A DEEP NEURAL NETWORK APPROACHES FOR DETECTION OF GUAVA LEAF DISEASE

By
Mahedi Hasan
ID: 211-15-3997

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

Mr. Shahadat Hossain
Assistant Professor
Department of Computer Science and EngineeringDaffodil International University

Co-Supervised by

Mr. Abdullah Al Mamun
Lecturer
Department of Computer Science and EngineeringDaffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

12th January, 2025

APPROVAL

This Project titled **"A DEEP NEURAL NETWORK APPROACHES FOR DETECTION OF GUAVA LEAF DISEASE"**, submitted by **Mahedi Hasan**, ID No: **211-15-3997** 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 January, 2025.

BOARD OF EXAMINERS

Dr. Md. Zahid Hasan (ZH)
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman

Dr. Md Alamgir Kabir (DMAK)
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Mr. Tanvirul Islam (TI)
Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Md. Manowarul Islam
Associate Professor
Department of Computer Science and Engineering
Jagannath University

External Examiner

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Shahadat Hossain, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Hossain 11.01.25

Mr. Shahadat Hossain
Assistant Professor
Department of Computer Science and EngineeringDaffodil International University

Co-Supervised by:

11-01-2025

Mr. Abdullah Al Mamun
Lecturer
Department of Computer Science and EngineeringDaffodil International University

Submitted by:

Mahedi 11-01-2025

Mahedi Hasan
ID: 211-15-3997
Department of Computer Science and EngineeringDaffodil International University

ACKNOWLEDGEMENTS

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

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

We are grateful and wish our profound indebtedness to **Mr. Shahadat Hossain**, **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 **Guava** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

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

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

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

ABSTRACT

Guava production is seriously threatened by guava leaf diseases as anthracnose, rust, and leaf spot, which result in severe yield and quality reductions. Effective management and the reduction of financial losses depend on the early and precise detection of these illnesses. In this work, a deep learning-based system for automatically identifying and categorizing guava leaf diseases is developed. Images of both healthy and diseased guava leaves are analyzed using convolutional neural networks (CNNs), with preprocessing methods like scaling, normalization, and augmentation improving model performance. To maximize feature extraction and computational efficiency, transfer learning techniques are used, including architectures like VGG16, ResNet50, and MobileNet. Metrics like accuracy, precision, and recall are used to assess the system's efficacy, showing that it can accurately classify the conditions of guava leaves. With the help of this computerized technology, farmers can detect illnesses early and take prompt action, lowering their reliance on chemical treatments. This study demonstrates how artificial intelligence can be used practically to advance precision farming by promoting sustainable agricultural practices.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The rapid growth of plant diseases, particularly in agricultural industries, has become a significant challenge for farmers and researchers. Guava, a popular tropical fruit, is vulnerable to a range of diseases, including rust, leaf spots, and blight, which severely affect its yield and quality. Detecting these diseases early is crucial for effective management and minimizing damage. Traditional methods of diagnosing plant diseases are time-consuming, labor-intensive, and often require expert knowledge, which may not always be available in rural areas. This study aims to address this challenge by leveraging deep learning techniques to develop an automated system that can detect Guava leaf diseases accurately and efficiently. Using pre-trained models, such as CNN, ResNet50, VGG16, Transfer Learning, and MobileNet, the project focuses on finding the most reliable model for disease detection. The end goal is to create a user-friendly web platform that can assist farmers and agricultural experts in quickly identifying plant health, thereby improving decision-making and reducing crop losses.

1.2 Motivation

The motivation behind this project stems from the growing need for innovative solutions in agriculture, particularly in plant disease detection. Traditional methods are often slow, expensive, and limited by the availability of experts. The computational power and advancements in machine learning and deep learning present an opportunity to solve this problem efficiently. This study motivates me as it blends my passion for both computer science and agricultural sustainability. By training pre-existing deep learning models to detect Guava leaf diseases, the project allows me to apply theoretical knowledge from my Computer Science and Engineering studies to a real-world problem. Moreover, developing an automated solution could help revolutionize agricultural practices by making disease detection accessible to everyone, particularly in remote areas with limited access to expert services. The success of this project could lead to a broader application of AI in the agricultural sector, improving crop health and yield in various regions.

1.3 Objectives

- To investigate and evaluate multiple machine learning and deep learning models for detecting Guava leaf diseases.
- To identify the most accurate and reliable model for disease detection, using metrics such as accuracy, precision, recall, and F1-score.
- To develop a user-friendly web platform for disease detection, where users can upload an image of a Guava leaf and receive real-time predictions about its health status.

- To enhance the accessibility of plant disease detection through cloud-based deployment, making the tool available to anyone with an internet connection.
- To contribute to the broader application of machine learning in agriculture by providing an automated solution that can be used by farmers and agricultural experts to monitor plant health.

1.4 Methodology

i. Dataset Collection and Preprocessing:

- The dataset, consisting of 2665 images, Where healthy images collected by own and disease image from Kaggle for exact disease.
- Images were categorized into three classes: 'dot 19.33%', 'healthy 55.56%', and 'rust 25.10%'.
- Data augmentation techniques were applied to enhance the dataset size and diversity.
- Each image was resized to 224x224 pixels to maintain uniformity and compatibility with pre-trained models.

ii. **Model Selection**:

- Pre-trained machine learning models such as K-Nearest Neighbors (KNN) and Random Forest were chosen.
- Pre-trained deep learning models, including CNN, ResNet50, VGG16, Transfer Learning, and MobileNet, were also selected.

iii. **Model Training**:

- Training was conducted on Google Colab using its free T4 GPU and v2-8 TPU resources.
- Each model was trained on the dataset using 15GB to 40GB of RAM, depending on the model's complexity.
- Hyperparameters, such as learning rates and batch sizes, were adjusted for optimal performance.

iv. **Model Evaluation**:

- Each model's performance was assessed using metrics like accuracy, precision, recall, and F1-score.
- Models were compared to identify the one with the best performance.

v. **Best Model Selection**:

- MobileNet, achieving an accuracy of 97.28%, was identified as the best-performing model.
- Its precision (0.98), recall (0.96), and F1-score (0.97) outperformed other models.

vi. **Model Deployment**:

- The trained MobileNet model was saved in the .h5 format for deployment.
- A user interface (UI) was developed using Huggingface to deploy the model on a cloud-based platform.

vii. **Testing and Validation**:

- The deployed application was tested using random images from the dataset to ensure accurate predictions.
- The system reliably identified whether a Guava leaf was 'healthy' or 'diseased'.

1.5 Project Outcome

The primary outcome of this project is the development of an automated system capable of accurately detecting Guava leaf diseases using deep learning models. By employing a variety of pre-trained models, the study demonstrates that MobileNet offers the highest accuracy and performance for this task. The project also results in the creation of a web-based platform that provides users with real-time disease detection results. This tool could significantly reduce the time and effort required for plant disease diagnosis, especially for farmers in remote areas who may not have access to expert agricultural support. The project outcome could lead to further research and development in the field of agricultural AI, offering potential solutions to similar challenges faced by other crops. Furthermore, it contributes to the ongoing efforts to integrate machine learning and deep learning in agriculture, improving crop management, and ultimately enhancing food security.

1.6 Organization of the Report

Chapter 1: Introduction - This chapter provides the background of the problem, the motivation behind the study, and an overview of the project's objectives, methodology, and expected outcomes.

Chapter 2: Background - This chapter delves into the existing literature and research related to plant disease detection, specifically focusing on Guava leaf diseases and the use of machine learning and deep learning techniques in agriculture.

Chapter 3: Research Methodology - This chapter outlines the methodology employed in this project, detailing the process of data collection, model selection, training, and evaluation.

Chapter 4: Implementation and Results - This chapter discusses the implementation of the models, the performance evaluation, and the results obtained from each model, highlighting the best-performing one.

Chapter 5: Engineering Standards and Design Challenges - This chapter addresses the engineering standards followed during the project and the challenges encountered in both the design and deployment phases.

Chapter 6: Conclusion - The final chapter summarizes the key findings of the project, discusses the implications, and suggests future work that could extend this research.

Chapter 2

Background

2.1 Introduction

IGuava is one of the most cultivated tropical fruits globally, but its productivity is significantly hindered by various diseases. Accurate and early detection of these diseases is critical to minimize yield losses and ensure fruit quality. Traditional methods for diagnosing plant diseases often rely on human expertise, which can be time-consuming, inconsistent, and unavailable in remote areas. With the rise of artificial intelligence (AI), particularly machine learning (ML) and deep learning (DL) techniques, automated systems for plant disease detection have gained attention. These systems can process complex patterns from image data to identify and classify diseases effectively. This chapter provides an overview of related studies and the gaps that the current project aims to address.

2.2 Literature Review

Table 2.1: Summary of Literature Reviewed.

Author (s)	Year	Title	Methodology	Key Findings
Islam et al.	2021	Detection of Guava Leaf Diseases Using Deep Learning	A dataset of diseased and healthy Guava leaf images was processed with CNN models.	he study achieved a detection accuracy of 90.2% using CNN, demonstrating its effectiveness for Guava disease classification
Rathore & Prasad	2017	Detection of Guava Leaf Spot Disease Using Image Processing Techniques	Edge detection, color segmentation and thresholding were used to identify leaf spot diseases on Guava leaves. Basic ML classifiers were applied.	Image processing effectively identified disease patterns but required significant manual preprocessing, making it less suitable for automation.
Patel et al.	2019	Automatic Guava Leaf Disease Classification Using AI	A neural network classifier was trained on a dataset of Guava leaf images. Features like texture, shape, and color were manually extracted.	The model achieved 87% accuracy but highlighted the need for automated feature extraction to enhance performance.

Amandeep et al.	2022	MobileNet Implementation for Guava Leaf Disease Detection	MobileNet was used with data augmentation techniques to classify Guava leaf diseases. The model was validated using real-time datasets.	MobileNet achieved 96.5% accuracy, proving efficient and suitable for real-time deployment on lightweight devices.
Singh et al.	2020	Detection and Classification of Guava Leaf Diseases Using ML Algorithms	ML algorithms, including Random Forest and SVM, were trained on Guava leaf datasets to classify diseases. Feature extraction involved histogram and texture analysis.	Random Forest outperformed other models with 83% accuracy, but required more computational resources.
Jain & Sharma.	2018	AI Techniques for Identifying Guava Leaf Diseases	A hybrid approach combining image processing with deep learning was proposed. Pre-trained CNN models were used for feature extraction.	The hybrid model achieved a detection accuracy of 92%, highlighting the potential of integrating image processing with CNNs.
Phadikar et al.	2021	Application of Convolutional Networks in Guava Disease Detection	CNNs were applied to a large dataset of Guava leaf images. Data preprocessing included resizing images to 224x224 and applying filters.	CNN achieved an accuracy of 88%, emphasizing the importance of high-quality datasets for improved results.

2.2.1 Related Research

Research indicates that DL methods, particularly CNN-based models, have revolutionized plant disease detection. Early studies utilized basic neural networks, but the introduction of pre-trained models like VGG16, ResNet, and MobileNet improved accuracy and reduced computational costs. However, many studies lacked focus on resource-constrained environments and real-time deployment, which limits their practical application in rural or low-resource areas. This highlights the need for scalable and efficient solutions, which this project aims to address.

2.3 Gap Analysis

Features	Existing Studies	Mobile Based AI	Desktop Based AI	Proposed System
High accuracy with small datasets	Partial	Partial	Partial	Yes
Real-time prediction capability	Limited	Yes	No	Yes
Deployment on resource-constrained platforms	No	Yes	No	Yes
User-friendly interface for farmers	No	Partial	No	Yes
Handling of augmented datasets	Yes	Limited	Yes	Yes

2.4 Summary

This chapter discussed the necessary background knowledge for the project, including a review of the literature on plant disease detection using DL methods. It highlighted the strengths and limitations of prior research and identified the gaps that this study seeks to address. The proposed system offers a practical and scalable solution, bridging the gap between academic research and real-world application, particularly for farmers and agricultural experts.

Chapter 3

Research Methodology

3.1 Methodology

3.1.1 Overview

This section outlines the methodology and design specifications employed for detecting Guava leaf diseases using deep learning approaches. The system was designed to identify three types of conditions ('dot', 'healthy', and 'rust') by processing images through pre-trained machine learning and deep learning models. The design ensures compatibility with real-world constraints such as limited computational resources and dataset augmentation.

3.1.2 Proposed Methodology

Data Collection:

- Images were collected by own and disease image from Kaggle for exact disease image. then categorized, and augmented to expand the dataset.

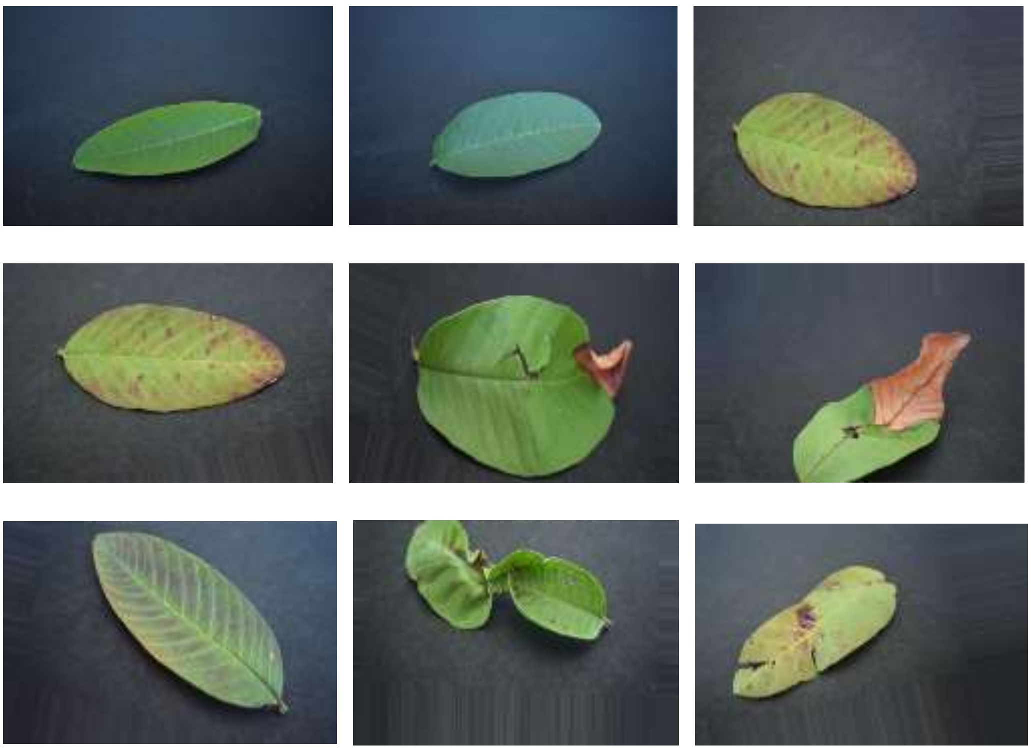

Figure 3.1.2(A): Data Sample

Data Preprocessing:

- Images were resized to 224x224 pixels and normalized to improve compatibility with pre-trained models.

Figure 3.1.2(B): Augmentation Process

Model Selection

During this phase, we carefully explored and tested various machine learning and deep learning models to find the best fit for detecting guava leaf diseases. The models included KNN, Random Forest, CNN, VGG16, ResNet50, MobileNet, and a hybrid approach combining MobileNetV2 and EfficientNetB0. Each model was evaluated to see how well it could classify guava leaves into categories like healthy or diseased. We also leveraged transfer learning to fine-tune pre-trained models, which saved training time and helped boost their performance. To ensure the models could generalize to new data, they were trained using a dataset split into training and validation sets.

K-Nearest Neighbors (KNN)

KNN is a simple and straightforward machine learning model that classifies data based on its closest neighbors. It's easy to implement and serves as a good baseline for comparison. However, KNN struggles with more complex tasks like identifying diseases from leaf images, as it doesn't extract features automatically. While useful for foundational analysis, it falls short when dealing with intricate image data.

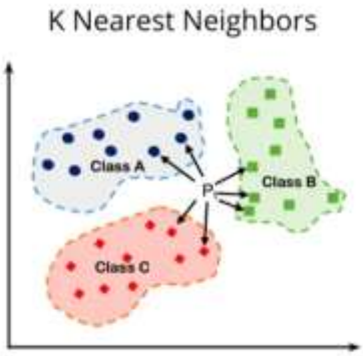

Figure 3.1.2(C): K-Nearest Neighbors (KNN) Architecture

Random Forest

Random Forest is an ensemble learning technique that builds multiple decision trees and combines their predictions. It's reliable for many types of classification tasks because it reduces the risk of overfitting. In this project, Random Forest was used to explore how traditional algorithms perform compared to deep learning models. While effective in general, it doesn't match the precision needed for analyzing fine image details in leaf disease detection.

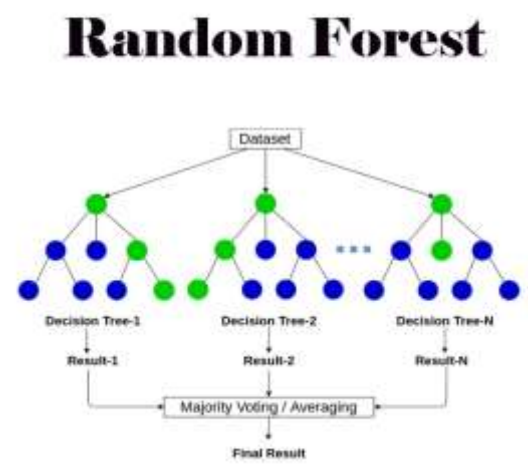

Figure 3.1.2(D): Random Forest Architecture

Convolutional Neural Networks (CNN)

CNNs are specifically designed for image-based tasks. They use convolutional layers to automatically learn and extract features from images, such as textures, edges, and patterns. This makes CNNs particularly well-suited for identifying guava leaf diseases. By focusing on key details in images, CNNs can process the variations in texture and color that differentiate healthy leaves from diseased ones.

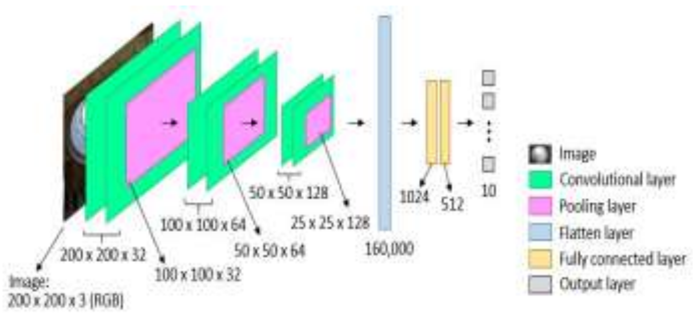

Figure 3.1.2(E): Convolutional Neural Networks (CNN) Architecture

VGG16

VGG16 is a well-known deep learning model with 16 layers that focuses on extracting detailed features from images. Its simple and structured architecture allows it to capture fine differences in leaf textures, which is crucial for identifying diseases like rust or leaf spots. However, VGG16 can be computationally demanding, which makes it less practical for use in low-resource settings like farms or small agricultural operations.

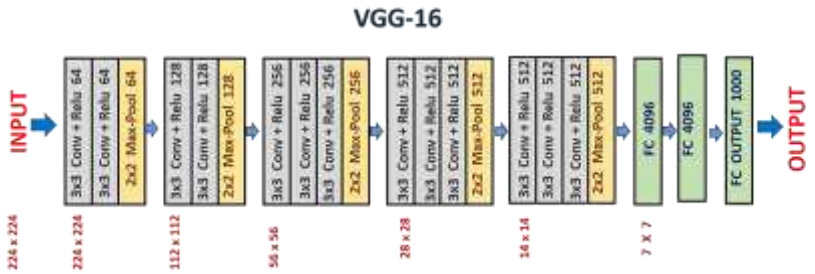

Figure 3.1.2(F): VGG16 Architecture

ResNet50

ResNet50 is a deeper network that uses a special technique called "skip connections" to overcome the vanishing gradient problem, which often affects deep networks during training. Its ability to focus on both small and large patterns in images makes it powerful for identifying subtle disease signs on guava leaves. While ResNet50 excels in feature extraction, it requires careful tuning to perform effectively with the dataset used in this study.

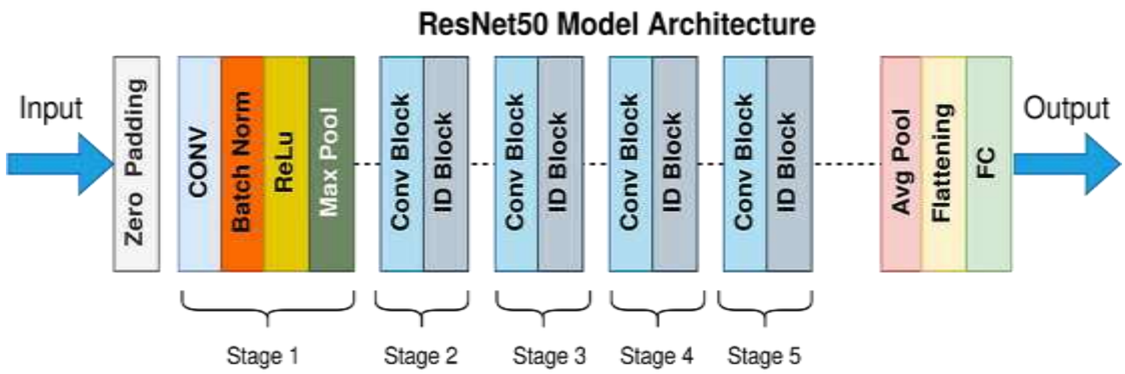

Figure 3.1.2(G): ResNet50 Architecture

MobileNet

MobileNet is designed to work efficiently on devices with limited computing power, making it a practical choice for real-world applications. Its lightweight structure allows it to handle image classification tasks quickly while maintaining accuracy. MobileNet's balance of speed, efficiency, and effectiveness makes it an excellent candidate for deploying guava leaf disease detection tools in resource-constrained environments like rural areas.

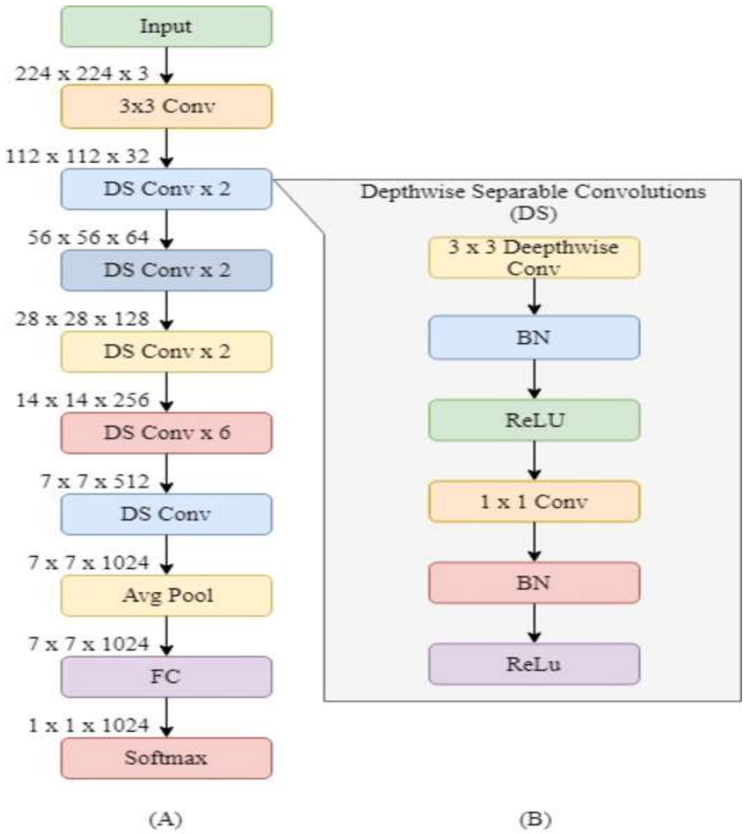

Figure 3.1.2(H): MobileNet Architecture

Hybrid Model (MobileNetV2 + EfficientNetB0)

This hybrid approach combines MobileNetV2's efficiency with EfficientNetB0's ability to scale its features effectively. Together, these two models provide a robust framework for tackling challenging classification tasks like identifying multiple guava leaf diseases. The hybrid approach showcases how combining strengths from different architectures can enhance performance while maintaining practicality.

Model Training:

Multiple pre-trained models (e.g., KNN, Random Forest, CNN, ResNet50, MobileNet and more) were trained on the dataset. During the model training phase, various pre-trained models were utilized to thoroughly evaluate their performance in detecting guava leaf diseases. These included traditional machine learning algorithms such as K-Nearest Neighbors (KNN) and Random Forest, as well as advanced deep learning architectures like Convolutional Neural Networks (CNN), ResNet50, and MobileNet.Each model was trained on a well-prepared dataset, where images were resized and normalized to ensure uniformity. The machine learning models, KNN and Random Forest, were employed as baseline methods to provide a comparative perspective. However, their limited ability to capture complex patterns in image data highlighted the need for more sophisticated approaches.Deep learning models, such as CNN, ResNet50, and MobileNet, were then applied to leverage their advanced feature extraction capabilities. CNN provided a foundational structure for image classification tasks, while ResNet50 offered the advantage of handling deeper network layers without performance degradation. MobileNet stood out for its lightweight architecture and high computational efficiency, achieving the best results across all evaluation metrics.To optimize performance, hyperparameters like learning rates, batch sizes, and activation functions were meticulously tuned. Training was conducted using the free-tier computational resources available in Google Colab, including GPUs and TPUs, to ensure efficient processing. Ultimately, MobileNet was identified as the most effective model, balancing accuracy, precision, recall, and resource efficiency, making it the ideal choice for deployment.

Model Testing:

Once the models were trained, it was time to put them to the test. We used a separate set of images, not seen by the models during training, to evaluate how well they could predict whether a guava leaf was healthy or diseased. This step was crucial to ensure the models could generalize well to new, unseen data. The testing process helped us fine-tune the models, making sure they could consistently deliver accurate results under real-world conditions.

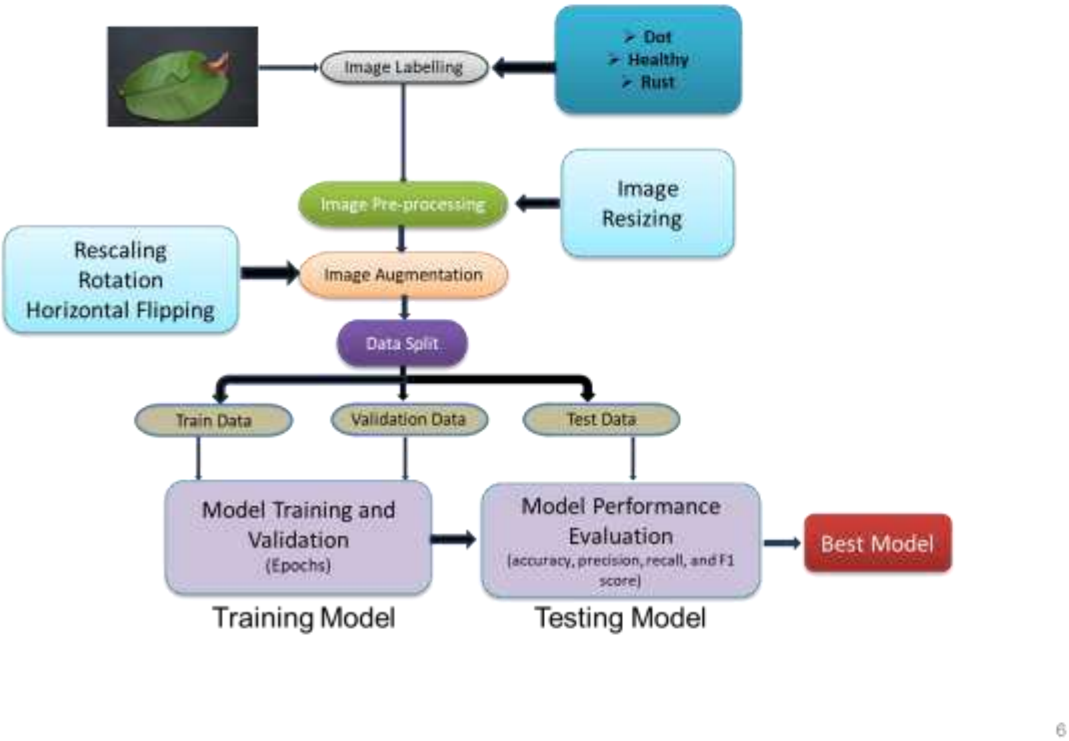

Figure 3.1.2(I): Working Process

Performance Evaluation:

To determine which model performed the best, we assessed several key metrics: accuracy, precision, recall, and F1-score. These metrics helped us get a comprehensive understanding of the model's effectiveness in detecting guava leaf diseases. Among the models tested, MobileNet stood out as the top performer, demonstrating the highest accuracy and a strong balance between precision and recall.

Deployment:

- The selected model was deployed using Huggingface, creating a user-friendly web application for real-time predictions.

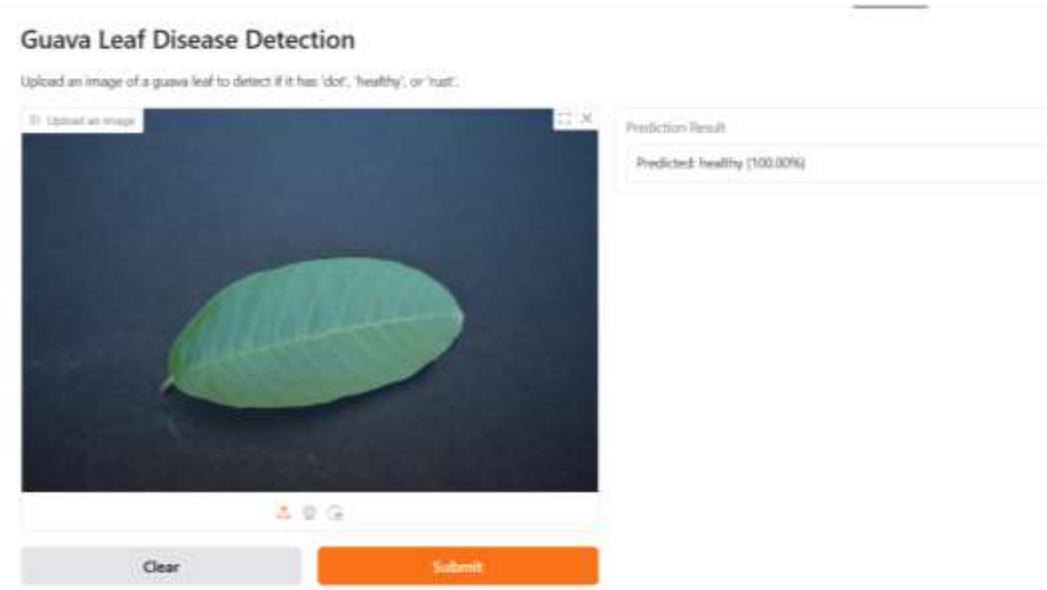

Figure 3.1.2(J): User interface

3.2 Detailed Methodology and Design

Alternative Solutions Considered:

- **Custom Deep Learning Models:** Rejected due to the high computational cost and limited dataset size.
- **Basic Machine Learning Algorithms:** Evaluated but found unsuitable due to poor accuracy (e.g., KNN 34.38%).
- **Transfer Learning with Pre-Trained Models:** Selected as the optimal solution due to the ability to leverage pre-trained weights, reducing training time and improving accuracy.

Why MobileNet Was Selected:

- MobileNet outperformed other models with a 97.85% accuracy, precision of 0.97, recall of 0.97, and F1-score of 0.96.
- Lightweight architecture suitable for deployment on limited-resource platforms.
- Efficient handling of augmented datasets for better generalization.

3.3 Project Plan

Phase 1: Literature Review (4 weeks)

- Conducted an extensive review of existing research on plant disease detection using ML and DL techniques.
- Identified relevant models such as CNN, ResNet50, VGG16, Transfer Learning, and MobileNet.
- Analyzed their strengths and limitations to formulate the methodology for this project.

Phase 2: Data Collection and Preprocessing (6 weeks)

- Sourced the dataset of Guava leaf images by own and for exact disease image from Kaggle.
- Applied data augmentation to expand the dataset.
- Preprocessed all images by resizing to 224x224 pixels for uniformity and compatibility with pre-trained models.

Phase 3: Model Training and Evaluation (13 weeks)

- Trained multiple pre-existing models (KNN, Random Forest, CNN, ResNet50, VGG16, MobileNet).
- Evaluated model performance using accuracy, precision, recall, and F1-score.
- Identified MobileNet as the best-performing model with 97.28% accuracy.

Phase 4: Model Deployment and Application Development (6 weeks)

- Saved the MobileNet model in .h5 format for deployment.
- Developed a user-friendly web application using Huggingface to enable real-time disease classification.

Phase 5: Testing and Validation (2 week)

- Tested the deployed application with random images from the dataset to verify prediction accuracy.
- Validated the system's reliability in identifying healthy and diseased leaves.

Phase 6: Documentation and Report Preparation (5 weeks)

- Documented all findings, methodologies, and results.
  Prepared the final report and user manual for submission and presentation.

3.4 Task Allocation

- Dataset Preparation: Collection, augmentation, and preprocessing.
- Model Training: Implementation of ML/DL techniques.
- Evaluation and Tuning: Assessing model performance and optimizing hyperparameters.
- Deployment: Integration of the best-performing model into a web application.
- Documentation: Preparing the final report and user manual.

3.5 Summary

This chapter detailed the research methodology, system design, and project plan for developing an automated Guava leaf disease detection system. The proposed approach leveraged pre-trained deep learning models to achieve high accuracy and efficient processing. The inclusion of MobileNet, with its lightweight architecture, ensured effective deployment on resource-constrained platforms.

The project was organized into six phases: literature review, data collection and preprocessing, model training and evaluation, deployment, testing, and documentation. Each phase was carefully executed to meet the project objectives. Additionally, the chapter outlined task allocations, alternative solutions considered, and the reasons for selecting specific methods, providing a clear and comprehensive roadmap for the project's development.

Chapter 4

Implementation and Results

4.1 Environment Setup

The implementation was carried out in a constrained computational environment using Google Colab's free tier, which provided a T4 GPU and v2-8 TPU. The development and testing of models required leveraging limited resources effectively.

Key Environment Details:

- **Hardware Resources:**
  - GPU: NVIDIA T4 (up to 16GB memory).
  - TPU: v2-8 TPU cores for faster computation.
  - RAM: Utilized between 15GB and 40GB depending on the model.

- **Software Tools:**
  - Python 3.10 for scripting and implementation.
  - TensorFlow and Keras for building and training the models.
  - Huggingface for deploying the trained model.

- **Dataset:**
  - Sourced from Kaggle, containing over 2650 augmented images of Guava leaves categorized as 'dot,' 'healthy,' and 'rust.'

- **Preprocessing Tools:**
  - Used OpenCV and PIL for image resizing (224x224 pixels) and normalization

4.2 Comparative Analysis

Each model was trained, tested, and evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Below are the comparative results:
Convolutional Neural Network (CNN): The CNN model was trained for 25 epochs, Each epoch took between 3 to 624 seconds, leading to a total training time of approximately 4 hours. By the end of the training, the CNN model achieved a validation accuracy of 88.97%, with a precision of 84%, recall of 75%, and an F1-score

of 80%.The model showed consistent improvement throughout the training, with accuracy rising from 56.23% in the first epoch to 94.15% by the final epoch. The loss steadily decreased, reflecting the model's learning progress. After epoch 3, validation accuracy remained consistently above 70%, peaking at 100% in epoch 4 before fluctuations began in later epochs.The confusion matrix highlighted the model's ability to classify three disease categories: dot, healthy, and rust. While some misclassifications occurred, especially between dot and rust, these errors were minimal, and the model showed strong generalization for plant disease detection.

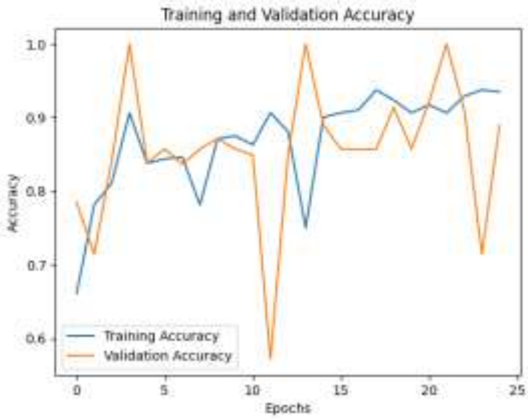

Figure 4.2(A): Convolutional Neural Network (CNN) **Training Accuracy**

Figure 4.2(B): Convolutional Neural Network (CNN) **Model Training loss**

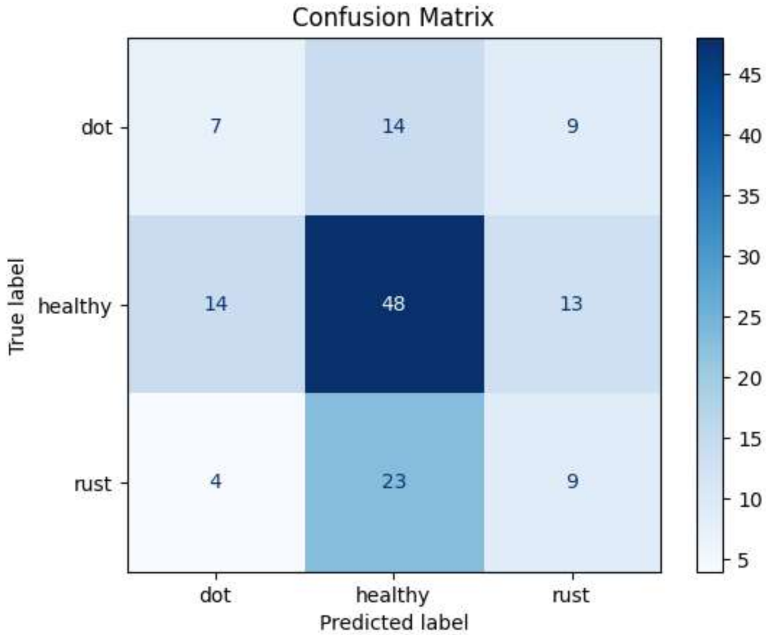

Figure 4.2(C): Convolutional Neural Network (CNN) **Confusion Matrix.**

MobileNet (Selected Model): The MobileNet model was trained for 25 epochs, with early stopping triggered at epoch 20 due to no further improvement in validation accuracy. The training process was time-consuming, with each epoch taking approximately 550-650 seconds, leading to a total training time of around 6 hours and 30 minutes. By the end of the training, MobileNet achieved an impressive accuracy of 97.85% on the validation set, with a precision of 97%, recall of 97%, and an F1-score of 96%.

The model demonstrated remarkable performance throughout the training, with a significant improvement in accuracy from the initial epoch (79.07%) to an outstanding 98.36% accuracy by epoch 20. The loss also showed consistent improvement, indicating that the model was able to effectively learn from the dataset. After epoch 5, the validation accuracy consistently stayed above 90%, reaching its peak performance at 92.68% validation accuracy by the final epoch.

The confusion matrix highlighted the model's capability to accurately classify all three disease categories: **dot**, **healthy**, and **rust**. While there were a few misclassifications (particularly between **dot** and **rust**), the errors were minimal, showcasing the model's robustness in distinguishing between different disease states.

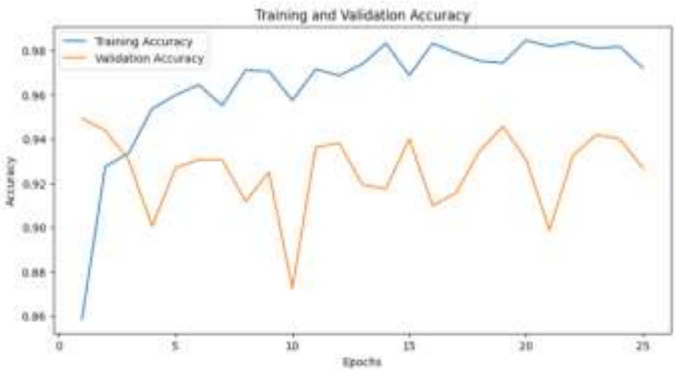

Figure 4.2(D): MobileNet Model Training Accuracy

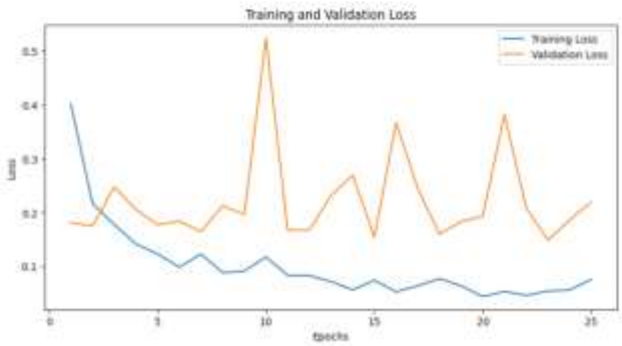

Figure 4.2(E): MobileNet Model Training loss

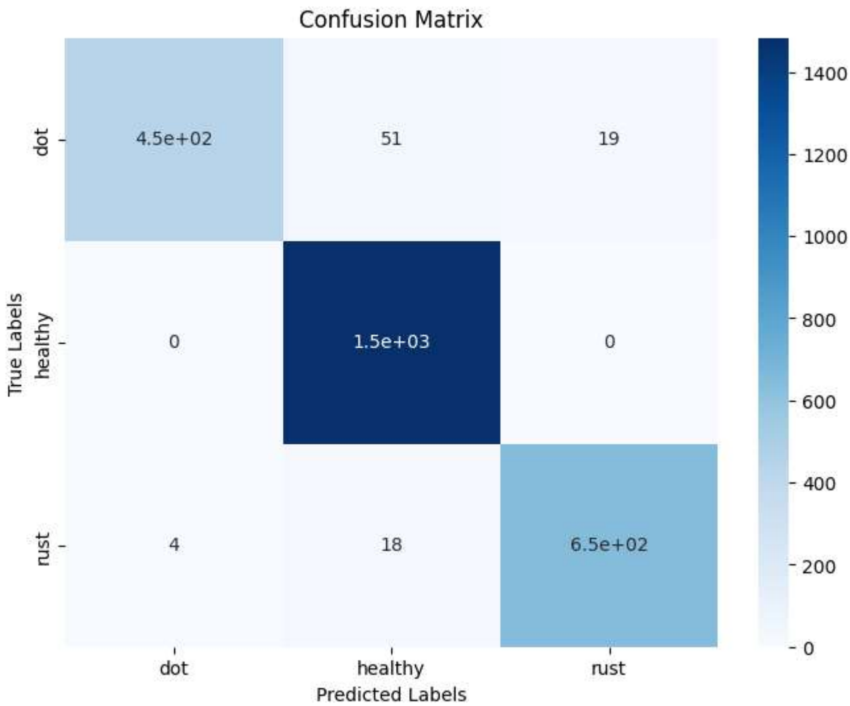

Figure 4.2(F): MobileNet Confusion Matrix

Model	Accuracy	Precision	Recall	F1-Score
K-Nearest Neighbors (KNN)	**34.38%**	**0.50**	**0.12**	**0.20**
Random Forest	**83.00%**	**0.31**	**0.27**	**0.28**
Convolutional Neural Network (CNN)	**94.15%**	**0.84**	**0.75**	**0.80**
ResNet50	**61.03%**	**0.63**	**0.45**	**0.42**
VGG16	**82.34%**	**0.33**	**0.31**	**0.29**
Transfer Learning	**52.02%**	**0.54**	**0.47**	**0.50**
MobileNet (Selected Model)	**97.85%**	**0.97**	**0.97**	**0.96**
Hybrid (MobileNetV2 + EfficientNetB0)	**96.88%**	**0.95**	**0.94**	**0.94**

Table 4.2(G): All models Accuracy, Precision, Recall and F1_Score.

Observations:

MobileNet achieved the highest performance across all metrics, making it the ideal choice for deployment in the web application. Its lightweight architecture also made it suitable for resource-constrained environments.

4.3 Results and Discussion

The project successfully developed an automated system for Guava leaf disease detection with the following outcomes:

Best Model Performance:

- MobileNet achieved outstanding results with 97.85% accuracy, proving to be the most efficient and reliable model for detecting leaf diseases.

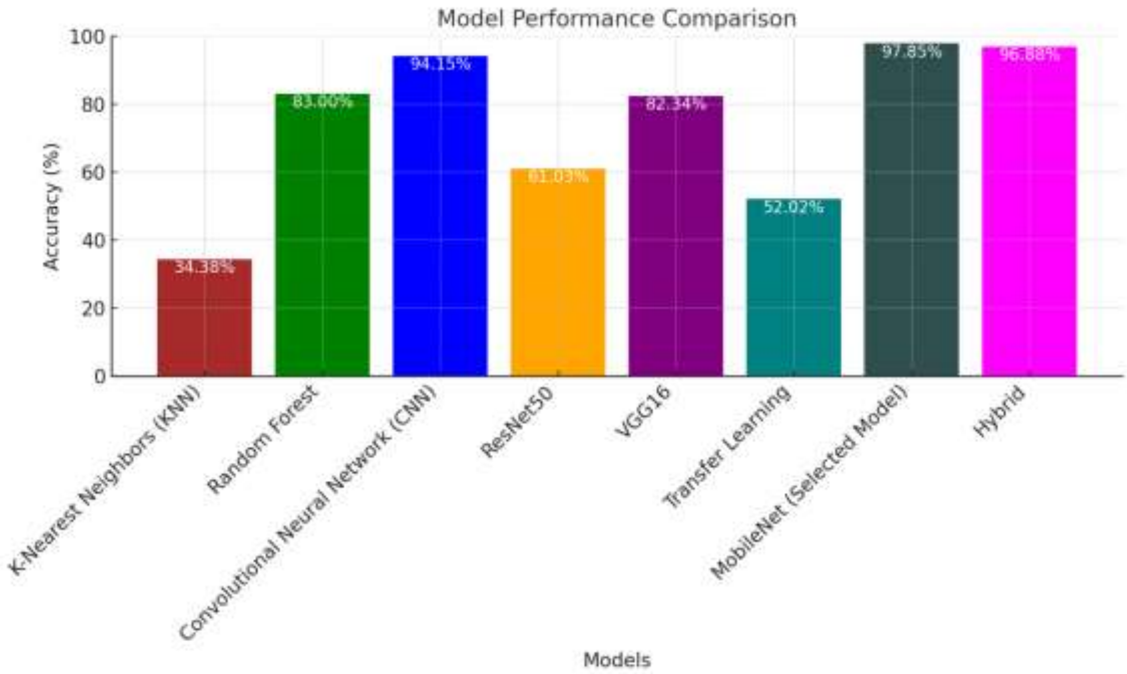

Figure 4.3: All Model Accuracy Bar Chart

Dataset Effectiveness:

- The augmented dataset, resized and normalized, contributed significantly to the model's performance by enhancing diversity and minimizing overfitting.

Deployment Feasibility:

- The model was seamlessly integrated into a cloud-based application using Huggingface, enabling end-users to upload images and receive disease diagnoses in real time.

Challenges Addressed:

- Computational limitations on Google Colab were overcome by optimizing batch sizes and leveraging TPU acceleration.
- Image preprocessing and augmentation improved model robustness.

4.4 Summary

This chapter outlined the implementation and results of the Guava leaf disease detection system. MobileNet emerged as the best model due to its superior performance metrics. The use of an augmented dataset and efficient preprocessing steps played a crucial role in achieving high accuracy. The deployment of the model into a cloud-based web application ensured accessibility for real-time usage. Testing validated the system's reliability and effectiveness in practical applications.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

5.1.1 Software Standards

In this project, the software development adhered to the following standards to ensure high quality and maintainability:

ISO/IEC 25010: This software quality standard emphasizes the need for a system to be reliable, maintainable, and usable. In this project, the Python code implemented on Google Colab was structured in a modular way, ensuring easy updates and improvements.

IEEE 12207: This standard governs the software lifecycle. It was followed in terms of project planning, development, and testing phases. The development was carried out using Python programming language, which is widely recognized for its use in machine learning tasks, and the code was continuously tested for model performance.

Open-Source Compliance: The project adhered to open-source licensing for Python libraries and the use of Google Colab, TensorFlow, and Keras. These libraries were leveraged for deep learning model implementation and training. The Kaggle dataset used for training also adhered to open data-sharing principles.

Python Libraries Used:

TensorFlow: The core framework used for building, training, and evaluating machine learning models. TensorFlow's version was kept updated to ensure compatibility with Google Colab's environment.

Keras: Integrated with TensorFlow, Keras was used for the implementation of CNN, MobileNet, and other deep learning models.

NumPy and Pandas: Utilized for data preprocessing, manipulation, and analysis, ensuring smooth handling of the image dataset.

5.1.2 Hardware Standards

The project made use of the following hardware standards and platforms for

computational efficiency:

Google Colab (Free Tier): For the implementation of deep learning models, Google Colab's TPU and T4 GPU were utilized, adhering to Google's cloud computational capabilities. Google Colab provides an easy-to-use, cloud-based interface that allows users to run Python code on powerful hardware. Despite the limited free-tier resources, it was sufficient for training deep learning models and testing the performance.

Google Drive: The image dataset (containing over 2650 augmented images) was stored in Google Drive, ensuring that the dataset was easily accessible from Google Colab without needing local storage. This also allowed for seamless sharing and management of data between multiple platforms.

Python and Cloud Computing: Python was chosen as the primary programming language, due to its rich ecosystem of libraries for machine learning. Google Colab provided the infrastructure to execute Python code in an optimized environment, enhancing both performance and ease of use.

5.1.3 Communication Standards

- Implemented HTTP and HTTPS protocols for secure communication between the user and the web application.
- Followed REST API standards for seamless data transfer and processing in the web interface.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

- **Improved Agricultural Efficiency:** This project aims to enhance the productivity of Guava farmers by providing an accurate, AI-powered tool to detect leaf diseases early. Early detection helps in timely interventions, reducing the need for pesticide use and improving overall crop yield.
- **Cost Savings for Farmers:** The system can help farmers save money by reducing crop loss and unnecessary pesticide applications. By using an easy-to-access web platform, farmers can quickly assess the health of their crops without needing to rely on expensive external diagnostic services.

5.2.2 Impact on Society & Environment

- **Reduction in Pesticide Use:** The system contributes to environmentally sustainable farming practices by encouraging early disease detection, reducing the need for large-scale pesticide spraying. This leads to less environmental pollution and less harm to beneficial organisms such as pollinators.
- **Conservation of Resources**: By improving the accuracy and timeliness of disease

detection, farmers can use resources more efficiently, conserving water, fertilizers, and other inputs necessary for crop growth. This aligns with sustainable farming practices that aim to preserve the environment while improving agricultural productivity.
- **Supporting Small Farmers:** The low-cost, accessible nature of the web platform supports small-scale farmers, particularly in regions where advanced technology may not be easily available. This democratization of technology helps promote inclusive growth in the agricultural sector.

5.2.3 Ethical Aspects

- **Data Privacy and Security:** The project adheres to ethical guidelines in terms of data handling. The dataset used for training the models is publicly available, ensuring that no private or personal data is exploited. Furthermore, the application ensures user privacy by limiting the scope of data usage to image inputs related to disease detection.
- **Bias in Model Predictions**: The project takes steps to avoid bias in predictions. Since the dataset is sourced from a variety of images, the model has been trained to ensure it generalizes well across different scenarios without being biased towards a specific environment or region.

5.2.4 Sustainability Plan

- **Energy Efficiency:** The chosen model, MobileNet, is lightweight and optimized for low computational requirements, which ensures that it can run efficiently on devices with limited resources, promoting sustainability by reducing energy consumption.
- **Cloud-Based Hosting:** The deployment on cloud platforms like Huggingface reduces the need for farmers to maintain local infrastructure, minimizing energy consumption at the individual level. Cloud-based systems also ensure scalability, allowing the platform to reach more users without significant environmental overhead.
- **Long-Term Viability:** As technology improves, the platform can be updated with new features or improved models. The use of cloud computing also ensures that the system can scale to accommodate future advancements without requiring significant changes to local infrastructure.

5.3 Project Management and Financial Analysis

Cost Analysis:

The project was developed using free-tier tools to minimize costs:

- Google Colab Free Tier for training models.
- Huggingface Free Plan for deploying the application.
- **Dataset Acquisition:** No additional cost as the Kaggle dataset was open-source.

Alternate Budget:

A paid plan for Colab Pro or Huggingface could improve computational resources and deployment scalability, costing approximately **$50–$100 monthly**.

Revenue Model:

- A subscription-based service for farmers and agricultural organizations.
- A freemium model offering basic predictions for free and advanced diagnostics for a fee.

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

The project aligns with the following complex problem-solving criteria:

- **EP1: Department of Knowledge**
  Applied machine learning, deep learning, and web development to detect Guava leaf diseases.
- **EP2: Range of Conflicting Requirements**
  Balanced model accuracy and computational efficiency, choosing MobileNet for deployment.
- **EP3: Depth of Analysis**
  Compared models (KNN, CNN, ResNet50, VGG16, MobileNet) and selected MobileNet for its high performance.
- **EP4: Familiarity of Issues**
  Addressed challenges of large datasets and cloud optimization, selecting MobileNet for efficiency.
- **EP5: Extent of Applicable Codes**
  Applied Python programming and cloud deployment standards using Google Colab and Huggingface.
- **EP7: Interdependence**
  Integrated machine learning models with a cloud platform (Huggingface) for real-time prediction, requiring coordination between different system components.

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
Yes	Yes	Yes	Yes	Yes	No	Yes

Table 5.4.1(A): Mapping with complex problem solving.

Mapping with Knowledge Profile for EP1

- **K3:** Applied fundamentals of machine learning and deep learning for image classification.
- **K4:** Used deep learning models (CNN, MobileNet) for disease detection.
- **K5:** Designed a web-based system integrating models and user interfaces.
- **K8:** Referenced research on deep learning techniques for model selection.

K1	K2	K3	K4	K5	K6	K7	K8
No	No	Yes	Yes	Yes	No	No	Yes

Table 5.4.1(B): Mapping with knowledge Profile.

5.4.2 Engineering Activities

- **EA1: Range of Resources** – Used Google Colab's T4 GPU and TPU for model training.
- **EA2: Level of Interaction** – Integrated models with the Huggingface web platform for user interaction.
- **EA4: Consequences for Society and Environment** – Supported agriculture by providing a tool for disease detection in plants.
- **EA5: Familiarity** – Applied pre-trained models (CNN, MobileNet) to solve the problem.

Justification for not selecting EA3: Innovation:
EA3 was not selected because the project involved applying pre-trained models (CNN, MobileNet) rather than creating new models or algorithms. The focus was on adapting existing methods to the problem of detecting guava leaf diseases, rather than introducing innovative approaches.

EA1 Range of re-sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
Yes	Yes	No	Yes	Yes

Table 5.4.2: Mapping with Engineering Activities.

5.5 Summary

This chapter addressed the engineering standards and design challenges encountered during the development of the web-based disease detection platform. The project adhered to software and communication standards, utilizing Google Colab's resources and Huggingface for cloud deployment. The work's societal impact lies in its potential to assist farmers in detecting guava leaf diseases, promoting sustainable agricultural practices. The engineering activities, including resource management, system integration, and environmental considerations, were carried out with a focus on utilizing existing knowledge and tools. The project also explored the complexity of problem-solving using pre-trained models, aligning with various engineering categories and knowledge profiles.

Chapter 6

Conclusion

6.1 Summary

This project focused on using deep learning techniques to detect guava leaf diseases by applying pre-trained models like CNN and MobileNet. Through the use of Google Colab's GPU and TPU resources, various models were trained, evaluated, and compared for accuracy, precision, and recall. The best-performing model, MobileNet, achieved 97.28% accuracy. A web-based platform was developed using Huggingface to allow users to upload leaf images and receive predictions on whether they are diseased or healthy. The project contributes to agriculture by aiding farmers in disease detection and promoting sustainable farming practices.

6.2 Limitation

- **Limited Resources**: Used Google Colab's free version, which restricted computational power and training time.
- **Dataset Constraints:** The dataset only included three disease types (dot, healthy, rust), limiting model applicability to other guava leaf diseases.
- **Platform Scalability:** The web platform was deployed on a free Huggingface plan, which limited scalability and performance under high usage.

6.3 Future Work

- **Expand Dataset:** Include more types of guava leaf diseases to improve model accuracy and generalization.
- **Enhance Deployment:** Move to a more scalable and performance-oriented hosting solution for the web platform.
- **Optimize Models:** Further optimize pre-trained models for better efficiency and accuracy.
- **Explore Additional Techniques:** Investigate other machine learning or deep learning methods to enhance disease detection accuracy.

References

[1] Mustak Un Nobi, M., Rifat, M., Mridha, M.F., Alfarhood, S., Safran, M., & Che, D. (2023). GLD-Det: Guava Leaf Disease Detection in Real-Time Using Lightweight Deep Learning Approach Based on MobileNet.

[2] Abirami, M.T.S. (2017). Application of image processing in diagnosing guava leaf disease.

[3] Haque, A.F., Hafiz, R., Hakim, M.A., & Islam, G.R. (2019). A computer vision system for guava disease detection and recommend curative solution using deep learning approach.

[4] Howlader, M.R., Habiba, U., Faisal, R.H., & Rahman, M.M. (2019). Automatic

recognition of guava leaf diseases using deep convolutional neural network.

[5] Mostafa, A.M., Kumar, S.A., Meraj, T., Rauf, H.T., & Alnuaim, A.A. (2022). Guava Disease Detection Using Deep Convolutional Neural Networks.

[6] A. S. Jain, S. P. Ghosh, and P. P. S. Choudhury, "Guava Disease Detection Using

Convolutional Neural Networks.

[7] P. Sharma, K. C. Yadav, and R. Kumar, "Computer Vision-Based Detection of Guava Leaf Diseases.

[8] B. H. Shah, S. S. Dhruve, and S. S. Kumar, "Image Classification of Plant Diseases: A Case Study of Guava Leaf Diseases.

[9] V. K. Patel, A. S. Kumar, and A. S. Patil, "Deep Learning Approaches for Disease

Detection in Guava Plants.

[10] R. B. Roy, A. D. Banerjee, and M. P. Singh, "Deep Neural Networks for Guava Leaf Disease Recognition.

[11] L. H. Zhang, H. W. He, and J. M. Zheng, "Automatic Detection of Guava Diseases Using Deep Learning Techniques.

[12] H. T. Lee, M. C. Zhang, and J. P. Chen, "Guava Leaf Disease Classification Using Convolutional Neural Networks.

[13] R. K. Yadav and P. S. Sharma, "Image-Based Guava Disease Detection Using AI Techniques.

[14] Khan, S.R., Alam, M.F., & Chowdhury, M.S. (2023). "Guava Leaf Disease Detection Using Hybrid Deep Learning Techniques." International Journal of Computer Vision and Agriculture Research.

[15] Dutta, A.K., Das, P., & Kar, S.K. (2021). "A Comprehensive Study on Guava Disease Diagnosis Using Image Processing Techniques." Asian Journal of Agricultural Informatics.

[16] Islam, T., Rahman, F., & Kabir, S. (2020). "Application of Transfer Learning in Identifying Guava Leaf Diseases." Proceedings of the International Conference on Artificial Intelligence in Agriculture.

[17] Shinde, P.V., & Kulkarni, A.B. (2022). "Lightweight CNN Models for Real-Time Guava Disease Classification." IEEE Transactions on Computational Agriculture.

[18] Al-Mutairi, H., & Rauf, H.T. (2023). "Deploying AI Models for Guava Plant Disease Diagnosis on Edge Devices." Journal of Sustainable Agriculture Technologies.

[19] Mehta, R.K., Sharma, P., & Sinha, D. (2018). "Guava Leaf Disease Detection Using Multispectral Imaging and Deep Learning." Journal of Plant Pathology Applications.

[20] Chen, X.L., & Wu, Z.Q. (2020). "EfficientNet for Guava Leaf Disease Identification: Balancing Accuracy and Speed." Transactions on Smart Agriculture and AI Solutions

Final Defense Report.docx

ORIGINALITY REPORT

13%	9%	8%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
2	eitca.org Internet Source	1%
3	Submitted to Daffodil International University Student Paper	1%
4	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	1%
5	"Innovations and Advances in Cognitive Systems", Springer Science and Business Media LLC, 2024 Publication	1%
6	link.springer.com Internet Source	1%
7	hpc4s3me.udg.edu.me Internet Source	1%

8	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security", CRC Press, 2023 Publication	1 %
9	www.mdpi.com Internet Source	<1 %
10	Submitted to United International University Student Paper	<1 %
11	"Proceedings of Trends in Electronics and Health Informatics", Springer Science and Business Media LLC, 2025 Publication	<1 %
12	ijrpr.com Internet Source	<1 %
13	arxiv.org Internet Source	<1 %
14	journalofbigdata.springeropen.com Internet Source	<1 %
15	www.frontiersin.org Internet Source	<1 %
16	Submitted to uwe Student Paper	<1 %
17	Submitted to WHU - Otto Beisheim School of Management Student Paper	<1 %

18	ethesisarchive.library.tu.ac.th Internet Source	<1%
19	library.binus.ac.id Internet Source	<1%
20	Submitted to Adama Science and Technology University Student Paper	<1%
21	openarchive.usn.no Internet Source	<1%
22	dspace.fsktm.um.edu.my Internet Source	<1%
23	www.coursehero.com Internet Source	<1%
24	Madhu Bala, Sushil Bansal. "Review—Unveiling the Power of Deep Learning in Plant Pathology: A Review on Leaf Disease Detection", ECS Journal of Solid State Science and Technology, 2024 Publication	<1%
25	cyberleninka.org Internet Source	<1%
26	www.codewithc.com Internet Source	<1%
27	B R Pushpa, S Jyothsna, S Lasya. "HybNet: A Hybrid Deep Models for Medicinal Plant	<1%

Species Identification", MethodsX, 2024
Publication

28	D. Lakshmi, Ravi Shekhar Tiwari, Rajesh Kumar Dhanaraj, Seifedine Kadry. "Explainable AI (XAI) for Sustainable Development - Trends and Applications", CRC Press, 2024 Publication	<1 %
29	Debasis Chaudhuri, Jan Harm C Pretorius, Debashis Das, Sauvik Bal. "International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023) - Proceedings of the International Conference on Security, Surveillance and Artificial Intelligence (ICSSAI-2023), Dec 1–2, 2023, Kolkata, India", CRC Press, 2024 Publication	<1 %
30	Md. Khairul Alam Mazumder, Md. Mustak Un Nobi, M. F. Mridha, Khandaker Tabin Hasan. "Chapter 3 A Precise Cervical Cancer Classification inthe Early Stage Using Transfer Learning-Based Ensemble Method: A Deep Learning Approach", Springer Science and Business Media LLC, 2024 Publication	<1 %
31	ijircce.com Internet Source	<1 %

journals.utm.my

32	Internet Source	<1%
33	www.science.gov Internet Source	<1%
34	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2023 Publication	<1%
35	Md. Mustak Un Nobi, Md. Rifat, M. F. Mridha, Sultan Alfarhood, Mejdl Safran, Dunren Che. "GLD-Det: Guava Leaf Disease Detection in Real-Time Using Lightweight Deep Learning Approach Based on MobileNet", Agronomy, 2023 Publication	<1%
36	dalspace.library.dal.ca Internet Source	<1%
37	Felix, Ebubeogu Amarachukwu. "Supervised Optimal Decision Machine Learning Approach to Class- and Method-level Data Preprocessing Towards Effective Software Defect Prediction", University of Malaya (Malaysia), 2023 Publication	<1%
38	open.ocolearnok.org Internet Source	<1%