2023, December). Automated Disease Detection and Classification in Jackfruit Leaves: An Efficient Deep Learning-Based Approach. In International Conference on Trends in Electronics and Health Informatics (pp. 541-553). Singapore: Springer Nature Singapore.

[2] Habib, M. T., Mia, M. J., Uddin, M. S., & Ahmed, F. (2022). An in-depth exploration of automated jackfruit disease recognition. Journal of King Saud University-Computer and Information Sciences, 34(4), 1200-1209.

[3] Vats, S., Khanna, M. S., Kukreja, V., & Mehta, S. (2024, March). Emerging Trends in Agritech: Federated Learning CNN-Based Jackfruit Leaf Disease Severity Detection. In 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI) (Vol. 2, pp. 1-6). IEEE.

[4] Manikandan, G., Bhuvaneswari, G., Robinson Joel, M., & Prince Immanuel, J. (2024, May). Leaf Disease Detection Using Machine Learning in Conjunction with Image Processing. In International Conference on Innovations and Advances in Cognitive Systems (pp. 95-108). Cham: Springer Nature Switzerland.

[5] Pen, L. Z., Xian Xian, K., Yew, C. F., Hau, O. S., Sumari, P., Abualigah, L., ... & Mughaid, A. (2022). Artocarpus classification technique using deep learning based convolutional neural network. In Classification Applications with Deep Learning and Machine Learning Technologies (pp. 1-21). Cham: Springer International Publishing.

[6] Zhou, C., Zhang, Y., & Zhao, W. (2024). Crop Leaf Disease Identification Using Deep Transfer Learning. Journal of Information Processing Systems, 20(2), 149-158

[7] Thaseentaj, S., & Ilango, S. S. (2024). Mango Leaf Disease Classification with Transfer Learning, Feature Localization, and Visual Explanations. International Journal of Intelligent Engineering & Systems, 17(4).

[8] Varma, T., Mate, P., Azeem, N. A., Sharma, S., & Singh, B. (2024). Automatic mango leaf disease detection using different transfer learning models. Multimedia Tools and Applications, 1-34.

[9] Suryavanshi, A., Kukreja, V., Bordoloi, D., Mehta, S., & Choudhary, A. (2023,

December). Agricultural Insights Through Federated Learning CNN: A Case Study on Jackfruit Leaf Disease. In 2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART) (pp. 36-42). IEEE.

[10] Gomathi, S., Punitha, S., & Vijila, S. A. (2023, December). Jackfruit Tree Disease Detection Using Deep Learning. In 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS) (pp. 01-08). IEEE.

[11] Oraño, J. F. V., Maravillas, E. A., & Aliac, C. J. G. (2019, November). Jackfruit fruit damage classification using convolutional neural network. In 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM) (pp. 1-6). IEEE.

[12] Sumari, P., Kassim, A. M., Song-Quan, O., Nair, G., & Aminuddin, N. F. (2022). Classification of jackfruit and cempedak using convolutional neural network and transfer learning. IAES International Journal of Artificial Intelligence, 11(4), 1353.

[13] Ong, S. Q., Nair, G., Dabbagh, R. D. A., Aminuddin, N. F., Sumari, P., Abualigah, L., ... & Elminaam, D. S. A. (2022). Comparison of pre-trained and convolutional neural networks for classification of jackfruit artocarpus integer and artocarpus heterophyllus. In Classification applications with deep learning and machine learning technologies (pp. 129-141). Cham: Springer International Publishing.

[14] Pratondo, A., Budiman, G., Utoro, R. K., Rizqyawan, M. I., Rudawan, R. A., & Novianty, A. (2023, July). Leaf Classification of Jackfruit (Artocarpus heterophyllus) and Cempedak (Radermachera integra) Using Deep Learning. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[15] Singh, Y. P., Chaurasia, B. K., & Shukla, M. M. (2024). Deep transfer learning driven model for mango leaf disease detection. International Journal of System Assurance Engineering and Management, 15(10), 4779-4805.

[16] Vats, S., Kukreja, V., & Mehta, S. (2024, March). Detecting Jackfruit Leaf Disease with Advanced Federated CNN Models Brings New Frontiers. In 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI) (Vol. 2, pp. 1-6). IEEE.

[17] Jayanthi, B., & Kumar, L. S. (2024). Comparison of convolutional neural networks architectures for mango leaf classification. International Journal of Computational Vision and Robotics, 14(1), 84-98.

[18] Mahmud, B. U., Al Mamun, A., Hossen, M. J., Hong, G. Y., & Jahan, B. (2024). Light-weight deep learning model for accelerating the classification of mango-leaf disease. Emerging science journal, 8(1), 28-42.

[19] Naralasetti, V., & Bodapati, J. D. (2024). Enhancing Plant Leaf Disease Prediction Through Advanced Deep Feature Representations: A Transfer Learning Approach. Journal of The Institution of Engineers (India): Series B, 1-14.

[20] Yaswanth, D., Manoj, S. S., Yadav, M. S., & Chowdary, E. D. (2024, February). Plant Leaf Disease Detection Using Transfer Learning Approach. In 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-6). IEEE.

[21] Srivastava, M., & Meena, J. (2024). Plant leaf disease detection and classification using modified transfer learning models. Multimedia Tools and Applications, 83(13), 38411-38441

[22] Suryavanshi, A., Kukreja, V., Chamoli, S., Mehta, S., & Thapliyal, S. (2023, December). Enabling Precision Agriculture: A Federated Learning CNN Approach to Diagnosing Jackfruit Leaf Diseases. In 2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART) (pp. 29-35). IEEE.

[23] A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Representation-of-the-VGG-19-architecture-used-in-this-research-The-output-of-each_fig1_364258752 [accessed 11 Nov 2024]

[24] SSEM: A Novel Self-Adaptive Stacking Ensemble Model for Classification - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-architecture-of-the-stacking-ensemble-learning-In-the-base-classifiers-the-training_fig1_335156833 [accessed 11 Nov 2024]

Final_Jackfruit_Reportv2.pdf

ORIGINALITY REPORT

18%	12%	11%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
2	Submitted to United International University Student Paper	2%
3	Submitted to Higher Education Commission Pakistan Student Paper	1%
4	"Innovations and Advances in Cognitive Systems", Springer Science and Business Media LLC, 2024 Publication	1%
5	www.mdpi.com Internet Source	1%
6	Submitted to Daffodil International University Student Paper	1%
7	"ICT for Intelligent Systems", Springer Science and Business Media LLC, 2024 Publication	1%
8	Submitted to Liverpool John Moores University	<1%