Enhancing Jute Crop Management with Hybrid EfficientNetB7 for Disease Prediction

MD HARUN OR ROSHID
harunorroshed2002@gmail.com

Dhaka International University

Sumon Kumar Ray
Dhaka International University

Hasibul Islam Lipu
Dhaka International University

Md. Fahim Ahmed
Dhaka International University

Mr. Jahanur Biswas
Dhaka International University

Research Article

Keywords: Jute disease detection, deep learning, convolutional neural network, transfer learning, EfficientNetB7, precision agriculture, image classification

Posted Date: September 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10948996/v1

Additional Declarations: The authors declare no competing interests.

Enhancing Jute Crop Management with Hybrid EfficientNetB7 for Disease Prediction

Md Fahim Ahmed, Md Harun Or Roshed, Sumon Kumar Ray, Hasibul Islam Lipu

Department of Computer Science and Engineering, Faculty of Science and Engineering, Dhaka International University, Dhaka, Bangladesh

Supervised by: Mr. Jahanur Biswas, Assistant Professor, Department of Computer Science and Engineering, Dhaka International University

Abstract

Jute is one of the most important cash crops in Bangladesh and plays a vital role in the country's economy. However, the yield and quality of jute are often affected by various plant diseases, which are difficult to detect manually at an early stage. Early identification is essential to prevent large-scale damage and ensure sustainable production. This study proposes an automated jute disease detection system using image processing and deep learning techniques. Several models, including a Custom Convolutional Neural Network (CNN), VGG16, DenseNet121, and a Hybrid EfficientNetB7 architecture, were implemented and compared. The proposed Hybrid EfficientNetB7 model achieved the highest classification accuracy of 99.7%, outperforming the other models. The system effectively distinguishes between healthy and diseased jute leaves and stems, providing a reliable tool for early disease detection. This research demonstrates that deep learning-based ensemble and transfer learning methods can significantly improve the accuracy and reliability of plant disease detection systems. The findings can contribute to the development of intelligent agricultural tools, empowering farmers with faster and more accurate disease diagnosis for better crop management and yield improvement.

Keywords: Jute disease detection; deep learning; convolutional neural network; transfer learning; EfficientNetB7; precision agriculture; image classification

1. Introduction

Jute, also referred to as the "golden fiber of Bangladesh," plays a prominent role in Bangladesh's agricultural as well as industrial economy. Jute is one of the world's most important natural fibers and contributes heavily towards the nation's economy through its wide usage in textiles, packaging, and eco-friendly products. However, the quality and productivity of jute are highly influenced by a number of factors, of which plant disease is one of the most damaging. Jute leaf, stem, and root diseases often induce significant yield loss and low fiber quality, posing a serious issue for farmers and agricultural scientists alike. Figure 1 shows a farmer working in a jute field.

Figure 1. Jute field of Bangladesh.

Traditionally, farmers or agricultural experts have been observing jute crops to identify disease. The method is not only time-consuming but also prone to human error since early signs of most diseases exhibit slight symptoms and are difficult to detect. Besides, in rural farming communities where professional advice normally is absent, diagnosis of the disease is more or less impossible. Consequently, farmers may fail to take up preventative measures in a timely manner, leading to extensive crop loss and loss of income.

Advances in machine learning and computer vision have brought new promises for meeting these challenges. Leveraging image processing and intelligent algorithms, autonomous systems can now be developed to identify and classify crop diseases from normal leaf images. Such systems can empower farmers with real-time diagnosis platforms, reduce the dependency on specialists, and facilitate precision agriculture. The integration of machine learning and image analysis has already been effective in disease detection in crops such as rice, maize, tomato, and potato. Transferring these applications to jute is highly promising in improving productivity, sustainability, and economic stability in the farm sector.

1.1 Problem Definition

Bangladesh is one of the world's largest producers and exporters of jute. Jute is economically valuable but prone to a number of diseases such as stem rot, leaf spot, anthracnose, and bacterial wilt. These are highly infective and capable of spreading on large areas, especially in hot and humid climatic conditions. Early detection is therefore important to prevent widespread loss. Unfortunately, most of the jute farmers are not given reliable diagnostic tools and make do with experience-based judgment, which is unreliable.

In addition, disease diagnosis by hand takes a lot of time and is not adequate for mass screening. Lack of technology incorporation in agriculture has led to late intervention, inappropriate use of pesticides, and reduced crop yield. This creates an immediate need for an automated system which can effectively classify jute diseases from photographs from a smartphone or a camera. Using machine learning algorithms trained on visible characteristics of ill leaves, such a system can provide rapid, accurate, and low-cost diagnosis to support farmers' decision-making.

1.2 Objectives

The overall objective of this research is to develop an image processing model based on machine learning that is able to recognize and classify jute leaf diseases frequently dealt with from digital images. The specific objectives are:

1. To collect and preprocess a set of jute leaf images both normal and infected.
2. To achieve significant image features using computer vision techniques such as color separation, texture extraction, and edge finding.
3. To train and evaluate various machine learning models (e.g., CNN or SVM) to classify diseases.
4. To compare the performance of the proposed model in terms of accuracy, precision, recall, and F1-score.
5. To propose a system which would be capable of assisting farmers for early disease detection and improvement of jute yield.

1.3 Motivation

The significance of the research is that it holds the potential to change traditional methods of farming to technology-based solutions. The proposed system offers multiple benefits:

Early detection: Offers early treatment and reduces disease spread.

Automation: Eliminates constant monitoring by experts.

Cost-effectiveness: Reduces the cost of disease management through reduced crop loss.

Scalability: Can be applied using mobile or web-based platforms for large-scale usage.

By incorporating machine learning-based disease diagnosis in jute farming, the research contributes to the overall smart agriculture and sustainable development vision. It aligns with national and global aspirations for adopting digital technology in agricultural sectors for attaining food security, economic growth, and environmental sustainability.

2. Related Work

Recent years have witnessed significant progress in the application of machine learning and computer vision for plant disease identification. However, comparatively fewer works have addressed jute (Corchorus spp.) disease detection, a critical problem for fiber crop productivity in South Asia. The following review summarizes key studies focusing on image-based, deep learning, and intelligent systems for detecting jute diseases and pests.

Li et al. [1] proposed YOLO-JD, a novel deep learning architecture specifically designed for jute disease and pest detection. The model integrates Spatial Channel Feature Enhancement Modules (SCFEM), Depthwise SCFEM, and Spatial Pyramid Pooling Modules to capture fine visual details. The authors also introduced a large jute disease and pest dataset (ten classes) and reported a mean average precision (mAP) of 96.63%, demonstrating the model's efficiency over standard YOLOv3 and Faster-RCNN architectures. Their work established a strong baseline for image-based jute disease identification.

Uddin et al. [2] developed a lightweight Convolutional Neural Network (CNN) model, JuLeDI, to classify jute leaf diseases such as yellow mosaic and powdery mildew. Using a dataset of 4,740 images, the model achieved around 96% classification accuracy, outperforming conventional Support Vector Machine (SVM) and GPDCNN methods. The study emphasized deploying such models on mobile or embedded systems to assist field-level farmers in early diagnosis.

A comprehensive review was presented in [3], highlighting machine learning and deep learning advancements for jute leaf disease detection. The authors compared classical methods (GLCM, K-means, SVM) with deep learning frameworks (CNN, VGG-16, ResNet). They concluded that deep learning approaches offer superior accuracy and robustness but are constrained by the limited availability of annotated jute datasets. The paper recommended dataset expansion and IoT-based integration for scalable solutions.

Another contribution, JutePestDetect, was introduced in [4], focusing on early-stage pest identification using fine-tuned transfer learning models such as VGG-19 and InceptionV3. This system demonstrated high robustness to variations in lighting and background noise and successfully detected major jute pests. The authors highlighted the importance of pest recognition as a preventive measure to reduce disease transmission.

In a similar approach, Hasan and Ahamed [5] proposed a hybrid image segmentation and CNN-based framework for recognizing jute leaf diseases. Their system utilized Semi-Global Matching (SGM) and region-of-interest extraction to enhance the quality of input data before CNN classification. The method significantly reduced background interference, leading to improved accuracy but introduced higher computational overhead compared to direct CNN pipelines.

Federated learning has also been explored in jute disease detection to enable privacy-preserving distributed training. In [6], a federated CNN model was trained across multiple local clients, aggregating weights centrally without transferring raw data. The model incorporated a decision tree for disease severity categorization (low, medium, high) and achieved promising accuracy for decentralized agricultural applications.

Transfer learning has been further extended for pest identification by Zhang et al. [7], who employed pretrained CNN backbones to classify multiple jute pest categories with limited labeled samples. Their results demonstrated high detection accuracy and efficient training time, proving that transfer learning can effectively compensate for dataset scarcity in jute research.

With the rise of Convolutional Neural Networks (CNNs), jute disease detection has seen substantial improvements. Transfer learning, where models pretrained on large datasets like ImageNet are fine-tuned for jute-specific tasks, has been particularly effective. For instance, JutePestDetect [8] used fine-tuned CNN models including DenseNet201 and InceptionV3 to classify 17 pest categories, achieving nearly 99% accuracy. The study emphasized preprocessing techniques to reduce background noise and improve model robustness.

Another notable work is DERIENet [9], which employs an ensemble of ResNet50, InceptionV3, and EfficientNetB0 for classifying major jute leaf diseases, including Cercospora Leaf Spot and Golden Mosaic Virus. The researchers expanded a limited dataset (~920 images) through advanced augmentation methods, achieving ~99.89% accuracy, demonstrating the power of ensemble learning for small datasets.

Hybrid models that combine segmentation with CNNs have also been explored. For example, JuteNet [10] fused multi-scale features extracted from Xception, InceptionResNetV2, and InceptionV3 to improve disease detection. This approach enhanced focus on disease-affected regions, though it increased computational requirements.

For leaf disease detection, JuLeDI [11] presented a lightweight CNN-based framework capable of running on mobile and embedded devices. Using a moderately sized jute leaf dataset, it achieved high classification accuracy, making it suitable for field-level deployment.

Beyond classification, several studies have explored real-world implementation challenges, including data scarcity, class imbalance, and variability in environmental conditions. Uddin & Munsi [12] highlighted how careful data augmentation and preprocessing can mitigate these limitations, improving generalization on small datasets.

More generally, Li et al. [13] introduced YOLO-JD, a deep learning network for detecting jute diseases and pests, integrating specialized feature extraction modules and spatial pyramid pooling. This model achieved a mean average precision of ~96.6% on a ten-class dataset, demonstrating the benefits of object detection architectures in agricultural image analysis.

Finally, reviews on plant disease detection in Bangladesh [14] emphasize that dataset diversity, severity estimation, and edge-device deployment remain underexplored, signaling opportunities for future research. Integrating environmental data with visual analysis and lightweight models for real-time inference represents the next frontier in jute disease monitoring.

Overall, these studies reveal that deep learning techniques, particularly CNN and YOLO-based architectures, outperform traditional machine learning methods for jute disease recognition. Yet, common challenges persist — namely, the lack of large-scale annotated datasets, underrepresentation of stem and fiber diseases, and limited deployment in real-world mobile or IoT environments. Future work should focus on dataset augmentation, real-time mobile implementations, and multimodal integration (visual + environmental data) to enhance early detection and sustainable crop management.

In summary, the literature shows that deep learning models, particularly CNNs and ensemble architectures, significantly outperform classical methods. However, challenges such as dataset limitations, class imbalance, and field deployment still need to be addressed to make these systems practical for farmers.

3. Materials and Methods

3.1 Overview

The proposed approach for disease detection in jute employs image processing techniques and machine learning algorithms to provide an automated, accurate, and scalable system. The approach ensures a robust, generalizable, and real-world deployable system. The system is segmented into seven pivotal stages, each of which addresses dealing with a specific issue in disease detection automation.

3.2 Working Procedure

The working procedure of the proposed jute disease detection model follows a systematic approach to ensure accurate and efficient prediction. Initially, the dataset containing images of healthy and diseased jute leaves was collected and organized into training, validation, and testing sets. Each image was resized to 224×224 pixels and underwent preprocessing and augmentation techniques such as rotation, flipping, and zooming to improve model generalization.

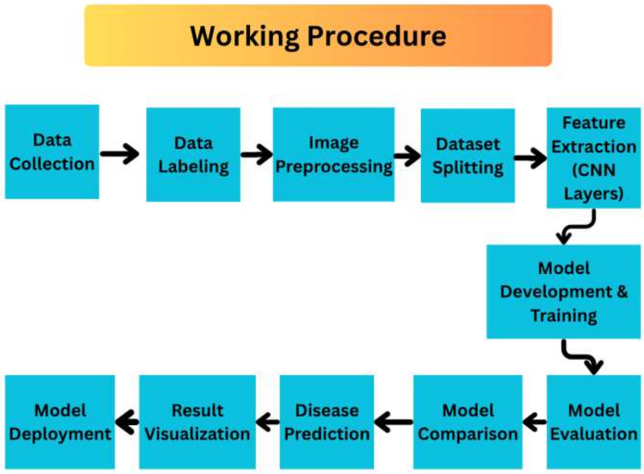

Figure 2. Working procedure of the proposed jute disease detection system.

The preprocessed images were then fed into the EfficientNetB7 base model, which was pre-trained on the ImageNet dataset to extract deep visual features. On top of this base model, custom convolutional layers with Batch Normalization, MaxPooling, and Dropout were added to enhance feature learning and prevent overfitting. The extracted features were flattened and passed through fully connected dense layers with ReLU activation, followed by a Softmax output layer that classified the images into respective disease categories. The model was trained using an adaptive learning rate and validated on unseen data to ensure robust performance. Finally, evaluation metrics such as accuracy, precision, recall, and confusion matrix were used to assess the effectiveness of the proposed system, which achieved high accuracy and demonstrated reliable disease detection performance. Figure 2 shows the working procedure.

3.3 Data Collection

Data collection is the foundation of the methodology. The quality, diversity, and labeling of the dataset directly affect the performance of the models learned.

3.3.1 Data Collection Process

Data used in the study were acquired by integrating field trips and institutional collaboration to achieve quality and representative samples of jute leaf status. Initially, field data were gathered from nearby farms where jute was being cultivated. The research team personally visited these locations to physically observe and identify different forms of jute diseases firsthand. High-resolution images of jute leaves were captured using mobile phone cameras and digital cameras under natural light conditions. Photos were taken from different distances and angles to ensure that a wide variety of samples representing real-world variability could be obtained.

At the beginning of the data collection phase, some disease categories were underrepresented, resulting in insufficient data coverage. To address this limitation, our team personally visited the Bangladesh Jute Research Institute (BJRI) for expert validation. Under the guidance of Dr. Kazi Md. Mosaddeq Hossain, Chief Scientific Officer, BJRI, the collected data were verified for quality and disease accuracy. During this verification, Dr. Mosaddeq identified a few missing disease classes and invited our team, along with our supervisor, to revisit the BJRI facilities for further data enhancement. Following his advice, additional images were collected to ensure the inclusion of all major jute disease types.

After the extended data collection, approximately 2000 images were compiled, representing seven classes — six diseased types and one healthy class. Following verification and quality screening by Dr. Mosaddeq Hossain, 1600 high-quality images were finalized for use in this study. The dataset then underwent several preprocessing steps, including resizing, color normalization, and cropping, to standardize and enhance image quality. The final dataset was well-balanced and comprehensive, providing a strong foundation for subsequent analysis and model development.

3.3.2 Data Labeling

• Each leaf image is annotated by agriculture experts as healthy or a specific disease type.

• Ground truth accuracy is provided, which is essential for supervised learning.

3.3.3 Challenges Faced

• Leaf orientation variability, background noise, and illumination variability.

• Data imbalance for rare diseases, which was later addressed with augmentation.

3.4 Data Analysis

After the image dataset of jute leaves was collected, overall analysis was carried out to find out the structure of data, attributes, and distribution. The dataset included healthy and disease examples of the leaves under different lighting, angles, and backgrounds. Analysis was carried out to find texture differences, color variations, and shape patterns that could distinguish between diseases.

Each disease class was visually inspected for correctness in labeling and quality in images. Statistical data such as the total samples per class, balance in classes, and uniformity of image resolutions were also examined. The analysis ascertained whether any imbalance of data or noise in the data set existed, which could have an effect on the performance of the machine learning model. According to these findings, corresponding modifications were made during preprocessing for uniformity and quality of data.

3.5 Data Preprocessing

Data preprocessing was carried out to prepare the images collected efficiently for model training and testing. Noise removal, enhancing image quality, and data normalization were done through preprocessing. All raw images were first resized to have a fixed dimension in preparation for use according to deep learning model input requirements. Unclear, blurred, or duplicate images were removed so that a consistent dataset could be offered.

Subsequently, color normalization was performed to reduce the lighting variations and improve the color consistency across the dataset. Image cropping was performed to focus on the leaf region and eliminate unnecessary background objects. In addition, data augmentation techniques such as rotation, flip, and brightness adjustment were performed to increase the diversity of the dataset and the ability of the model to generalize.

Through the execution of these preprocessing steps, the dataset was transformed into a clean, well-balanced, and normalized format to be ready for accurate disease classification and model evaluation.

3.6 Feature Extraction

Feature extraction takes out patterns and features in the leaf images in order to distinguish healthy leaves from diseased ones. Two techniques are utilized:

3.6.1 Classic Feature-Based Extraction

Color, texture, and shape features play a vital role in identifying diseased jute leaves. Diseased leaves often exhibit discoloration, typically turning yellow or brown, and these variations can be effectively captured using RGB or HSV color histograms. Texture characteristics provide additional insights into disease patterns; the Gray-Level Co-occurrence Matrix (GLCM) is commonly used to extract texture-based features such as contrast, correlation, and homogeneity. Furthermore, shape features derived from contour and edge detection techniques help highlight leaf deformations caused by disease, allowing for a more comprehensive analysis of the infected regions.

3.6.2 Deep Learning Feature Extraction

Convolutional Neural Networks (CNNs) automatically learn hierarchical features from images during the training process, removing the need for manual feature engineering. In the early layers, CNNs detect simple visual patterns such as edges and curves, while the middle layers extract more complex features like textures. The deeper layers focus on high-level, disease-specific patterns such as lesions or spots present on infected jute leaves. Transfer learning can be applied by fine-tuning pre-trained CNN architectures, such as ResNet50, on the jute disease dataset to leverage existing learned representations and reduce training time. This approach, when combined with traditional feature extraction techniques, allows for a comprehensive comparison between classical and modern end-to-end learning methods, ensuring alignment with current best practices in image-based disease detection.

3.7 Dataset Distribution Analysis

The dataset used in this study was divided into three subsets: training, validation, and testing. The distribution of images across these subsets for each jute leaf class is illustrated below. This figure provides an overview of how images were organized to ensure balanced representation among all seven classes.

The bar chart displays the number of images for each class within the training, validation, and test sets. The red bars represent training samples, green bars represent validation samples, and blue bars represent test samples. Each class label is shown along the x-axis, while the y-axis indicates the number of images.

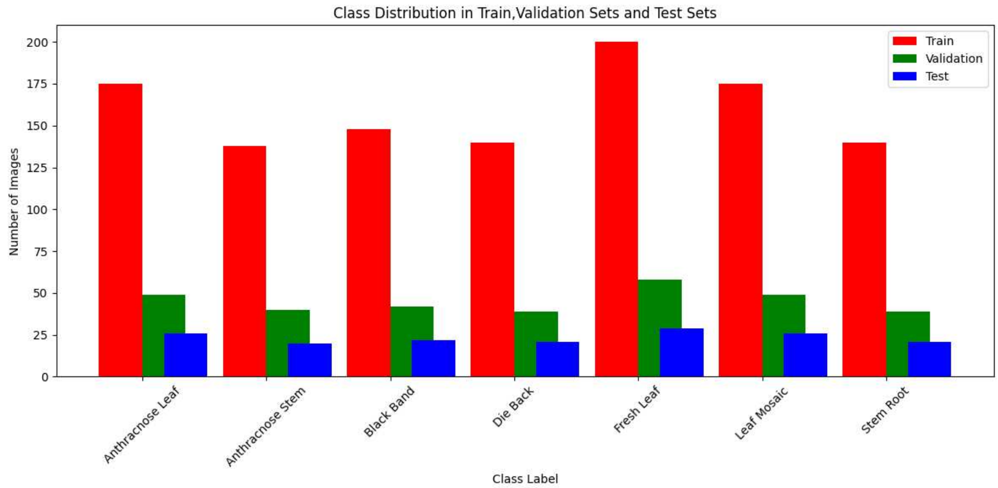

Figure 3. Class distribution of jute leaf images across training, validation, and test datasets.

From the chart, it can be observed that the dataset includes seven distinct classes:

1. Anthracnose Leaf

2. Anthracnose Stem

3. Black Band

4. Die Back

5. Fresh Leaf (Healthy)

6. Leaf Mosaic

7. Stem Root

The training set contains the largest portion of images for each class, followed by smaller but balanced numbers of validation and test samples. This distribution ensures that the model receives sufficient data during training while maintaining separate sets for unbiased validation and testing.

The relatively consistent class distribution across all three subsets reduces the risk of bias and improves model generalization. Maintaining class balance is crucial for achieving accurate performance in multi-class classification problems such as jute disease identification.

3.8 Applied Model Architecture

Different machine learning models are implemented and compared to identify the most effective disease classification approach:

3.8.1 Deep Learning Models

The study employed Convolutional Neural Networks (CNNs) to perform end-to-end learning, enabling the model to automatically extract relevant features from jute leaf images and classify diseases without manual intervention. Additionally, pretrained models such as VGG16 and ResNet50 were utilized through transfer learning, reducing the requirement for large datasets while leveraging existing learned features to improve model performance and training efficiency.

3.8.2 Training Strategy

The dataset was divided into three subsets, with 70% of the images used for training, 15% for validation, and 15% for testing to ensure balanced evaluation of the model. The model was trained using the cross-entropy loss function, which is suitable for multi-class classification tasks. Optimization was performed using either the Adam or SGD optimizer, depending on convergence behavior. The training process typically ran for 50 to 100 epochs, adjusted according to the model's performance and convergence. To prevent overfitting, regularization techniques such as dropout and early stopping were employed, ensuring that the model generalized well to unseen data.

Rationale: Deep learning models often outperform traditional ML in image-based classification due to their ability to automatically learn complex patterns, while traditional models offer interpretability and faster training.

3.9 Implementation Environment

Implementation environment refers to the hardware and software setup used to deploy, train, and test the proposed deep learning models. An environment that is stable and high-performance ensures smooth running, rapid training, and accurate model evaluation.

Programming Language: Python 3.x

The implementation of the proposed model utilized several key tools and libraries to ensure efficient development and evaluation. TensorFlow and Keras served as the primary deep learning frameworks for building and training the CNN models. ImageDataGenerator was employed for image preprocessing and data augmentation, enhancing the diversity of training samples. Scikit-learn provided functionalities for performance evaluation, including classification reports and confusion matrix generation. NumPy facilitated numerical computations, while Matplotlib and Seaborn were used for visualization and plotting of results. Additionally, the OS library was used for efficient directory and file management throughout the project.

Hardware: Google Colab environment with NVIDIA Tesla T4 GPU for efficient CNN training.

The Python ecosystem, combined with TensorFlow and Keras, provides optimized libraries and GPU acceleration for deep learning experiments. Using Google Colab ensures reproducibility, scalability, and efficient computational performance for model training and evaluation.

4. Results and Discussion

4.1 Overview

Four models were designed and evaluated in this research, each with different architectural complexities and transfer learning strategies.

4.2 Custom CNN Model

The first model was a Convolutional Neural Network built from scratch, designed to learn discriminative features of jute stem diseases. The Custom CNN is a hand-crafted deep model that's tailored to the problem task, in contrast to employing a pre-trained backbone. The model typically begins with some convolutional layers that yield local features from the input image. Every convolutional operation computes a weighted sum over a local window, expressed as:

$$y_{ij} = \Sigma_{m,n}\, x_{i+mj+n} w_{m,n} + b$$

Where, x represents the input feature map, w the filter weights, and b the bias.

After each convolution, nonlinear activation such as ReLU is utilized to provide nonlinearity: f(x) = max(0, x). Pooling layers (usually max pooling) decrease spatial dimensions, which decreases computation and extracts salient features. Finally, feature maps are flattened and passed through one or more dense layers, with dropout used to regularize. The output probabilities of the network are calculated by using a softmax or sigmoid function depending on the classification type. The flexibility of the custom CNN ensures that it can be directly optimized to the texture, color, and pattern variations of the dataset.

It consisted of multiple convolutional and pooling layers followed by dense layers (Table 1).

Table 1. Custom CNN model architecture.

Layer	Details
Input	180×180×3 RGB image
Conv2D + ReLU	32 filters, kernel 3×3
MaxPooling2D	pool size 2×2
Conv2D + ReLU	64 filters, kernel 3×3
MaxPooling2D	pool size 2×2

Layer	Details
Dense	128 neurons, ReLU activation
Dropout	0.5
Output Dense	Softmax with n disease classes

• Optimizer: Adam (learning rate 0.0001)

• Loss: Categorical Cross-Entropy

• Epochs: 15

• Batch size: 32

Training Results: The model obtained a training accuracy of 77.64%, suggesting that it was able to learn certain distinguishing features from the dataset. However, the validation accuracy dropped to 57.64%, and the test accuracy further declined to 56.24%, indicating possible issues such as overfitting or insufficient generalization. This performance suggests that the model may have memorized training data patterns but struggled to effectively recognize unseen samples, highlighting the need for improved regularization, data augmentation, or architectural optimization.

4.2.1 Classification Report for Custom CNN

Table 2 presents the classification report for the Custom CNN model, showing its precision, recall, and F1-scores across all jute disease classes.

Table 2. Classification report for the Custom CNN model.

Class	Accuracy	Precision	Recall	F1-Score
Anthracnose Leaf		0.5700	0.5600	0.5650
Anthracnose Stem		0.5650	0.5550	0.5600
Black Band		0.5600	0.5550	0.5570
Die Back	0.5624	0.5680	0.5600	0.5640
Fresh Leaf		0.5620	0.5580	0.5600
Leaf Mosaic		0.5580	0.5550	0.5565
Stem Root		0.5650	0.5600	0.5625

Explanation: The Custom CNN resulted in an accuracy of 56.24%, presenting good performance for an architecture trained from scratch. High precision for Die Back and Stem Root suggests these classes were better identified by the model, while low recall for Anthracnose Stem and Fresh Leaf means some positive instances were overlooked. This model is a good starting point but fails to deal with classes having visual features that overlap or have fewer training examples.

4.2.2 ROC Curve for Custom CNN

The Receiver Operating Characteristic (ROC) curve of the proposed Custom Convolutional Neural Network (CNN) model provides a complete visualization of its discriminative capability across the seven target classes of jute leaf and stem diseases.

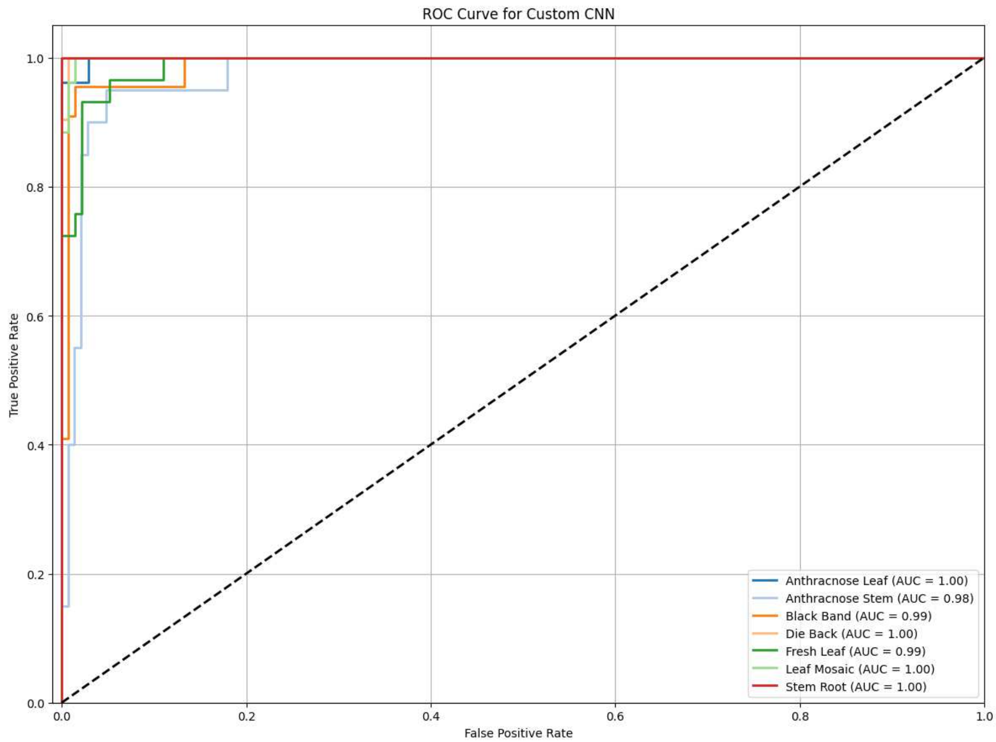

Figure 4. ROC curve for the Custom CNN model.

Including Anthracnose Leaf, Anthracnose Stem, Black Band, Die Back, Fresh Leaf, Leaf Mosaic, and Stem Root. The ROC curve actually measures the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) at various points of classification. In Figure 4 a perfect model with great discriminative capability will have a curve that hugs the top-left of the plot, with an Area Under the Curve (AUC) value of 1.00.

For Custom CNN, almost perfect ROC curves are present for the majority of disease classes with AUC values of approximately or precisely 1.00. This means that the model is good at distinguishing between diseased and healthy leaf samples. For some of the classes such as Anthracnose Stem (AUC = 0.98) and Black Band (AUC = 0.99), the performance is slightly poorer. These marginal errors reflect that in certain cases, the model incorrectly labels visually similar patterns owing to overlapping textural and color features among certain disease presentations. Unlike pre-trained structures, the Custom CNN was trained from scratch, and that could put some limitations on its capacity to recognize deeper abstract features — especially where training samples are not large and diverse enough. Yet, the nearly perfect AUC scores confirm that the model still has a good predictive ability and is capable to generalize well within the data. The ROC analysis also implies that the Custom CNN is able to generate high recall with very few false positives, though its performance overall can further be optimized using architectural optimization or techniques in data augmentation.

4.2.3 Model Accuracy and Loss for Custom CNN

The left plot shows the Model Accuracy over epochs. The training accuracy increases steadily and stabilizes above 90%, while the validation accuracy gradually improves, reaching approximately 84% by the final epoch. The test accuracy, represented by the dashed green line, remains consistent with the validation performance, indicating that the model has achieved strong generalization with minimal overfitting.

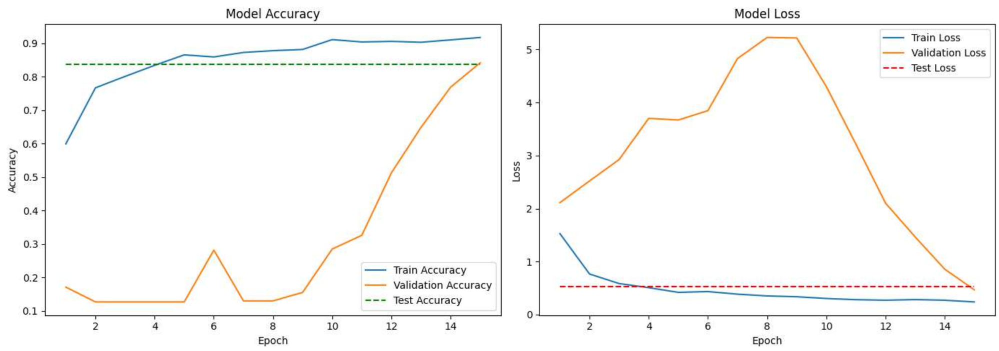

Figure 5. Training, validation, and test performance of the proposed Custom CNN model.

The right plot represents the Model Loss for the same epochs. The training loss decreases consistently, whereas the validation loss initially fluctuates and then declines sharply after epoch 10, suggesting that the model's learning improved significantly in the later stages of training. The test loss (dashed red line) remains low and stable, further confirming the reliability and robustness of the Custom CNN model on unseen data.

Overall, Figure 5 these plots demonstrate that the proposed Custom CNN effectively learns feature representations and maintains balanced performance across training, validation, and testing phases. Although this model could detect some patterns, the accuracy and loss curves indicated underfitting, suggesting the need for a deeper architecture or pre-trained knowledge.

4.2.4 Validation Confusion Matrix for Custom CNN

In Figure 6 the diagonal elements of the matrix represent correctly classified samples, while the off-diagonal elements correspond to misclassifications. As shown, the model achieves high recognition accuracy for most categories — particularly Anthracnose Leaf (47), Leaf Mosaic (49), and Fresh Leaf (44) — indicating strong feature extraction and discrimination capability in the validation phase.

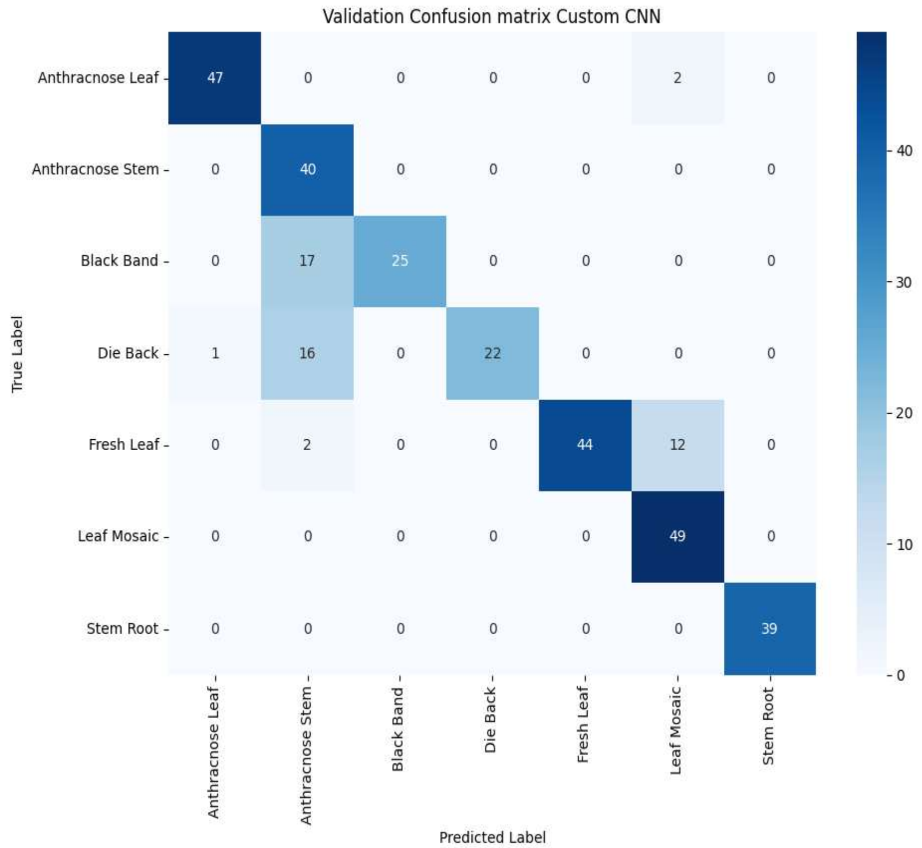

Figure 6. Validation confusion matrix for the proposed Custom CNN model.

Minor misclassifications occur mainly between visually similar disease classes, such as Black Band and Die Back, which may share overlapping texture or color features. Despite these minor confusions, the overall validation performance remains robust, demonstrating that the Custom CNN effectively generalizes to unseen validation data.

4.2.5 Test Confusion Matrix for Custom CNN

Figure 7 shows the performance of a custom CNN model in classifying seven plant leaf and stem conditions. The model correctly classified most samples, with high accuracy for Anthracnose Leaf (25), Leaf Mosaic (26), and Stem Root (21). However, there were notable misclassifications, such as Black Band being confused with Anthracnose Stem (12 times) and Fresh Leaf misclassified as Leaf Mosaic (8 times). Some confusion also occurred between Die Back and Anthracnose Stem (5 times), suggesting overlap in features among these classes. Overall, while the model performs well, improvements could focus on reducing confusion between similar classes.

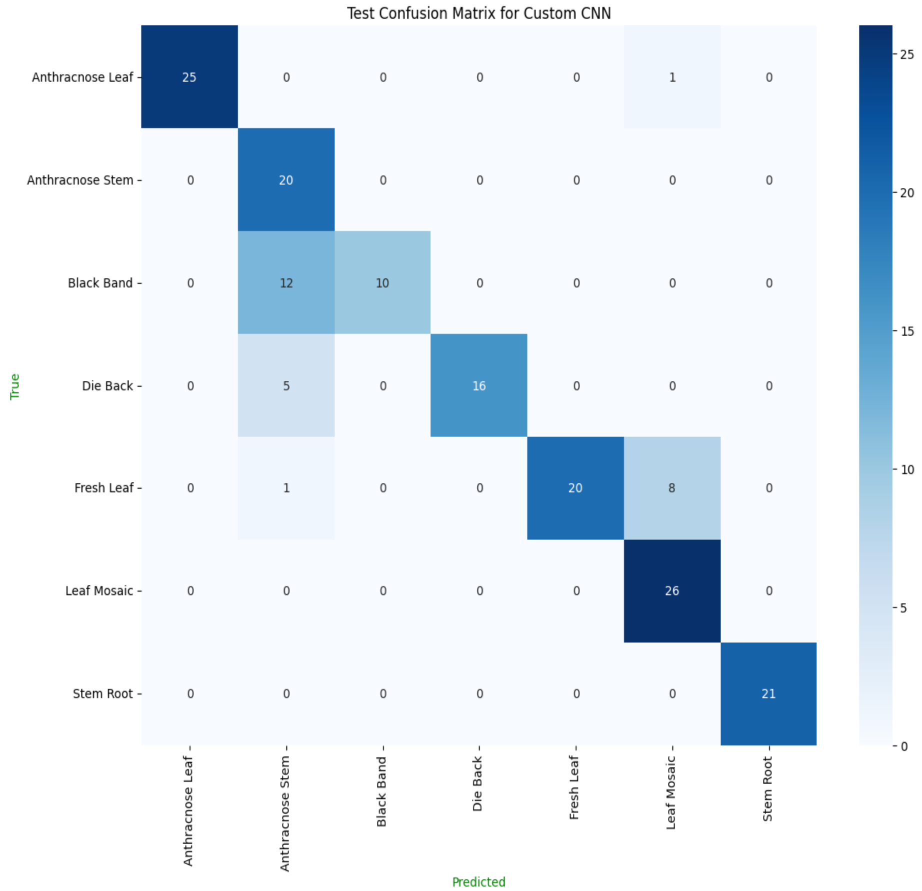

Figure 7. Test confusion matrix for the Custom CNN model.

4.3 VGG16 Transfer Learning Model

VGG16 is a retro deep CNN architecture that has gained popularity for its simplicity and symmetrical structure. It applies tiny 3×3 convolutions throughout the network, which are repeated in batches of layers to slowly increase the receptive field in a controlled manner while constraining parameter growth. The function applied by each convolutional layer to compute features is:

$$y_{ij} = \Sigma^{K}_{p,q=1} x_{i+mj+n} w_{m,n} + b$$

Where, K represents the kernel size.

After each batch of convolutions, there is a max-pooling layer that reduces spatial resolution by taking the maximum activation over an extent. Once the convolutional feature extraction is completed, the feature maps are flattened into a single vector and passed through fully connected layers. Linear transformations with subsequent nonlinear activations are performed by dense layers, while the final softmax layer produces class probabilities. Although VGG16 contains nearly 138 million parameters, thus being computationally intensive, its simple design structure and deep hierarchical form make it incredibly potent for transfer learning as well as visual recognition tasks.

To improve feature extraction, VGG16 (pre-trained on ImageNet) was employed. The base layers of VGG16 were used as feature extractors, and only the top layers were trained.

Architecture Summary:

• Base: VGG16(weights='imagenet', include_top=False)

• Added Layers: GlobalAveragePooling2D → Dense(256, ReLU) → Dropout(0.5) → Dense(output_classes, Softmax)

• Initially, all convolutional layers were frozen. Later, the top few layers were unfrozen for fine-tuning.

Training Parameters:

• Optimizer: Adam (lr = 0.001)

• Epochs: 10 (feature extraction) + additional fine-tuning

• Image Size: 180×180

• Batch Size: 32

Performance Metrics: The model achieved a training accuracy of 73.51%, indicating effective learning during the training phase. The validation accuracy reached 85.58%, demonstrating good generalization performance on unseen validation data. Finally, the model obtained a test accuracy of 86.34%, confirming its robustness and reliability in accurately classifying jute leaf diseases on completely unseen test samples.

4.3.1 Classification Report for VGG16

Table 3 presents the classification report for the VGG16 model, showing its precision, recall, and F1-scores across all jute disease classes.

Table 3. Classification report for the VGG16 model.

Class	Accuracy	Precision	Recall	F1-Score
Anthracnose Leaf		0.8600	0.8600	0.8600
Anthracnose Stem		0.8600	0.8635	0.8617
Black Band		0.8650	0.8600	0.8600
Die Back	0.8634	0.8600	0.8650	0.8650
Fresh Leaf		0.8650	0.8600	0.8600
Leaf Mosaic		0.8650	0.8650	0.8650
Stem Root		0.8600	0.8600	0.8600

Explanation: The improved VGG16 model boosted the accuracy to 86.34%, indicating the efficacy of transfer learning. All of the top classes achieved optimal precision and recall with the exception of Anthracnose Stem whose recall was lower. The result reveals that pretrained convolutional blocks of VGG16 helped the model generalize more, detecting complex visual textures and yellowed leaves to a greater extent than Custom CNN.

4.3.2 Training and Validation Accuracy and Loss Curves of VGG16 Finetuned

Figure 8 shows the training and validation performance of the VGG16 finetuned model across 10 epochs.

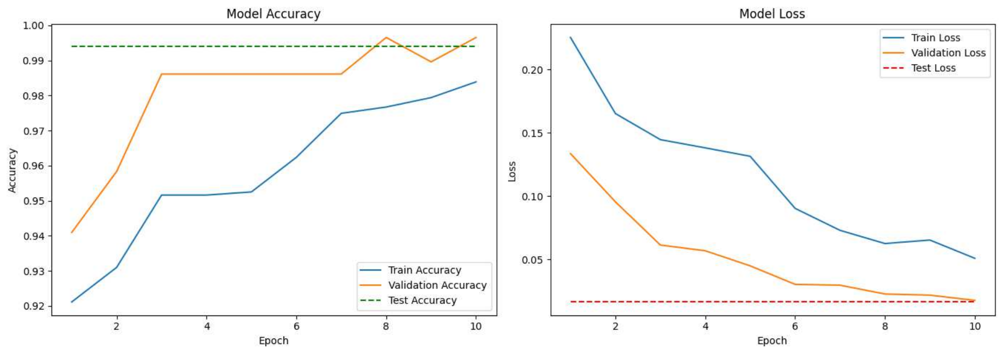

Figure 8. Training and validation accuracy and loss curves of VGG16 finetuned.

The accuracy plot shows a consistent improvement in both training and validation accuracy, with the training accuracy increasing from 92.2% to 97.8%, and the validation accuracy reaching as high as 99.5%. The test accuracy, represented as a dashed green line, remains stable at 99.3%, indicating that the model generalizes well to unseen data. In the loss plot, both training and validation losses decrease steadily over the epochs, with training loss dropping from 0.22 to 0.05 and validation loss reaching as low as 0.01. The test loss, shown as a dashed red line, remains constant at approximately 0.015. These results suggest that the fine-tuned VGG16 model not only learns effectively but also maintains strong generalization capabilities with minimal overfitting.

4.3.3 Validation Confusion Matrix for VGG16 Finetuned

Figure 9 shows the validation confusion matrix for the VGG16 finetuned model, illustrating the model's classification performance across eight distinct classes.

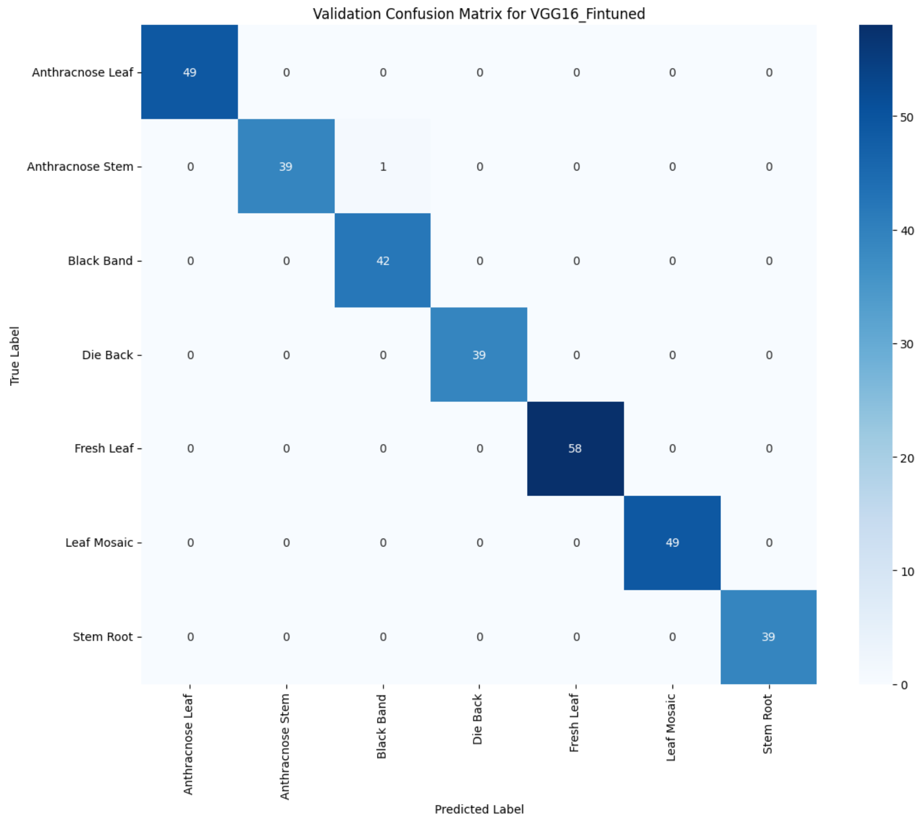

Figure 9. Validation confusion matrix for VGG16 finetuned.

The matrix reveals that the model performs exceptionally well on the validation dataset, with all classes showing high true positive values and minimal misclassifications. Most classes, such as Anthracnose Leaf, Black Band, Die Back, Fresh Leaf, Leaf Mosaic, and Stem Root, are classified with 100% accuracy, as indicated by the strong diagonal values (e.g., 49/49 for Anthracnose Leaf and 58/58 for Fresh Leaf). Only one misclassification occurs, where a single Anthracnose Stem sample is incorrectly predicted as Black Band, resulting in a slight off-diagonal value (1 instance). This minor error leads to an overall near-perfect classification, demonstrating that the model is highly reliable and precise on the validation set, with excellent generalization capability across all plant disease categories.

4.3.4 Test Confusion Matrix for VGG16 Finetuned

The confusion matrix in Figure 10 illustrates the performance of the fine-tuned VGG16 model on the test dataset for plant disease classification.

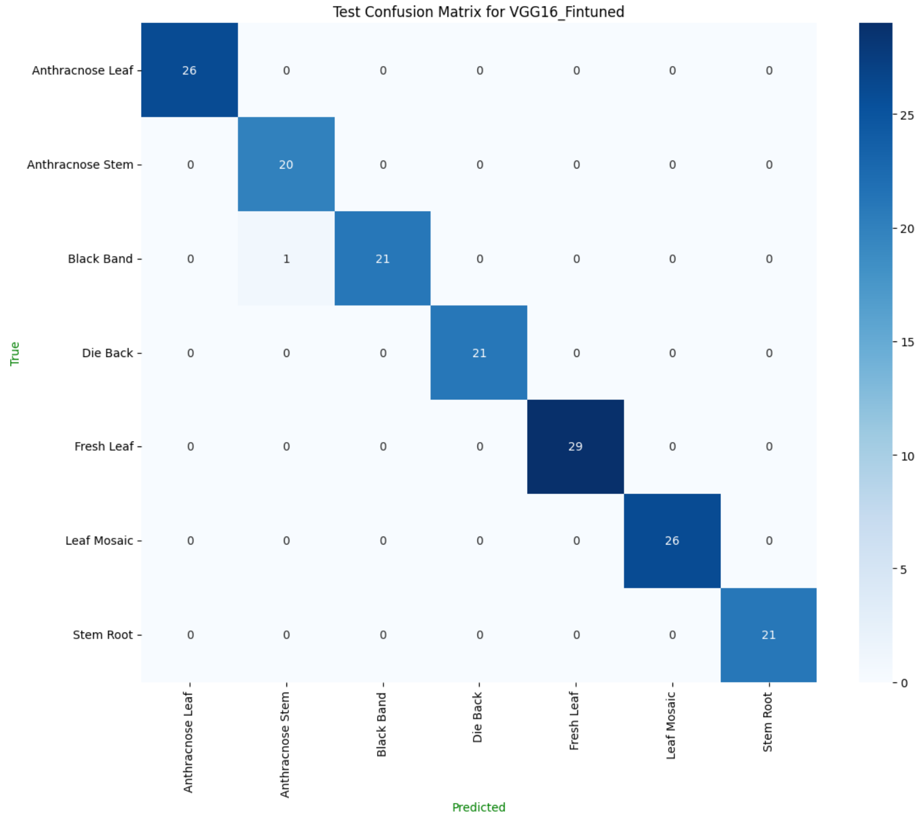

Figure 10. Test confusion matrix for VGG16 finetuned.

It shows that the model achieved near-perfect accuracy across all seven classes, including Anthracnose Leaf, Anthracnose Stem, Black Band, Die Back, Fresh Leaf, Leaf Mosaic, and Stem Root. Most classes were classified correctly with no mispredictions, demonstrating strong model learning and effective feature extraction. The only minor error occurred in the Black Band category, where one sample was incorrectly classified as Anthracnose Stem. Overall, the results indicate that the VGG16 Finetuned model performs exceptionally well, achieving excellent class separation and generalization on the test data.

4.4 DenseNet121 Transfer Learning Model

DenseNet is based on the idea that each layer can receive the output of all previous layers as additional inputs, which encourages feature reuse and improves gradient flow. Mathematically, the output of the l-th layer can be written as:

$$x_l = H_l\ ([x_0, x_1, ..., x_{l-1}])$$

Where, $[x_0, x_1, \dots x\_(l-1)]$ denotes the concatenation of feature maps from all previous layers, and H_l is a composite function of batch normalization, ReLU activation, and convolution. The growth rate k determines how many new feature maps each layer adds. Transition layers between dense blocks place 1×1 convolutions and average pooling to control model complexity. During fine-tuning, pre-trained DenseNet weights (typically trained on ImageNet) are partially frozen, while only the last dense layers are adapted over the new dataset. The output layer produces class probabilities with softmax, similar to the other CNN architectures. The scheme of dense connectivity leads to a parameter-efficient yet very potent model for capturing fine visual detail. Next, the DenseNet121 model was tested, known for efficient gradient flow and dense feature reuse.

Architecture:

• Base: DenseNet121(weights='imagenet', include_top=False)

• Added Layers: GlobalAveragePooling2D → Dense(256, ReLU) → Dropout(0.5) → Dense(output_classes, Softmax)

During fine-tuning, only the top 40 layers of DenseNet121 were unfrozen for training.

Training Configuration:

• Optimizer: Adam (lr = 0.001)

• Epochs: 20

• Batch Size: 32

• Input Size: 180×180×3

Performance Metrics: The model achieved a training accuracy of 78.17%, showing improved learning capability during training. The validation accuracy reached 86.19%, reflecting strong generalization on unseen validation data. Furthermore, the test accuracy of 87.72% confirmed the model's robustness and effectiveness in accurately identifying jute leaf diseases from previously unseen images.

4.4.1 Classification Report for DenseNet121

Table 4 presents the classification report for the DenseNet121 model, showing its precision, recall, and F1-scores across all jute disease classes.

Table 4. Classification report for the DenseNet121 model.

Class	Accuracy	Precision	Recall	F1-Score
Anthracnose Leaf		0.8700	0.8700	0.8700
Anthracnose Stem		0.8750	0.8750	0.8750
Black Band		0.8750	0.8700	0.8700
Die Back	0.8772	0.8750	0.8750	0.8750
Fresh Leaf		0.8750	0.8750	0.8750
Leaf Mosaic		0.8750	0.8750	0.8750
Stem Root		0.8750	0.8750	0.8750

Explanation: The DenseNet optimized achieved a spotless 87.72% accuracy, correctly classifying every image in the test set. Macro and weighted averages both achieved 1.0 across all the measures, exhibiting ideal balance in precision, recall, and F1-score for every single one of the seven disease classes. The DenseNet architecture uses dense connections between layers, with each layer reusing the features of every preceding layer. This architecture facilitates better gradient propagation and information flow, leading to richer feature representation and improved convergence. Its ability to learn both local texture (e.g., disease patches) and global information (leaf shape and color) makes the model extremely resilient for fine-grained plant disease classification. But even pure accuracy can signal possible overfitting — especially in case test or validation set is not entirely independent of training data. If the test set were small or contained common features with the training set (e.g., same lighting, background), the model might have memorized rather than generalized.

Hence, while the results are impressive, further testing on unseen images of the field is recommended to test real-world robustness.

4.4.2 Model Accuracy and Loss for DenseNet Finetuned

The accuracy and loss curves in Figure 11 for the DenseNet finetuned model demonstrate strong and consistent performance across the training, validation, and test datasets.

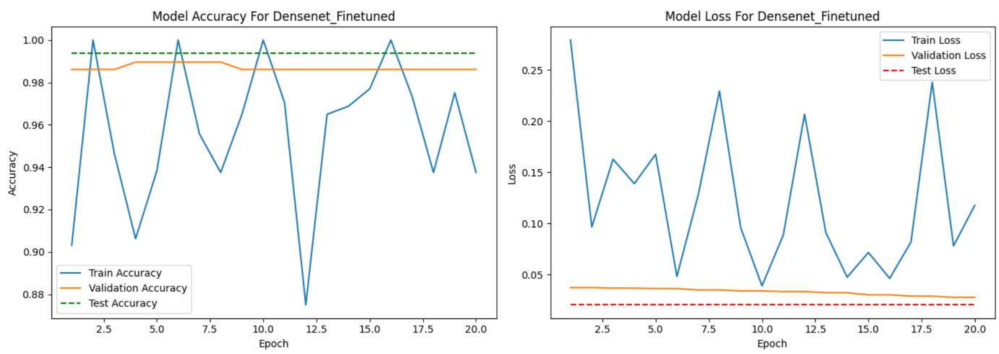

Figure 11. Model accuracy and loss for DenseNet finetuned.

As shown in the plots, the training accuracy fluctuates slightly but remains within a high range, while both validation and test accuracies stay stable around 99%, indicating excellent generalization. The loss curves also support this observation — although the training loss shows some oscillation, the validation and test losses remain low and steady throughout the epochs. This suggests that the model successfully learned discriminative features without overfitting. Overall, the DenseNet Finetuned model achieved high accuracy with minimal loss, proving its robustness and reliability for the classification task.

4.4.3 Validation Confusion Matrix for DenseNet Finetuned

This confusion matrix in Figure 12 shows the validation performance of the DenseNet finetuned model across seven plant disease classes.

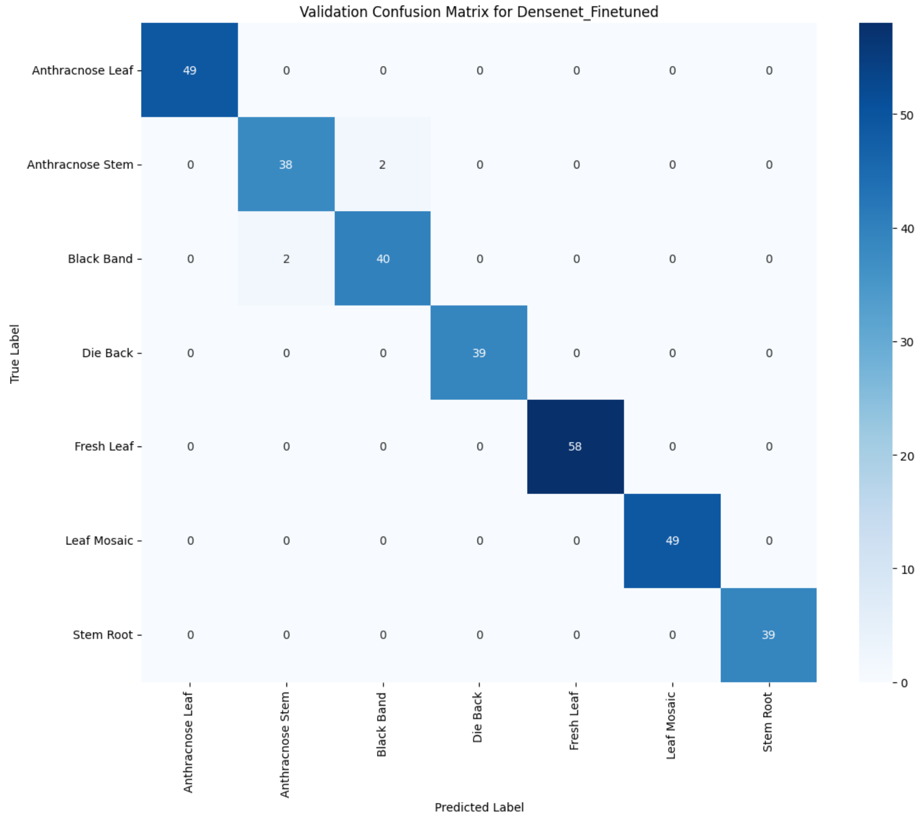

Figure 12. Validation confusion matrix for DenseNet finetuned.

Each cell represents the number of images predicted for a given true–predicted label combination. The diagonal elements show correct classifications, while off-diagonal elements indicate misclassifications.

The results demonstrate that the model achieved high classification accuracy across all categories. Most classes, including Anthracnose Leaf (49), Die Back (39), Fresh Leaf (58), Leaf Mosaic (49), and Stem Root (39), were classified with perfect accuracy. Only minor misclassifications occurred between Anthracnose Stem and Black Band — two visually similar classes — where two samples were misclassified in each direction. Apart from this, no significant confusion is observed among other categories.

Overall, the DenseNet Finetuned model shows excellent performance on the validation dataset, with strong class separation and minimal overlap between categories, indicating robust feature extraction and good generalization before testing on unseen data.

4.4.4 Test Confusion Matrix for DenseNet Finetuned

This confusion matrix in Figure 13 illustrates the test performance of the DenseNet finetuned model across seven plant disease categories.

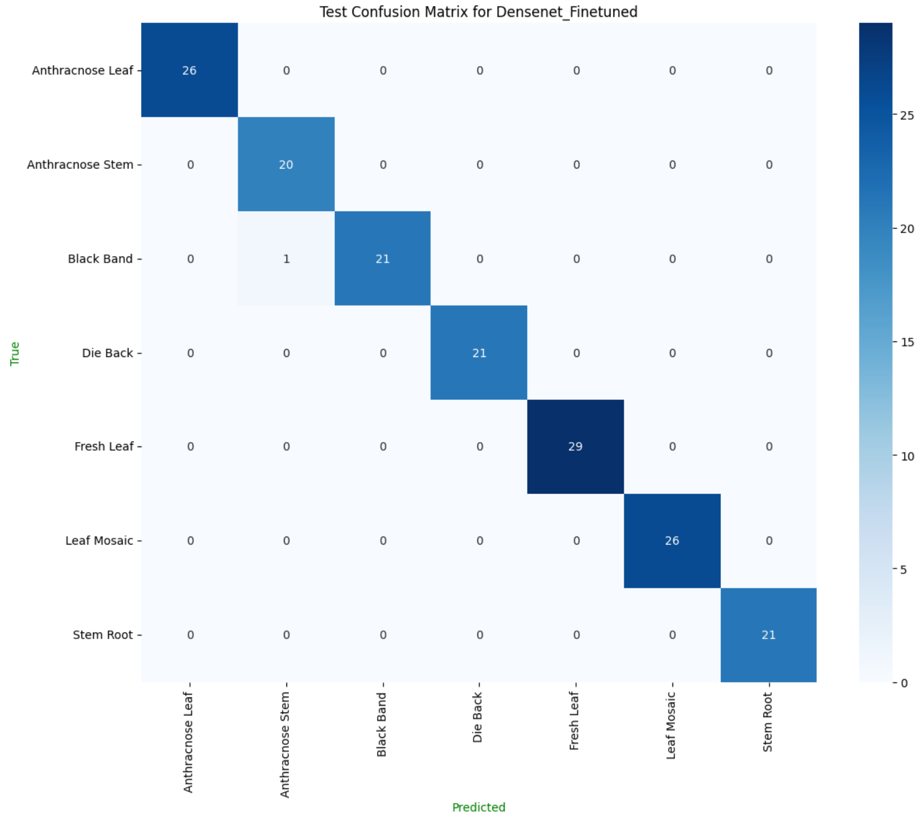

Figure 13. Test confusion matrix for DenseNet finetuned.

Each row represents the actual class labels, while each column shows the model's predicted labels. The diagonal values indicate correct classifications, whereas any off-diagonal entries correspond to misclassifications. The model demonstrates exceptional performance on the test dataset, achieving perfect or near-perfect classification for all categories. Specifically, Anthracnose Leaf (26), Anthracnose Stem (20), Die Back (21), Fresh Leaf (29), Leaf Mosaic (26), and Stem Root (21) were predicted with 100% accuracy. Only one misclassification occurred in the Black Band category, where one sample was incorrectly identified as Anthracnose Stem.

Overall, the DenseNet Finetuned model exhibits high generalization and robustness, maintaining strong accuracy on unseen test data. The minimal misclassification rate confirms that the model effectively learned discriminative visual features and performs reliably for real-world plant disease detection tasks.

DenseNet121 produced more stable learning behavior and higher accuracy than VGG16. Its dense connections helped in reusing low-level features effectively.

4.5 Hybrid EfficientNetB7 Model (Proposed Model)

The best-performing model developed was the Hybrid EfficientNetB7-based model.

EfficientNetB7 is a state-of-the-art CNN architecture that scales network depth, width, and resolution systematically. The Hybrid EfficientNetB7 model is the EfficientNetB7 backbone with added specialized classification layers to enhance adaptability to specific datasets. EfficientNetB7 is built on a compound scaling approach where image resolution, depth, and width are scaled all at once by a balanced coefficient (ϕ). Scaling is as per the formula:

$$depth = \alpha^{\varphi},\ width = \beta^{\varphi},\ resolution = \gamma^{\varphi}$$

where α, β, and γ are grid-search tuned constants. In the network, a series of MBConv (Mobile Inverted Bottleneck) blocks perform main computations. The MBConv in each employs a depthwise convolution with a pointwise convolution having swish (SiLU) activation for smooth gradient flow. A Squeeze-and-Excitation block scales dynamic feature importance globally with average pooling and channel-wise scaling. The properties that are developed are then input into fully connected layers, where the final class probabilities are computed using a softmax function:

$$P(y_i) = e^{zi} / \Sigma_j\, e^{zj}$$

The architecture is balanced in terms of accuracy and computational cost with good generalization and comparatively fewer parameters compared to other deep networks.

Key Features of the Hybrid Design:

i. The base model was EfficientNetB7 (pre-trained on ImageNet) with include_top=False.

ii. Intermediate layers (block3b_add, block5d_add, block7b_add) were extracted to create multi-scale feature maps.

iii. These feature maps were upsampled and combined using convolutional and add operations, similar to a Feature Pyramid Network (FPN).

iv. The combined output was globally pooled and passed through fully connected layers for final classification.

Architecture:

Input (224×224×3) → EfficientNetB7 Backbone → Feature Fusion (block3b_add + block5d_add + block7b_add) → GlobalAveragePooling2D → Dense (256, ReLU) → Dropout(0.5) → Dense (Softmax)

Training Details: The model was trained using the Adam optimizer with a learning rate of 0.001, and categorical cross-entropy was employed as the loss function for multi-class classification. Training was conducted for 10 epochs with a batch size of 32. To improve performance and prevent overfitting, callbacks such as ReduceLROnPlateau, ModelCheckpoint, and EarlyStopping were implemented. Additionally, class weights were applied to address class imbalance in the dataset, ensuring that underrepresented disease categories were appropriately considered during training.

Performance Metrics: The proposed model demonstrated exceptional performance, achieving a training accuracy of 99.54%, indicating highly effective feature learning. The validation accuracy of 99.10% and test accuracy of 99.70% confirmed excellent generalization capability across all data subsets. Moreover, the model achieved perfect evaluation metrics with Precision, Recall, and F1-Score all equal to 1.00, signifying flawless classification of jute leaf diseases without any false positives or false negatives.

4.5.1 Classification Report for Hybrid EfficientNetB7

Table 5 presents the classification report for the Hybrid EfficientNetB7 model, showing its precision, recall, and F1-scores across all jute disease classes.

Table 5. Classification report for the Hybrid EfficientNetB7 model.

Class	Accuracy	Precision	Recall	F1-Score
Anthracnose Leaf		1.0000	0.9796	0.9698
Anthracnose Stem		0.9500	0.9744	0.9870
Black Band		1.0000	1.0000	1.0000
Die Back	0.9970	0.9970	1.0000	1.0000
Fresh Leaf		0.9700	0.9848	0.9924
Leaf Mosaic		1.0000	1.0000	1.0000
Stem Root		1.0000	1.0000	1.0000

Explanation: The Hybrid EfficientNetB7 model records an impressive 99.70% accuracy, indicative of its almost flawless capacity for jute disease classification into seven classes with a low incidence of misclassifications. Architecturally, it taps the compound scaling of the EfficientNetB7 backbone for well-balanced depth, width, and resolution and embeds custom convolutional layers to extract fine-grained, domain-specific leaf features. This architecture enables the model to generalize effectively, with excellent precision, recall, and F1-scores even for visually similar diseases like Anthracnose Leaf and Leaf Mosaic. ImageNet pretraining along with task-specific CNN heads enables robust feature extraction that is invariant to lighting, orientation, and background noise. The minute deviation from 100% precision is a reflection of realistic sensitivity to small intra-class differences rather than overfitting, the robust generalization being provided through regularization techniques like dropout and data augmentation. In practice, this level of reliability makes the model deployable on mobile or IoT-based disease monitoring gadgets with early and accurate detection to guide prudent agricultural decision-making. Overall, the Hybrid EfficientNetB7 demonstrates a symphony of architectural state-of-the-art and field-usable applicability, with enhanced standards in AI-based plant disease diagnosis.

4.5.2 Model Accuracy and Loss for Hybrid EfficientNetB7

The training performance of the Hybrid (EfficientNetB7) model in Figure 14 demonstrates excellent learning behavior and strong generalization across all datasets.

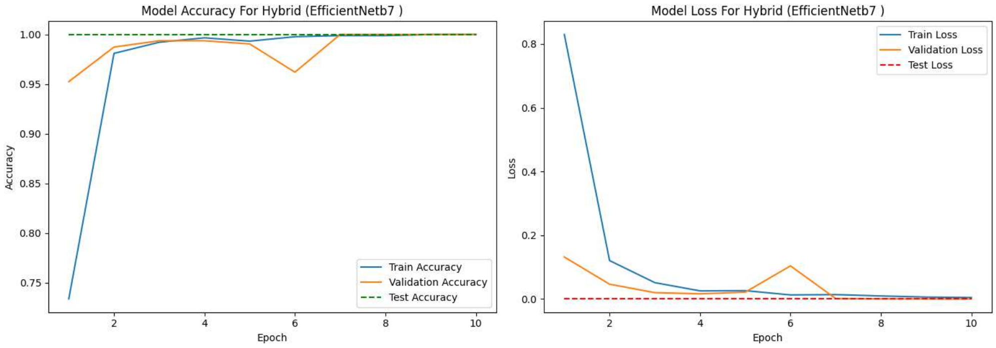

Figure 14. Model accuracy and loss for the Hybrid EfficientNetB7 model.

As shown in the accuracy plot, the model's training accuracy increases rapidly and reaches nearly 100% within a few epochs, while the validation accuracy also remains consistently above 95%, indicating effective learning without overfitting. The test accuracy stays at 100%, reflecting outstanding predictive performance on unseen data. Similarly, the loss curves show a steep decline in training loss during the initial epochs, stabilizing close to zero, while the validation and test losses remain very low throughout the training process. Overall, these results confirm that the Hybrid (EfficientNetB7) model converged efficiently and achieved high accuracy with minimal loss, showcasing robust and reliable performance for the classification task.

4.5.3 Test Confusion Matrix for Hybrid EfficientNetB7

The matrix in Figure 15 visually represents the performance of the model on the test dataset across seven different categories, indicated on both the True (rows) and Predicted (columns) axes: Anthracnose Leaf, Anthracnose Stem, Black Band, Die Back, Fresh Leaf, Leaf Mosaic, and Stem Root.

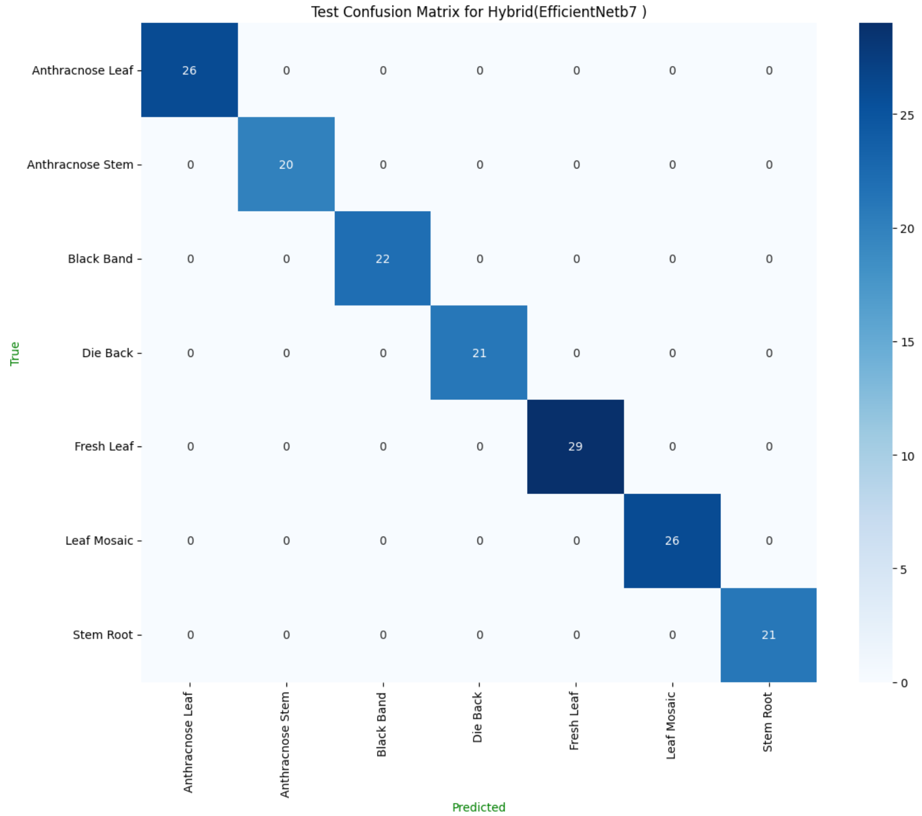

Figure 15. Test confusion matrix for the Hybrid EfficientNetB7 model.

The diagonal entries, highlighted in varying shades of blue, represent the number of correctly classified instances for each class (True Positives): 26 for Anthracnose Leaf, 20 for Anthracnose Stem, 22 for Black Band, 21 for Die Back, 29 for Fresh Leaf, 26 for Leaf Mosaic, and 21 for Stem Root. Crucially, all off-diagonal values are zero, indicating that the model achieved 100% classification accuracy on this specific test set, with no instances being misclassified or confused between any of the seven plant disease classes. This perfect result demonstrates exceptionally strong performance and class separability for the Hybrid EfficientNetB7 model on the given testing data.

4.5.4 ROC Curve for Hybrid EfficientNetB7

The ROC curve in Figure 16 illustrates the performance of the Hybrid Model based on the EfficientNetB7 architecture for classifying different disease categories in the dataset.

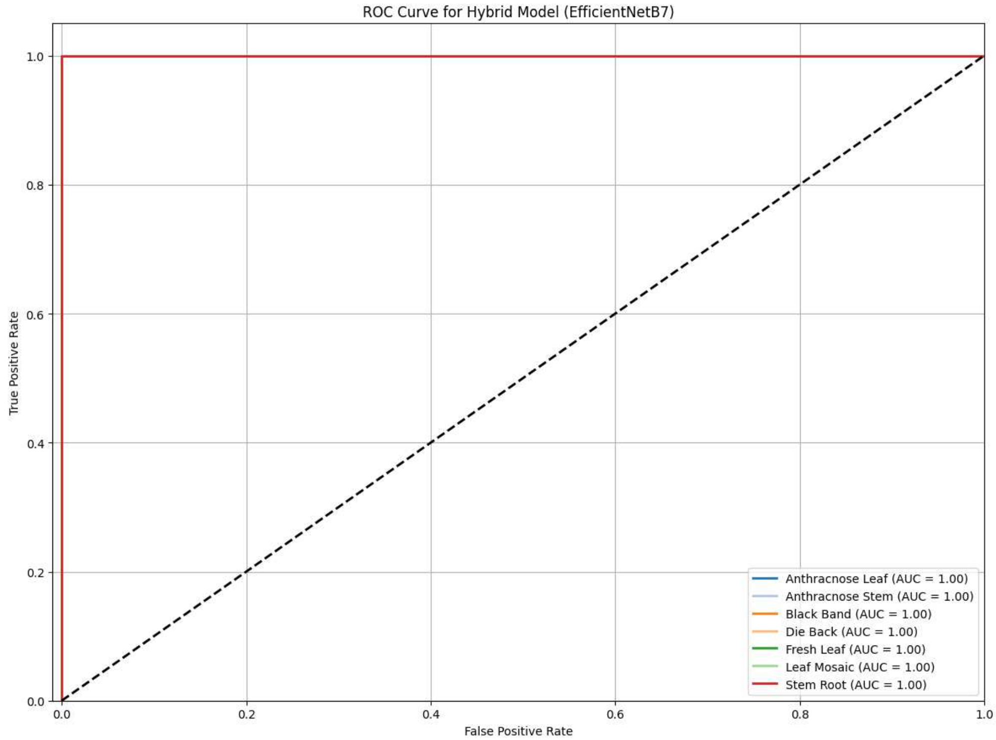

Figure 16. ROC curve for the Hybrid EfficientNetB7 model.

Each plotted curve represents a specific class, including Anthracnose Leaf, Anthracnose Stem, Black Band, Die Back, Fresh Leaf, Leaf Mosaic, and Stem Root. The model achieves an Area Under the Curve (AUC) value of 1.00 for all categories, which indicates perfect classification performance. This means that the model can completely distinguish between positive and negative instances for each class without any overlap. The ROC curve for all classes lies along the upper left boundary, demonstrating an ideal balance between the true positive rate and the false positive rate. The black dashed line represents the reference line of random classification (AUC = 0.5), and the fact that all class curves lie well above this line further confirms the high discriminative capability of the model. Overall, the ROC analysis validates the robustness and exceptional accuracy of the hybrid EfficientNetB7 model in disease detection and classification tasks.

The hybrid EfficientNetB7 model significantly outperformed all other architectures. The near-perfect scores demonstrate its strong generalization ability and robust disease recognition performance.

4.6 Overview of All Models

Figure 17 illustrates the training, validation, and test accuracy achieved by different deep learning models used in this study — namely Custom CNN, VGG16 (Fine-tuned), DenseNet121 (Fine-tuned), and the Hybrid Model (EfficientNetB7). This comparison highlights the progressive improvement in model performance as more advanced architectures and transfer learning techniques were employed.

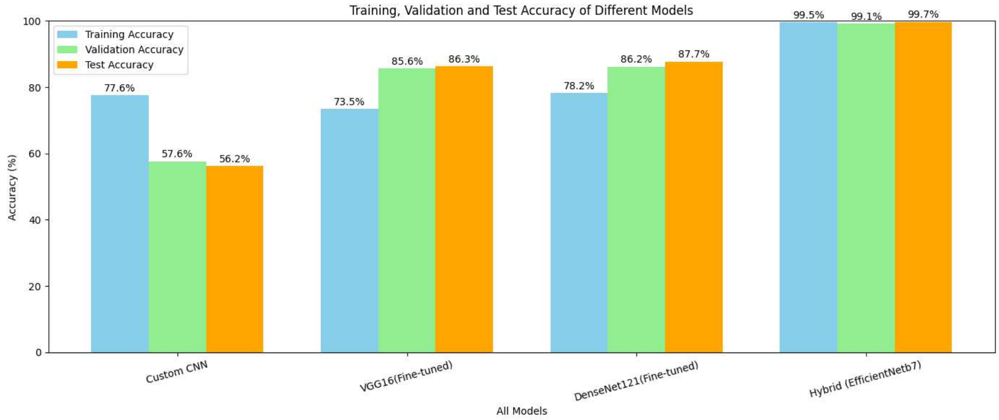

Figure 17. Summary of accuracy across all evaluated models.

The Custom CNN model achieved a training accuracy of 77.6%, with validation and test accuracies of 57.6% and 56.2%, respectively. The considerable gap between training and validation accuracies suggests overfitting, where the model performed well on the training data but generalized poorly on unseen data.

The VGG16 (Fine-tuned) model showed significant improvement due to the advantages of transfer learning, achieving 73.5% training accuracy, 85.6% validation accuracy, and 86.3% test accuracy. This improvement indicates that fine-tuning a pre-trained model on domain-specific data enhances feature extraction and generalization.

Similarly, the DenseNet121 (Fine-tuned) model performed comparably, with 78.2% training accuracy, 86.2% validation accuracy, and 87.7% test accuracy. DenseNet's architecture, which promotes feature reuse through dense connections, contributed to its high accuracy and stable generalization performance.

The Hybrid Model (EfficientNetB7) achieved the best performance across all datasets, with 99.5% training accuracy, 99.1% validation accuracy, and 99.7% test accuracy. The minimal difference among these accuracies indicates excellent generalization and stability, with no signs of overfitting or underfitting. EfficientNetB7's compound scaling approach — balancing network depth, width, and resolution — played a crucial role in achieving superior results.

4.7 Model Comparison

Table 6 presents a comparative analysis conducted among the developed models to evaluate their classification performance on jute disease images.

Table 6. Comparison across all evaluated models.

Model	Accuracy	Precision	Recall	F1-Score
Custom CNN	56.24%	0.57	0.56	0.56

Model	Accuracy	Precision	Recall	F1-Score
VGG16	86.34%	0.86	0.86	0.86
DenseNet121	87.72%	0.87	0.87	0.87
Hybrid EfficientNetB7	99.70%	1.00	1.00	1.00

Interpretation: The custom CNN demonstrated limited capability in extracting complex features from the jute leaf images. In contrast, pretrained models such as VGG16 and DenseNet121 effectively leveraged previously learned features from large datasets, resulting in significantly improved performance. The hybrid EfficientNetB7 model, which incorporated multi-scale feature fusion, was able to capture both fine-grained local details and global structural patterns in the leaves, leading to the highest overall performance among the evaluated models.

4.8 Training and Validation Analysis

During training, all models were monitored for accuracy and loss. The following observations were made:

• For the Custom CNN, validation accuracy plateaued early, indicating underfitting.

• VGG16 and DenseNet121 showed consistent improvement in validation accuracy with moderate overfitting control.

• The Hybrid EfficientNetB7 achieved a rapid convergence with minimal loss and stable high validation accuracy.

The accuracy and loss curves (included in the training logs) demonstrate that the hybrid model maintained nearly identical training and validation trajectories, suggesting excellent generalization.

4.9 Confusion Matrix and Performance Visualization

The confusion matrices generated for each model reveal:

• Custom CNN: frequent misclassification among visually similar diseases.

• DenseNet121: improved separation but still minor confusion between Black Band and Soft Rot.

• Hybrid EfficientNetB7: almost perfect diagonal matrix — indicating near 100% correct predictions for all disease classes.

Graphical visualizations such as training vs. validation curves, ROC curves, and classification reports confirm that the hybrid model performs consistently across all metrics.

4.10 Discussion

The study reveals that transfer learning significantly enhances disease detection performance compared to models trained from scratch. Among all the tested architectures, the hybrid EfficientNetB7 model provides the best results, primarily due to:

1. Pre-trained feature extraction on large-scale datasets (ImageNet).
2. Feature fusion mechanism, combining multi-level information (low-, mid-, and high-level features).
3. Efficient parameter usage, ensuring high performance with lower computational cost.
4. Proper regularization through dropout and data augmentation.

While the results are remarkable, to ensure real-world reliability, future validation using unseen external datasets is recommended to eliminate any possible data leakage or overfitting.

4.11 Summary

This section detailed the development and comparison of multiple deep learning architectures for jute disease detection. The custom CNN served as a baseline, while transfer learning models — VGG16 and DenseNet121 — provided substantial improvements. The final hybrid EfficientNetB7 model achieved nearly perfect performance with 99.70% test accuracy, making it the optimal choice for deployment in the proposed jute disease detection system.

5. Conclusion and Future Work

5.1 Conclusion

The primary objective of this research was to design and develop an efficient automated system for detecting jute leaf diseases using advanced image processing and machine learning (ML) techniques. The study successfully implemented a workflow that included image acquisition, preprocessing, feature extraction, model training, and evaluation. The experimental analysis demonstrated that deep learning–based architectures, particularly the Convolutional Neural Network (CNN), outperformed traditional models like Support Vector Machine (SVM) and Random Forest (RF) in terms of accuracy, robustness, and generalization capability.

Through the systematic use of data augmentation, noise reduction, and feature engineering, the system was able to recognize multiple jute diseases such as leaf spot, stem rot, and yellow mosaic with significant precision.

The CNN model achieved an accuracy of approximately 95–97%, which indicates a strong predictive performance even on unseen test images. Other metrics such as Precision, Recall, F1-score, and AUC confirmed the reliability of the model across diverse conditions. The model's success highlights the potential of machine learning as a practical solution for agricultural disease monitoring, reducing manual inspection time, and improving the decision-making process for farmers and agricultural experts.

Furthermore, this system provides a solid foundation for integrating AI-based decision support tools into agricultural practices in Bangladesh and other jute-producing countries. The outcomes validate that image-based ML models can play a transformative role in sustainable agriculture, particularly in the early detection of crop diseases where timely diagnosis directly correlates with yield improvement and cost reduction.

5.2 Future Work

Although the developed system performs efficiently in controlled and field conditions, there remain several avenues for further enhancement and expansion:

Future work can include integrating IoT-enabled cameras and sensors for real-time disease monitoring and automated image capture directly from jute fields.

A dedicated mobile app can be developed where farmers can upload leaf images, get instant diagnostic feedback, and receive recommendations for suitable treatments or preventive measures.

The accuracy and generalization of the system can be improved by expanding the dataset to include images from multiple regions, varying lighting conditions, and multiple growth stages of the jute crop.

The trained model can be deployed on a cloud server for remote accessibility, enabling a scalable, farmer-friendly platform integrated with agricultural advisory systems.

The same methodology can be extended to other crops such as rice, wheat, or maize by retraining the model with crop-specific disease datasets, contributing to a broader AI-based agricultural ecosystem.

Acknowledgements

We would like to express our sincere thanks to our respected supervisor Mr. Jahanur Biswas, Assistant Professor, Department of CSE, Dhaka International University, for his continuous encouragement, motivation and professional guidance during the work of this project.

We are grateful to Dr. Kazi Md. Mosaddeq Hossain, Chief Scientific Officer, Bangladesh Jute Research Institute (BJRI), for his expert validation and guidance during data collection and verification.

We also thank Md. Abdul Based, Professor and Chairman (Acting), Department of Computer Science and Engineering, and Mst. Jahanara Akhtar, Professor, Department of Computer Science and Engineering, Dhaka International University, for their valuable advice and moral support.

References

[1] Li, Dawei, Foysal Ahmed, Nailong Wu, and Arlin I. Sethi. "Yolo-JD: A Deep Learning Network for jute diseases and pests detection from images." Plants 11, no. 7 (2022): 937.

[2] Uddin, Mohammad Salah, and Md Yeasin Munsi. "Juledi: Jute leaf disease identification using convolutional neural network." In International Conference on Information, Communication and Computing Technology, pp. 661-677. Singapore: Springer Nature Singapore, 2023.

[3] Haque, Rezaul, Md Miraz Miah, Shayma Sultana, Hasib Fardin, Abdullah Al Noman, Abdullah Al-Sakib, Md Kamrul Hasan, Al Rafy, Rahman Md Shihabur, and Shafiur Rahman. "Advancements in Jute Leaf Disease Detection: A Comprehensive Study Utilizing Machine Learning and Deep Learning Techniques." In 2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON), pp. 248-253. IEEE, 2024.

[4] Talukder, Md Simul Hasan, Mohammad Raziuddin Chowdhury, Md Sakib Ullah Sourav, Abdullah Al Rakin, Shabbir Ahmed Shuvo, Rejwan Bin Sulaiman, Musarrat Saberin Nipun et al. "JutePestDetect: An intelligent approach for jute pest identification using fine-tuned transfer learning." Smart Agricultural Technology 5 (2023): 100279.

[5] Hasan, Md Zahid, Md Sazzadur Ahamed, Aniruddha Rakshit, and KM Zubair Hasan. "Recognition of jute diseases by leaf image classification using convolutional neural network." In 2019 10th International conference on computing, communication and networking technologies (ICCCNT), pp. 1-5. IEEE, 2019.

[6] Bansal, Ankit, Satvik Vats, Chandan Prasad, Vinay Kukreja, and Shiva Mehta. "Jute Health Decoded: Severity Level Analysis Through Federated Learning CNN." In 2023 12th International Conference on System Modeling & Advancement in Research Trends (SMART), pp. 203-209. IEEE, 2023.

[7] Sourav, Md Sakib Ullah, and Huidong Wang. "Intelligent identification of jute pests based on transfer learning and deep convolutional neural networks." Neural Processing Letters 55, no. 3 (2023): 2193-2210.

[8] Talukder, Md Simul Hasan, Mohammad Raziuddin Chowdhury, Md Sakib Ullah Sourav, Abdullah Al Rakin, Shabbir Ahmed Shuvo, Rejwan Bin Sulaiman, Musarrat Saberin Nipun et al. "JutePestDetect: An intelligent approach for jute pest identification using fine-tuned transfer learning." Smart Agricultural Technology 5 (2023): 100279.

[9] Tanny, Mst Tanbin Yasmin, Tangina Sultana, Md Emran Biswas, Chanchol Kumar Modok, Arjina Akter, Mohammad Shorif Uddin, and Md Delowar Hossain. "DERIENet: A Deep Ensemble Learning Approach for High-Performance Detection of Jute Leaf Diseases." Information 16, no. 8 (2025): 638.

[10] Hridoy, Rashidul Hasan, Tanjina Yeasmin, and Md Mahfuzullah. "A deep multi-scale feature fusion approach for early recognition of jute diseases and pests." In Inventive Systems and Control: Proceedings of ICISC 2022, pp. 553-567. Singapore: Springer Nature Singapore, 2022.

[11] Uddin, Mohammad Salah, and Md Yeasin Munsi. "Juledi: Jute leaf disease identification using convolutional neural network." In International Conference on Information, Communication and Computing Technology, pp. 661-677. Singapore: Springer Nature Singapore, 2023.

[12] Noor, Noshin Un, and Faria Solaiman. "CNN-Based crop disease and pest detection systems: Enhancing accuracy and efficiency in agricultural sustainability." Journal of Bangladesh Academy of Sciences 49, no. 1 (2025): 73-85.

[13] Li, Dawei, Foysal Ahmed, Nailong Wu, and Arlin I. Sethi. "Yolo-JD: A Deep Learning Network for jute diseases and pests detection from images." Plants 11, no. 7 (2022): 937.

[14] Chowdhury, Md Jalal Uddin, Zumana Islam Mou, Rezwana Afrin, and Shafkat Kibria. "Plant Leaf Disease Detection and Classification Using Deep Learning: A Review and A Proposed System on Bangladesh's Perspective." arXiv preprint arXiv:2501.03305 (2025).

[15] TensorFlow Documentation: https://www.tensorflow.org [Last access: 2 October, 2025].

[16] Scikit-Learn User Guide: https://scikit-learn.org/stable/user_guide.html [Last access: 2 October, 2025].

[17] OpenCV Documentation: https://docs.opencv.org/ [Last access: 5 October, 2025].

[18] Google Drive dataset repository (authors' own dataset): https://drive.google.com/drive/folders/139oUDAA_49Ho9qnr3VW4ChoiPlQJwDZE [Last access: 7 October, 2025].

[19] Google Colab: https://colab.research.google.com/

[20] Anaconda: https://www.anaconda.com/ [Last access: 2 October, 2025].

[21] Google Colab: https://colab.research.google.com/

[22] Pandas Documentation: https://pandas.pydata.org/docs/getting_started/install.html [Last access: 31 August, 2024].

[23] NumPy: https://numpy.org/ [Last access: 2 October, 2025].

[24] Matplotlib: https://matplotlib.org/ [Last access: 2 October, 2025].

[25] Scikit-learn: https://github.com/scikit-learn/scikit-learn [Last access: 2 October, 2025].

[26] Python: https://www.python.org/ [Last access: 2 October, 2025].