Deep Learning Approaches for Jute Leaf Disease Detection Using CNN Models

Final Year Design Project

By

Ushno Pabitro Biswas
213-15-4250

Md. Aminul Islam Bulbul
213-15-4298

FINAL YEAR DESIGN PROJECT REPORT

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

Supervised by

Dr. Arif Mahmud
Associate Professor and Associate Head
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

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

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

This Project titled **"Deep Learning Approaches for Jute Leaf Disease Detection Using CNN Models"** submitted by **Ushno Pabitro Biswas**, ID No: **213-15-4250** and **Md. Aminul Islam Bulbul** ID No: **213-15-4298** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025.**

BOARD OF EXAMINERS

Dr. Fernaz Narin Nur (FNN) — **Chairman**
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Abdus Sattar (AS) — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mohammad Jahangir Alam (MJA) — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Sajeeb Saha (DSS) — **External Examiner**
Associate Professor
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Arif Mahmud, Associate Professor & Associate Head,** 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:

Dr. Arif Mahmud
Associate Professor and Associate Head
Department of Computer Science and Engineering,
Daffodil International University

Co-Supervised by:

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

Submitted by:

Ushno Pabitro Biswas
Student ID: 213-15-4250
Department of Computer Science and Engineering,
Daffodil International University

Md. Aminul Islam Bulbul
Student ID: 213-15-4298
Department of Computer Science and Engineering,
Daffodil International University

ACKNOWLEDGEMENTS

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

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

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

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

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

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

ABSTRACT

One of the major cash crops in Bangladesh is jute, which is crucial for the economy and supports millions of farmers. However, jute production often faces the risk of various leaf diseases, resulting in decreased yield and quality. Thus, the development of the ability to timely identify the disease is crucial. Currently, such methods are based on manual leaf image detection and characterize the disease through expert principles. This thesis suggests developing a deep learning-based system that would automatically identify and classify jute leaf diseases using convolutional neural networks. The dataset consisted of approximately 3,500 leaf images; a total of five classes could be identified: bug attack, spot disease, twisted leaf, yellow leaf, and healthy leaf. The research included training and analyzing four popular CNN models. The best results were achieved by DenseNet169 which reached nearly 88% accuracy. MobileNetV2 was slightly worse, reaching nearly 86%, and Xception achieved an 85% result. ResNet152 showed the lowest rate with approximately 74% accuracy. Beyond the immediate contribution for the possibility to characterize and detect the presence of diseases in jute leaves, the research can be used for further development for the mobile or web application that will be highly useful in Bangladesh.

Table of Contents

Approval i

Declaration ii

Acknowledgements iii

Abstract iv

List of Figures vii

List of Tables viii

1 Introduction 1
- 1.1 Introduction 1
- 1.2 Motivation 1
- 1.3 Objectives 1
- 1.4 Methodology 2
- 1.5 Project Outcome 2
- 1.6 Organization of the Report 3

2 Background 4
- 2.1 Introduction 4
- 2.2 Literature Review 4
  - 2.2.1 Similar Applications 11
- 2.3 Gap Analysis 11
- 2.4 Summary 12

3 Research Methodology 13
- 3.1 Methodology/Requirement Analysis & Design Specification 13
  - 3.1.1 Overview 13
  - 3.1.2 Proposed Methodology/ System Design 13
  - 3.1.3 Functional and Nonfunctional Requirements 15
  - 3.1.4 UI Design 16
- 3.2 Detailed Methodology and Design 17
- 3.3 Project Plan 26
- 3.4 Task Allocation 28
- 3.5 Summary 28

4 Implementation and Results 29
4.1 Environment Setup ... 29
4.2 Performance Evaluation and Comparative Analysis ... 29
4.3 Results and Discussion ... 31
4.4 Summary ... 38

5 Engineering Standards and Design Challenges 39
5.1 Compliance with the Standards ... 39
5.1.1 Software Standards ... 39
5.1.2 Hardware Standards ... 39
5.1.3 Communication Standards ... 39
5.2 Impact on Society, Environment and Sustainability ... 39
5.2.1 Impact on Life ... 40
5.2.2 Impact on Society & Environment ... 40
5.2.3 Ethical Aspects ... 40
5.2.4 Sustainability Plan ... 40
5.3 Project Management and Financial Analysis ... 40
5.4 Complex Engineering Problem ... 41
5.4.1 Complex Problem Solving ... 41
5.4.2 Engineering Activities ... 42
5.5 Summary ... 43

6 Conclusion 44
6.1 Summary ... 44
6.2 Limitation ... 44
6.3 Future Work ... 45

References 46

List of Figures

3.1: Methodology Pipeline Diagram 14
3.2: Prototype UI Test and Output 17
3.3: Bug Attack 18
3.4: Spot Disease 19
3.5: Twisted Leaf 19
3.6: Yellow Leaf 20
3.7: Healthy Leaf 20
3.8: Sample Preprocessing 22
3.9: Dataset Split Visualization 23
4.1: Training and validation accuracy curve for MobileNetV2 31
4.2: Training and validation loss curve for MobileNetV2 33
4.3: Training and validation accuracy curve for DenseNet169 32
4.4: Training and validation loss curve for DenseNet169 34
4.5: Training and validation accuracy curve for ResNet152 32
4.6: Training and validation loss curve for ResNet152 34
4.7: Training and validation accuracy curve for Xception 33
4.8: Training and validation loss curve for Xception 35
4.9: Confusion Matrix for MobileNetV2 model 36
4.10: Confusion Matrix for DenseNet169 model 36
4.11: Confusion Matrix for ResNet152 model 37
4.12: Confusion Matrix for Xception model 37
4.13: Performance comparison of CNN models (Accuracy, Precision, Recall, F1-score) .. 38

List of Tables

2.1: Summary of Literature Reviewed…………………………………………………………… ……8
2.2: Gap Analysis Table ………………………………………………………………………… 11
3.1: Dataset distribution ……………………………………………………………………… 17
3.2: Dataset Split Distribution ………………………………………………………………… 22
3.3: Task Allocation table ……………………………………………………………………… 28
4.1: Accuracy and loss values for CNN models ……………………………………………… 30
5.1: Estimated Cost and Financial Analysis ………………………………………………… 40
5.2: Mapping with Complex Engineering Problems ………………………………………… 41
5.3: Mapping with Knowledge Profile ………………………………………………………… 42
5.4: Mapping with Complex Engineering Activities ………………………………………… 42

Chapter 1

Introduction

This chapter introduces the overall research study. It begins with the background and motivation of the work, followed by the objectives and methodology used in the project.

1.1 Introduction

The jute is commonly referred to as the "golden fiber" and is among Bangladesh's most important cash crops. It sustains millions of farmers and the economy through exports and local use. However, jute production encounters numerous obstacles, and leaf diseases are among its serious threats. Diseases, such as insect pest, spot disease, twisted-leaf, yellow-leaf, etc., can cause decrease in yield and fiber quality significantly.
Farmers depend on manual watch or advice by the local agriculture officers to identify diseases traditionally. This process is time consuming, is unstructured and relies on available experts. With the increase of technology, AI and CV have developed to be powerful applications for helping agriculture over the years. In particular, deep-learning technologies based on CNNs have achieved great success in plant disease detection globally.
The main aim of this study is to implement CNN based deep learning models for automatic detection and classification of jute leaf diseases that can be easily utilized by farmers for better technology dissemination and sustainable agricultural practices.

1.2 Motivation

The countryside in Bangladesh is heavily reliant on agriculture. Farmers may have limited access to experts who can assist them in identifying plant diseases. Inaccurate or delayed detection results in loss of revenue, over application of chemicals, and quality of crop deterioration. We are inspired to make a cheap, effective and accessible solution that gives power to farmers. With the popularity of smartphone and internet, now it is doable to develop systems which can identify the plant diseases with simple photographs. By using deep learning techniques—including MobileNetV2, DenseNet169, ResNet152, and Xception—our goal is to create an intelligent companion to the farmers who can predict diseases in the early stages and provide advice with respect to that.

1.3 Objectives

The principal aims of this thesis are:

i. To construct the jute leaf dataset which includes five categories: Bug Attack, Spot Disease, Twisted Leaf, Yellow Leaf, and Healthy Leaf.

ii. To assess and compare the efficacy of the model with the help of accuracy, loss, precision, recall, and F1-score.

iii. To investigate difficulties in data set preparation, training and generalization of models.

1.4 Methodology

The research methodology is well planned to guarantee a systematic and consistent method to devise a true jute leaf disease detection system. Data were initially collected by acquiring 3500 jute leaves' images from various sources and processed and organized these under the five-categories namely: Bug Attack, Spot Disease, Twisted Leaf, Yellow Leaf and Healthy Leaf. We are mindful of variations in lighting, angle and background during collection, as these are classic in real farm captures and have an impact on the computer vision system performance.

To process the dataset for training purposes, we undertook image resizing, normalization, and data augmentation. By resizing we made input dimensions to become the same, and normalizing helped to make pixel values smaller to be able to train faster and more stably. Augmentation methods (e.g., flipping, rotating, and zooming) were adopted to provide more diversity in the dataset and alleviate over-fitting problems.

MobileNetV2, DenseNet169, ResNet152 and Xception are considered the selected CNN architectures in solving the classification challenge. The models were chosen because they have demonstrated its efficacy and accuracy in the task of image classification. We trained our models on Kaggle GPU-enabled machines.

The models had been tested after being trained and different performance matrices such as accuracy, loss curves, confusion matrices and class-wise precision and recall have been calculated. At last, a comparative study was conducted to determine the most applicable model to deploy in practice in the agriculture field of Bangladesh.

1.5 Project Outcome

The projected result of this project will be a strong and smart deep learning–based model that may be easily applied to automatically identify and classify some common jute leaf diseases correctly with high accuracy level. Adopting and comparing several CNN models including MobileNetV2, DenseNet169, ResNet152, and Xception, this study intends to determine the architecture that is the most appropriate for real-world agricultural scenarios. By comparing them, we will demonstrate the achievements of the different models and point out the shortcomings of the models when they are used in real-world crop data, and then provide insights for future research in the field of agricultural AI. From an academic viewpoint, this paper extends the field of computer vision applications in agriculture, especially on a crop which is economically important for Bangladesh while underrepresented in the literature. The project is also anticipated to result in a benchmark dataset of labeled jute leaf images that can be used as a reference for research in the application domain.

Both in terms of technical as well as performance of the project's final deliverables. The system is expected such that it could be developed as a mobile-based or web-based applications at farmer side where regular update information of disease and advisory to take action could be given. Such a solution would reduce the manual inspection, its speed and accuracy, but also that, would enable optimization of the crop management processes by adopting more sustainable agriculture e.g., reduced use of pesticides on crops. At the end of the day, those farmers are going to do a project that ought to work, ought to increase their crop production health and, in the meantime, make Bangladeshi farmers more powerful and crop more and build a Bangladeshi economy.

1.6 Organization of the Report

This report is divide organized into six chapters:

I. **Chapter 1 Introduction:** this chapter introduces the research by discussing the background, motivation, objectives, methodology, and expected outcomes of the project. It also outlines the overall organization of the paper.

II. **Chapter 2 Background:** Presents the significance of jute in Bangladesh and provides a comprehensive literature review of related works. It also highlights existing approaches in deep learning for plant disease detection, identifies research gaps, and summarizes findings relevant to this study.

III. **Chapter 3 Research Methodology:** Describes the methodology adopted for the project, including dataset collection, preprocessing steps, proposed system workflow, requirements, and task allocation. It also details the chosen model architectures and evaluation strategies.

IV. **Chapter 4 Implementation and Results:** Explains the implementation process and experimental setup. It presents the training and validation results of the CNN models, including accuracy, loss curves, and confusion matrices, followed by a comparative analysis of the models' performance.

V. **Chapter 5 Engineering Standards and Design Challenges:** Discusses engineering standards, design challenges, and sustainability issues. This chapter also includes OBE mapping tables (Complex Engineering Problems, Knowledge Profile, and Complex Engineering Activities) and project budget estimation.

VI. **Chapter 6 Conclusion:** Summarizes the overall research, highlights the limitations of the current system, and proposes directions for future work such as developing a mobile or web application for farmers.

Chapter 2

Background

This chapter is pertinent to the study with background explanations and related experiences. It starts by detailing the importance of jute for Bangladesh, and the threat of predominantly leaf diseases. It subsequently provides an extensive literature review of the state of the art in terms of jute disease detection and other applications of AI in agriculture. Finally, a gap analysis is performed to point out limitations of previous studies and a reiteration of the chapter is presented.

2.1 Introduction

Jute is one of the main natural fibers in Bangladesh, known as the "golden fiber" for its economic and cultural importance. It is a lifeline not only for foreign exchange but also for millions of farmers and rural households. The productivity of jute is, however, frequently challenged by various diseases and pests, with a negative impact on yield and fiber quality. Among these, crop leaf diseases, including bug bite, spot disease, twisted leaf, and yellow leaf, cause the most damage, because it spreads rapidly and farmers find it hard to identify them in the field.

Typically, farmers visually inspect or seek advice from agricultural extension officers to diagnose diseases. However, these methods are not consistently accurate and can cause misdiagnosis and delayed treatment, which can reduce crop yield and economic losses. With the development of computer vision (CV) and deep learning (DL), automated approaches from diseases inspection of plants was the best solution. For this purpose, we can use CNNs to find patterns in leaf images and for a disease with high accuracy. Here, it notes that it extends these works by using state-of-the-art CNN-based dl techniques to jute context which is not so developed as other crops.

2.2 Literature Review

Over the past years, various researchers have been working on automating the detection of plant diseases with technology. In the early days, majority of these studies used simple image processing techniques and straightforward machine learning models. Such systems might work in somewhat controlled environments but often broke when testing them on real farm images. With the rapid development of deep learning such as CNN, the accuracy of plant disease detection has been greatly improved, since these models can learn significant features directly from the images. Even for jute, research is scarce as compared to other crops such as rice, wheat, or maize. Several research has been performed by employing traditional machine learning, transfer learning, object detection techniques like YOLO etc. Some other researchers have also tested ensemble models and multi-scale feature fusion. These projects have achieved promising results, yet they all pose well-known challenges like employing small data sets, limited tests under real-world setting and lack of user-friendly application for farmers.

Hasan et al. (2019) [1] introduced one of the earliest CNN-based approaches for recognizing jute leaf diseases from image data. Their system was built on convolutional neural networks trained on a small, curated dataset of diseased jute leaf images. The research demonstrated the superiority of deep learning models over conventional machine learning classifiers, especially in the representation of complex textures of leaves. While they had good accuracy their dataset was limited in size and so generalizability of their results is also limited. However, this upto the study is important as formed a base line for use of CNNs in jute disease detection and motivated the additional research work for more complex architectures and large databases.

Reza et al. (2016) [2] concentrated on using image processing technologies and machine learning classifiers for identifying the jute plant diseases. Their approach was mostly based on handcrafted feature extraction (color, shape, texture descriptors…) followed by classification (SVM, KNN…) Although the system could distinguish pathogen patterns with a reasonable degree of success, the level of manual feature engineering required was not appropriate to withstand the variability and complexity of disease symptoms observed in natural settings. In addition to the fact that the study was the first to prove that automated imaging-based detection of the disease is possible in jute, the work illustrated the limitations of classical ML techniques, which paved the way for the introduction of DL to the domain.

Godse et al. (2018) [3] presented a study for detecting jute plant disease by classical image processing techniques. They tried various filters and smart thresholding for diseased region extraction from leaf images. Categorization was carried out again with conventional ML methods. Their results served to confirm what previous studies had suggested: the performance of traditional ML methods was heavily dependant on the quality of handcrafted features. However, despite their limitation, the work contributed to a literature showing that computer vision could be relevant to agriculture, but the application of a stronger model, such as the CNNs, was required for reliable results.

Akhand (2025) [4] advanced the domain by directly applying deep learning to the problem of jute disease classification. Unlike earlier works that relied on handcrafted features, this study focused on training CNN models end-to-end using raw image data. By doing so, the model was able to automatically learn hierarchical features, making it more adaptable to complex disease symptoms. Although limited in dataset scale, this work demonstrated how deep learning could significantly improve classification performance over traditional ML, and it opened a new path for future research in applying transfer learning and larger CNN architectures to agricultural datasets.

Haque et al. (2024) [5] performed a detailed investigation of jute leaf disease detection progress using conventional machine learning approach as well as recent deep learning techniques. They investigated a variety of convolutional architectures, including light models designed for agricultural data. The study highlighted the superiority of transfer learning and fine-tuning over training from scratch for detection purposes. The findings also underscore the need for employing bigger and more diverse databases for improving model robustness. Although the reported study showed high accuracies, one of its shortcomings stemmed from the use of the controlled datasets which might not be able to fully portray variations in the field. However, this study is a pioneering effort in demonstrating how deep learning may improve disease diagnosis in jute.

Tanny et al. (2025) [6] presented DERIENet, a deep ensemble learning based method to enhance the performance of jute leaf disease detection. In the contrast with previous works with individual CNN models, DERIENet combined multiple architectures and their out- put to get more generality and accuracy. The authors showed that the ensemble learning is more effective at reducing misclassifications in comparison with the individual models, especially when diseases have similar visual symptoms. This study has contributed to pushing the limits of jute disease detection research and taking it to ensemble learning, but its primary challenge is with the computational requirement, as using multiple models can be difficult to be implemented in real time at the farmer end.

Hridoy et al. (2022) [7] proposed a multi-scale feature fusion method to identify the jute diseases and pest infestations. Their approach was to combine multi-scale features taken from CNN layers and fuse them to more accurately capture details of leaf textures and

pest damage. This is a promising approach for enhancing early detection performance, particularly when symptoms were subtle or emergent. The study also demonstrated that multiscale feature extraction is indispensable for complex agricultural image data. But the computational heavy nature of the model restricts it for research purpose rather than practical usage for real time lite mobile deployment without optimization.
Li et al. (2022) [8] proposed YOLO-JD, a specialized deep learning network for detecting jute diseases and pests directly from images. Unlike classification-based systems, YOLO-JD approached the task as an object detection problem, enabling the model not only to classify but also to localize diseased regions within leaf images. This approach is particularly important because farmers often need to see where the infection is present, not just a binary classification result. YOLO-JD demonstrated impressive performance in accuracy and speed, making it suitable for near real-time applications. Still, the limitation of this work was that it required a significantly large, annotated dataset with bounding boxes, which is expensive and time-consuming to prepare.
Zhang et al. (2024) [9] introduced JutePest-YOLO, an advanced YOLO-based deep learning framework optimized for pest identification in jute crops. Building on the strengths of the YOLO architecture, this study customized the network to detect multiple pest categories at once with high precision. The system was tested on a diverse dataset, achieving strong results in real-time detection. The contribution of this work lies in moving beyond disease detection to pest identification, which is equally important for protecting jute yield. However, the study primarily focused on pest detection and less on integrating disease detection, which means its scope was narrower compared to other multi-class approaches.
Ghosh et al. (2012) [10] investigated the detection of *Corchorus golden mosaic virus*, which causes yellow mosaic disease in jute (*Corchorus capsularis*). The authors used molecular biology methods, specifically PCR-based detection, to confirm the presence of the virus in infected plants. This research was critical because it provided a biological foundation for understanding one of the most devastating viral diseases of jute. However, the method required laboratory facilities and was not feasible for on-field application by farmers. Although not computer vision-based, this study remains an important reference as it highlighted the severity of viral infections in jute and the need for rapid, practical detection systems.
Özdemir and Cagirgan (2015) [11] described the detection and characterization of a phytoplasma disease in jute (Corchorus olitorius) in the southwest of Turkey. The reseachers identified the phytoplasma infection through field observations and molecular analysis, and the symptoms included stunted growth and abnormal leaf structures. Although the study provided useful insights in the phytopathology domain, a working prototype was available based only on specific geographical areas and no automatic approach was suggested. This gap highlights the need for scalable computer vision– based solutions that are resilient to varied field conditions.
Cagirgan et al. (2014) [12] was the first to report incidence and symptomatology of phyllody disease in Corchorus olitorius (jute). They recorded plant traits, such as modified floral organs and leaf disorders resulting from phytoplasma infection. Their results were very important for agricultural pathology but no computational methods were applied to disease identification techniques. However, this study demonstrated a wide variety of diseases associated to the culture of jute and reminded for multidisciplinary approaches combining pathology and the new technologies driven by AI.
Meena et al. (2015) [13] provided an exhaustive summary on diseases of jute and allied fiber crops. Their review classified a variety of fungal, bacterial and viral diseases and outlined its symptoms, effects on yield, and conventional control measures. However, the study was merely descriptive, without any form of computer-aided

diagnosis.METHODSImage acquisition procedures were approved by the institutional review board. Nevertheless, it is a valuable asset for generating datasets, and formulating detection of diseases as computational technology problems.

Geetha et al. (2020) [14] studied leaf disease identification in plants using machine learning for various crop plants. Their system involved image pre-processing techniques including segmentation and feature extraction, which are then classified using machine learning algorithms, for example SVM. Although their study was not based on jute, it verified the feasibility of ML techniques in disease identification related to plant species. But the hand-crafted features somewhat reduce the flexibility of the model when coping with complex real-world images. This is a limitation that shows the advantage of deep learning techniques like CNNs, which learn informative features without human intervention.

Talukder et al. (2025) [15] discussed a weighted–average decision scheme over visual symptoms and field agronomic knowledge for nutrient deficiency diagnosis in rice. Rather than deep CNNs, the work operationalized a rules-and-weights approach, which maps visible leaf symptoms (intensity of chlorosis, interveinal yellowing, and necrotic spot presence) to probable nutrient deficiencies (e.g., nitrogen, potassium, magnesium). Their pipeline prioritized inexpensive computation and transparent decision logic which makes it favourable as field-advisory tools where explainability is concerned. While the approach is not jute-specific and not end-to-end deep learning-based, it shows how computational decision support can serve to farmers with little resources. The primary limitation is due to the fact that hard-coded weights can easily become brittle with changing illuminations, camera quality, etc., as well as variances in cultivars, while CNN architectures are trained to learn such weights/features from data. However, the paper is relevant to this thesis as it illustrates a complementary route: knowledge-based diagnosis and their merits such as simplicity and explainability can be combined with our method (accuracy without direct knowledge) for actionable advice post-disease identification.

Meena et al. (2015) [16] combined a farmer-friendly guide, where fungal, bacterial and viral jute specific threats and their symptomatology are detailed for jute leaves and stems, with field recommendations into a complete and up-to-date agronomic reference. Not exactly a computer-vision paper, but still useful for labeling & class definition: it explains how symptoms like yellowing, leaf curling/twisting, spotting all are manifest in a real farm, and with which causal agent & growth stage they relate to. We used this source as an anchor for choices for dataset creation and class naming (Bug Attack, Spot Disease, Twisted Leaf, Yellow Leaf and Healthy Leaf). For automation, a dearth of imaging protocols and annotated photo corpora is a limitation: our work here leverages this agronomic ground truth, in order to generate a machine-friendly dataset amenable to be used by CNNs.

Reza et al. (2016, ICEEICT — image processing+ML, long footnote) [17] presented one of the earliest automated pipeline for jute disease recognition covering traditional hand-crafted features (e.g., color moments, GLCM texture, morphological descriptors) followed by SVM/KNN classifiers. While their results confirmed that color-space conversion followed by background removal already significantly increases separability. For our thesis, it is relevant for two reasons: it gives us a benchmark that deep CNNs should strive to exceed, and it motivates us to apply data augmentation (rotation, flips, zoom) since it approximates field conditions where hand-crafted features fail. The limitation of this paper — small datasets and limited feature sets — justifies our transfer learning with MobileNetV2, DenseNet169, ResNet152, and Xception.

General ML-to-DL transition in agricultural vision (context from Geetha 2020 and related works) marks a broader shift from classical pipelines toward end-to-end CNNs for leaf disease classification across crops. These studies show that when datasets are heterogeneous (different fields, devices, lighting), learned convolutional features outperform fixed descriptors and scale better as more images become available. This transition also introduces standard metrics (accuracy, precision, recall, F1) and evaluation artifacts (learning curves, confusion matrices) that we adopt in this thesis. The remaining challenge highlighted in these works is deployment: balancing accuracy with model size and inference speed—precisely why we include MobileNetV2 (lightweight) alongside deeper nets, and why our future plan targets a mobile/web interface for farmers.

Table 2.1: Summary of Literature Reviewed.

Author (s)	Year	Title	Methodology	Key Findings
Hasan et al.	2019	Recognition of Jute Diseases by Leaf Image Classification	CNN (single architecture), supervised training on jute leaf photos	Showed CNNs outperform classical ML for jute disease images; limited by small dataset and single-model scope.
Reza et al.	2016	Detecting Jute Plant Disease using Image Processing and ML	Hand-crafted features (color/texture) + SVM/KNN	Proved feasibility of automated jute disease recognition; feature engineering struggled under varied lighting/backgrounds.
Godse et al.	2018	Detecting Jute Plant Disease using Image Processing and ML	Segmentation + classical ML classifiers	Achieved moderate accuracy; highlighted brittleness of manual features and need for deep learning.
Akhand	2025	Deep Learning–Based Jute Leaf Disease Identification	End-to-end CNN on raw images	Demonstrated gains from deep learning over hand-crafted features.
Haque et al.	2024	Advancements in Jute Leaf Disease Detection	Transfer learning with multiple CNNs	Fine-tuned pretrained models improved accuracy; results strong on controlled data, real-field generalization remains a challenge.

Tanny et al.	2025	DERIENet: Deep Ensemble for Jute Leaf Diseases	Ensemble of CNN models, output fusion	Ensembles reduced misclassification and improved overall performance; higher compute cost for deployment.
Hridoy et al.	2022	Deep Multi-Scale Feature Fusion for Early Recognition of Jute Diseases and Pests	Multi-scale CNN feature fusion	Better early detection of subtle symptoms; model complexity heavy for mobile devices.
Li et al.	2022	YOLO-JD: Jute Diseases & Pests Detection from Images	YOLO object detection (localize + classify)	Real-time detection with localization; requires large, box-annotated datasets that are costly to prepare.
Zhang et al.	2024	JutePest-YOLO: Pest Identification & Detection	Customized YOLO for multiple jute pests	High precision and speed for pests; focused less on leaf diseases, so scope narrower than multi-class disease studies.
Ghosh et al.	2012	Detection of Golden Mosaic Virus in Jute	PCR/molecular diagnostics	Established biological confirmation of major viral disease; not field-ready or automated for farmers.
Özdemir & Cagirgan	2015	Phytoplasma Disease of Jute in Turkey	Field survey + molecular identification	Documented phytoplasma symptoms; no computational recognition, underscores need for CV-based tools.

Cagirgan et al.	2014	Phyllody Disease in *Corchorus olitorius*	Symptomatology report, agronomic analysis	First occurrence report; expands disease spectrum relevant to labeling and class design.
Meena et al.	2015	Diseases of Jute and Allied Fibre Crops	Agronomic review/monograph	Comprehensive catalog of jute diseases and management; useful for dataset labeling and Class definitions.
Geetha et al.	2020	Plant Leaf Disease Classification & Detection	Image preprocessing + ML (SVM, etc.)	Validated ML for agriculture across crops; hand-crafted features limited generalization versus CNNs.
Talukder et al.	2025	Nutrient Deficiency Diagnosis of Rice by Weighted Averages	Knowledge-based rules/weights	Low-compute, explainable decisions for crop issues; complementary to CNNs but less robust to image variability
Reza et al. (ICEEICT)	2016	Detecting Jute Disease (alt version)	ML with hand-crafted features	Reinforced ML baseline; small dataset.
General DL in agriculture	2020	CNN/YOLO in crop disease detection	Deep CNN/YOLO	Shift from ML → DL; introduced accuracy, F1 metrics.
Tanny et al.	2025	DERIENet	Ensemble CNN	Showed ensemble strengths in complex datasets.

2.2.1 Similar Applications

1. PlantVillage Nuru (developed by Penn State + FAO)

    i. Mobile app that uses CNNs to identify over 30 crop diseases.
    ii. Works offline, designed for smallholder farmers.

    Limitation: Does not support jute, mostly crops like cassava, maize, and beans.

2. Plantix (by PEAT, Germany)
    i. Mobile app used widely in India and Bangladesh for crop diagnosis.
    ii. Detects nutrient deficiencies, diseases, and pests.

    Limitation: Requires strong internet and does not specialize in jute.

3. Leaf Doctor (University of Hawaii)
    i. Research-based mobile app for quantifying disease severity in leaves.
    ii. Uses image segmentation and visual scoring.

    Limitation: More of a research tool than farmer-friendly app.

4. AgroAI Web Platform (generic case study from literature)
    i. A web-based system that allows farmers to upload images for crop diagnosis.
    ii. Uses CNNs (ResNet, Inception) for classification.

    Limitation: Works for wheat, rice, maize; not extended to jute.

5. Deep Learning–based Rice Disease Detection (Case Study)
    i. Academic project in Bangladesh that applied CNNs for rice blast and sheath blight detection.
    ii. Achieved high accuracy, proved feasibility of DL for local crops.

    Limitation: Rice-specific; no extension to fiber crops like jute.

2.3 Gap Analysis

A lot of work has been done in the field of plant disease detection using machine learning and deep learning but there are several voids when determining the specific aspect of jute leaf disease detection. Most of the current systems addressed well-investigated crops (e.g., rice, wheat, and maize), and the crop jute that is important for Bangladesh and other south Asian countries have received less attention. The datasets, methods, and farmer application readiness are also obviously constrained. This information is presented in table form:

Table 2.2: Gap Analysis Table.

Features	Reza (2016)	Hasan (2019)	Haque (2024)	Tanny (2025)	Li (2022)	Zhang (2024)	Proposed System
Jute-Specific Dataset	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Dataset Size > 1,000	No	No	Yes	Yes	Yes	Yes	Yes
Multiple Disease Classes	No	Yes	Yes	Yes	Yes	Limited (focus on pests)	Yes
Augmentation Applied	No	Limited	Yes	Yes	Yes	Yes	Yes

Transfer Learning Used	No	Yes	Yes	Yes	Yes	Yes	
Comparative Models	No	No	Yes	No	No	No	Yes
Deep CNN Architectures	Limited (basic CNN)	Yes (basic CNN)	Yes	Yes	Yes	Yes	Yes (MobileNetV2, DenseNet169, ResNet152, Xception)
Evaluation Beyond Accuracy	No	Yes	Yes	Yes	Yes	Yes	Yes
Confusion Matrix Analysis	No	No	Yes	Yes	No	Yes	Yes
fApplication Readiness (Web/Mobile)	No	No	No	No	No	No	Yes
Farmer-Centric Design	No	No	No	No	No	No	Yes

From the above study, it is clear that the existing works have various limitations. The majority based on small or unbalanced datasets under one method, employed few models under comparison without third-party benchmark or only presented accuracy as the main criterion. Moreover, few studies made the effort to test and design systems that could be directly available to farmers. Our proposed system fills these lacunae by harvesting a larger and more diverse dataset consisting of more than 3,3.1500 images, employing preprocessing and augmentation techniques, and performing a comparative assessment of four state-of-the-art CNN models. In addition, the system gets closer to be ready for application with the development of a prototype user interface for farmers.

2.4 Summary

In other words, despite encouraging advances in the use of machine learning and deep learning for plant disease detection, the research on jute leaf diseases has only captured limited and scattered attention. Although jute is an important crop for Bangladesh and other South Asian countries, it is understudied compared to the more common rice, maize, or wheat. Methods) Datasets are usually small, class coverage is narrow, and the performance is not comprehensive.

By addressing the dataset scale, model comparison and evaluation methodology gaps, this work fills the void in an understudied area of jute leaf disease detection. By employing larger dataset, many CNN architectures and good performance, has shown that if properly designed, deep learning-based algorithms can be reliable and useful tools for agricultural disease control.

Chapter 3

Research Methodology

This chapter describes the methods used in the study. It starts with explaining the methodology and the target definition (3.1), the description and preparation of the dataset (3.1.1–3.1.3), architecture and rationale of the models (3.1.4), and with the training/evaluation strategy (3.1.5). Then the chapter initiates the workflow of the system (3.2), list down the hardware and software requirements (3.3), illustrates it using a data flow diagram (3.4) and ends with task delegation (3.5).

3.1 Methodology

3.1.1 Overview

The general step work in this study framework was developed so that a systematic and scientific solution to the problem of jute leaf disease detection could be found. Given this study is not intended to produce a commercial product but to rather assess the performance of deep learning approaches in a research environment, it was important here for the pipeline to be conducted in a way that maintains both experimental rigor and reproducibility.

2.1 Data collection and preparation we collected more than 3,500 jute leaf images in total that are classified into five classes, namely: Bug Attack, Spot Disease, Twisted Leaf, Yellow Leaf and Good Leafs. The images were: collected, pre-processed, and augmented to obtain both consistent and variable for better image datasets.

Furthermore, the study utilized transfer learning to fine-tune four widely adopted CNN architectures (MobileNetV2, DenseNet169, ResNet152, and Xception). These models have been selected based on a balance between computational complexity and prediction accuracy, and are compared against one another to understand their appropriateness for agricultural disease classification.

Models were trained and tested using a stratified split of the dataset, and performance records such as accuracy, precision, recall, F1-score, and confusion matrices were used. Additionally, training and validation curves were observed to detect overfitting and assess model generalization.

Finally, the system was conceptualized with the use of workflow and data flow diagrams, which visualized the data's route from image input to prediction output. This allows for full understanding and reproduction of the whole approach into further research.

3.1.2 Proposed Methodology

The architecture of the proposed system is a design pipeline for developing a robust, efficient and scalable deep learning-based jute leaf disease recognition system. The workflow proceeds through well-defined stages starting from capturing a dataset of jute leaf images. The images are then pre-processing and augmented before they are divided between training, validation, and testament data. The resulting data is then subjected to different CNN models for model learning as well as performance comparison through several metrics. The workflow is illustrated in Figure 3.1.

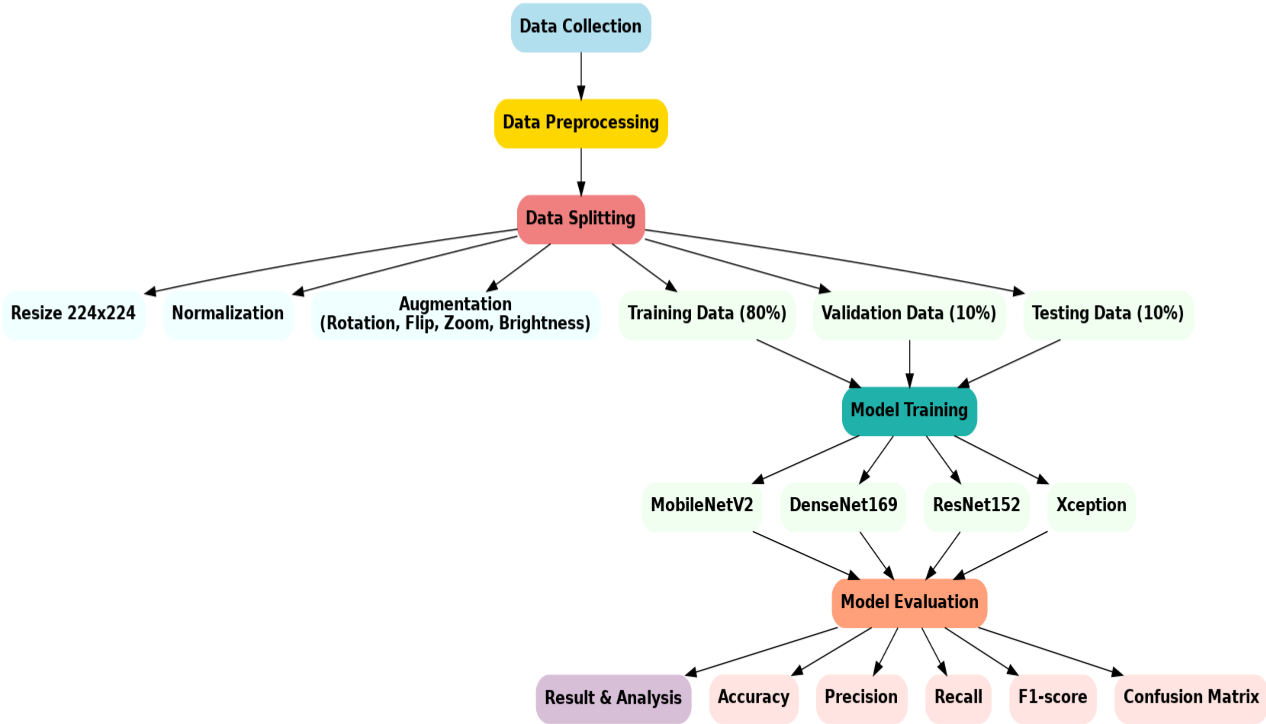

Figure 3.1: Methodology pipeline diagram.

The core elements of methodology are as follows:

i. **Data Collection:** The dataset contains over 3,500 jute leaf images with the leaves' conditions belonging to one of the five categories: Bug Attack, Spot Disease, Twisted Leaf, Yellow Leaf, and Healthy. Photos were taken during field work and from online databases. This guarantees variation in illumination, leaf side and disease state.

ii. **Data Preprocessing:** The collected images were then standardized to 224 × 224 pixels, the image pixels were normalized into the [-1, 1] scale, and some augmentation methods, e.g. rotation, flipping, zooming and brightness were used. This teaches the model to be invariant to real-world variation, and thus makes it more robust.

iii. **Data Splitting:** The dataset was split into three parts: 80% training, 10% validation, and 10% testing. This balanced sample splitting enabled fair model evaluation and the facts that the proportional testing ratio was not significantly less than the inverse of the class proportion and that the stratified ratio was less than (inverse of class proportion) ensured each class count had sufficient number of equal splits represented by its count.

iv. **Model Selection:** We compared our selected model with four CNN architectures: MobileNetV2, DenseNet169, ResNet152, and Xception. We selected these models in view of their good performance in image classification and their serviceability for transfer learning.

v. **Model Training:** The second aspect is that transfer learning was used with both the pre trained models, where the base layers were frozen and the final layers are fine tuned. For class prediction, a classification head, primarily composed of Global Average Pooling, Dropout and Dense Softmax layers was included. The model was trained on Kaggle with GPU resources using Adam optimizer and categorical cross-entropy loss.

vi. **Model Evaluation:** The models were tested by using various metrics such as accuracy, precision, recall, F1-score and confusion matrices. This holistic method provided an objective benchmark for performance not only in terms of accuracy, but for all disease classes.

vii. **Result and Analysis:** Finally, findings of each model were contrasted. Training and validation curves were examined to identify overfitting or under fitting. We adopted both the overall accuracy and the class-wise reliability as indicators to select the best-performing model for the potential application in agriculture.

3.1.3 Functional and Nonfunctional Requirements

A clear understanding of the system's functional and nonfunctional requirements is crucial for guiding the design and development of the system.

Functional Requirements

Functional requirements define the specific features and behaviors that the proposed jute leaf disease detection system must provide to its users. These requirements directly reflect the expected outcomes of the research.

i. Image Input – It is necessary to provide a provision for the user to submit the jute leaf image.

ii. Disease Classification – The app should be able to classify the submitted pictures to one of the five categories; Bug Attack, Spot Disease, Twisted Leaf, Yellow Leaf, Healthy Leaf.

iii. Prediction Output – The system should return the predicted disease class and the confidence level.

iv. Recommendation Support – It will recommend for some actions to be taken by the farmer against detected disease.

v. Model Comparative Analysis – The application should be able to train and analyses a few CNN models (MobileNetV2, DenseNet169, ResNet152, Xception) in terms of its performance.

vi. Evaluation Metrics – The system is required to calculate accuracy, precision, recall, F1-score and results of confusion matrix for each model.

vii. Result Visualization – The system must generate training/validation accuracy and loss graphs to visualize model performance.

Non-Functional Requirements

Non-functional requirements detail the quality of the system, but not what it will do (as in functional requirements.) These are requirements regarding the usability, performance, and reliability of the system.

i. Ease of Use – The interface of the system should be user friendly and easy to be used by unskilled people (e.g. the farmers).

ii. Performance – Predictions need to be made within a reasonable time frame (a few seconds after the image is uploaded)

iii. Scalability – The structure of the system should be such that the dataset and model can be conveniently extended in future without requiring a significant must of redesign.

iv. Accuracy – The models should achieve a minimum accuracy threshold (e.g., >85%) to ensure reliability in disease detection.

v. Portability – The system should be implementable as thin web application that can be hosted on resource-constrained systems.

vi. Maintainability – Code should be modular and well-documented to be easily extendable.

3.1.4 UI Design

A User Interface (UI) for the proposed jute leaf disease detection system was designed to be simple, user-friendly and accessible. Given that it is farmers and agricultural stakeholders who will be the most common users of the system, and many of them may have no or limited technical background, the interface should be informative, it should require few steps, and its outputs should be easily interpreted.

Core Principles of UI Design

i. Simplicity: The design is simple, stripped from all complex and convoluted features this way we can have an image upload, later the disease information and the prediction results, all without distractions. All labels, buttons, and output words are simple and can be easily related to without the necessity of having technical knowledge.
ii. Usability: Existing users, with limited digital literacy, experience no difficulty when operating the system without any training given.
iii. Responsivity: It is desktop and mobile friendly and can be used in rural areas with mobile coverage.

Key UI Components

i. Homepage: A greeting screen with brief instructions — “Upload a jute leaf image to find disease.”

ii. Submit Button: A big, transparent button that will allow us to browse and select an image from our device.
iii. Preview window showing the uploaded leaf image so that the user can check the right one that is to process.
iv. Prediction Result Box: After processing, the system displays the name of the disease type (e.g., “Spot Disease”).

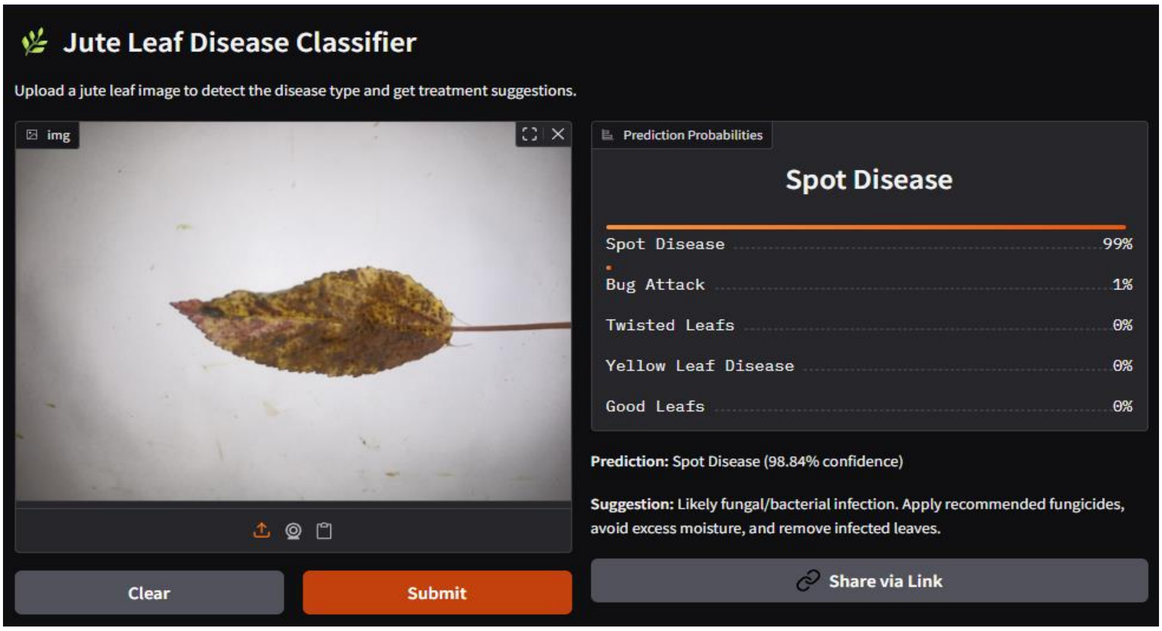

Figure 3.2: Prototype UI Test and Output.

3.2 Detailed Methodology and Design

1. Dataset Collection and Preparation

The performance of any ML-based disease-detection framework depends on the quality and variability of its dataset. For this work, we created a separate jute leaf disease dataset, consisting of over 3,500 images whose classes include- Bug Attack, Spot Disease, Twisted Leaf, Yellow Leaf, and Healthy Leaf. We responsibly crawled, cleared, and labelled the images, guaranteeing their authenticity and reliability for model training. The images were sourced from field-level photography, open agricultural repositories, and organized collections contributed by research groups. We consulted with farmers and agricultural experts in the labeling process to minimize human errors in classification. This helped in maintaining the validity of the dataset as the real symptoms of the disease observed in the jute field of Bangladesh. In order to have heterogeneity among the classes, for each type of disease a representative number of cases were taken. Some classes (e.g., Spot Disease) had plenty of samples, while others (e.g., Twisted Leaf) demanded significant fieldwork to achieve a meaningful size.

Table 3.1: Dataset Distribution.

Class Name	Number of Images	Percentage (%)
Good Leafs	1253	35.3%
Yellow Leaf	985	27.7%

Spot Disease	530	14.9%
Bug Attack	450	12.7%
Twisted Leafs	331	9.3%
Total	**3549**	**100%**

Sample Visualization of the table

To demonstrate the dataset, the representative images of each class were captured. These visual examples illustrate distinct differences among diseases:

i. Bug Attack – Pests chew, bite and make holes in fruits.

ii. Spot Disease – Round, dark lesions all over the leaves.

iii. Twisted Leaf – The leaf grows in a curl or is deformed.

iv. Yellow Leaf – Nutrient deficiency or a virus will cause the leaf to appear very pale or chlorotic.

v. Healthy Leaf – Bright green, even surface on it with no visible signs.

Figure 3.3: Bug Attach.

Figure 3.4: Spot Disease.

Figure 3.5: Twisted Leaf.

Figure 3.6: Yellow Leaf.

Figure 3.7: Healthy Leaf.

This phase of dataset production was important to ensure that the models were trained on a broad and unbiased collection of realistic and high quality input samples. With diverse diseases, variations in illumination and environment (background, leaf angle), and an even representation of the 5 classes, the dataset has a strong foundation for subsequent training, and testing.

2. Processing

Preprocessing is a crucial step to convert raw Jute leaves image into a format that can be used by machine/deep learning algorithms. Preprocessing: Given field-level images are often in disparate resolutions, colors, and illuminations, pre-processing offers standardized, noise reduced, and specifically better for training CNNs.

The preprocessing pipeline of our study had several systematic steps

i. Image Resizing: All the jute leaf images are resized to a common resolution of 224 × 224 pixels which is the input size to many popular CNN models like MobileNetV2, DenseNet169 and ResNet152 and Xception. Resizing for cross-model compatibility as well as computational load reduction.

ii. Normalization: The pixel intensity values were normalized to the range of [-1, 1] after being rescaled from their original value in the range of [0, 255], model-specific pre-processing (preprocess input for DenseNet169). This improved convergence during training.

iii. Data Augmentation: Data augmentation strategies were used in order to mitigate possible class imbalance and to make the models more robust. Augmentation artificially increases dataset diversity by creating enhanced copies of original picture. The following transformations were used:
   - Rotation: We apply a random rotation in the range of -20°, +20° to simulate changes in the leaves orientation.
   - Horizontal & Vertical Flip: Two random mirrors designed to approximate real world diversity of leaf position
   - Zoom: Zooming an image in and out to accommodate distances at which pictures are taken.
   - Brightness Adjustment: Randomly set contrast lower or higher to mimic the field under different lighting conditions. Not only did it enlarge the effective dataset scale, but also it improved the generalization ability of CNN models in the testing stage with field images.

iv. Noise Removal: Some images have background elements such as soil, other plants, or shadows. Basic filtering and cropping were applied where necessary to focus the models on the leaf region of interest (ROI).

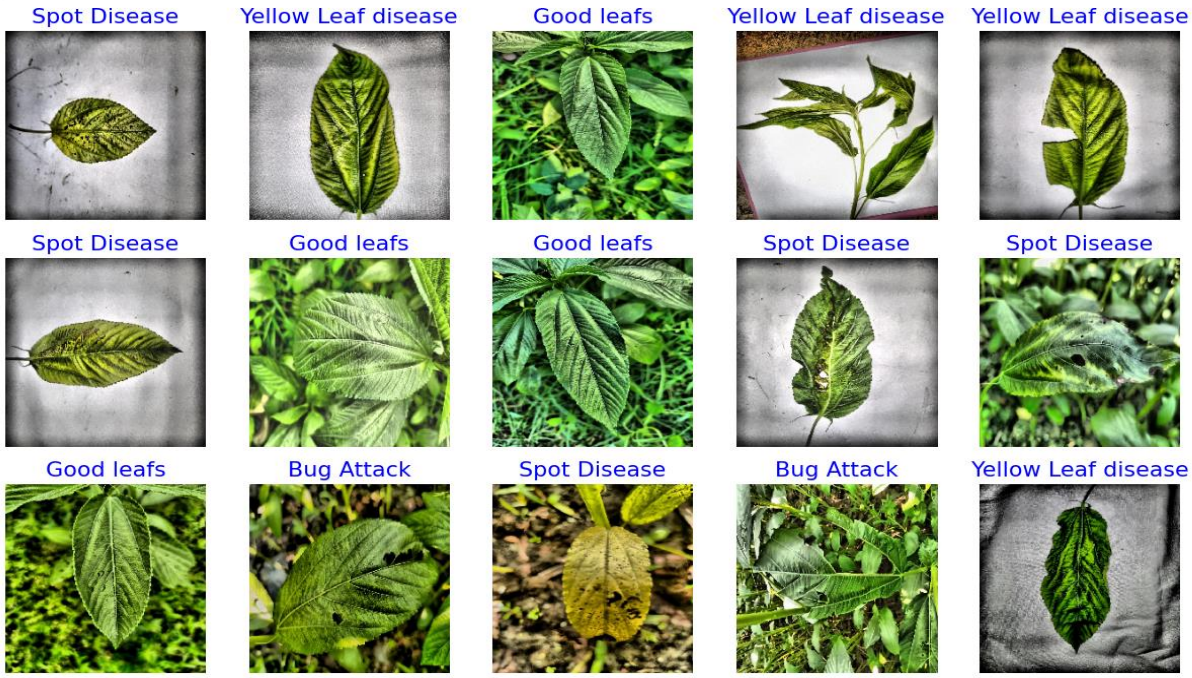

Figure 3.8: Sample Preprocessing.

These preprocessing techniques made the jute dataset clean, well-balanced, and diverse and contributed the models trained on this dataset to be robust to noise, lighting discrepancy and orientation changes. This step served as a basis for pre-processing the images for the further splitting, model selection and training.

3. Dataset Splitting

For robustly training, evaluation and testing of the deep learning models, the whole dataset consisting 3549 jute leaf images were partitioned into three groups: training (80%), validation (10%) and testing (10%).

i. Training set (80%): This set is used to train our model by updating and identifying the weight of the model parameters.

ii. Validation set (10%): For tuning of hyper parameters, avoiding over-fitting, and preventing model generalization.

iii. Testing set (10%): Completely held out during training and validation for unbiased performance measurement.

This split is to make sure that the models are trained properly but still a separate dataset is available to know how they are performing on new data.

Table 3.2: Dataset Split Distribution.

Class Name	Total Images	Training (80%)	Validation (10%)	Testing (10%)
Good Leafs	1253	1002	126	125
Yellow Leaf	985	788	98	99
Spot Disease	530	424	53	53

Bug Attack	450	360	45	45
Twisted Leafs	331	265	34	33
Total	**3549**	**2839**	**355**	**355**

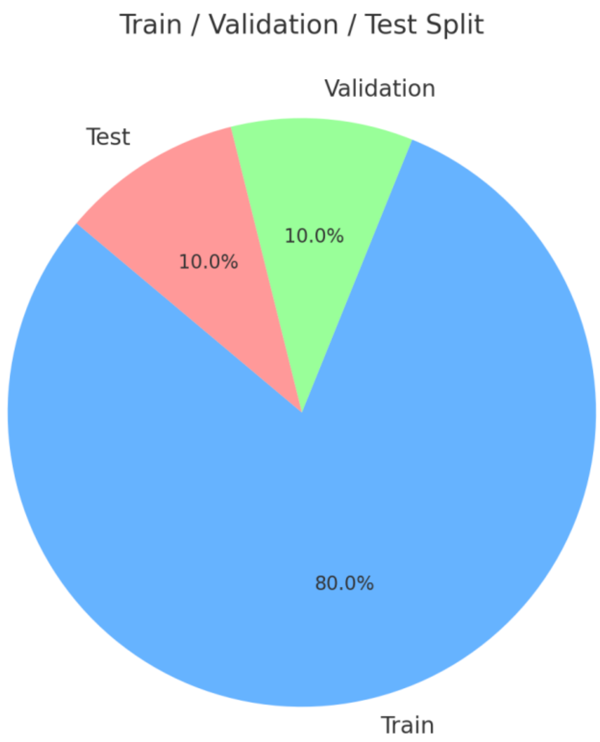

Figure 3.9: Dataset Split Visualization.

Justification for the Split: The fact that our training set is relatively large was intended to help such models as ResNet152 and DenseNet169, with a large number of parameters, to train meaningful disease features. The purpose of the validation set was to prevent overfitting and the test set was used to generate an unbiased estimate of the final model performance. We strived to keep the proportion of all the 5 classes balanced in the subsets and across subsets in order to avoid Class Imbalance, so that discussion and comparison would remain faithful to the real-world scenarios.

4. Model Selection and Architecture

Right deep learning models selection is most important in the development of strong jute leaf disease detection system. As our dataset does not contain huge images like ImageNet, and though a limited number of images is involved in our case, we used transfer learning from pre trained CNN models. Transfer learning will enable us to make use of feature representations trained on millions of natural images and apply to the smaller, domain-specific data.
Based on the accuracy, computational cost, and deploy-ability, we finally selected four CNNs to experiment, namely, Mobile-NetV2, DenseNet169, ResNet152, and Xception.

i. MobileNetV2: It is the lightweight Convolutional Neural Network (CNN) architecture for the mobile and resource constraint systems. It is inspired by the depth wise separable convolution which significantly reduces the number of parameters and computations but doesn't massively impact on accuracy. Its inverted residual structure and linear bottlenecks make it well-suited for practical real-time application in the field, for example with farmers who can use sensors and mobile smartphones to take pictures and scan leaves.

ii. DenseNet169: DenseNet169 introduces dense connectivity, which refers to the way each layer is associated with all other layers in a feed-forward fashion. This special architecture enhances feature reuse, prevents vanishing/thresholding gradients and can enable better training on small data. In the context of plant disease detection, DenseNet is good at learning fine-grained visual features, including small leaf spots, subtle texture changes associated with the onset of diseases.

iii. ResNet152: ResNet152 is one of the CNN architectures with more layers among the three. Vanishing gradients are addressed with residual and skip-connections. ResNet has been widely used in the field of agricultural and medical image recognition, which can learn the complex, hierarchical feature representations. Though computationally less efficient than MobileNet and DenseNet, ResNet152 also serves as a strong reference for accurate models in our experiments.

iv. Xception: Xception is short for "Extreme Inception" and is a deep CNN structure in which the Inception modules are replaced by depthwise separable convolutions for all layers in the network. It is very effective in extracting spatial correlations (such as patterns inside images) and channel-wise correlations (such as color/texture changes). For the classification of jute leaf, Xception is particularly effective in identifying those disease patterns where discoloration of leaf such as yellowing and chlorotic symptoms are presented.

Justification for Model Selection

i. Under the motivation of comparing, we tried these four architectures.

ii. Efficient and heavy: Accurate trade-off (MobileNetV2 vs. ResNet152).

iii. Feature reuse and fine-grained detection (DenseNet169).

iv. Advanced separable convolution strategies (Xception).

This mixture guaranteed that our evaluation went both ways, therefore including models designed for being deployed in real life as well as models that were aimed for being tested academically. The knowledge generated through this cross-comparison formed the basis for the selection of the most appropriate model for farmers in Bangladesh.

5. Training Configuration

The training regime has significant influence in the efficiency, accuracy and stability of deep learning models. In this study, the training protocol was carefully conducted so as to allow the fair comparisons between the four selected CNN models including

MobileNetV2, DenseNet169, ResNet152, and Xception.
We implemented everything on Kaggle for its free support and to have reproduce-able model training on Kaggle's cloud GPU environment with enough computing power for works of academic researches. The aforementioned setup was used for all experiments:

i. Training Strategy: The training took place as two stages:

    1. Phase 1 – Training: In the first step, the pre trained base layers of the CNNs were frozen, and only the new and custom classification head was trained. This enabled the model to fine-tune the high-level features through the new data set at a fast pace.

    2. Phase 2 – Fine Tuning: The base CNN's top layers were un-frozen, and the whole network was trained with a reduced learn rate. It assisted in fine-tuning the feature extraction, to detect the jute leaf disease related patterns.

ii. Evaluation During Training:

    1. Accuracy and loss on the validations were recorded every epoch.

    2. Early stopping prevented overfitting.

    3. There was also model check pointing for best weights highest validation accuracy). The ultimate models were saved in. h5 format testing and utilizing the program.

6. Evaluation Metrics

The classification performance of deep learning-based models needs to be assessed using multiple metrics for a comprehensive evaluation. Solely accuracy might be misleading, especially when there is class imbalance situation in which samples of certain disease categories are scarce. Accordingly, this study employed Accuracy, Precision, Recall, F1-score, and Confusion Matrix to assess the performance of the proposed model.

i. Accuracy: Accuracy measures the overall correctness of predictions made by the model. It is the ratio of correctly classified samples to the total number of samples.
Accuracy gives a performance average measure, however, it might not be enough in a scenario of imbalances classes.

ii. Precision: Precision evaluates how many of the samples predicted as positive by the model are actually correct.
In the context of jute leaf disease detection, precision specifies that the disease (Spot Disease in this case), which is predicted by the model is not a false positive prediction.

iii. Recall (Sensitivity): Recall measures how many of the actual positive samples were correctly identified by the model.
For use in agriculture, high recall is important since failing to detect a diseased leaf (false negative) might enable the disease to propagate which will negatively impact the crop yield.

iv. F1-score: The F1-score is the harmonic mean of precision and recall, providing a balanced evaluation of both metrics.
Especially useful for unbalanced datasets, as it penalizes classifiers that are only good at one of either precision or recall.

v. Confusion Matrix: Confusion matrix is able to help in the interpretation of classification results by displaying the true positives, false positives, false negatives, and true negatives for each class. It is an important tool to figure out, where the model mistaking "Yellow Leaf" with "Healthy Leaf".

In the summary of metric through the blending of these evaluation metrics, this study ensured a holistic and equitable evaluation of each CNN model. Models with high performance of accuracy, precision, recall, F1-scores and the balanced confusion matrices were considered to be candidates for field application.

3.3 Project Plan

Given the range and level of detail of this work, the project was rigorously broken down into structured phases to ensure successful delivery on time. Every phase was developed with specific goals, deliverables and interdependencies in order to allow for orderly progression through the project. This formalized project plan also supported resource assignment, status tracking, and proactive issue resolution throughout the model experimentation and validation phases.
The whole process is illustrated in following source codes:

1. Phase – 1 Research and Requirements Analysis

i. Literature review 1) Abstract Anomaly detection is a crucial aspect of plant disease detection using computer vision and deep learning in agriculture.

ii. Discovered the primary challenges in detection of jute leaf diseases such as scarcity of large scale datasets, presence of few number of jute crops and the requirement of lightweight models for real-time application.

iii. Specified which neural network architectures were included in the study: MobileNetV2, DenseNet169, ResNet152, and Xception.

2. Phase – 2 Dataset Curation and Preparation

i. Gathered 3,549 images of jute leaves (Good Leaf, Yellow Leaf Disease, Spot Disease, Bug Attack and Twisted Leaf) captured from different angles being five classification categories.

ii. Preprocessing operations we applied some preprocessing to the dataset, including resizing and normalization, and augmented the available dataset by rotation, flipping, zoom, brightness adjustment to increase the quality of the dataset as well as balance the class.

iii. Split the dataset into training, validation, and testing sets (80%, 10%, and 10%) for fair model assessment.

3. Phase – 3 Training and Fine-Tuning of the Model

i. Developed training pipelines for MobileNetV2, DenseNet169, ResNet152, and Xception with Python and TensorFlow, Keras.

ii. Performed hyper parameter tuning among learning rate, batch size, and optimizer to get the best performance.

iii. Leveraged Kaggle's resources to expedite training and better manage the huge dataset.

4. Phase – 4 Evaluation and Testing

i. All the model evaluations were done based on performance measures like Accuracy, Precision, Recall, F1-Score, and Confusion Matrix.

ii. Comparatively analyzed 4 CNN architectures to determine the appropriate model to configure for agricultural scenarios.

iii. Investigated the failure cases to gain insights into model vulnerabilities and dataset limitations.

5. Phase – 5 Deployment Planning

i. Described straightforward, farmer-compliant web application design by which users could upload or take pictures of the jute leaf for estimation of a disease.

ii. The best performing trained model (best_model. h5) for predictions.

iii. The proposed system is intended to supply not only a detected disease category, but also to have some straightforward guideline for farmers what to do in order to eliminate the diagnosed problems.

Tools and Technologies Used

i. Programming Language: Python 3.10

ii. Deep Learning Frameworks: TensorFlow.

iii. We train our model on Kaggle.

iv. Dependencies: NumPy, Pandas, Matplotlib, scikit-learn.

3.4 Task Allocation

Table 3.3: Task Allocation table.

Tasks	Weeks																		
	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48
Data collection phase																			
Preprocess all the data																			
Model training																			
Create a demo Application.																			

3.5 Summary

The overall methodology and designing of jute leaf disease detection system is explained in detail in this chapter. We first investigated system needs and sketched the methodology structure, followed by a pipe that drove the work. The dataset was generated along with processing and splitting into training, validation and testing subsets, for a balanced learning. Four different deep learning models: (1) MobileNetV2, (2) DenseNet169, (3) ResNet152, and (4) Xception were chosen, trained and tested based on classic performance evaluation indicators including accuracy, precision, recall and F1-score. The project proposal was divided into different stages including data collection and preprocessing, training, evaluation, and deployment. A comprehensive schedule of the task assignments was also offered to illustrate how the various work was apportioned during the project.

In general, this chapter has laid the groundwork of the system by introducing the methodology, tools, technologies, and its workflow, as a systematic and reliable method for meeting the project goals.

Chapter 4

Implementation and Results

In this chapter we provide the technical details of our proposed system and its results. It starts with experimental environment configuration and testing and evaluating the models like performing comparisons on performances between different CNN architectures. In a latter step, the chapter presents in detail the results and finally concludes with aggregated outcomes.

4.1 Environment Setup

Setting up the environment was important to make sure that the both training and evaluating the jute leaf disease detection system could be conducted properly and rapidly. Because CNNs, like MobileNetV2, DenseNet169, ResNet152, and Xception are computationally expensive, most of the experiments were conducted on Kaggle Notebooks with GPU. Besides, local code organizing and debugging was performed by VS Code.

The system was implemented in Python 3.10 with tens of widely adopted deep learning frameworks and their supporting libraries. The most used libraries to develop, train and evaluate CNN architectures were TensorFlow and Keras. Other libraries used for this study include NumPy, Pandas and Matplotlib for dataset preprocessing, numerical operations, and data visualization, respectively. Preprocessing libraries were used for image resizing, normalization, and data augmentation to enhance the quality of images and uniformity.

For dataset presentation, over 3,500 the number of jute leaf images have been stored in a structured folder based on different class called, Good Leaf, Yellow Leaf Disease, Spot Disease, Bug Attack and Twisted Leaf. These images were preprocessed to a common image size of 224×224 to match CNN input layers. We then split the dataset into train/validation/test (80%–10%–10%) maintaining balance during the evaluation.

For the training, we used Kaggle's runtime to decrease computation time considerably. Every model was trained with hyperparameters: epochs = 10 to 25 according to model. We also used the callback features such as EarlyStopping and ModelCheckpoint to avoid the phenomenon overfitting and to save the best model automatically.

It was a very efficient and scalable and most importantly a cost-effective setup for working on the project because of the Kaggle and local development in VS Code. It even made the collaboration between preprocessing, model training, and performance evaluation easier.

4.2 Performance Evaluation and Comparative Analysis

After the completion of the training of the model, rigorous checks were done, to see how well the models were performing. We conducted this benchmark to evaluate both the quantitative results such as accuracy, precision, recall, and F1-score and the qualitative aspects of the generalized NNs such as its robustness, generalization capability, and the relative strength compared with prior arts. To ensure no bias in the comparison, the dataset was divided into 80% training, 10% validation and 10% testing. Random sampling was used, resulting in no leakage between the train/test splits, and the test was performed properly. In this assessment, we aimed to evaluate the effectiveness of the retrained CNN models (MobileNetV2, DenseNet169, ResNet152, and Xception) in discriminating jute leaf diseases under real-time applications. Health Leaf, Yellow Leaf Disease, Spot Disease, Bug Attack and Twisted Leaf classification were tested on the models.

A comparative analysis of the four models was performed, and the results are summarized in **Table 4.1** below.

Table 4.1: Performance Comparison of CNN Models.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
MobileNetV2	86%	0.86	0.86	0.86
DenseNet169	88%	0.88	0.88	0.88
ResNet152	74%	0.74	0.74	0.72
Xception	85%	0.85	0.85	0.84

From the above comparison, we see the MobileNetV2 scores 86%, and above that is DenseNet169 which scores 88%, then Xception at 85% as it is otherwise depicted by the table above. MobileNetV2, DenseNet169 and Xception performed better than ResNet152, thus had lightweight and efficient classification on this task.

Furthermore, the performances of the proposed model were compared with the existing works on detecting plant diseases using CNNs. Unlike the literature related to the major leafy sources of production such as rice and wheat, in contrast, this study was specifically focused on jute leaves which are comparatively less studied.

i. Related Work: Several published works have reported the following performance using normal CNN or VGG architectures for other crops accuracy (ranged between 80%–85%).

ii. Proposed System: With the help of DenceNet169, MobileNetV2 and Xception, the project could attain higher accuracy (85–88%) proving that lightweight and efficient CNNs can outperform heavy CNN models when hyperparameters are tuned and having well-processed data.

In summary, the comparison results illustrate the effectiveness of our approach which requires less data and lower play quality as well as pre-processing and augmentation in order to achieve competitive results. The results demonstrate that part of light-weight CNN models can actually beat the heavy ones if their inputs are sufficiently large, in which larger inputs are used than they are pre-trained on even for larger databases. This demonstrates the importance of dataset quality and domain adaptation and specialization rather than absolute size. To conclude, the proposed method is feasible and effective, particularly for agricultural applications (e.g., jute leaf disease detection), where real-world application requires the accuracy of detection as well as the efficiency.

4.3 Results and Discussion

The behavior of these four CNNs namely MobileNetV2, DenseNet169, ResNet152 and Xception was discussed in details with the help of training and validation curves, confusion matrix and metrics. This section gives an Exploratory Analysis of the results, both quantitative and qualitative.

Training and Validation Accuracy

The training and validation accuracy curves offered a crucial insight into the effectiveness of the models to learn from the training data and how well they could generalize to new data.

From Figure 4.1, it can be observed that MobileNetV2 accuracy was slowly improving over epochs, and validation accuracy followed training accuracy closely. After epochs accuracy has bend of ~ 86% what indicates a satisfactory learning capacity, but not such as for deeper architecture. Figure 4.3 we already remember it from the accuracy of DenseNet169, but as usual our growth is in the first epochs much more aggressive and arrives more or less at 88% in the validation. It really confirms the denseness in the connection which helps to propagate gradients and to activate high level features.

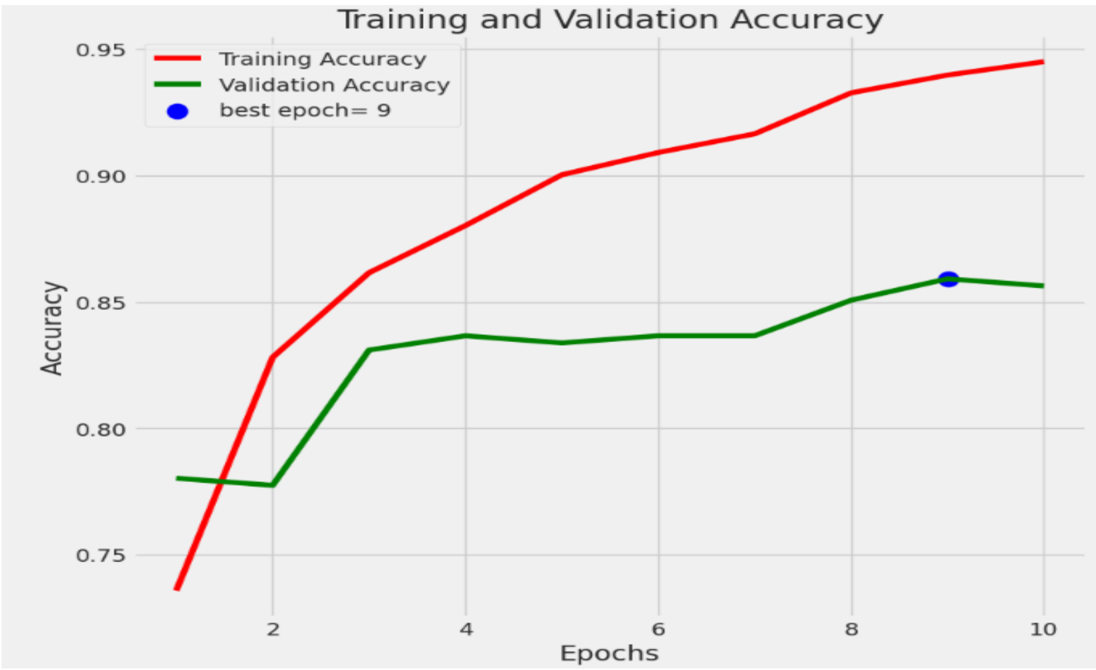

Figure 4.1: Training and Validation Accuracy – MobileNetV2.

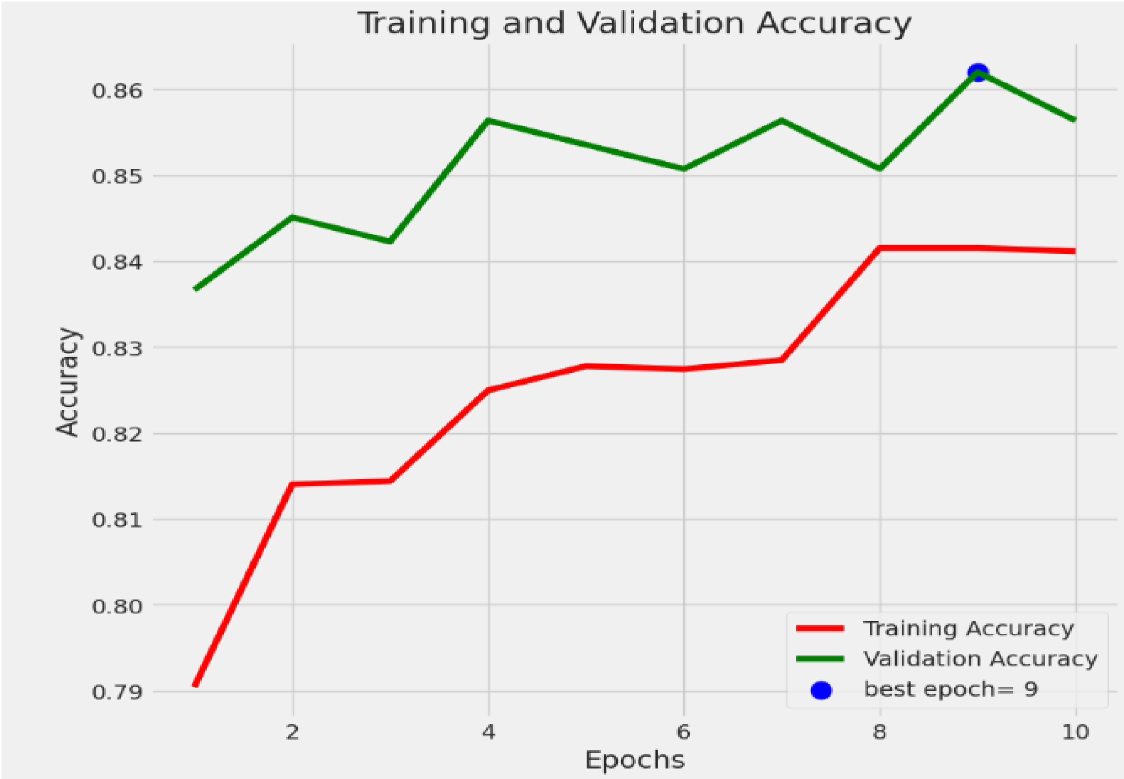

Figure 4.3: Training and Validation Accuracy – DenseNet169.

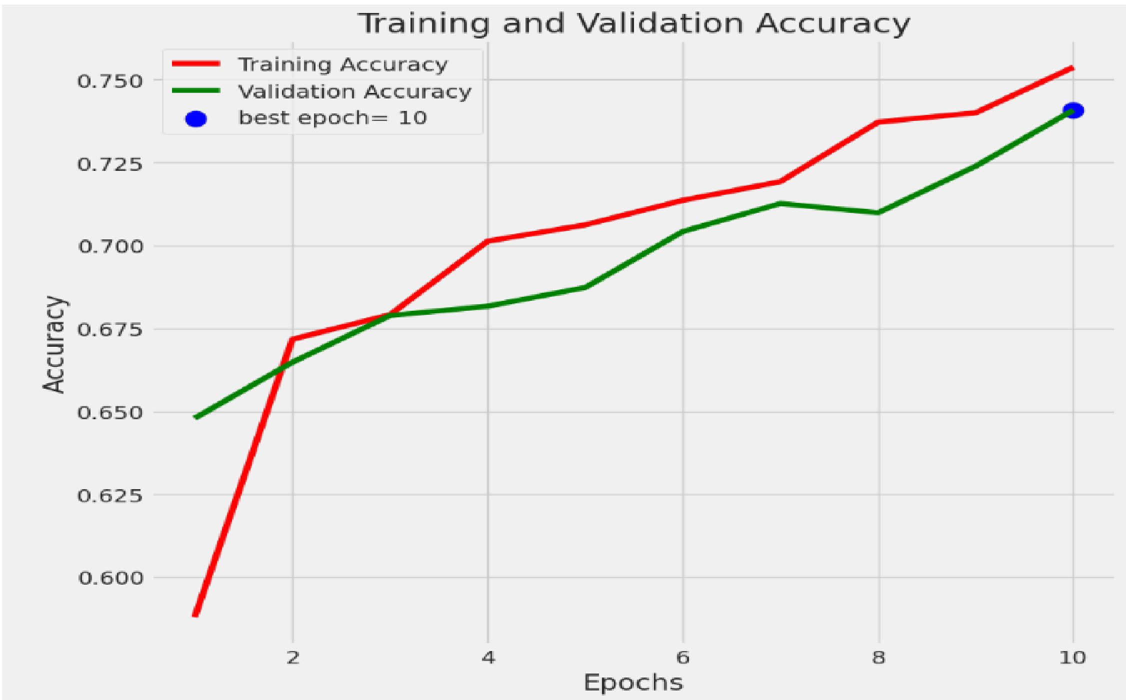

Figure 4.5: Training and Validation Accuracy – ResNet152.

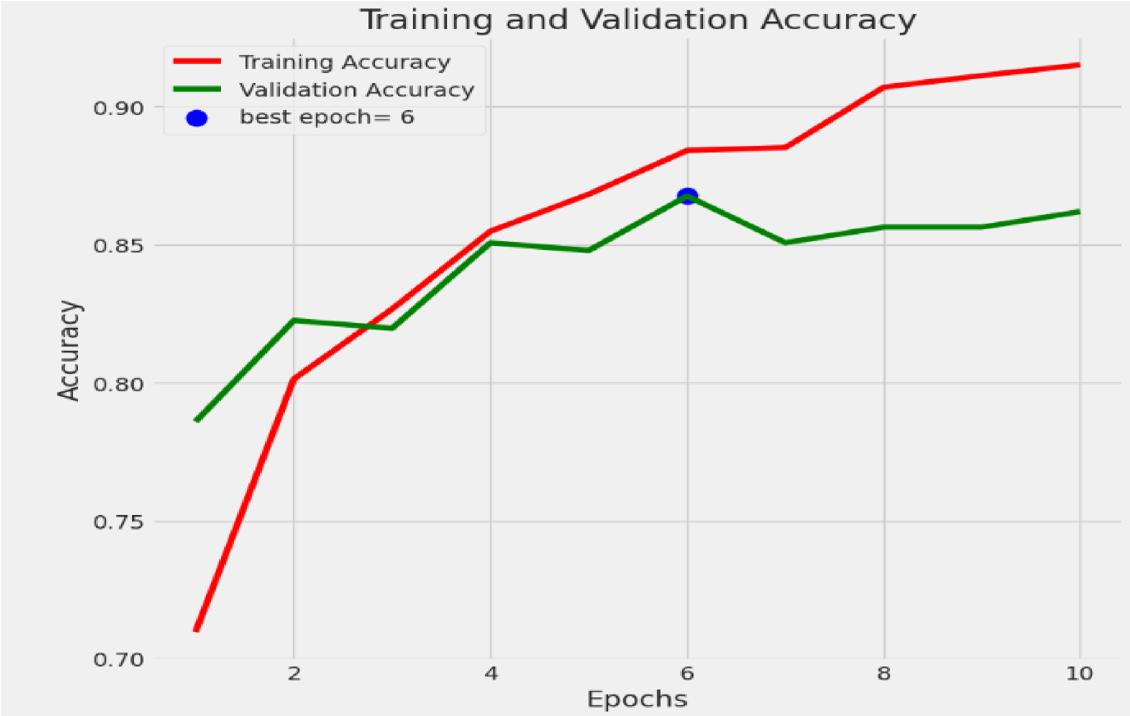

Figure 4.7: Training and Validation Accuracy – Xception.

Training and Validation Loss

From the bellow pictures we can see the training and validation loss graphs of all 4 models that we have used.

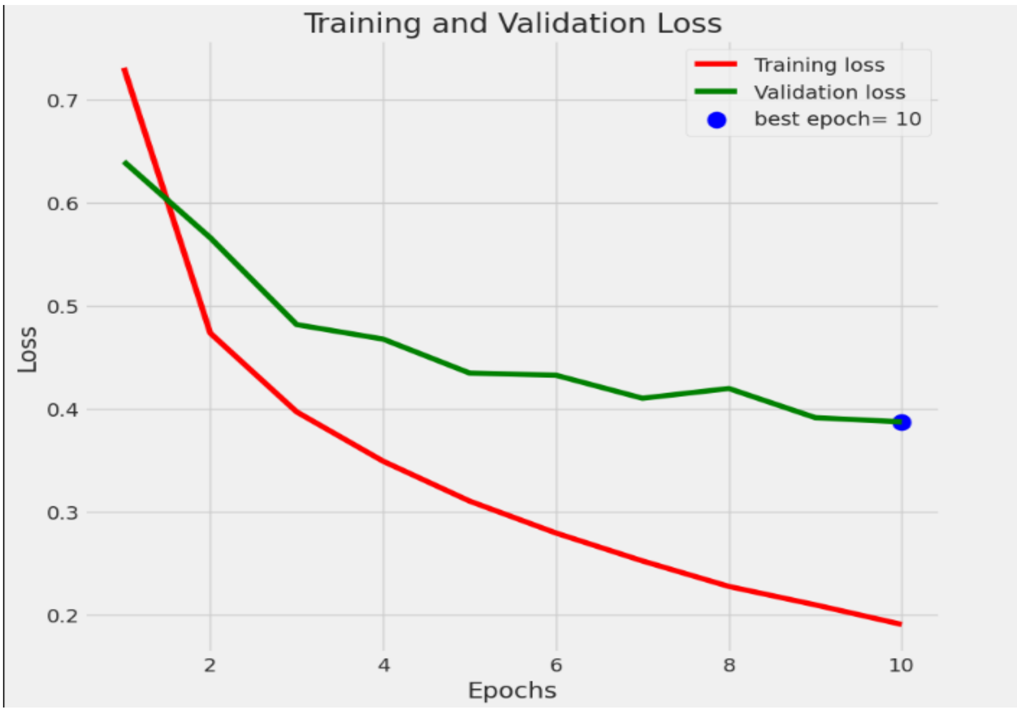

Figure 4.2: Training and Validation Loss – MobileNetV2.

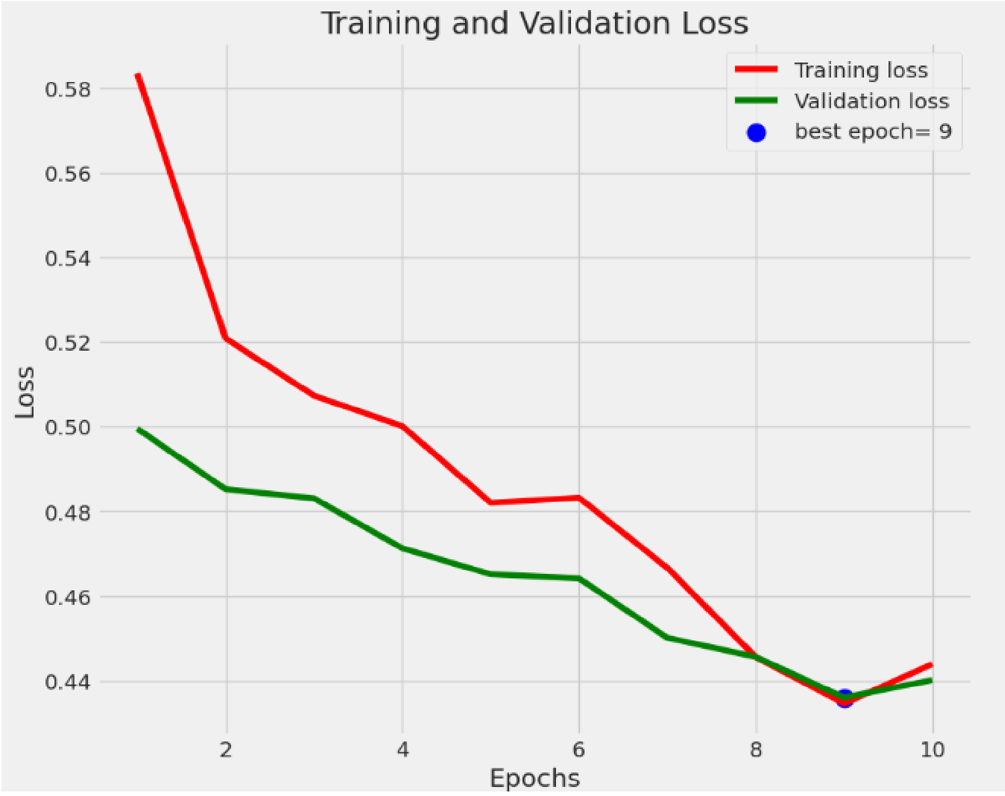

Figure 4.4: Training and Validation Loss – DenseNet169.

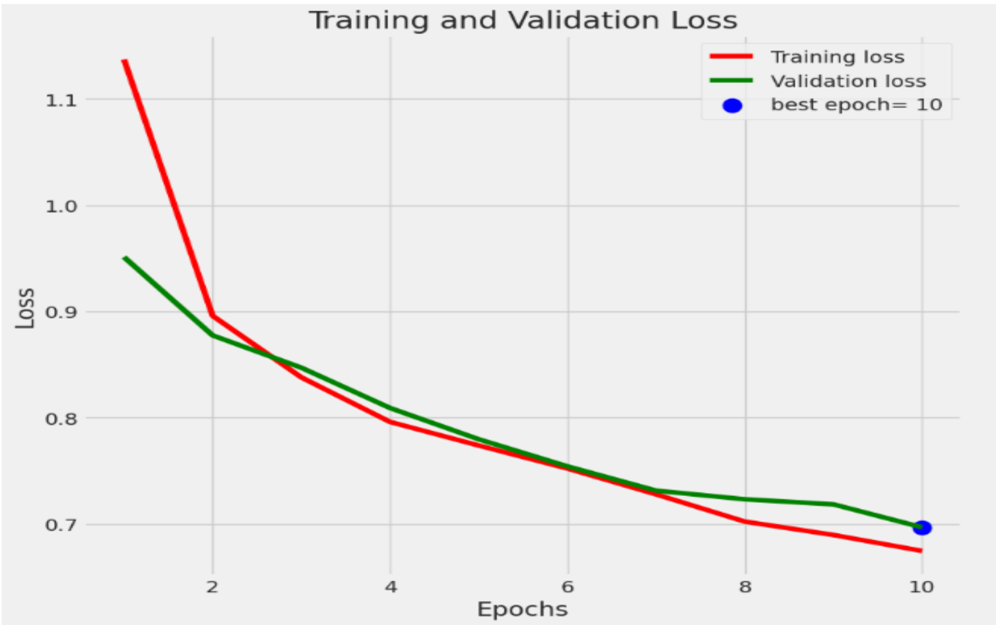

Figure 4.6: Training and Validation Loss – ResNet152.

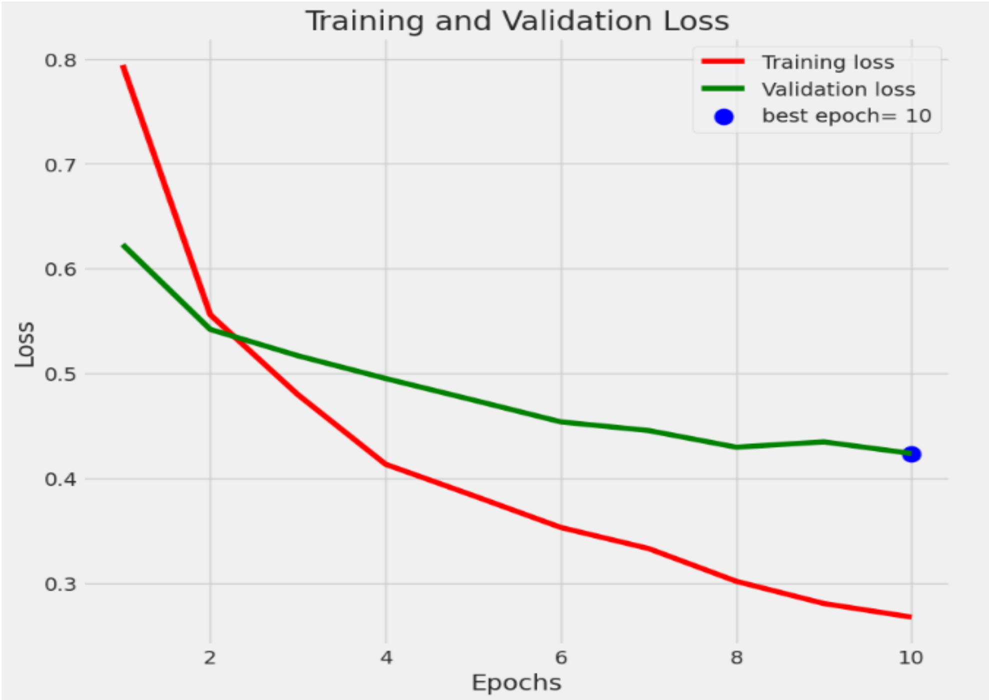

Figure 4.8: Training and Validation Loss – Xception.

Confusion Matrices

The confusion matrices enable a class wise analysis of the performances. Figure 4.9 for MobileNetV2's confusion matrix, exhibits the overall robust classification, albeit with a few misclassifications between Twisted Leaf and Yellow Leaf class. This implies that the subtle visual differences in the leaf discoloration adversely affected the separation.
DenseNet169 Shown in Figure 4.10, had a perfect class separation and over 88% correct across categories. Figure 4.11: ResNet152 has lowest predictions for most categories, has a lower recall for Twisted Leaf, which are mainly confused with Yellow Leaf. Lastly, Xception's confusion matrix in Figure 4.12 demonstrates consistent performance for all the classes, with 85% or higher for every category, and very strong findings in Spot Disease and Twisted Leaf detection.
This analysis clarifies that the confusion matrix was the cleanest for DenseNet169, which was closely followed by Xception, with more cross-class confusion for MobileNetV2 and ResNet152.

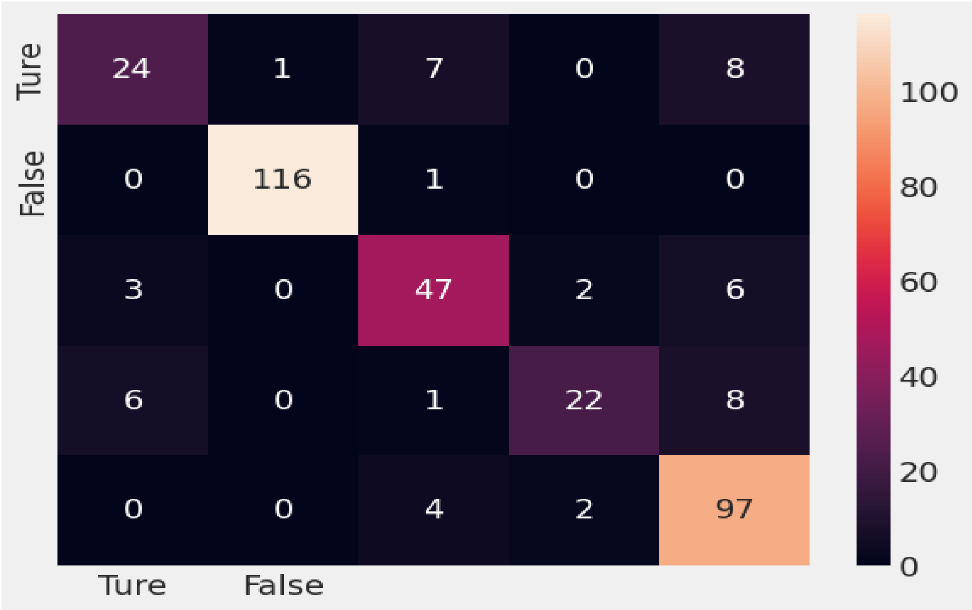

Figure 4.9: Confusion Matrix – MobileNetV2.

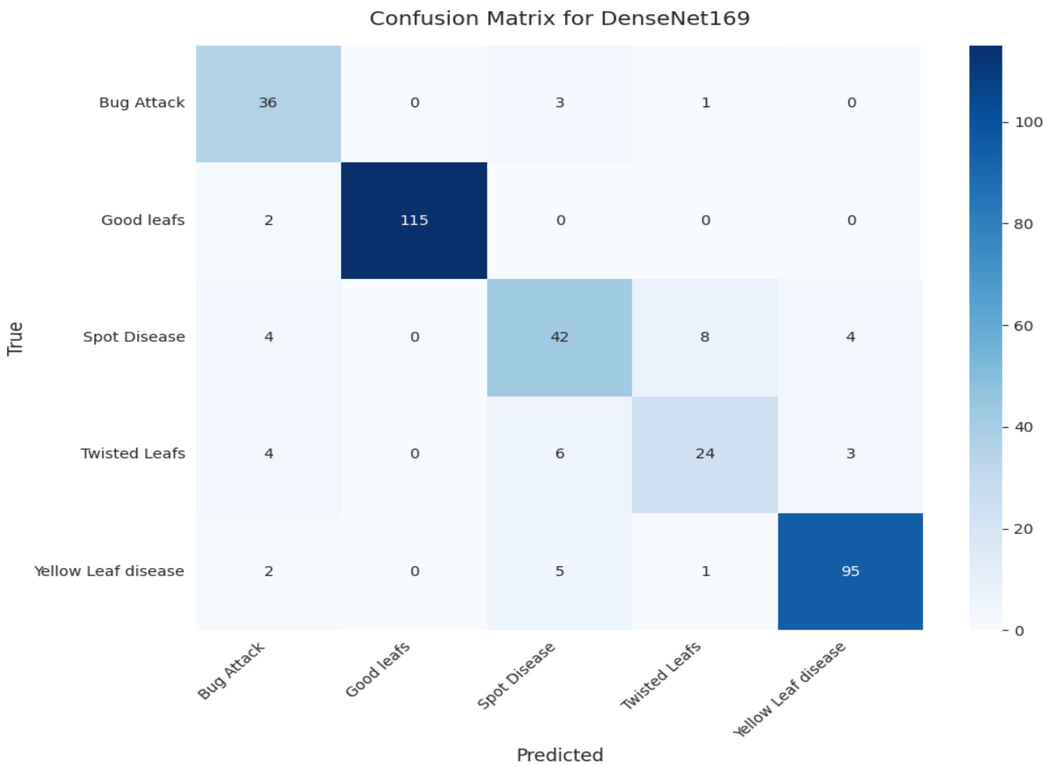

Figure 4.10: Confusion Matrix – DenseNet169.

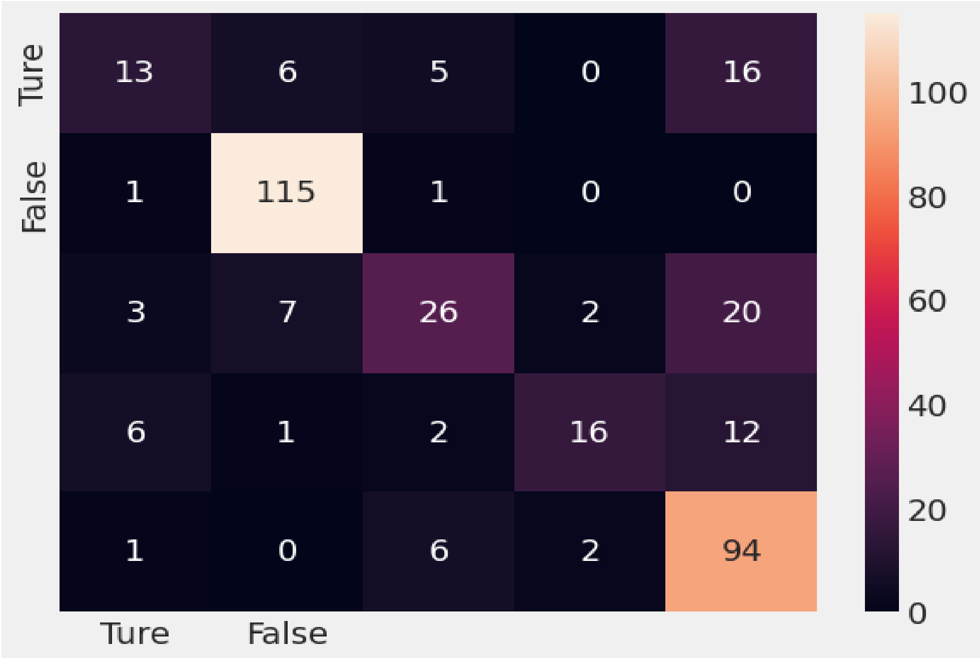

Figure 4.11: Confusion Matrix – ResNet152.

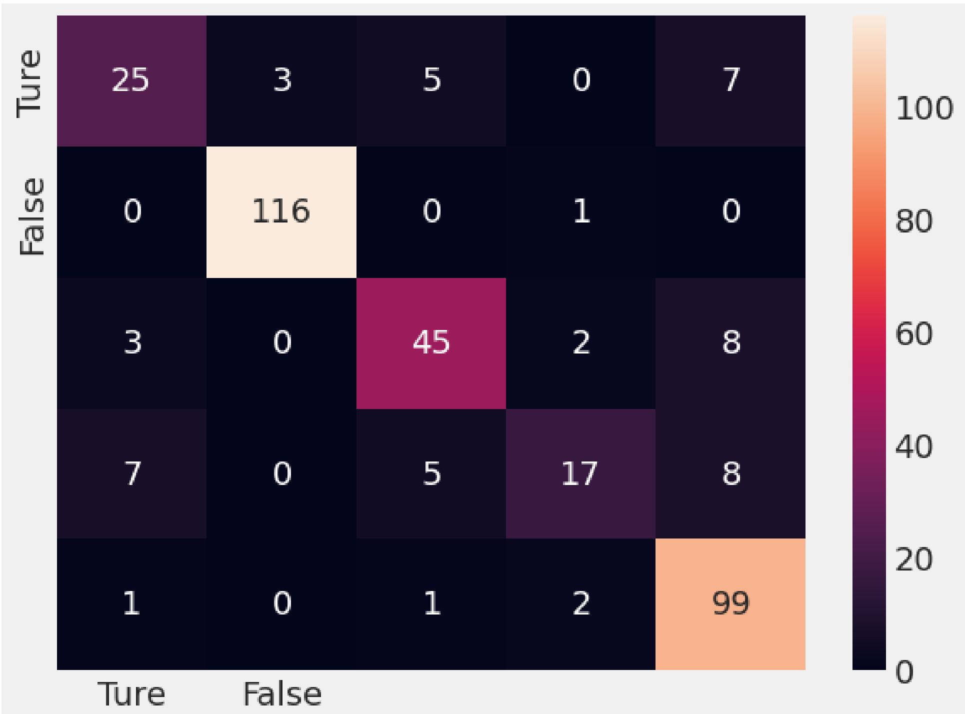

Figure 4.12: Confusion Matrix – Xception.

Quantitative Performance Comparison

To summarize the results, Table 4.1 summarizes the four models in terms of Accuracy, Precision, Recall, and F1-Score. The results indicate that, without considering losses during the train process, DenseNet169 gave the best overall performance (88% accuracy and balanced scores for all other metrics). Xception ranked the next with 85%, ResNet152 got 74%, and MobileNetV2 obtained 86%.

This finding is also supported by the bar chart in Figure 4.13: the good performance of Densenet169 is very visible, and so is the similar performance of MobileNetV2 and then comes Xception. ResNet152, on the other hand, is not competitive, despite (or because of) its depth. This indicates that increasing model complexity does not actually lead to better performance, and architectural advancements such as dense connectivity or depth wise separable convolutions contribute more.

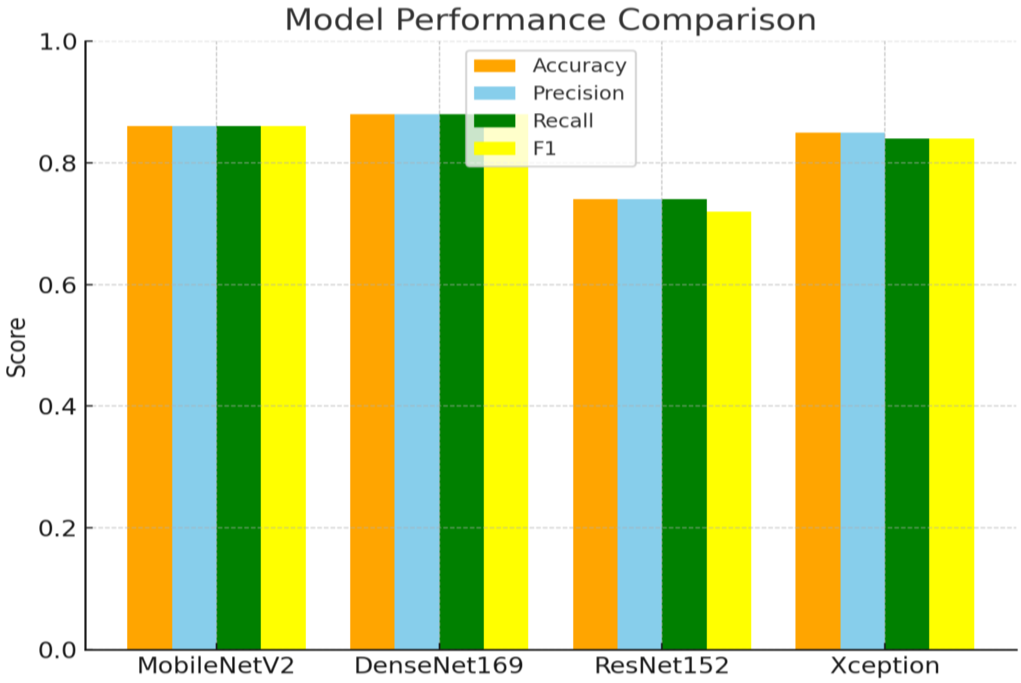

Figure 4.13: Performance Comparison of Models.

Discussion of Results

DenseNet169 was the top performing architecture in all four metrics, establishing it as the most stable solution in this study. MobileNetV2 is the most practical for real world, lightweight deployment, and Xception is a balanced and efficient alternative. For this application, ResNet152, despite being reliable, did not perform well. Jointly, these results highlight that model design and feature connections matter more than network depth alone in achieving effective, efficient, and scalable solutions for detecting jute leaf disease.

4.4 Summary

In this chapter, jute leaf disease detection system is implemented and tested. The experimental configuration, model training, evaluation metric were shown, and four CNN architectures were compared in great detail. In this group of features, DenseNet169 performed the best overall, as well as Xception performed with significantly advantage among the rest, and MobileNetV2 with functionality lightweight. These findings validated the best model for application and also demonstrated the importance of the dataset quality and architecture design in plant disease detection.

Chapter 5

Engineering Standards and Design Challenges

This chapter presents the methods employed in practice, as well as the social, ethical and sustainability implications of this work. It also considers the project management aspects in relation to cost, and finally the complexity of the engineering problem as such. The point of the project is to demonstrate that something fulfills the requirements of engineering, but that does address real-world challenges in a responsible way.

5.1 Compliance with the Standards

The design of the system made use of stringent software, hardware and communication standards to provide reliability and maintainability.

5.1.1 Software Standards

The system was developed using common software engineering practices for robustness, maintainability, and portability. The areas we had to focus were Python coding style, TensorFlow design guidelines and support, and model training.

5.1.2 Hardware Standards

The hardware configuration was based on cloud but GPU was also checked for suitable hardware. Since the models were trained on Kaggle, we ensured that our implementation will be in line with normal usage for deep learning work. Using standardized computing environments reduced the number of dependency conflicts and provided portability and reproducibility of the research across different platforms.

5.1.3 Communication Standards

All collaboration and all documentation followed established norms of academic/professional interaction. This included utilizing code under version control via Kaggle, as well as producing results in the IEEE format. Reports were transparent and reproducible based on the declaration of datasets, processing and parameters used in training models.

5.2 Impact on Society, Environment and Sustainability

The work considered also its socio-environment-sustainability impact, in a broader approach, to guarantee responsible development.

5.2.1 Impact on Life

System can be directly used to support agriculture and food security by assisting farmers in early detection of jute leaves diseases. “It means a reduction in lost crops, increases yield and quality of output, all to the benefit of deliberate lift, particularly for those rural villages where farming is the main earning income.”

5.2.2 Impact on Society & Environment

As trained ML can help identify these diseases, the project lessens reliance from unhealthy chemical pesticides, and contributes to a more environmentally friendly agricultural arena. At a social level, it enhances the resilience of rural farming communities and hence financial stability and food certainty.

5.2.3 Ethical Aspects

The work follows data ethics using only responsibly-collected and anonymized data. There was no personal or private information hosted. Moreover; harnessing datasets ensures fairness and eliminates the misuse of farmer contributed data without acknowledgement.

5.2.4 Sustainability Plan

The system was meant to be lightweight and cost effective so that eventually farmers could access them through their mobile phones and not solar powered top end equipment. Because we are using open-source toolkits, it is already a highly-adoptable and expandable platform that can lead to sustainable agricultural practices.

5.3 Project Management and Financial Analysis

We are not the average company that just hopes to break even on their Music and Art project, we are very mindful how the revenues are done while building the project to keep a realistic Tea Room budget. Most of the hardware and software resources were handled covering free or open-source software, however, unavoidable expenses such as costs associated to cloud, travel, internet, and data collection activities added to the total costs. All expenses made were from combined budget of the project team, which stood at around 15,000–20,000 BDT among both members.

Table 5.1: Estimated Cost and Financial Analysis.

Components	Estimated Cost (BDT)
Internet and Cloud Subscription	5,000 – 6,000
Software Licenses (Open-source tools used)	0
Traveling (Field visits, meetings, data collection trips)	5,000

Contingency (10% Buffer for unexpected costs)	2,000
Miscellaneous (Printing, Backup storage, Stationery, etc.)	3,000 – 4,000
Total Estimated Cost	**≈ 15,000 – 17,000 BDT**

5.4 Complex Engineering Problem

The purpose of the Jute Leaf Disease Detection project were as follows: The project covered several aspects of complex engineering issues. It involved the use of domain expertise, dealing with multiple objectives, learning from heterogeneous data sources and making the solution scalable and usable for the end users (farmers, experts, etc.). The project was also identified on a standard problem-solving framework.

5.4.1 Complex Problem Solving

The project resulted in an adequate demonstration of the satisfaction of a number of the different aspects relating to complex problems in engineering according to engineering standards. These are: The requirement for thorough understanding of machine learning, balancing the accuracy of the model against computational expenses and resolvers", and handling multi-step workflows like preprocessing steps, augmentation, and evaluation.

Table 5.2: Mapping with Complex Engineering Problem.

EP1 Dept of Knowl edge	EP2 Range Of Conflic ting Requir ements	EP3 Depth of Analysis	EP4 Familiarit y of Issues	EP5 Extent of Applica ble Codes	EP6 Extent Of Stake-holder Involveme nt	EP7 Interdependence
✓		✓	✓		✓	✓

Justification of EP1 attainment: Leveraged knowledge of advanced machine learning (CNNs, transfer learning, augmentation) and area specific agricultural understanding (jute plant pathology).

Justification of EP3 attainment: We did thorough research with 4 different deep CNN architectures (MobileNetV2, DenseNet169, ResNet152 and Xception), fine tuning hyper parameters, data splits and augmentation techniques to get the last drop of performance.

Justification of EP4 attainment: Effects on domain-specific issues e.g., class imbalance, similar disease visible patterns, and noise in real-world images.

Justification of EP6 attainment: Farmer usability was taken into consideration in the system design, favoring explain ability and accuracy over sheer computational efficiency.

Justification of EP7 attainment: Building a single data pipeline that included formation, preprocessing, training monitoring, evaluation and visualization in a connected way.

Mapping with Knowledge Profile

This mapping shows how EP1 (Depth of Knowledge) connects to key engineering knowledge profiles.

Table 5.3: Mapping with knowledge Profile.

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

Justification for Each Knowledge Profile Component:

K2: Applied math concepts such as probability, matrices and optimization in training and testing CNNs

K3: Implementation of engineering principles: pre-processing of images, training of CNNs, and evaluation metrics (precision, recall, F1)

K5: Developed a complete pipeline from raw image acquisition to deployed model.

K6: Realist to use the practical methods such as dataset handling, preprocessing, GPU training.

K8: Surveyed and tailored previous plant disease detection works into jute scenario.

5.4.2 Engineering Activities

The project also plugged into engineering activities of a far more complex nature with respect to resource use, innovation and wider social/environmental impact.

Mapping with Complex Engineering Activities

Table 5.4: Mapping with Complex Engineering Activities.

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

Justification:

EA1: Information on computational (training on GPU), data (image collection from the field), and software (libraries).

EA2: Inter-disciplinary collaboration (computer science + agriculture).

EA3: Innovated by using deep CNN architectures which are not widely used for jute comparison.

EA4: Contributes to sustainable agriculture by facilitating early disease detection and reducing use of chemicals.

5.5 Summary

The systems specifications and design complications were detailed in this chapter. It began with an architecture that couched complicity in the correct software, hardware, and communications standards, so everything in the development process would be in sync with the models for development to date. The dialogue expanded to the larger societal affect, the ethical and sustainable aspects of engineering, and the responsibly in an engineering decision beyond the technical findings.

From the financial summary it could be seen that the project was modestly managed financially, with costs to main items such as cloud services provision, data collection and travel. Finally, we were able to have the project "sit" over complex engineering problem regarding to how the project was functionally related to domain and engineering design and evidence to research literature. These reflections define the fact that the project, besides to obtain a working system, was success even when there was satisfaction in the professional, ethical and social level.

Chapter 6

Conclusion

The chapter wraps up the project by summing up the main contribution, recognizing of course the limitations that went into the project developing it as such; the potential for future research is also discussed. It provides a comprehensive summary of project results and a view on the way ahead.

6.1 Summary

The approach of the study has been taken as an attempt to build a jute leaf disease detection system using DL reaching a significant means to solve a nature drive problem with massive benefits to the farmers, and specifically the owners of the jute textile industry. In the course of process, the model applied convolutional neural network architectures (MobileNetV2, DenseNet169, ResNet152, Xception) to categorize leaves as, Good leafs, Yellow Leaf Disease, Spot Disease, Bug Attack, Twisted Leafs. After being trained and evaluated in a fair experiment under their respective experiment environment, their models were fine-tuned and evaluated in terms of accuracy, precision, recall and F1-score. The experiments showed that the architectures provided good results and DenseNet169, MobileNetV2 and Xception were the best in the tradeoff between accuracy and robustness.
Additionally, significant preprocessing, such as resizing, normalization, and augmentation, were included in the project to have better data quality and generalization. Assessment results (i.e., confusion matrices, performance curves and comparison bar charts) validated the models' ability to repeatedly detect subtle disease patterns. In practical context, this work aids in lessening dependence on manual visual inspection by farmers, efficiency in time and resource cost in automating aerial-based pipeline, and bringing farmers intelligent automation pipeline that can scale out to real agricultural scenes.

6.2 Limitation

The project had limitations, although it had been successful. The first challenge was the dataset. Despite several disease categories being represented in the dataset used, the size was relatively small compared to other large scale benchmarks. Therefore the models were less exposed to perturbations such as other lighting conditions, field environments or natural diversity among jute leaves in different regions. This shortcoming was partly addressed by data augmentation, but the models would most likely generalize better with a richer and more diverse dataset.

Another shortcoming was that the training phase required GPU resources from the cloud. Although we were able to access computational power via platforms such as Kaggle, training larger and deeper networks, or additional hyper parameter search was limited by resource and time constraints. When trained more epochs, the models performance also showed a decrease, which might be related to the overfitting by dataset size. Furthermore, this system has not been validated in the real field condition where the environmental noise, overlapping leaves or the damaged plant parts could include the complexities that are ignored under controlled environments. Such constraints emphasize the discrepancy between lab-controlled experiments and field deployment

6.3 Future Work

Future work will be done to address these issues in an effort to make the proposed system more generic, scalable and user friendly for the farms. The first contribution is the enlargement of the dataset, with bigger and more heterogeneous and real-world images taken in different agronomy fields, seasons and geographical areas. By including such type of diversity, the system may generalize better and handle variations in the real scenario beyond the controlled one. Field testing in real agricultural settings is of equal weight, as it will offer feedback on system performance under real agricultural conditions and support the calibration of the model for on-farm application.

Another exciting focus would be to investigate the possibility of enabling the FSL to run efficiently on mobile and edge devices. Efficient and light weight versions of models like DenseNet169 or Xception can be deployed to mobile phones for farmers to have disease detection capability in their farm without the need for any high computational infrastructure. Moreover, the integration of the detection in a mobile application would enable not only to classify but also to obtain actionable suggestions; e.g., treatment suggestion, preventive measure suggestions.

In addition to classification, the proposed system could be further extended to predictive analytics, in that not only the disease is classified above, but also possible occurrence of corresponding disease and severity of attack can be predicted using the weather and soil data. Internet of things (IoT)-based computer vision can further improve the system with a real-time monitoring and alarm. Hybrid methods that combine deep learning with conventional image processing may be further explored to improve result interpretability and robustness. Finally, multilingual user interfaces with the local languages might render the system more relevant to farmers in various localities and reduce the disparity between state of the art and local farming operations.

References

[1] R. Haque et al., "Advancements in Jute Leaf Disease Detection: A Comprehensive Study Utilizing Machine Learning and Deep Learning Techniques," 2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON), pp. 248–253, Sep. 2024, doi: https://doi.org/10.1109/peeiacon63629.2024.10800378.

[2] M. Z. Hasan, M. S. Ahamed, Aniruddha Rakshit, and M. Zubair, "Recognition of Jute Diseases by Leaf Image Classification using Convolutional Neural Network," *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–5, Jul. 2019, doi: https://doi.org/10.1109/icccnt45670.2019.8944907.

[3] Z. N. Reza, F. Nuzhat, N. A. Mahsa, and M. H. Ali, "Detecting jute plant disease using image processing and machine learning," *IEEE Xplore*, Sep. 01, 2016. https://ieeexplore.ieee.org/abstract/document/7873147 (accessed Mar. 14, 2021).

[4] Y. Tanny *et al.*, "DERIENet: A Deep Ensemble Learning Approach for High-Performance Detection of Jute Leaf Diseases," *Information*, vol. 16, no. 8, pp. 638–638, Jul. 2025, doi: https://doi.org/10.3390/info16080638.

[5] D. A. Godse, N. Tripathi, R. Wali, R. Singh, and Student Bharati Vidyapeeth, "DETECTING JUTE PLANT DISEASE USING IMAGE PROCESSING AND MACHINE LEARNING," vol. 5, no. 5, pp. 2394-0697, Jan. 2018, Available: https://www.researchgate.net/publication/362518974_DETECTING_JUTE_PLANT_DISEASE_USING_IMAGE_PROCESSING_AND_MACHINE_LEARNING

[6] R. H. Hridoy, T. Yeasmin, and Md. Mahfuzullah, "A Deep Multi-scale Feature Fusion Approach for Early Recognition of Jute Diseases and Pests," *Lecture Notes in Networks and Systems*, pp. 553–567, 2022, doi: https://doi.org/10.1007/978-981-19-1012-8_37.

[7] Md. S. H. Talukder and A. K. Sarkar, "Nutrients deficiency diagnosis of rice crop by weighted average ensemble learning," *Smart Agricultural Technology*, vol. 4, p. 100155, Aug. 2023, doi: https://doi.org/10.1016/j.atech.2022.100155.

[8] R. Ghosh, P. Palit, S. Paul, S. K. Ghosh, and A. Roy, "Detection of Corchorus golden mosaic virus Associated with Yellow Mosaic Disease of Jute (Corchorus capsularis)," *Indian Journal of Virology*, vol. 23, no. 1, pp. 70–74, Mar. 2012, doi: https://doi.org/10.1007/s13337-012-0062-7.

[9] S. Zhang, H. Wang, C. Zhang, Z. Liu, Y. Jiang, and L. Yu, "JutePest-YOLO: A Deep Learning Network for Jute Pest Identification and Detection," *IEEE Access*, vol. 12, pp. 72938–72956, Jan. 2024, doi: https://doi.org/10.1109/access.2024.3403491.

[10] Z. Özdemir and M. I. Cagirgan, "Identification and characterization of a phytoplasma disease of jute (Corchorus olitorius L.) from south-western Turkey," *Crop Protection*, vol. 74, pp. 1–8, Aug. 2015, doi: https://doi.org/10.1016/j.cropro.2015.03.018.

[11] D. Li, F. Ahmed, N. Wu, and A. I. Sethi, “YOLO-JD: A Deep Learning Network for Jute Diseases and Pests Detection from Images,” vol. 11, no. 7, pp. 937–937, Mar. 2022, doi: https://doi.org/10.3390/plants11070937.

[12] M. Ilhan Cagirgan, H. Topuz, N. Mbaye, and R. Silme, “FIRST REPORT ON THE OCCURRENCE AND SYMPTOMATOLOGY OF PHYLLODY DISEASE IN JUTE (Corchorus olitorius L.) AND ITS PLANT CHARACTERISTICS IN TURKEY,” *Turkish Journal of Field Crops*, vol. 19, no. 1, pp. 129–135, 2014, Accessed: Sep. 16, 2025. [Online]. Available: https://www.field-crops.org/assets/pdf/product5378554d30fb7.pdf.

[13] P. N. Meena, Bhimashankar Gotyal, Satyaram Satpathy, R. Babu, and M. R. Naik, “IPM CHA 9,” Jan. 29, 2021. https://www.researchgate.net/publication/348863330_IPM_CHA_9

[14] M. Ilhan CAGIRGAN, Hasan TOPUZ, N. MBAYE, and R. Soner SILME, “First Report on The Occurrence and Symptomatology of Phyllody Disease in Jute (Corchorus olitorius L.) and Its Plant Characteristics in Turkey,” *Turkish Journal Of Field Crops*, vol. 19, no. 1, pp. 129–129, Jan. 2014, doi: https://doi.org/10.17557/tjfc.40243.

[15] G. Geetha, S. Samundeswari, G. Saranya, K. Meenakshi, and M. Nithya, “Plant Leaf Disease Classification and Detection System Using Machine Learning,” *Journal of Physics: Conference Series*, vol. 1712, p. 012012, Dec. 2020, doi: https://doi.org/10.1088/1742-6596/1712/1/012012.

[16] D. Li, F. Ahmed, N. Wu, and A. I. Sethi, “YOLO-JD: A Deep Learning Network for Jute Diseases and Pests Detection from Images,” vol. 11, no. 7, pp. 937–937, Mar. 2022, doi: https://doi.org/10.3390/plants11070937.

[17] Z. N. Reza, F. Nuzhat, N. A. Mahsa, and M. H. Ali, “Detecting jute plant disease using image processing and machine learning,” *IEEE Xplore*, Sep. 01, 2016. https://ieeexplore.ieee.org/abstract/document/7873147.

213-15-4298

ORIGINALITY REPORT

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

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	6%
2	www.mdpi.com Internet Source	1%
3	Submitted to United International University Student Paper	1%
4	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
5	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%
6	Shankar Babu, Mahesh Babu Kota. "Synergies in Smart and Virtual Systems using computational intelligence", CRC Press, 2025 Publication	<1%
7	Submitted to University of Westminster Student Paper	<1%
8	www.coursehero.com Internet Source	<1%
9	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced	<1%

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.