# SWEET PUMPKIN LEAF DISEASE DETECTION APPROACH WITH DEEP LEARNING ALGORITHM

**BY**

**Name: Md. Amir Khasru**
**ID: 241-25-048**

**Final Year Research Based Project Report**

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

**Supervised By**

**Mr. Md. Sadekur Rahman**
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

**DAFFODIL INTERNATIONAL UNIVERSITY**

**DHAKA, BANGLADESH**

**24 May, 2025**

# APPROVAL

This Project titled **"SWEET PUMPKIN LEAF DISEASE DETECTION APPROACH WITH DEEP LEARNING ALGORITHM"**, submitted by **Md. Amir Khasru** and 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 MSc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **24-05-2025**.

## BOARD OF EXAMINERS

**Dr. Arif Mahmud** — **Chairman**
**Associate Professor and Associate Head**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

**Dr. Fizar Ahmed** — **Internal Examiner**
**Associate Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

**Dr. Md Alamgir Kabir** — **Internal Examiner**
**Assistant Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

**Nazibur Rahman** — **External Examiner**
**Technical Lead, Database Administrator**
Wipro, Telenor - Grameen Phone Account
Dhaka, Bangladesh

# DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Md. Sadekur Rahman, 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:**

20.5.25

**Mr. Md. Sadekur Rahman**
Assistant Professor
Department of CSE
Daffodil International University

**Co-Supervised by:**

**Dr. Arif Mahmud**
Associate Professor and Associate Head
Department of CSE
Daffodil International University

**Submitted by:**

20.05.25

**Md. Amir Khasru**
ID: 241-25-048
Department of CSE
Daffodil International University

# ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty for His divine blessing making it possible for us to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Assistant Professor Mr. Md. Sadekur Rahman** Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Deep Learning*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to the **Head of the Department of CSE Dr. Sheak Rashed Haider Noori**, **Associate Professor and Associate Head Dr. Arif Mahmud**, for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

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

# ABSTRACT

Agriculture stands as a cornerstone of global food security and economic progress, particularly vital where large populations depend on farming for sustenance. Among the widely cultivated crops, sweet pumpkin holds prominence due to its nutritional value and economic significance. However, its susceptibility to various leaf diseases poses significant threats, impacting both yield and quality. Current disease detection methods rely on visual inspection by agricultural experts, prone to human error and time-intensive, limiting their scalability for extensive farming operations.

This research addresses these challenges through the development of an automated deep learning system tailored for sweet pumpkin leaf disease detection. Leveraging convolutional neural networks (CNNs), known for their robust image classification capabilities, the study employed pre-trained models such as DenseNet121, VGG16, VGG19, MobileNet, and InceptionV3. Among these, MobileNet demonstrated the highest accuracy at 91%, followed closely by DenseNet121 at 90%, VGG16 at 88%, InceptionV3 at 86%, and VGG19 at 85%.

The system includes a user-friendly interface built on Streamlit, enabling agricultural professionals and farmers to upload leaf images for instant analysis, facilitating early disease diagnosis and informed decision-making. This application of artificial intelligence not only enhances agricultural productivity but also promotes sustainability by enabling proactive disease management. Moving forward, the developed solution shows potential for adaptation to other crops and integration into smart agricultural technologies, promising further advancements in agricultural practices and food security worldwide.

# TABLE OF CONTENTS


| Contents | Page No |
|---|---|
| Approval | ii |
| Declaration | iii |
| Acknowledgment | iv |
| Abstract | v |
| **CHAPTER 1: INTRODUCTION** | **1-5** |
| 1.1 Overview | 1 |
| 1.2 Background and Present State | 1-2 |
| 1.3 Problem Statement | 2 |
| 1.4 Objectives | 3 |
| 1.5 Scope and Limitations | 3-4 |
| 1.6 Report Organization | 4-5 |
| 1.7 Summary | 5 |
| **CHAPTER 2: LITERATURE REVIEW** | **6-10** |
| 2.1 Overview | 6 |
| 2.2 Related Works | 6-7 |
| 2.3 Comparison between existing works | 8 |
| 2.4 Open Issues | 9 |
| 2.5 Summary | 10 |
| **CHAPTER 3: METHODOLOGY/ REQUIREMENT ANALYSIS & DESIGN SPECIFICATION** | **11-20** |
| 3.1 Overview | 11 |
| 3.2 Proposed Methodology/ System Design | 12-17 |
| 3.3 Hardware/ Software Requirement | 18 |
| 3.4 Project Management and Financial Analysis | 18-19 |
| 3.5 Summary | 20 |

**CHAPTER 4: IMPLEMENTATION** **21-25**
4.1 Overview 21
4.2 Train Model/ Prototype Design 21-23
4.3 System Testing/ Model Evaluation 24-25
4.4 Summary 25
**CHAPTER 5: RESULT AND ANALYSIS** **26-40**
5.1 Overview 26
5.2 Experimental/ Simulation Result 26-31
5.3 Performance/ Comparative Analysis 31-40
5.4 Summary 40
**CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY** **41-44**
6.1 Impact on Life 41
6.2 Impact on Society & Environment 41-42
6.3 Ethical Aspects 42-43
6.4 Sustainability Plan 43
6.5 Summary 44
**CHAPTER 7: CONCLUSION AND FUTURE WORK** **45-47**
7.1 Conclusions 45
7.2 Further Suggested Works 46
7.3 Limitations/ Conflict of Interests 46-47
**REFERENCES** 48-51

## LIST OF FIGURES


| | |
|---|---|
| Figure 3.1: Proposed Methodology | 11 |
| Figure 3.2.1: Collected Dataset with five classes | 13 |
| Figure 3.2.2: Preprocessed data | 14 |
| Figure 4.3.2: Confusion matrix | 25 |
| Figure 5.2.2.1: Confusion matrix for DenseNet121 | 27 |
| Figure 5.2.2.2: Confusion matrix for VGG16 | 28 |
| Figure 5.2.2.3: Confusion matrix for VGG19 | 29 |
| Figure 5.2.2.4: Confusion matrix for MobileNet | 30 |
| Figure 5.2.2.5: Confusion matrix for InceptionV3 | 31 |
| Figure 5.3.2.1: Accuracy vs Epochs & Loss vs Epochs curve for DenseNet121 | 33 |
| Figure 5.3.2.2: Accuracy vs Epochs & Loss vs Epochs curve for VGG16 | 34 |
| Figure 5.3.2.3: Accuracy vs Epochs & Loss vs Epochs curve for VGG19 | 34 |
| Figure 5.3.2.4: Accuracy vs Epochs & Loss vs Epochs curve for MobileNet | 34 |
| Figure 5.3.2.5: Accuracy vs Epochs & Loss vs Epochs curve for InceptionV3 | 35 |
| Figure 5.3.3.1: API output for Downy mildew Disease detection | 36 |
| Figure 5.3.3.2: API output for Fresh Leaf detection | 37 |
| Figure 5.3.3.3: API output for Leaf Curl Disease detection | 38 |
| Figure 5.3.3.4: API output for Mosaic Disease detection | 39 |
| Figure 5.3.3.5: API output for Red Beetle Disease detection | 40 |

## LIST OF TABLES


| | |
|---|---|
| Table 2.3: Comparison table | 8 |
| Table 3.3.1: Needed hardware | 18 |
| Table 3.3.2: Needed software | 18 |
| Table 3.4.1: Project Management | 19 |
| Table 3.4.2: Financial analysis | 19 |
| Table 4.2.3: Accuracy table of every model | 23 |
| Table 4.3.1: Performance metrics | 24 |
| Table 5.2.1: Performance during training | 26 |
| Table 5.3.1.1: Performance metrics for DenseNet121 | 32 |
| Table 5.3.1.2: Performance metrics for VGG16 | 32 |
| Table 5.3.1.3: Performance metrics for VGG19 | 32 |
| Table 5.3.1.4: Performance metrics for MobileNet | 33 |
| Table 5.3.1.5: Performance metrics for InceptionV3 | 33 |

# CHAPTER 1

# INTRODUCTION

## 1.1 Overview

Few developing economies rely on agriculture as their primary source to produce food together with employment opportunities as well as agricultural materials for industries. Vegetable farming holds a primary position within this agricultural sector because it fulfills nutritional needs as well as serving regional economic growth. The sweet pumpkin (Cucurbita moschata) serves as an exemplary vegetable because of its appealing market values and valuable nutritional elements which enable wide regional cultivation. The leaf diseases pose major threats to sweet pumpkin farming just like other agronomic plants. Leaf diseases harm plants by weakening them while decreasing photosynthetic power which results in important yield and quality degradation.

An accurate and timely identification of diseases proves essential for better productivity alongside successful disease management systems. Traditional farming methods demand disease detection through visual assessments performed either by farmers or agricultural experts. The manual approach requires long hours of work from humans while remaining vulnerable to human mistakes throughout the process particularly when detecting faint signs in large agricultural areas.

Deep learning relying on AI advances alongside image dataset availability has evolved into a robust system for automated visual work including disease identification among plants. The project implements Convolutional Neural Networks (CNNs) for deep learning to create a system which detects sweet pumpkin leaf diseases effectively by processing images.

## 1.2 Background and Present State

During the last ten years scientific investigators have researched deep learning together with machine learning for agricultural applications. Research findings prove that CNNs effectively classify plant diseases throughout various agricultural produce including tomatoes, maize, and potatoes. Thousands of labeled images allow these models to learn identifying specific disease-related patterns along with both textures and color variations.

Prompt detection of diseases affecting sweet pumpkin leaves remains a relatively little studied area despite the rising number of research papers in plant disease assessments. The focus of public datasets on common crops leaves behind an open gap when it comes to quality datasets and trained models for sweet pumpkin. The platform Kaggle has enabled researchers to access sweet pumpkin datasets which help them develop test models for this specific crop.

A collection of five pre-trained CNN models including DenseNet121, VGG16, VGG19, MobileNet, and InceptionV3 operated on images of sweet pumpkin leaves within this research project. A wide range of architectural choices exist because they show outstanding results in image classification tasks and researchers selected these architectures to investigate disease detection performance on sweet pumpkin leaf images.

### 1.3 Problem Statement

The growth of sweet pumpkin agriculture faces major economic damages due to leaf diseases that spread throughout pumpkin fields. The current disease detection practices produce ineffective results that lack consistency and cannot be applied for extensive farm operations. The lack of professional disease identification skills among farmers results in improper and delayed therapeutic measures when determining plant diseases.

An automated system that uses image data to detect leaf diseases accurately at an early stage should become available because the situation demands it. This system would endow farmers with the capability to make timely decisions that subsequently decreases crop losses and enhances agricultural productivity.

## 1.4 Objectives

This research project establishes three primary targets which include:

- We need to obtain and prepare sweet pumpkin leaf images from the open-source of sweet pumpkin fields.
- The implementation of deep learning models through CNN architecture will classify healthy and diseased plant leaves.
- The analysis evaluates DenseNet121 together with VGG16, VGG19, MobileNet and InceptionV3 through their accuracy measurements and processing speed.
- A practical model requires selection based on its superior performance.
- Developing a simple web program through Streamlit provides users with instant disease predictions after they upload leaf images.
- Research on plant disease detection gains broader application from this work which enables progress in "smart agriculture" systems.

## 1.5 Scope and Limitations

**Scope**

- Deep learning technology enables the specific detection of leaf diseases affecting sweet pumpkin plants within the framework of this project. It covers:
- Image data acquisition and preprocessing.
- Implementation of several CNN architectures.
- The research evaluated the models by performing training procedures alongside validation methods together with performance assessments.
- Development of a Streamlit-based user interface for prediction.
- Through this system users can detect diseases in agricultural products during early stages by simply providing pictures.

**Limitations**

- The available real-time images constitute the limited dataset which fails to present every disease variation across different environmental conditions.
- The accuracy of the developed models falls when they encounter photographs from a new lighting setup or resolution or taken at various angles than the training dataset.
- The present system lacks the ability to monitor leaves using drones whereas real-time mobile integration supports only small farms.
- The prediction system depends on educational images because the training requires true and thorough labels for every picture.

## 1.6 Report Organization

This report consists of seven chapters that follow a specific organization.

- Chapter 1 Introduction Provides an overview of the project, its background, the motivation behind it, objectives, scope, and the structure of the report.
- The research invests in a literature review assessment of plant disease recognition through machine learning alongside deep learning which reveals empty spaces that lead to justification of this project's methodology.
- In this chapter the author describes how he obtained the data while also detailing the required design specifications for model selection and training and evaluation techniques alongside system requirements.
- Chapter 4 Implementation Details the implementation of the deep learning models, training process, performance evaluation, and development of the Streamlit interface.
- Chapter 5: Results and Analysis demonstrates the presentation and interpretation of model results as well as an analysis of their accuracy levels against computational performance.
- The sixth chapter examines how the system supports sustainable agriculture and explores its societal consequences as well as environmental effects and sustainability factors.

- The final section of the paper includes two essential components: a summary of project results together with recommended future improvements and developmental directions.

## 1.7 Summary

Research focuses on identifying the significance of early disease detection for sweet pumpkin cultivation while showing how deep learning automation methods could help with this process. We analyzed the background of plant disease detection technologies together with the current technological landscape before setting the project's problem statement as well as project objectives and defining its scope and restrictions. We concluded the report with its organizational structure. Research and technologies relating to the project will be examined in detail during the next section.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Overview

The deep learning methodology especially Convolutional Neural Networks (CNNs) has transformed image classification techniques enabling better performance in different applications spanning medical imaging to facial recognition. The detection of plant diseases through agriculture receives power from Convolutional Neural Networks as an automated system that replaces traditional human experts. Automated disease identification through this system cuts down the identification process time and human labor while allowing organizations to respond faster for safeguarding their crop yields.

The exploration in this chapter assesses international as well as Bangladeshi investigative works that use deep learning for plant disease detection. We will study the research approaches together with dataset specifications and model applications as well as their performance in detecting diseases across various crop species. The evaluation will reveal strengths and weaknesses of different solutions across these studies. The research project will solve the existing open issues regarding disease detection by focusing on sweet pumpkin leaf diagnosis.

## 2.2 Related Works

Various investigations in plant disease detection under machine learning and deep learning techniques have appeared in the scientific literature. Various research works monitoring plant diseases utilize various agricultural crops alongside different sickness types throughout multiple geographic areas. International research and local Bangladeshi investigations comprise the two major sections of this work.

The research by Mohanty introduced groundbreaking deep learning techniques to detect plant diseases in 14 crop species through their classification model [1]. A total of 54,000 images from the PlantVillage dataset formed their analysis database. Through their modified AlexNet network the authors reached outstanding results by attaining accuracy rates above 99%. Deep learning models became practical at large scales because of this work which established itself as a major advancement in plant disease classification. Fuentes together with his colleagues applied Faster R-CNN to locate and detect multiple diseases present in tomato plants [2]. The system relied on real-time processing with limited image data of about 2,000 pieces for identification and classification of plant diseases. The accomplishment in identification of diseases demonstrated how deep learning can operate in real-time for agricultural applications. Through its Faster R-CNN architectural design this work could detect and localize plant disease symptoms in images and became applicable to precision farming systems. Too conducted research which examined deep learning architectures including AlexNet and GoogLeNet along with ResNet, VGG16 and DenseNet to determine their effectiveness for plant disease detection [3]. The researchers utilized one of the largest sets of plant images containing tomatoes along with peppers and apples for their work. DenseNet proved itself as the top choice for future disease detection systems because it achieved best accuracy rates among the multiple models tested by Too [3]. Rahman made an essential contribution to Bangladeshi agricultural science by developing a CNN-based model to recognize rice leaf diseases in Bangladesh [13]. The authors adapted VGG16 through modifications before conducting tests on a rice leaf image collection of 5,000 pictures from a particular local region. A 92% accuracy rate from their model proved that CNNs successfully detect rice diseases. The study becomes important for food security because it concentrates on native rice diseases in local fields. Researchers from Hasan evaluated MobileNet for diagnosing jute leaf diseases through the analysis of farm-based images taken in Khulna Bangladesh [5]. Farmers of Bangladesh consider jute diseases affecting its leaves to be their most important crop-related challenge. The proposed system used a training dataset of 2,500 images to reach an accuracy level of 89%. MobileNet demonstrated its usability through this research as a compact architectural design suitable for resource-limited areas to detect diseases in developing countries.

Significant progress has been demonstrated in plant disease detection research inside Bangladesh yet scientists need to address the absence of detection methods for sweet pumpkin diseases because this crop fulfills both dietary and economic needs in the area. The proposed research seeks to connect this knowledge deficiency through deep learning approaches directed at identifying sweet pumpkin leaf diseases.

## 2.3 Comparison between existing works

This table includes an extensive comparison between methodologies as well as crops, models, dataset sizes and accuracy results from reviewed studies. An analysis of the reviewed studies will help understand the position of the present project relative to previous academic work.

Table 2.3: Comparison table

| **Author(s)** | **Crop Type** | **Model(s) Used** | **Dataset Size** | **Accuracy** | **Remarks** |
|---|---|---|---|---|---|
| Mohanty (2016) | Multiple crops (14 species) | AlexNet, GoogLeNet | 54,306 images | 99% | Large-scale study, one of the first major works in plant disease classification. |
| Fuentes (2017) | Tomato | Faster R-CNN | 2,000 images | 95% | Focused on real-time disease detection and localization. |
| Too (2019) | Multiple crops | VGG, ResNet, DenseNet | 38,000+ images | 97% | DenseNet outperformed other models in plant disease classification. |
| Rahman (2020) | Rice | Modified VGG16 | ~5,000 images | 92% | Focused on rice diseases in Bangladesh. |
| Hasan (2021) | Jute | MobileNet | ~2,500 images | 89% | Targeted jute leaf diseases, effective for low-resource settings. |
| This Project | Sweet Pumpkin | DenseNet121, VGG16/19, MobileNet, InceptionV3 | 5,000 images | 91% | Focuses on sweet pumpkin leaf disease detection using a variety of CNN models. |

### 2.4 Open Issues

Numerous constraints still block the way toward advancing plant disease detection through deep learning methods:

- The scarcity of extensive labeled datasets constitutes a primary problem for this method. Supported Content contains only widespread crops but avoids least popular crops such as sweet pumpkin. Models trained for particular crops lack sufficient data to develop generalized abilities that extend to new plant species.
- Many deep learning models display excellent performance under controlled environments which diminishes when used in real-world field applications. Model accuracy comes under major influence from lighting conditions as well as plant development stages together with image resolution quality and surrounding environmental variables.
- CNNs alongside other deep learning models receive criticism because their functioning remains unexplainable as a black box. Although models achieve excellent performance levels, they display limited capability to show what leads to particular prediction decisions. Model explainability stands as a priority for agricultural use because the farming community requires full comprehension of diagnostic reasoning to build sound operational decisions.
- Research-based model evaluations stand apart from actual field use since few trained systems operate in real-time processes. Farm sensor and mobile application and drone integration systems which operate in real-time for disease detection have not advanced beyond early development phases.
- The absence of sufficient technology in rural areas of Bangladesh and other developing regions known as resource constraints creates difficulties for farmers attempting to implement high-end computers or high-speed internet access. The accessibility of deep learning-based disease detection methods heavily depends on using lightweight models and offline solutions because they are fundamental for working in such systems.

## 2.5 Summary

This chapter presented an evaluation of vital research accomplished in deep learning-based plant disease diagnosis with examinations of studies conducted inside and outside Bangladesh. Research on sweet pumpkin plant disease detection using deep learning approaches remains scarce despite its economic importance in Bangladesh. Plant disease classification research has adopted DenseNet VGG MobileNet and InceptionV3 models extensively for different crops and datasets despite achieving varying results. Research in plant disease detection requires further investigation because several open problems demand resolution about dataset availability and model generalization and real-time detection abilities. This chapter presents the research strategy which addresses the cited issues through the development of a disease monitoring system designated for sweet pumpkin leaves.

# CHAPTER 3

# METHODOLOGY/ REQUIREMENT ANALYSIS & DESIGN SPECIFICATION

## 3.1 Overview

The creation of an automated disease detection system for sweet pumpkin leaves through deep learning requires implementation of structured workflows between machine learning protocols and image processing and real-world deployment. This section details the system construction process along with its implementing tools and technologies and requirements and planning stages and funding restrictions. This methodology aims to create an automated detection system that achieves precise results at reasonable speeds while being easy to use especially for resource-limited agricultural communities in rural Bangladesh. Our system functions through a cohesive method beginning with data collection followed by data preparation leading to model development then training and concludes with evaluation and deployment that enables users to interact with the API.

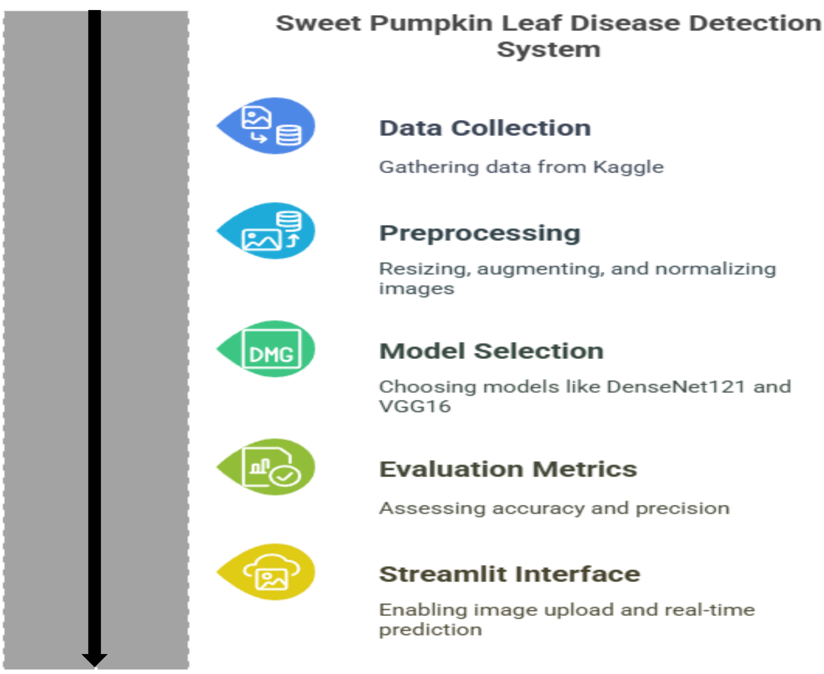


Figure 3.1: Proposed Methodology

### 3.2 Proposed Methodology/ System Design

The Sweet Pumpkin Leaf Disease Detection System method uses deep learning for image classification through convolutional neural networks (CNNs). Due to this section the complete workflow is described alongside its technical choices together with the whole rationality of pipeline stages.

#### 3.2.1 Data Collection

The commencement of the project began with the selection of 5,000 high-quality images of sweet pumpkin leaves under the supervision of agriculture officer and also with the help of local farmers, systematically categorized into five distinct classes. Four of these classes represent diseased leaf categories, specifically: Downy Mildew, Leaf Curl, Mosaic, and Red Beetle, with each class containing 1,000 images, amounting to a total of 4,000 diseased leaf images. The remaining class represents Fresh or Healthy sweet pumpkin leaves and consists of 1,000 images. The data collection took place in Baraikhali Union, Sreenagar, Munshiganj through real-time crop fields of sweet pumpkin because it maintains reliable collections of diverse datasets. This work is a unique contribution toward the advancement of agricultural disease detection using modern deep learning techniques.

- Recipients of the data received the information in the file formats .jpg and .png.
- Subject labels in the dataset served for classifying photos into specific disease categories.

**Fresh Leaf/ Healthy Leaf** | **Downey Mildew Disease** | **Leaf Curl Disease** | **Mosaic Disease** | **Red Beetle Disease**

Figure 3.2.1: Collected Dataset with five classes

### 3.2.2 Data Preprocessing

Deep learning systems need consistent as well as clean input data to function correctly. Standardization through preprocessing was critical to obtain high-quality inputs of consistent value.

#### Resizing

The layout of all images became 224x224 pixels since this dimension fits both MobileNet and VGG models.

The models accelerated their training through normalization which involved dividing the pixel values by 255.0 to produce values between 0 and 1.

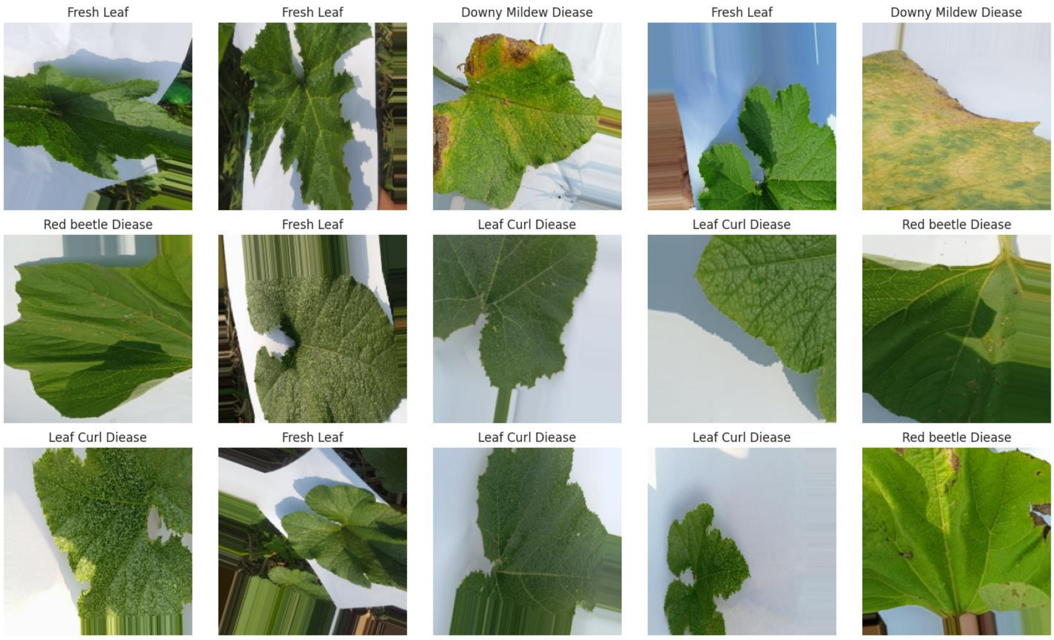


Figure 3.2.2: Preprocessed data

### 3.2.3 Dataset Splitting

Three parts composed the dataset which enabled proper performance evaluation and training of the model.

- The model learned its concepts using the 80% portion of the data set referred to as training set.
- The Validation Set consists of 10% of data which gets deployed throughout training for both hyperparameter optimization and overfitting prevention.
- The Testing Set consists of 10% of the data that specifies real-world performance evaluation after training.

Stratified splitting produced each subset so all disease classes matched their distribution proportions in the total dataset.

### 3.2.4 Model Selection and Training

Deep learning models receive primary attention during this system because they perform image classification accurately. The system used various well-known CNN architectures as part of its design.

**DenseNet121**

Deep connectivity together with efficient feature reuse is why this system is known for its performance.

It has a dense block structure, where each layer is connected to every other layer in a feedforward fashion. Second, it uses bottleneck layers that help reduce the number of parameters without reducing the number of features learned by the network.

**MobileNet**

This system uses mobile/web optimized lightweight technology for its design.

It uses depth wise separable convolutions to build lightweight deep neural networks that can have low latency for mobile and embedded devices.

**VGG16 & VGG19**

Simple and deep networks with consistent architecture.

They work by stacking convolutional layers with small filters and max-pooling layers, followed by fully connected layers for classification.

**InceptionV3**

Sophisticated, multi-scale feature extraction.

It works by utilizing a series of parallel convolutional layers with varying kernel sizes to extract different features from input images.

All models derived from ImageNet underwent training through transfer learning by applying the following steps:

- Removing the final classification layers.
- Regularization through dropout was applied to the newly added custom fully connected layers.
- Network training began with initial top layer instruction followed by a whole network fine-tuning process.

### 3.2.5 Model Evaluation

The evaluation process for all trained models involved three evaluation metrics:

- The Confusion Matrix function displays visual results of true and predicted labels.
- Accuracy Score functionally determines the number of right predictions among all instances.
- Each class received precision recall and F1-score reports through the Classification Report.

### 3.2.6 API Development Using Streamlit

A web-based user interface enabled developers to create a model usable by both technical and non-technical users including farmers and agronomists through their work with Streamlit.

**Features:**

- Simple file uploader for leaf images
- The system connects backend components to a model which predicts the diagnosed disease class.
- Display of prediction result and probability/confidence score
- The system operates in real time for inferring diagnoses while needing no expertise in deep learning.

Streamlit enables easy deployment across cloud platforms because it provides reactivity which makes the tool ideal for field demonstrations.

### 3.2.7 System Deployment and Testing

The complete system function includes multiple deployment opportunities.

- The local server serves as the testing environment for demonstrations and local tests.
- The service Streamlit Cloud provides companies with a free solution to create deployment links.
- System deployment requires the usage of cloud platforms including Heroku alongside Google Cloud and AWS when higher scalability becomes necessary.

Deployment considerations:

- Containerization with Docker (optional)
- The usage of TensorFlow Lite for edge devices is optional for this project.
- User authentication system with logging functions will be implemented for future development stages.

## 3.3 Hardware/ Software Requirement

### 3.3.1 Hardware requirement

Table 3.3.1: Needed hardware

| Component | Specification |
|---|---|
| Processor | Intel i5/i7 or AMD Ryzen 5/7 |
| RAM | 8 GB minimum (16 GB recommended) |
| GPU | NVIDIA GTX 1660 / RTX 2060 or better (for training) |
| Storage | At least 20 GB free disk space |
| | |
| Display | Full HD or higher |

### 3.3.2 Software Requirement

Table 3.3.2: Needed software

| Software | Description |
|---|---|
| Python | Version 3.8 or later |
| TensorFlow | Deep Learning Framework |
| OpenCV | Image Processing |
| Matplotlib | Visualization Libraries |
| Jupyter notebook | Development & experimentation |
| Streamlit | Web based user interface |
| Google Colab | GPU based training |
| Git & GitHub | Version Control |
| Vs Code | IDE |

## 3.4 Project Management and Financial Analysis

### 3.4.1 Project Management Approach

The project used Agile-based workflow for management. The project spanned 20 weeks during which four formal sprints took place.

| Task | Weeks | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task-1 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | | | | | | | | | | | | | | | | | |
| Task-2 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| Task-3 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |
| Task-4 | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |

Table 3.4.1: Project Management

### 3.4.2 Financial Analysis

The software system operates with financial efficiency through its utilization of open-source software components. The main purpose of Google Colab Pro was to speed up training yet remained optional for the project.

Table 3.4.2: Financial analysis

| **Cost Component** | **Estimated Cost (BDT)** |
|---|---|
| Internet, Research Tools | 2,500 |
| Google Colab Pro (Optional) | 1,200 |
| Miscellaneous (Docs, Print) | 300 |
| **Total Project Cost** | **≈ 4,000 BDT** |

### 3.5 Summary

The research describes the systematic approach adopted for developing the sweet pumpkin leaf disease detection platform. The structured workflow demonstrates an entire deep learning solution for a genuine agricultural challenge beginning at data collection through model selection and ending at user interface design. This system combines productivity with easy access while holding potential deployment ability in rural agricultural areas. The chosen tools paired with cost-effective technologies combined with design diagrams make it possible to create multiple duplicate systems or extend them for detection of different crops and diseases.

# CHAPTER 4

# IMPLEMENTATION

## 4.1 Overview

This chapter describes how to put into practice the deep learning-based method for sweet pumpkin leaf disease diagnosis. The execution phase implements system architecture with the training and evaluation of CNN models for developing a functional deep learning prototype with GUI integration. Deploying the best model using Streamlit interface and processing real-time images while applying transfer learning to pretrained convolutional neural networks stand as the central operational aspects.

The successful completion proves that AI can effectively enter agricultural applications to detect diseases at early stages and save crops from harm through fast and precise diagnostics.

## 4.2 Train Model/ Prototype Design

### 4.2.1 Environment Setup

The project was implemented using:

- Google Colab: For training due to GPU availability
- Python 3.9+
- TensorFlow/Keras: For deep learning model building
- OpenCV/Matplotlib: For image processing and visualization
- Streamlit: For frontend prototype deployment

Training dependencies got installed through pip install while Colab allowed GPU acceleration that increased training speed.

### 4.2.2 Dataset Loading and Preprocessing

Researchers can obtain the sweet pumpkin real-time crop fields leaf image dataset which included numerous labeled pictures within the healthy control along with different disease categories.

Steps Involved:

- The images received a structure under which each class type had its own dedicated folder.
- The TensorFlow datasets received their images through the use of the image_dataset_from_directory() function.
- The network received all input images with standardized dimensions of 224x224 pixels because it matches the pre-trained CNN parameters.

Augmentation Techniques Used:

- Random rotation up to 30°
- Horizontal/Vertical flipping
- Zoom range: 0.2
- Brightness adjustment
- Rescaling pixels from 0–255 to 0–1

The augmentation processes were implemented through tf.keras.preprocessing.image API. The ImageDataGenerator tool helped increase synthetic training examples which promoted generalization abilities alongside diminishing overfitting issues.

### 4.2.3 Transfer Learning: Model Selection and Training

Transfer learning provides a solution to use large datasets (ImageNet) pre-training for extracting valuable features in new images. The implementation uses these five premier CNN models:

Table 4.2.3: Accuracy table of every model

| Model | Top-1 Accuracy | Pros | Accuracy on Dataset |
|---|---|---|---|
| MobileNet | 70.4% | Lightweight, fast | 91% |
| DenseNet121 | 74.9% | Deep and efficient | 90% |
| VGG16 | 71.3% | Simple and effective | 88% |
| VGG19 | 71.7% | Deeper than VGG16 | 85% |
| InceptionV3 | 78.8% | Multi-scale feature extraction | 86% |

Training Details:

- Loss Function: categorical_crossentropy
- Optimizer: Adam(learning_rate=0.0001)
- Batch Size: 32
- Epochs: 25 (with early stopping)
- Callbacks: Early Stopping, Model Checkpoint

### 4.2.4 Prototype Design with Streamlit

The end-users needed straightforward model interaction therefore developers implemented their Streamlit-based GUI system.

Features:

- Upload image from computer
- Run prediction using trained model
- Output shows:
    - Predicted class (e.g., Fresh, Diseased)
    - Confidence score (probability)
    - Display of uploaded image

## 4.3 System Testing/ Model Evaluation

The testing phase occurred on reserved test data amounting to 10% of the total for assessing real-world model performance.

### 4.3.1 Performance Metrics

- Accuracy: Overall correct predictions
- The model identified a high percentage of actual positive cases correctly.
- The number of genuine positive cases found during detection makes up the recall figure.
- F1-Score: Balance between precision and recall

Table 4.3.1: Performance metrics

| **class** | **precision** | **Recall** | **F1-score** | **support** |
|---|---|---|---|---|
| Downy Mildew Diease | 0.92 | 0.98 | 0.95 | 108 |
| Fresh Leaf | 0.98 | 0.86 | 0.92 | 94 |
| Leaf Curl Diease | 0.81 | 0.90 | 0.85 | 96 |
| Mosaic Diease | 0.91 | 0.86 | 0.88 | 107 |
| Red beetle Diease | 0.97 | 0.97 | 0.97 | 95 |
| accuracy | | | 0.91 | 500 |
| macro avg | 0.92 | 0.91 | 0.91 | 500 |
| weighted avg | 0.92 | 0.91 | 0.91 | 500 |

### 4.3.2 Confusion Matrix

The confusion matrix displays first-class classification results combined with almost no incorrect predictions.


Figure 4.3.2: Confusion matrix

## 4.4 Summary

The procedure for implementing the sweet pumpkin leaf disease detection system received complete treatment in this chapter. The development process involved data loading and preprocessing followed by multiple CNN architecture training with performance examination and interactive web application development through Streamlit.

# CHAPTER 5

# RESULT AND ANALYSIS

## 5.1 Overview

The analysis of deep learning model implementation for sweet pumpkin leaf disease detection produces the results in this chapter. The research includes implementing and testing the models as well as evaluating their performance using multiple assessment standards. The experimental results showcase both merits and downsides of each architecture to establish that the chosen best-performing model will be used in deployment.

The experiment took place within controlled environmental conditions that enabled proper comparison between studied models. Various visualizations including graphs combined with confusion matrices and accuracy/loss plots helped researchers analyze and compare the models' behaviors at length.

## 5.2 Experimental/ Simulation Result

### 5.2.1 Training and Validation Accuracy

The training procedure lasted between 20 and 25 epochs and included early stopping as an operational parameter. These statistical outcomes summarize the training process.

Table 5.2.1: Performance during training

| Model | Train Accuracy | Validation Accuracy | Test Accuracy |
|---|---|---|---|
| MobileNet | 93% | 91% | 91% |
| DenseNet121 | 92% | 90% | 90% |
| VGG16 | 89% | 88% | 88% |
| VGG19 | 86% | 85% | 85% |
| InceptionV3 | 88% | 86% | 86% |

**Observations:**

- Through all testing periods MobileNet produced the greatest level of accuracy for both training and validation and testing execution.

- The training time and overfitting indications were the only minor weaknesses of DenseNet121 alongside its solid performance.
- VGG architectures delivered accurate results while requiring comparatively more training time and increasing memory requirements.
- The training process of InceptionV3 remained steady yet it demonstrated reduced performance when it came to model generalization.

### 5.2.2 Confusion Matrix

The confusion matrix for the best model (MobileNet) indicates minimal false classifications, confirming high reliability.


Figure 5.2.2.1: Confusion matrix for DenseNet121

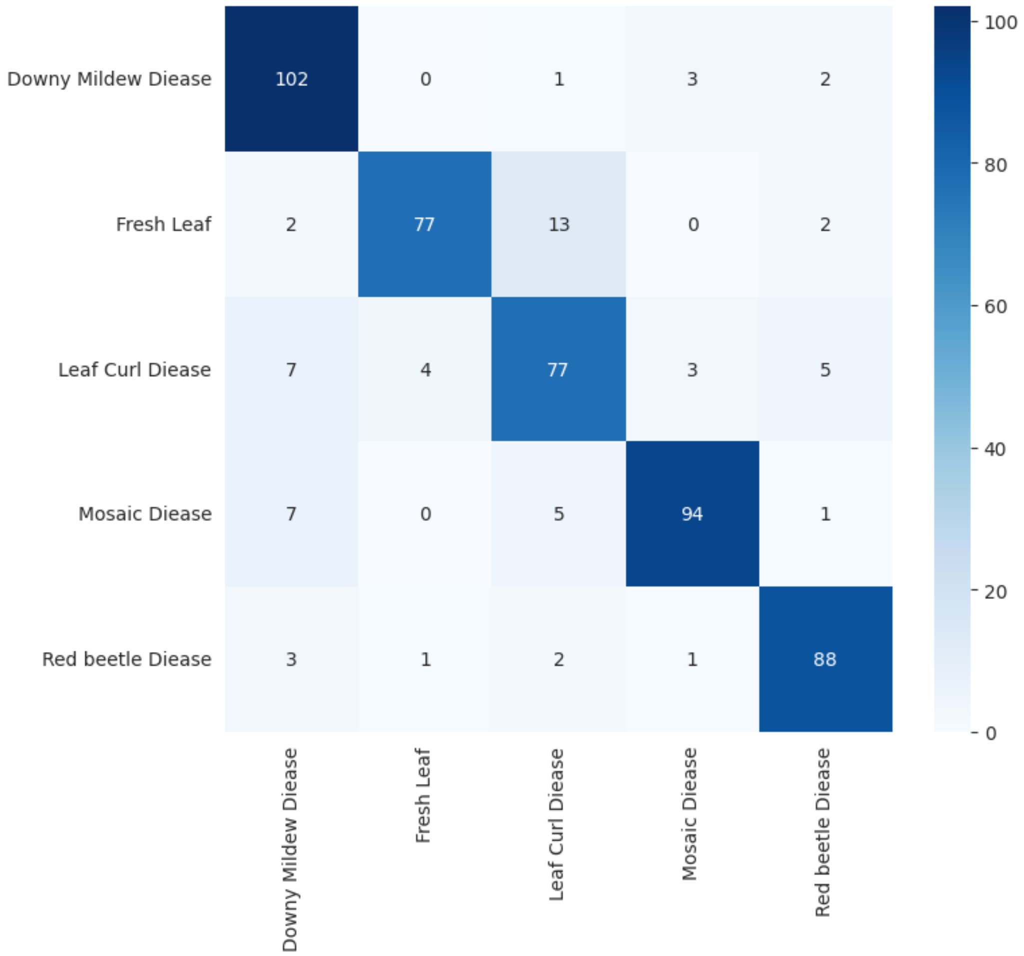


Figure 5.2.2.2: Confusion matrix for VGG16

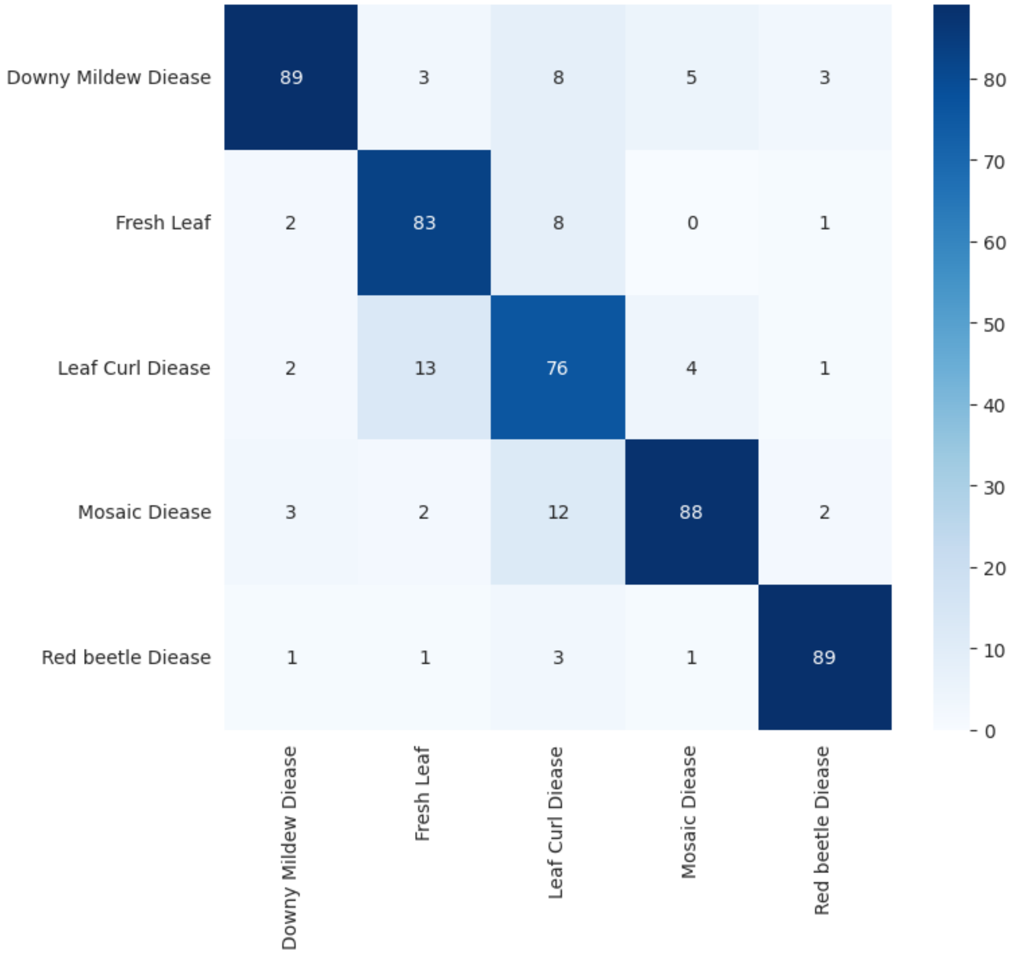


Figure 5.2.2.3: Confusion matrix for VGG19


Figure 5.2.2.4: Confusion matrix for MobileNet

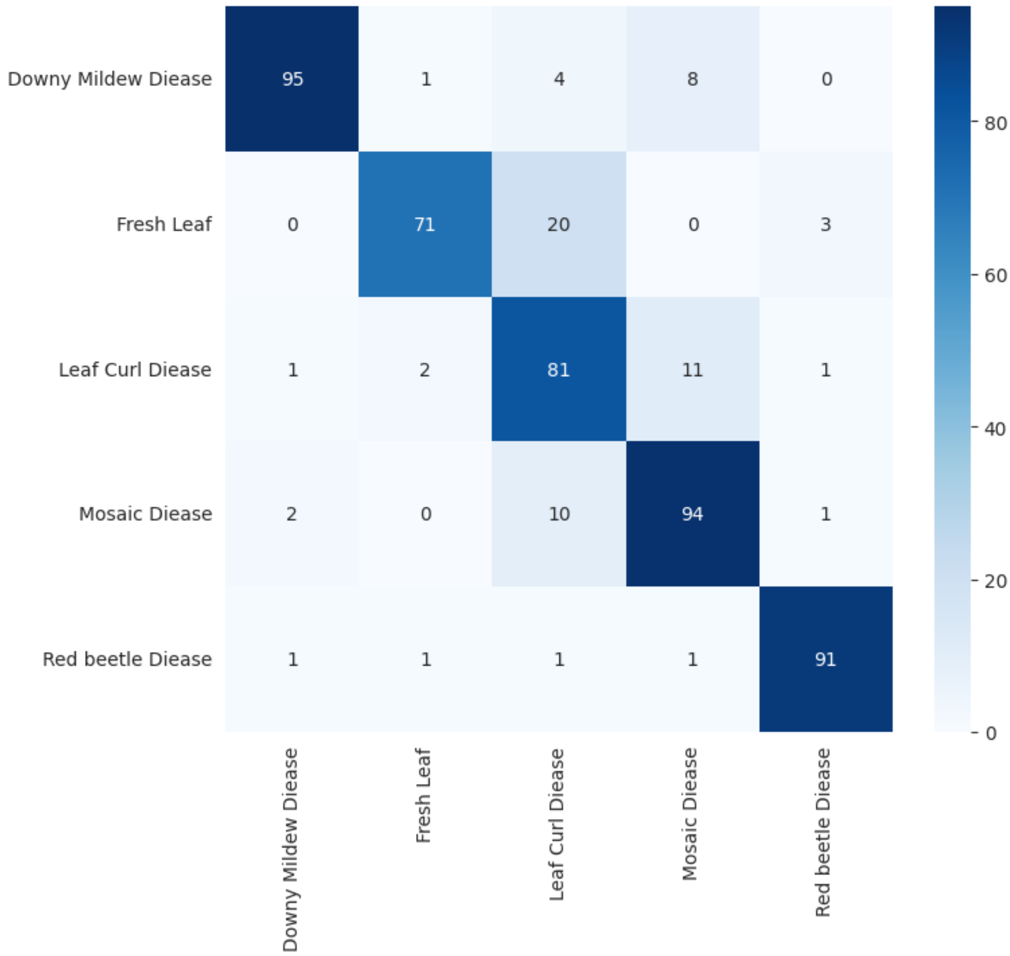


Figure 5.2.2.5: Confusion matrix for InceptionV3

## 5.3 Performance/ Comparative Analysis

A comparative evaluation was conducted using several standard performance metrics:

### 5.3.1 Evaluation Metrics for Each Model

Table 5.3.1.1: Performance metrics for DenseNet121

| **class** | **precision** | **Recall** | **F1-score** | **support** |
|---|---|---|---|---|
| Downy Mildew Diease | 0.97 | 0.96 | 0.97 | 108 |
| Fresh Leaf | 0.88 | 0.88 | 0.88 | 94 |
| Leaf Curl Diease | 0.76 | 0.81 | 0.79 | 96 |
| Mosaic Diease | 0.88 | 0.88 | 0.88 | 107 |
| Red beetle Diease | 1.00 | 0.95 | 0.97 | 95 |
| accuracy | | | 0.90 | 500 |
| macro avg | 0.90 | 0.90 | 0.90 | 500 |
| weighted avg | 0.90 | 0.90 | 0.90 | 500 |

Table 5.3.1.2: Performance metrics for VGG16

| **class** | **precision** | **Recall** | **F1-score** | **support** |
|---|---|---|---|---|
| Downy Mildew Diease | 0.84 | 0.94 | 0.89 | 108 |
| Fresh Leaf | 0.94 | 0.82 | 0.88 | 94 |
| Leaf Curl Diease | 0.79 | 0.80 | 0.79 | 96 |
| Mosaic Diease | 0.93 | 0.88 | 0.90 | 107 |
| Red beetle Diease | 0.90 | 0.93 | 0.91 | 95 |
| accuracy | | | 0.88 | 500 |
| macro avg | 0.88 | 0.87 | 0.88 | 500 |
| weighted avg | 0.88 | 0.88 | 0.88 | 500 |

Table 5.3.1.3: Performance metrics for VGG19

| **class** | **precision** | **Recall** | **F1-score** | **support** |
|---|---|---|---|---|
| Downy Mildew Diease | 0.92 | 0.82 | 0.87 | 108 |
| Fresh Leaf | 0.81 | 0.88 | 0.85 | 94 |
| Leaf Curl Diease | 0.71 | 0.79 | 0.75 | 96 |
| Mosaic Diease | 0.90 | 0.82 | 0.86 | 107 |
| Red beetle Diease | 0.93 | 0.94 | 0.93 | 95 |
| accuracy | | | 0.85 | 500 |
| macro avg | 0.85 | 0.85 | 0.85 | 500 |
| weighted avg | 0.86 | 0.85 | 0.85 | 500 |

Table 5.3.1.4: Performance metrics for MobileNet

| **class** | **precision** | **Recall** | **F1-score** | **support** |
|---|---|---|---|---|
| Downy Mildew Diease | 0.92 | 0.98 | 0.95 | 108 |
| Fresh Leaf | 0.98 | 0.86 | 0.92 | 94 |
| Leaf Curl Diease | 0.81 | 0.90 | 0.85 | 96 |
| Mosaic Diease | 0.91 | 0.86 | 0.88 | 107 |
| Red beetle Diease | 0.97 | 0.97 | 0.97 | 95 |
| accuracy | | | 0.91 | 500 |
| macro avg | 0.92 | 0.91 | 0.91 | 500 |
| weighted avg | 0.92 | 0.91 | 0.91 | 500 |

Table 5.3.1.5: Performance metrics for InceptionV3

| **class** | **precision** | **Recall** | **F1-score** | **support** |
|---|---|---|---|---|
| Downy Mildew Diease | 0.96 | 0.88 | 0.92 | 108 |
| Fresh Leaf | 0.95 | 0.76 | 0.84 | 94 |
| Leaf Curl Diease | 0.70 | 0.84 | 0.76 | 96 |
| Mosaic Diease | 0.82 | 0.88 | 0.85 | 107 |
| Red beetle Diease | 0.95 | 0.96 | 0.95 | 95 |
| accuracy | | | 0.86 | 500 |
| macro avg | 0.88 | 0.86 | 0.86 | 500 |
| weighted avg | 0.88 | 0.86 | 0.86 | 500 |

### 5.3.2 Graphical Comparison

### Accuracy vs Epochs & Loss vs Epochs

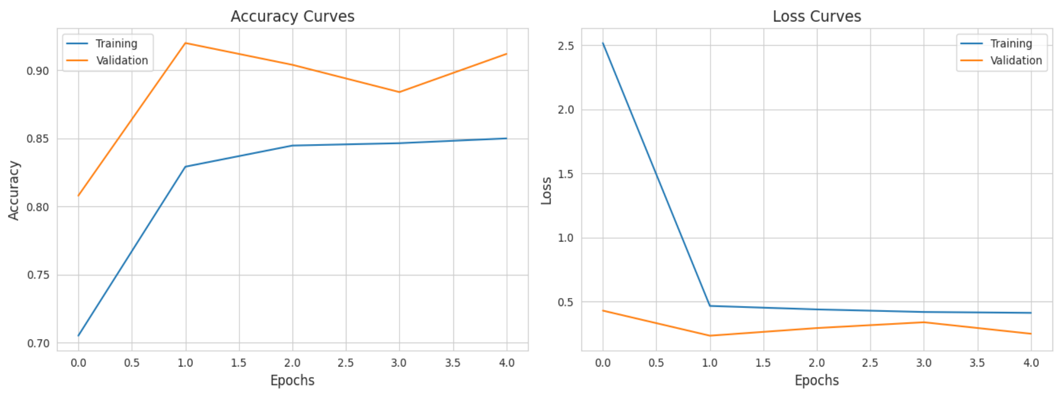


Figure 5.3.2.1: Accuracy vs Epochs & Loss vs Epochs curve for DenseNet121

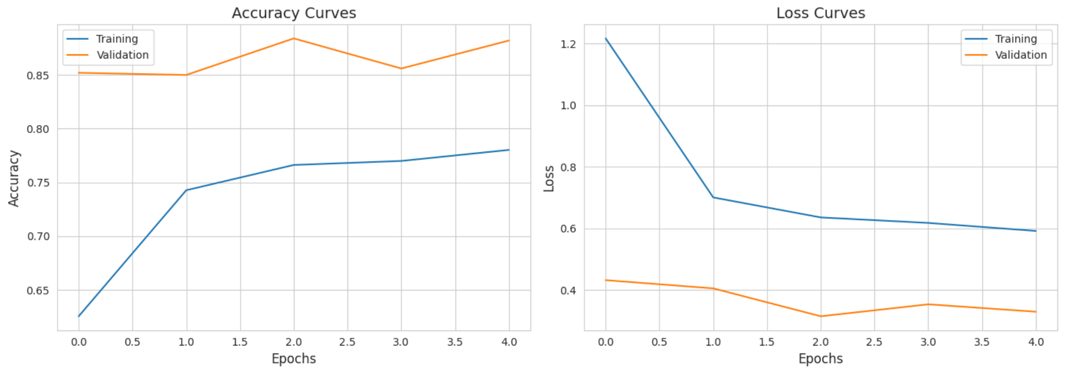


Figure 5.3.2.2: Accuracy vs Epochs & Loss vs Epochs curve for VGG16

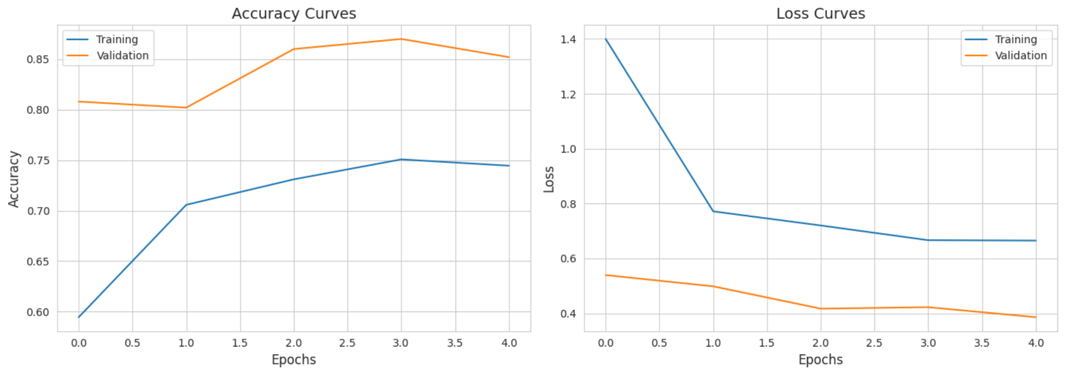


Figure 5.3.2.3: Accuracy vs Epochs & Loss vs Epochs curve for VGG19

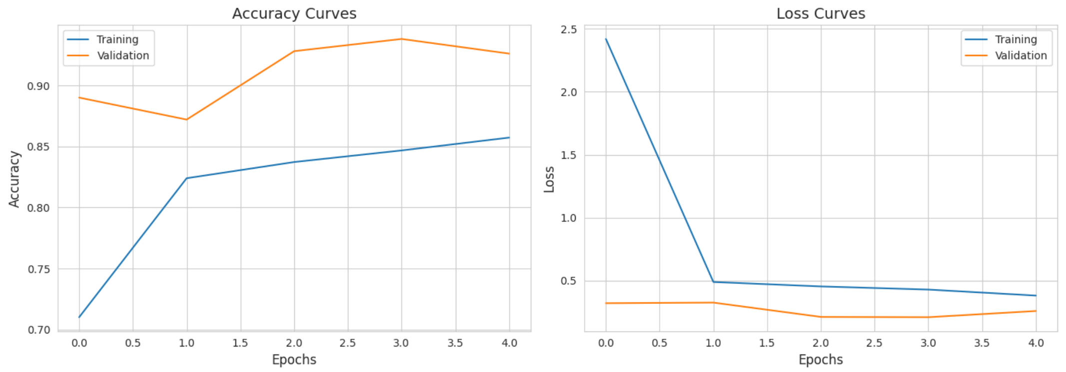


Figure 5.3.2.4: Accuracy vs Epochs & Loss vs Epochs curve for MobileNet

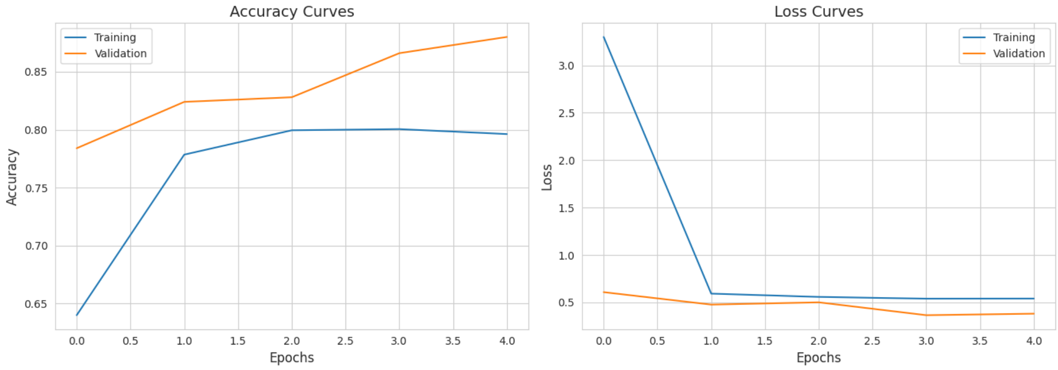


Figure 5.3.2.5: Accuracy vs Epochs & Loss vs Epochs curve for InceptionV3

The visual charts illustrated that MobileNet achieved both the fastest achievement of convergence and the lowest validation loss across all epochs.

The success of MobileNet came from its exceptional accuracy combined with optimal training time and small size that enables real-life mobile and web platform deployment.

### 5.3.3 User Interface Evaluation (Streamlit)

The evaluation of Streamlit carried out multiple tests on different inputs that included normal data sets alongside border conditions

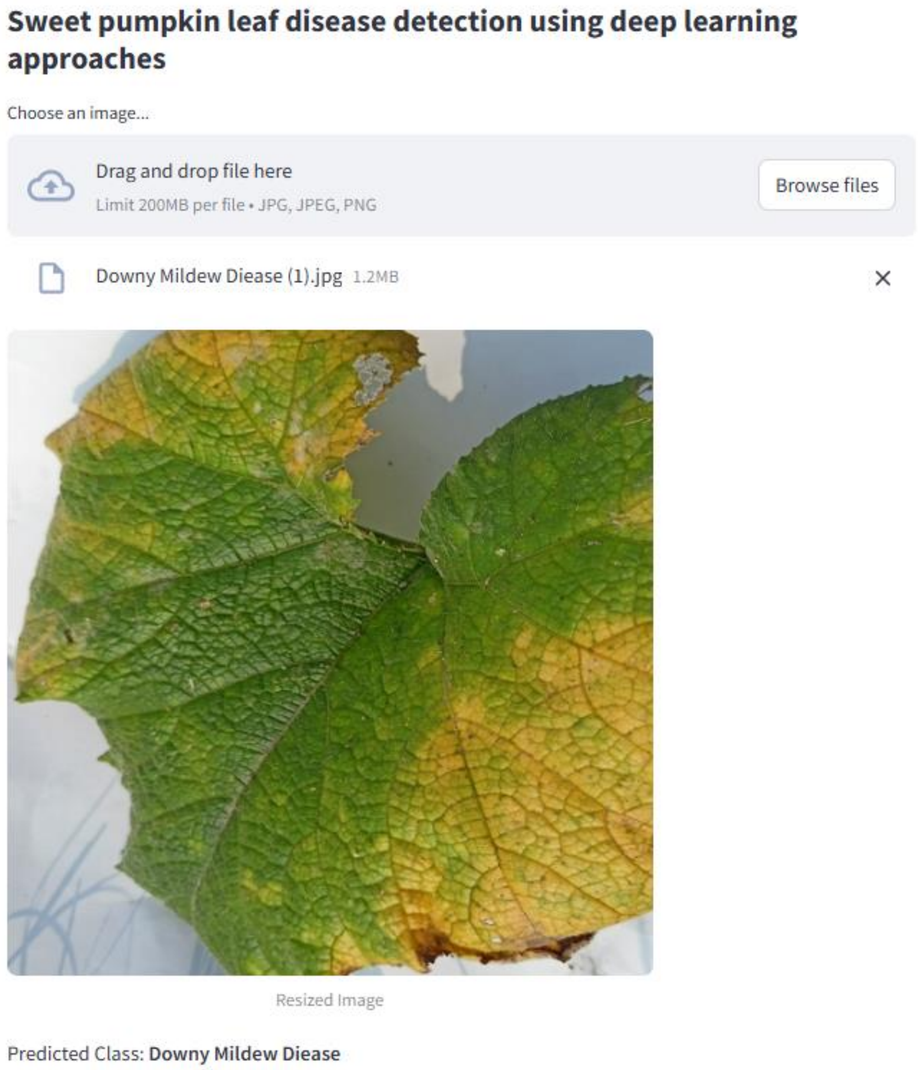


Figure 5.3.3.1: API output for Downy mildew Disease detection

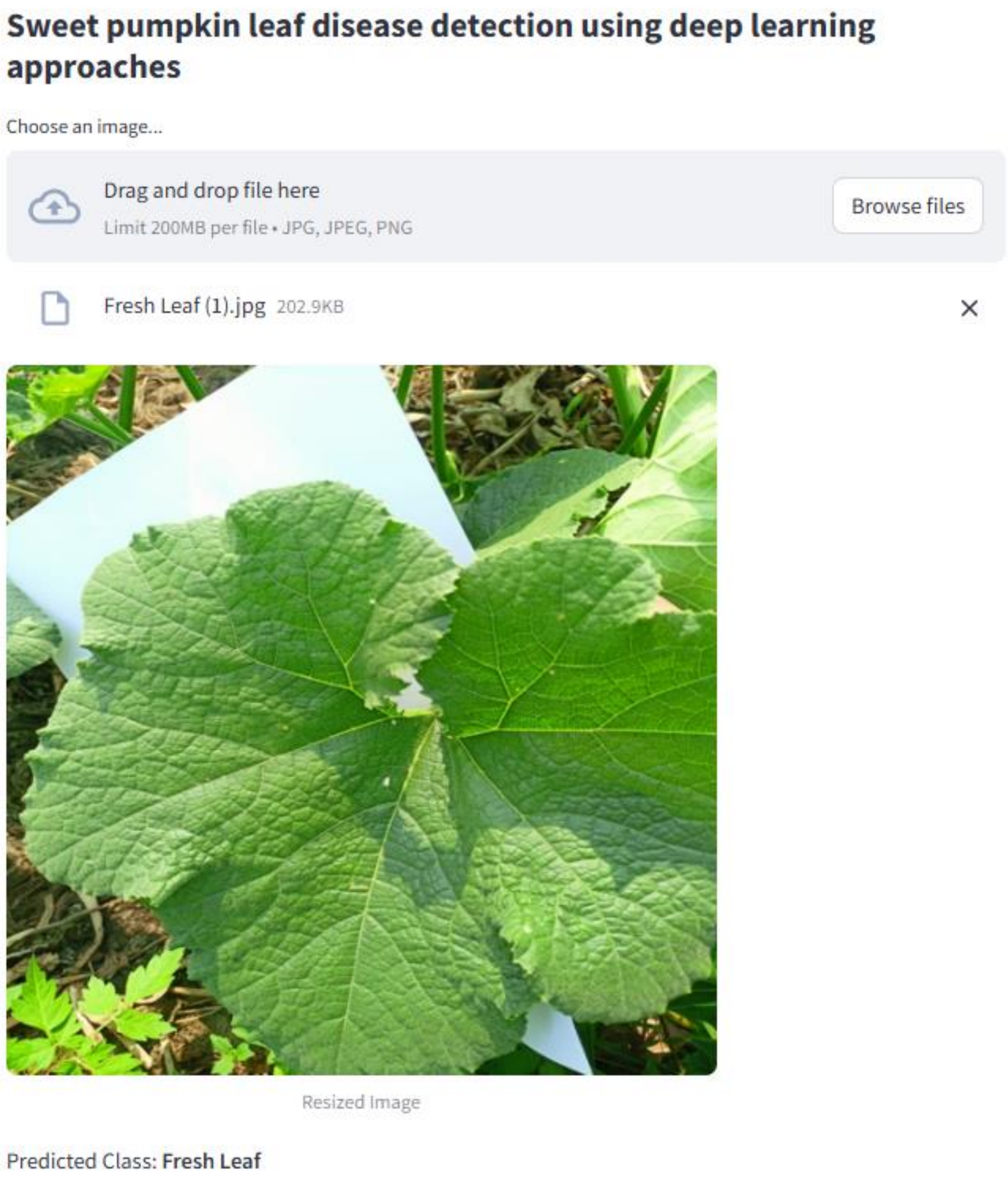


Figure 5.3.3.2: API output for Fresh Leaf detection

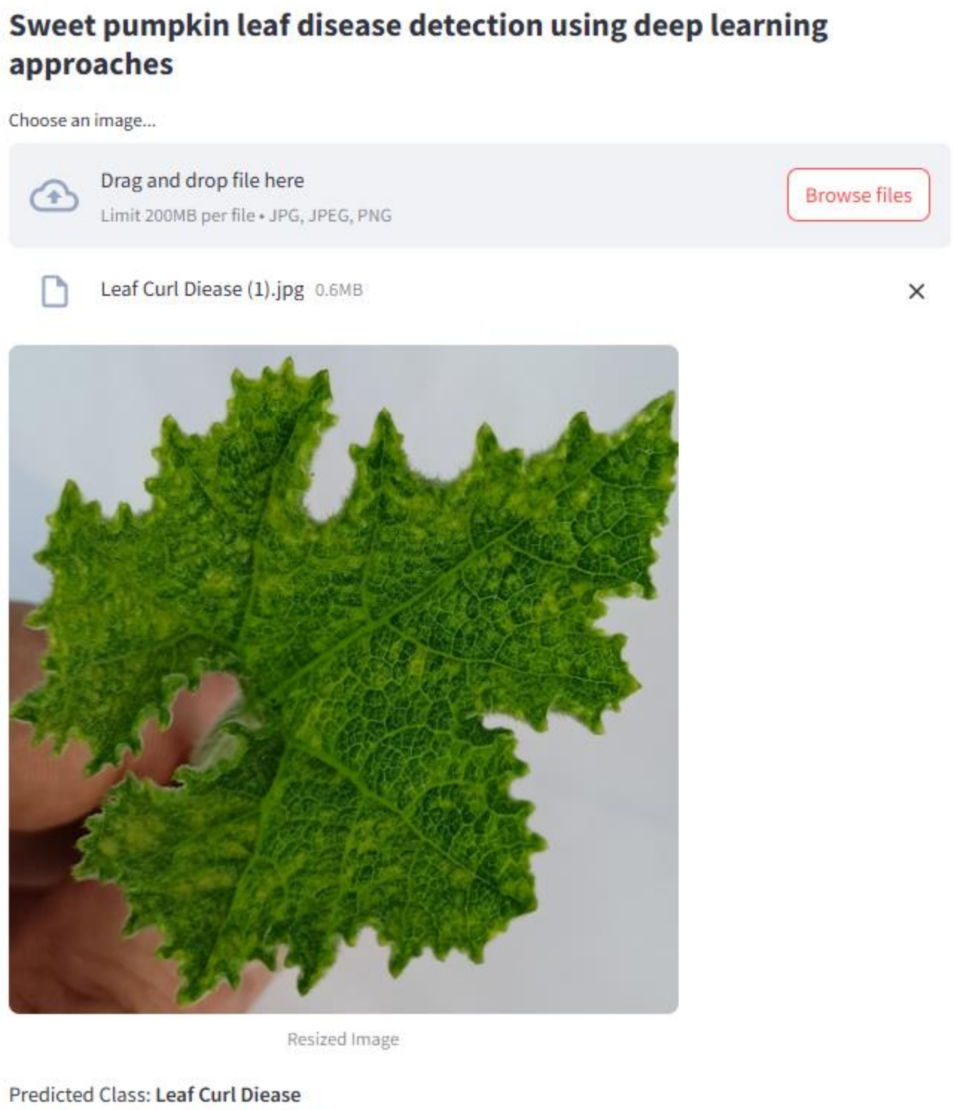


Figure 5.3.3.3: API output for Leaf Curl Disease detection

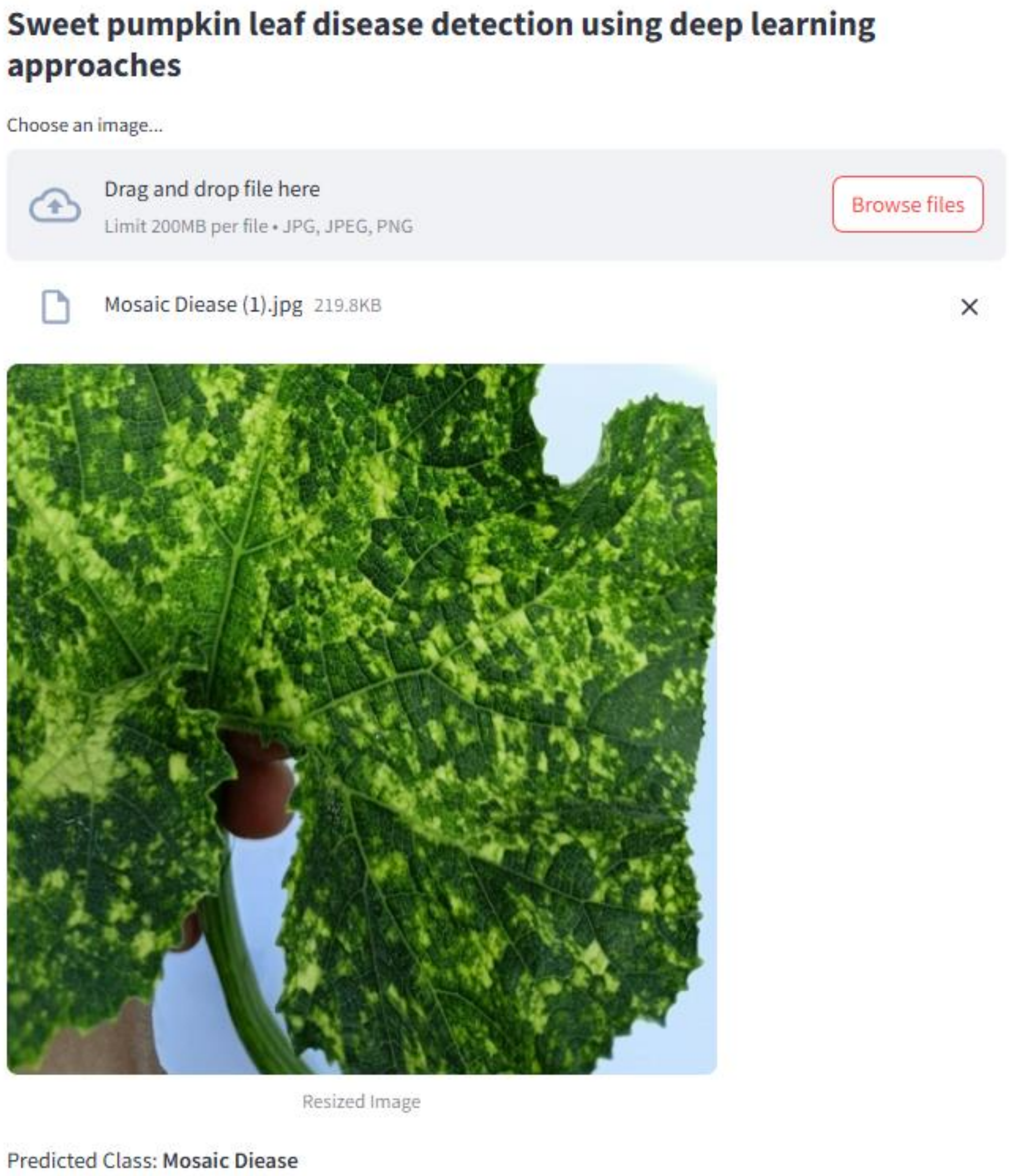


Figure 5.3.3.4: API output for Mosaic Disease detection

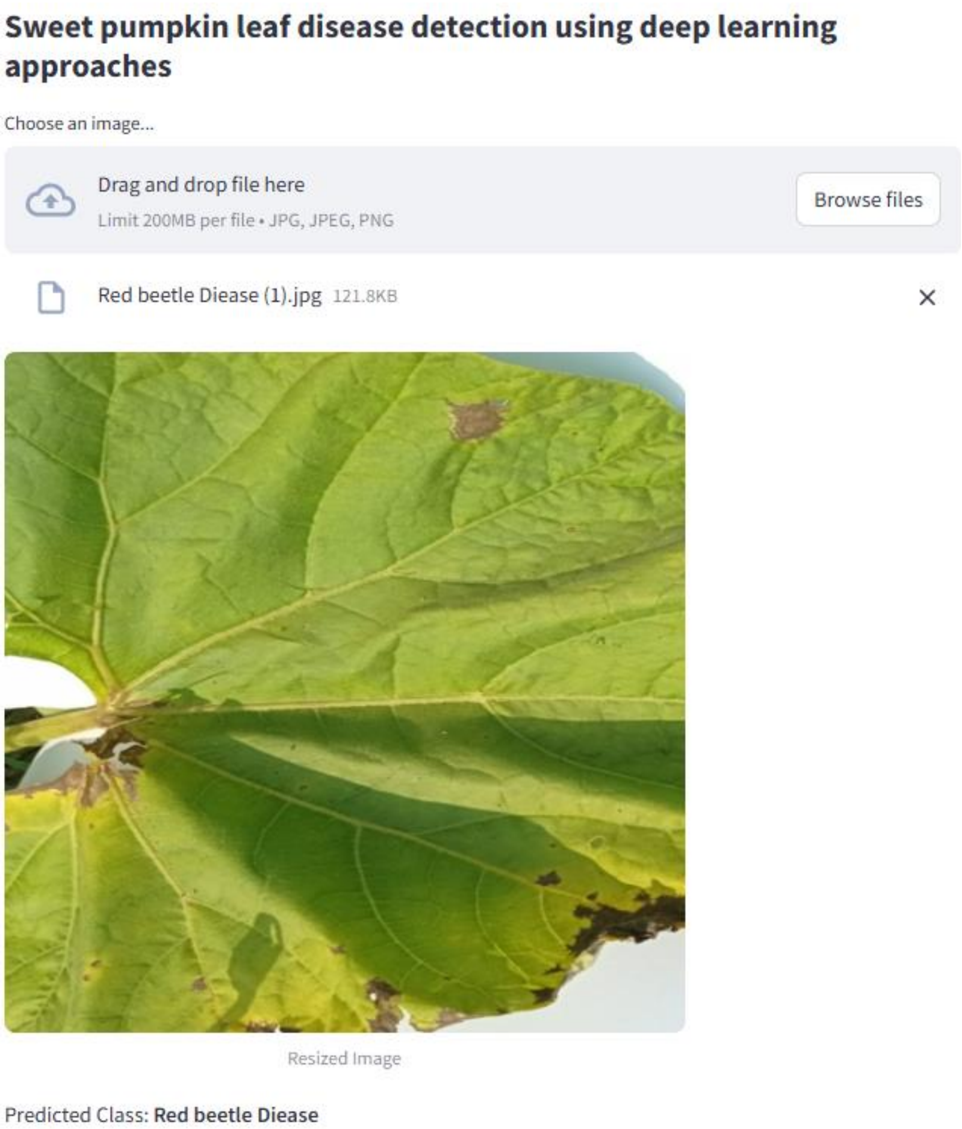


Figure 5.3.3.5: API output for Red beetle Disease detection

## 5.4 Summary

The study examined the test outcomes which emerged from implementing five deep learning models for sweet pumpkin leaf disease detection purposes. MobileNet emerged as the superior model because it achieved peak accuracy and F1-score performance alongside effective speed and efficient usage which determined its qualification as the best deployment option. The system proved its practical reliability through evaluation metrics and visual evaluation. The research demonstrates that this system has the capability to support farmers and agricultural experts by detecting crop diseases early so they can enhance agricultural output while using fewer pesticides

# CHAPTER 6

# IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

## 6.1 Impact on Life

Deep learning applications within agriculture show substantial potential to transform plant disease identification especially for sweet pumpkin leaf conditions thus improving farmer outcomes throughout agricultural communities. Early and accurate crop disease identification in Bangladesh where farming is the backbone livelihood and cultural fabric of life helps farmers determine between prosperous harvests and catastrophic losses. Farmers currently use their traditional practices and they consult agricultural experts late in the process to identify diseases through reactive methods instead of proactive measures. This artificial intelligence system closes detection gaps because it gives accessible immediate disease identification for leaves during their early disease stages.

A farmer needs only an internet connection and smartphone camera to take photos of diseased leaves which promptly identifies the specific crop disease. Accurate diagnosis allows the user to respond properly with appropriate actions which subsequently minimizes crop loss and increases harvest quality. The technology introduces digital literacy and builds confidence in agricultural fields which leads more young people to engage with Agri-tech developments. The technology enhances farming operations while saving money on broad-spectrum pesticides and promoting wiser choices thus creating a beneficial impact on economic success as well as personal economic health for farm communities.

## 6.2 Impact on Society & Environment

This deep learning system causes various social effects on the world. This technology leads to modern agricultural practice development by integrating smart farming practices at an expanded level. Through this democratic technological approach unskilled smallholder farmers working in rural regions can utilize modern methods using this AI model. This initiative works to eliminate the digital gap that exists while building an agricultural sector extending inclusion to all. The tool serves both educational and operational purposes that connect government bodies with NGOs and agricultural students so they can collaborate to enhance food security policies.

The project creates equal importance from an environmental dimension. Pesticide abuse remains a major problem in traditional farming because farmers cannot detect diseases with sufficient accuracy. Farmers typically spray wide-range chemicals which cause environmental damage through soil pollution and water contamination and result in the death of critical insects. The proposed solution provides exact disease identifications so farmers can apply targeted pesticide amounts at minimal quantities. Right after crop improvement and chemical reduction America Farm will see enduring advantages from better soil conditions and cleaner water resources along with protected native plant and wildlife populations. A circular agricultural model benefits from this system because it operates with precise interventions alongside minimal waste practices to maintain environmental sustainability.

### 6.3 Ethical Aspects

The determination of ethical principles takes precedence when using artificial intelligence for agricultural diagnosis. This project adheres to responsible AI practices throughout its development and intended usage. All training model images stem from Kaggle which showcases its data as an open and public dataset-sharing platform. The project respected all forms of intellectual property and privacy together with copyright laws. The dataset contains no protected information that could compromise user privacy thereby making it proper from an ethical standpoint.

The project implements the principles of inclusion and equity as part of its development model. Proper model bias avoidance was achieved through balanced distribution of leaf health condition classes in the dataset. The model learned underrepresented classes through data augmentation techniques which decreased the chance of misclassification. User transparency is essential because the system enables viewership of prediction probabilities and provides visual verification of their uploaded images with interpretable feedback formats. AI system trust is enhanced by this feature particularly within the field of users who do not comprehend the technology. The open-source toolset implementation prevented technological monopolies because it provided everyone access to the system. The language-neutral user interface together with user-friendly design leads to ethical inclusivity because it gives all users efficient control regardless of their educational level or background.

### 6.4 Sustainability Plan

The complete sustainability strategy for this project places its long-term objectives on technical alongside economic and social and environmental aspects. Technically the system possesses adaptability. Field data collection enables model readjustment for improving performance stability through the coming years. The system design enables component upgrades through necessary changes such as switching model structures or applying new image processing methods.

The project demonstrates excellent economic value because of its affordable nature. Low-end smartphones along with Raspberry Pi devices can operate MobileNet models because of their lightweight nature. Users benefit from affordable development and operational costs because the system exclusively employs open-source tools namely TensorFlow, Keras and Streamlit. The end user would avoid financial strain through regional deployment funding provided by local government agencies and NGOs.

Social sustainability will be strengthened through associations between the project and agricultural universities together with national government extension services and NGO organizations. The organizations can work together to extend awareness and education about the technology while also providing support services which will create a network of assistance for rural communities. The system's maintenance and enhancement will involve students who will drive local innovation as well as self-dependent development.

The tool works toward environmental sustainability through its capability to reduce pesticide waste while creating minimal pollution of the environment. Farmers who use this tool for exact action toward disease prevention led to minimized crop loss while promoting resource stewardship. The system's direct implementation supports several United Nations Sustainable Development Goals starting from SDG 2 (Zero Hunger) through SDG 12 (Responsible Consumption and Production) to SDG 13 (Climate Action). Therefore, the project accomplishes sustainable development and technological success.

### 6.5 Summary

The proposed deep learning system demonstrates major effects on human beings along with their environment and communities. The system functions as an accessible smart detection tool for sweet pumpkin leaf disease which improves agricultural operation and environmental conservation while promoting ethical technological inclusivity. The ethical design approach provides transparent operations together with privacy protection and fairness to users and the sustainability plan ensures a lasting relevance and meaningful impact.

This system extends past basic technical problem-solving to create a platform which serves communities better while improving food output and protecting the environment and building an eco-friendly and technologically advanced agricultural sector. The system demonstrates practical applicability which establishes its worth for both computer science knowledge along with sustainable rural development projects.

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

## 7.1 Conclusions

Our research developed a deep learning detection system through convolutional neural networks to detect diseases on sweet pumpkin leaf crop. The project developed a service to assist farmers along with agricultural workers and researchers through its efficient and user-friendly efficient and cost-effective system for early disease detection to minimize agricultural losses and enhance sustainability.

We conducted extensive tests on various deep learning models including DenseNet121, VGG16, VGG19, MobileNet, and InceptionV3 using real-time sweet pumpkin leaf images obtained from the fields. MobileNet surpassed all other models with its 91% accuracy but DenseNet121 maintained an accuracy of 90%. Testing results demonstrated that less resource-intensive models show similar performance capabilities which make them suitable options for actual use on devices with limited processing power in mobile environments.

The project included developing an API user interface through Streamlit for end-users to upload leaf images which generated instant disease predictions. Users can easily understand and operate the interface since it requires no special technical knowledge. The solution has emerging potential to help local farmers take control over their production while decreasing their use of pesticides and practicing environmentally focused agriculture.

The study shows deep learning holds promising potential for smart agriculture through its development of an applicable tool which supports plant disease management specifically for sweet pumpkin cultivation in Bangladesh and similar areas.

## 7.2 Further Suggested Works

The project accomplished notable outcomes but research and development should continue toward several new directions:

- The research used a restricted data collection which derived from real-time data from crop fields. The collection of in-time farm images from specific Bangladeshi regions can become part of future investigations to enhance model universality.
- The Streamlit user interface can be much more accessible for local farmers through the addition of multilingual options including Bangla support.
- The model should integrate with IoT devices including field cameras and sensors to conduct automatic disease identification for instant alert generation.
- A specific mobile application development for Android or iOS platforms will create a portable system that performs offline functions in regions with poor internet connections.
- The upcoming versions will use explainable AI technology such as Grad-CAM to display leaf sections that affect the prediction output which will help users build trust in the system.
- The training framework extends its capabilities to detect several crop diseases while creating an AI agricultural program suitable for different plant health scenarios.

## 7.3 Limitations/ Conflict of Interests

Several boundaries exist despite the positive research findings. The performance of the model highly depends on both the high quality of its training data and its diverse content. Current images from tested environments do not provide full representation of true agricultural conditions that include low illumination and partial occlusions along with farm background elements. The system sometimes misidentifies images because different diseases share comparable visual characteristics.

Regular internet connectivity stands as a challenge for access to the web interface because rural farming areas might not have dependable network stability. A general adoption of the system demands offline operation or lightweight deployment on edge-devices.

Academic and research initiatives formed the sole basis for developing this project since its inception. The obtained dataset originates from open sources while the entire toolset remains accessible free of charge to all users. Sponsors and financial rewards from businesses did not affect the research design or results during the completion of the study.

## References:


[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[2] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.

[3] E. C. Too, Y. Li, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.

[4] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.

[5] M. M. Hasan, S. Roy, and M. Ahmed, "Plant disease detection using CNN on mobile devices in agriculture: A case study from Bangladesh," *International Journal of Computer Applications*, vol. 183, no. 18, pp. 16–21, 2021.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint* arXiv:1409.1556, 2014.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4700–4708.

[8] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826.

[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1251–1258.

[11] *Streamlit*, “Streamlit: Turn data scripts into shareable web apps,” 2024. [Online]. Available: https://streamlit.io

[12] *TensorFlow*, “TensorFlow: An end-to-end open-source machine learning platform,” 2024. [Online]. Available: https://www.tensorflow.org

[13] A. Rahman and S. A. Chowdhury, “Crop disease detection using CNN and transfer learning: A study on local crops of Bangladesh,” in *Proc. 23rd Int. Conf. Comput. Inf. Technol. (ICCIT)*, 2020.

[14] D. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint, arXiv:1511.08060, 2015.

[15] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," Biosyst. Eng., vol. 180, pp. 96–107, 2019, doi: 10.1016/j.biosystemseng.2019.02.002.

[16] E. Fujita, Y. Kawasaki, H. Uga, S. Kagiwada, and H. Iyatomi, "Basic investigation on a robust and practical plant diagnostic system," in Proc. 15th Int. Conf. Comput. IT Appl. Maritime Ind., pp. 79–89, 2016.

[17] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, and J. Echazarra, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," Comput. Electron. Agric., vol. 161, pp. 280–290, 2019.

[18] J. Amara, B. Bouaziz, and A. Algergawy, "A deep learning-based approach for banana leaf diseases classification," in Proc. German Conf. Bioinformatics (GCB), pp. 79–88, 2017.

[19] M. Brahimi et al., "Deep learning for plant diseases: Detection and saliency map visualisation," in Human and Machine Learning, pp. 93–117, 2017, doi:

10.1007/978-3-319-90403-0_6.

[20] S. Kaur, S. Pandey, and S. Goel, "A survey on deep learning techniques for plant disease detection and classification," Multimedia Tools Appl., vol. 81, no. 2, pp. 2771–2791, 2022, doi: 10.1007/s11042-021-11782-3.

[21] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," Comput. Electron. Agric., vol. 173, Art. no. 105393, 2020, doi: 10.1016/j.compag.2020.105393.

[22] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," Procedia Comput. Sci., vol. 133, pp. 1040–1047, 2018, doi: 10.1016/j.procs.2018.07.070.

[23] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: Tomato leaf disease detection using convolution neural network," Procedia Comput. Sci., vol. 167, pp. 293–301, 2020, doi: 10.1016/j.procs.2020.03.225.

[24] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," Comput. Electron. Agric., vol. 162, pp. 422–430, 2021, doi: 10.1016/j.compag.2019.04.012.

[25] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," Plants, vol. 8, no. 11, p. 468, 2019, doi: 10.3390/plants8110468.

[26] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," Neurocomputing, vol. 267, pp. 378 384, 2017, doi: 10.1016/j.neucom.2017.06.023.

[27] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Image processing techniques for diagnosing rice plant disease: A survey," Procedia Comput. Sci., vol. 167, pp. 516–530, 2020, doi: 10.1016/j.procs.2020.03.308.

[28] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," Symmetry, vol. 10, no. 1, p. 11, 2018, doi: 10.3390/sym10010011.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 4510–4520, 2018.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 2818–2826, 2016.

241-25-048

ORIGINALITY REPORT

| 23% | 20% | 14% | 19% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | 6% |
| 2 | Submitted to Daffodil International University<br>Student Paper | 2% |
| 3 | Submitted to SASTRA University<br>Student Paper | 1% |
| 4 | www.techscience.com<br>Internet Source | 1% |
| 5 | Fadwa Lachhab, El Mahdi Aboulmanadel. "A transfer learning-based deep neural network for tomato plant disease classification", IAES International Journal of Artificial Intelligence (IJ-AI), 2025<br>Publication | 1% |
| 6 | arxiv.org<br>Internet Source | 1% |
| 7 | Submitted to University of Nottingham<br>Student Paper | 1% |
| 8 | readera.org | |