International Journal of Statistical Sciences *ISSN 1683-5603*
Vol. 25(2), November, 2025, pp 143-157
DOI: https://doi.org/10.3329/ijss.v25i2.85777
© 2025 Dept. of Statistics, Univ. of Rajshahi, Bangladesh

Detection of Mango Leaf Disease using Convolution Neural Network: A Comparative Analysis

Rasna Begum[1], Yasir Arafat[1], Md. Saifur Rahman[1] and Md. Kaderi Kibria[1*]

[1]Department of Statistics, Hajee Mohammad Danesh Science and Technology University, Dinajpur-5200, Bangladesh

[*]Correspondence should be addressed to Md. Kaderi Kibria
(Email: kibria.stt@tch.hstu.ac.bd)

[Received July 15, 2025; Accepted October 10, 2025]

Abstract

Mango leaf diseases pose significant challenges to the agricultural industry, requiring effective and efficient detection methods to ensure the health and productivity of mango crops. This study aims to perform a comparative analysis of various Convolutional Neural Network (CNN) architectures for the detection and classification of mango leaf diseases. The dataset consists of 4,000 images, with 500 images per class. Images were preprocessed through resizing and normalization to enhance model generalization. All images in the dataset are resized to a uniform size of 224 × 224 pixels and normalized to the range [0, 1] by dividing each pixel value by 255. The dataset was split into training, validation and testing sets in an 80:10:10 ratio. The performance of several CNN models was evaluated based on accuracy, computational efficiency, and generalization ability. The DenseNet201 model outperformed others, achieving the highest average accuracy (98.10%) and proved effective for automated disease detection in mango crops. Gall Midge was identified as the most difficult disease to classify with average accuracy of 93.41%, while Cutting Weevil and Bacterial Canker showed the highest classification accuracy, 99.51% and 99.06% respectively. Computational efficiency was a key consideration, as some models, like ResNet152V2, while accurate, required substantial computational resources, limiting their feasibility in low-resource environments. This research contributes to the advancement of automated disease detection in agriculture, offering a pathway to more efficient mango leaf disease management systems.

Keywords: CNN, Mango Leaf Disease Detection, DenseNet201, Computational Efficiency, Agricultural Automation.

AMS Classification: 62P10, 68T07, 68T09.

1. Introduction

Mango (*Mangifera indica*) is a commercially significant fruit crop cultivated extensively across tropical and subtropical regions with major producers including India, Thailand, Brazil, and Mexico [1]. Beyond its global economic value, mango farming is a key source of income for millions. However, productivity is increasingly compromised by leaf diseases such as anthracnose, bacterial canker, dieback and infestations from pests like aphids and gall midges [2]. These problems reduce photosynthetic capacity, affect fruit quality and lead to substantial economic losses [3]. Traditional disease management dependent on chemical treatments and manual inspection is labor-intensive, costly and often inaccessible to smallholders [4].

Deep learning, particularly CNNs, offers an effective alternative for plant disease identification through image-based classification. Mohanty et al. (2016) and Ferentinos (2018) demonstrated high accuracy (over 99%) using models like GoogLeNet and VGGNet on large agricultural datasets [5,6]. Enhancements such as data augmentation (e.g., CutMix, MixUp) have further improved model generalization, as seen in works by Yadav et al. (2022) and Pandian et al. (2019) [7,8]. Transfer learning has also proven valuable, Sagar and Jacob (2021) and Sholihati et al. (2020) reported strong performance using pre-trained models like ResNet50 and VGG16 [9,10]. EfficientNet (Genaev et al., 2021) and custom architectures like PlantDiseaseNet (Turkoglu et al., 2022) continue to advance classification performance, even in multi-disease contexts [11,12]. Real-time applications, including robotic systems (Durmus et al., 2017) and hybrid models with ensemble techniques (Adekunle, 2019), further highlight the practicality and scalability of CNN-based approaches [13,14]. While much progress has been made in crops such as tomato, potato, apple and rice. However, studies focused specifically on mango remain limited. In this study, we evaluated several CNN architectures for mango leaf disease classification to identify the most accurate and computationally efficient models. It also explores classification difficulty across disease types to guide the development of robust and scalable disease management tools for mango cultivation.

2. Methodology

2.1 Data Source, Collection and Pre-processing

This study utilized the MangoLeafBD dataset, sourced from the Mendeley repository [https://data.mendeley.com/datasets/hxsnvwty3r/1], comprising 4,000 images evenly distributed across eight classes seven mango leaf diseases and one healthy class [15]. Each class contains 500 images, ensuring dataset balance. Images varied in size and format, necessitating preprocessing to standardize dimensions and enhanced model performance during training and evaluation. The dataset underwent preprocessing to enhance model performance and ensure compatibility with CNN architectures. All images were resized to 224 × 224 pixels to standardize input dimensions, a common requirement for models like VGG16 and ResNet [16,17]. All images were normalized by scaling pixel values to the [0, 1] range. This standardization ensures stable and efficient training of the CNN models [18].

2.2 Data Partition

The dataset was divided into training, validation and testing sets using an 80:10:10 ratio. Stratified sampling was employed to maintain equal class representation in each subset. The training set was used to optimize model weights, the validation set monitored model performance and guided hyperparameter tuning and the testing set evaluated final model performance on unseen data.

2.3 Model Selection and Training

This study evaluates six CNN architectures such as Xception, VGG16, ResNet152V2, InceptionResNetV2, DenseNet201 and MobileNetV2 selected for their established performance in image classification tasks. Each model contributes unique architectural features and computational strategies to improve classification accuracy and efficiency in detecting mango leaf diseases [18]. The Adam optimizer was employed to train all CNN models with an initial learning rate of 0.0001 which is ensuring gradual and stable weight updates. This optimizer adapts the learning rate for each parameter individually to prevent drastic changes that could destabilize training. All models were trained for 5 epochs. This number was sufficient to learn essential patterns in the dataset while minimizing the risk of overfitting. All models were initialized with weights pre-trained on the ImageNet dataset. Transfer learning was applied by freezing the base layers initially and then fine-tuning the deeper layers to adapt the models to the mango leaf disease dataset.

2.3.1 Xception

Xception employs depthwise separable convolutions, which decompose standard convolutions into depthwise and pointwise operations, reducing parameters and improving efficiency [19].
Depthwise convolution:

$$Y_{depthwise} = X * K_{depthwise}$$

Pointwise convolution:

$$Y_{pointwise} = Y_{depthwise} * K_{pointwise}$$

It also uses residual connections to ease gradient flow in deep networks:

$$Y = F(X) + X$$

2.3.2 VGG16

VGG16 uses a uniform architecture with small 3×3 convolutional kernels and max pooling layers [16]. The basic convolution operation is:

$$Y = f(X * W + b)$$

where, X is the input feature map, W is the weight (filter), b is the bias term and f is typically ReLU:

$$ReLU(Y) = \max(0, Y)$$

This configuration allows for deep feature extraction and is widely compatible with transfer learning applications.

2.3.3. ResNet152V2

ResNet152V2 utilizes residual learning to mitigate vanishing gradient issues in deep networks [20].
Residual block:

$$Y = F(X, \{W_i\}) + X$$

where, X is the input to the block, $F(X, \{W_i\})$ is the transformation applied by the convolutional layers and Y is the output after adding the residual connection.

Batch normalization: It also integrates batch normalization to stabilize and accelerate training

$$\hat{X} = \frac{X - \mu}{\sigma} \times \gamma + \beta$$

where, X is the input to the batch normalization layer, μ and σ are the mean and variance of the batch, γ and β are learnable scale and shift parameters.

2.3.4 InceptionResNetV2

InceptionResNetV2 is a combination of the Inception and ResNet architectures, where the inception modules are combined with residual connections [21].

Inception Modules: These modules have parallel convolutional operations with different kernel sizes ($1 \times 1,\ 3 \times 3,\ 5 \times 5$) and pooling operations where the outputs are concatenated together.

$$Y = [Conv_1(X),\ Conv_3(X), Conv_5(X), MaxPool(X)]$$

Residual connections:

$$Y_{final} = F(X) + X$$

where F(X) is the transformed input (as in standard CNN layers), and Y_{final} is the output after the residual connection.

2.3.5 DenseNet201

DenseNet is a deep convolutional neural network where each layer is connected to every other layer in a feed-forward fashion. DenseNet201 refers to the version with 201 layers. It is composed of multiple dense blocks, where each layer within a block receives the concatenated output of all previous layers [22].

$$Y_l = H_l([X_0, X_1, \dots., X_{l-1}])$$

where, Y_l is the output of the l^{th} layer, H_l is the transformation applied by the layer and $[X_0, X_1, \dots., X_{l-1}]$ represents the concatenation of all previous layers' feature maps and, which typically consists of a convolution operation and an activation function.

To manage complexity, bottleneck layers reduce dimensionality:

1×1 bottleneck convolution:

$$X_{bottleneck} = Conv_1(X)$$

where X is the input feature map, $Conv_1$ represents the 1×1 convolution, which reduces the number of channels, $X_{bottleneck}$ is the output after the 1x1 convolution.

3×3 convolution**:**

$$X_{out} = Conv_3(X_{bottleneck})$$

where, $Conv_3$ represents the 3x3 convolution applied after the bottleneck, X_{out} is the output after the 3x3 convolution, which is then concatenated with other feature maps in the dense block.

2.3.6 MobileNetV2

Designed for efficiency, MobileNetV2 uses inverted residuals and linear bottlenecks with depthwise separable convolutions [23].

Depthwise + Pointwise convolution:

$$Y_{inverted} = Conv_1(DepthwiseConv(X))$$

where X is the input feature map, and $Y_{inverted}$ is the output after passing through the sequence of convolutions.

For activation, it uses ReLU6, a capped version of ReLU:

$$RelU6 = \min(\max(0, x), 6)$$

This design achieves a favorable trade-off between speed and accuracy, particularly suitable for deployment in mobile or low-resource environments.

2.4 Model Training Strategy

The training approach integrates transfer learning, custom classifier design, and hyperparameter optimization to build effective deep learning models for image classification.

1. Transfer Learning

Transfer learning uses models pre-trained on large datasets (e.g., ImageNet) to reduce training time and improve performance on smaller, domain-specific datasets.

Let:

f_θ: pre-trained model with parameters θ

D_tt: target dataset.

Fine-tuning optimizes:

$$\theta^* = \arg min_\theta \, L(f_\theta(D_t))$$

Strategies:

Feature extraction: Freeze CNN layers and train a new classifier on $h = f_\theta(x)$, then predict:

$$y = g_\phi(h)$$

Fine-tuning: Unfreeze deeper layers and update parameters:

$$\theta' = \theta - \eta \nabla_\theta L(f_\theta, g_\phi, D_t)$$

2. Custom Classifier

A dense layer and softmax output were added for classification:

Fully connected layer:

$$y = \sigma(Wx + b)$$

Softmax output (for class probabilities):

$$P(y = i|X) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

3. Training Configuration

Optimizer: Adam **(Adaptive Moment Estimation)** was used as optimizer which adapts the learning rate for each parameter individually, helping prevent drastic weight updates that could destabilize the learning process. The learning rate is typically initialized to a small value, such as η=0.0001, to ensure gradual and stable weight updates.

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{v}_t}{\sqrt{\hat{s}_t + \epsilon}}$$

Loss Function: Categorical cross-entropy:

$$Loss = -\sum_{i=1}^{n} y_i log(\hat{y}_i)$$

where, y_i is the actual label (one-hot encoded) and $\hat{y}_i$ is the predicted probability. The model aims to minimize this loss by adjusting weights during training, thereby improving the accuracy of class predictions. Batch size refers to the number of samples processed before the model updates its parameters. A batch size of 32 is chosen as a balance between computational efficiency and stable training. An epoch represents one complete pass through the entire dataset during training. The number of epochs was selected as 5 that provides enough iterations for the model to learn essential patterns in the data while preventing overfitting.

2.5 Evaluation Metrics

To assess the performance of classification models, key evaluation metrics such as accuracy, precision**,** recall**,** F1-score, and AUC are used. These metrics help evaluate model correctness, sensitivity, and discriminative power, especially in multi-class settings.

Accuracy: Measures the proportion of correct predictions out of all predictions:

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

Precision: Indicates the proportion of correctly predicted positive samples among all predicted positives:

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

Recall (Sensitivity): Represents the proportion of actual positives correctly identified:

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

F1-Score: The harmonic mean of precision and recall, useful for imbalanced class distributions:

$$\text{F1} - \text{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

AUC (Area Under the Curve): Measures the area under the ROC curve, which plots TPR vs. FPR across thresholds. A higher AUC indicates better class separation ability:

$$AUC = \int_0^1 True\ Positive\ Rate\ (TPR)\ \ d(False\ Positive\ Rate\ (FPR))$$

Where, True Positive Rate (TPR) $= Recall = \frac{TP}{TP+FN}$, False Positive Rate (FPR) $= \frac{FP}{FP+TN}$.

3. Results

3.1 Class-Specific Accuracy

The performance comparison of six CNN models in classifying anthracnose-infected and healthy mango leaves reveals high overall accuracies across all models (see Table 1). Among them, MobileNetV2 achieved the highest accuracy, recording 99.00% across training, validation and test sets, indicating its superior generalization capability and robustness in distinguishing between diseased and healthy leaves. ResNet152V2 followed closely with 98.20% training, 98.00% validation and 98.00% test accuracy, demonstrating consistent and reliable performance. Similarly, DenseNet201 and Xception also performed well, with test accuracy of 97.50% and 97.00%, respectively. InceptionResNetV2 yielded 96.50%, suggesting a slightly lower but still strong classification ability. On the other hand, VGG16 exhibited the lowest performance among the evaluated models, with an accuracy of 91.50% on both validation and test sets.

Table 1: Comparison of training, validation and test accuracies on Anthracnose vs. healthy leaves

Model	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
Xception	97.82	97.00	97.00
VGG16	91.51	91.50	91.50
ResNet152V2	98.20	98.00	98.00
InceptionResNetV2	96.67	96.50	96.50
DenseNet201	97.50	97.50	97.50
MobileNetV2	99.00	99.00	99.00

In the classification task distinguishing bacterial canker-infected mango leaves from healthy ones, all six CNN models demonstrated high accuracy indicating their strong discriminatory capabilities (see Table 2). Among them, MobileNetV2 achieved the highest overall performance, with 99.74% training, 99.69% validation, and 99.65% test accuracy, highlighting its robustness and excellent generalization. InceptionResNetV2 and ResNet152V2 also performed exceptionally well with test accuracies of 99.40% and 99.39%, respectively, reflecting their suitability for high-precision leaf disease detection. Xception followed closely with 99.23% test accuracy, while DenseNet201 achieved a slightly lower performance with 98.23% on the test set, showing a noticeable drop between training and validation/test accuracy. VGG16 had the lowest accuracy among the models with 97.50% on both validation and test sets, though still indicating strong classification ability. These results confirm the effectiveness of deep CNN models particularly MobileNetV2 in accurately identifying bacterial canker in mango leaves.

Table 2: Comparison of training, validation, and test accuracies on Bacterial Canker vs. healthy leaves

Model	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
Xception	99.32	99.24	99.23
VGG16	97.96	97.50	97.50
ResNet152V2	99.43	99.38	99.39
InceptionResNetV2	99.49	99.41	99.40
DenseNet201	99.25	98.24	98.23
MobileNetV2	99.74	99.69	99.65

In the task of differentiating cutting weevil-infected mango leaves from healthy ones, all six CNN models demonstrated remarkably high performance, with test accuracy exceeding 99% in most cases (see Table 3). MobileNetV2 once again emerged as the top-performing model, achieving 99.92% training, 99.90% validation, and 99.90% test accuracy, underscoring its exceptional capability to generalize across datasets. DenseNet201 followed closely, with a near-perfect 99.84% test accuracy, indicating robust performance and minimal overfitting. InceptionResNetV2 and ResNet152V2 also exhibited strong results, with 99.70% and 99.42% test accuracies, respectively. Xception and VGG16 achieved slightly lower yet highly competitive test accuracy of 99.20% and 99.04%, respectively. The consistently high accuracies across all models suggest that the visual patterns of cutting weevil infection are highly distinguishable from healthy leaves, and modern CNN architectures are well-equipped to detect them with excellent precision.

Table 3: Comparison of training, validation, and test accuracies on Cutting Weevil vs. healthy leaves

Model	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
Xception	99.24	99.22	99.20
VGG16	99.07	99.05	99.04
ResNet152V2	99.48	99.43	99.42
InceptionResNetV2	99.73	99.71	99.70
DenseNet201	99.86	99.85	99.84
MobileNetV2	99.92	99.90	99.90

For the classification of dieback-infected versus healthy mango leaves, all six CNN models demonstrated strong performance, though with slight variation in generalization capabilities (see Table 4). DenseNet201 achieved the highest accuracy, with 99.68% training, 99.63% validation, and 99.63% test accuracy, indicating both excellent learning and consistent generalization. InceptionResNetV2 and ResNet152V2 also performed reliably, with test accuracies of 99.50% and 99.37%, respectively, reinforcing their effectiveness in fine-grained leaf disease classification. Xception yielded 98.95% test accuracy, reflecting robust performance, while MobileNetV2 scored slightly lower at 98.50%, though still maintaining high accuracy. VGG16, while effective, again demonstrated comparatively lower performance, with 96.50% accuracy on both validation and test sets. These results suggest that deeper and more complex CNN architectures such as DenseNet201 and InceptionResNetV2 are particularly well-suited for detecting dieback symptoms, offering high precision in distinguishing diseases from healthy leaves.

Table 4: Comparison of training, validation, and test accuracies on Dieback Disease vs. healthy leaves

Model	**Training Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**
Xception	99.13	98.97	98.95
VGG16	97.05	96.50	96.50
ResNet152V2	99.43	99.38	99.37
InceptionResNetV2	99.54	99.50	99.50
DenseNet201	99.68	99.63	99.63
MobileNetV2	98.76	98.50	98.50

In the classification of gall midge-infected versus healthy mango leaves, the performance of the CNN models varied more noticeably compared to other disease categories (see Table 5). Xception delivered the highest performance, with 98.70% training and 98.50% accuracy on both validation and test sets indicating strong learning and generalization. ResNet152V2, DenseNet201, and MobileNetV2 also showed solid performance, with test accuracies of 96.50%, 96.00%, and 96.50%, respectively, suggesting their effectiveness in capturing disease-specific features despite slight drops in validation accuracy. However, InceptionResNetV2 exhibited a significant performance decline, with a test accuracy of 91.50% while VGG16 recorded the lowest performance across all models at 81.50%, suggesting that this architecture may not effectively capture the visual complexity of gall midge symptoms. The relatively lower performance of some models on this disease class may indicate that gall midge symptoms are visually less distinct or more challenging to differentiate from healthy leaves, thereby requiring more sophisticated feature extraction capabilities.

Table 5: Comparison of training, validation, and test accuracies on Gall Midge vs. healthy leaves

Model	**Training Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**
Xception	98.70	98.50	98.50
VGG16	81.68	81.50	81.50
ResNet152V2	97.32	96.50	96.50
InceptionResNetV2	94.36	91.50	91.50
DenseNet201	97.69	96.00	96.00
MobileNetV2	96.60	96.50	96.50

In distinguishing powdery mildew-infected mango leaves from healthy ones, all CNN models achieved high levels of accuracy, with particularly strong performance from the more advanced architectures (see Table 6). MobileNetV2 achieved the best overall results, with 99.57% training, 99.49% validation, and 99.50% test accuracy, demonstrating outstanding generalization and robustness. Xception, ResNet152V2, InceptionResNetV2 and DenseNet201 all achieved identical or nearly identical accuracies, with 98.50% on both validation and test sets, indicating reliable and consistent performance across these deeper architectures. VGG16, while still effective, lagged behind the other models, reaching 94.00% accuracy on both validation and test sets. This relatively lower performance suggests that simpler models like VGG16 may be less capable of capturing the fine-grained features required to accurately detect powdery mildew symptoms.

Table 6: Comparison of training, validation, and test accuracy on Powdery Mildew vs. healthy leaves

Model	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
Xception	98.93	98.50	98.50
VGG16	94.66	94.00	94.00
ResNet152V2	98.93	98.50	98.50
InceptionResNetV2	98.93	98.50	98.50
DenseNet201	98.89	98.50	98.50
MobileNetV2	99.57	99.49	99.50

In the classification of honeydew-infected mango leaves versus healthy leaves, the CNN models demonstrated high but slightly varied performance levels (see Table 7). Xception achieved the highest overall accuracy, with 97.85% training and 97.50% accuracy on both validation and test sets, indicating strong generalization and effective feature learning. MobileNetV2 also performed competitively, attaining 97.00% accuracy on both validation and test sets, confirming its consistency and reliability. ResNet152V2 and DenseNet201 followed closely with test accuracies of 96.50% and 96.00%, respectively, suggesting solid but slightly lower generalization compared to the top models. InceptionResNetV2 showed moderate performance, with a 95.00% test accuracy, while VGG16 recorded the lowest results among all models, achieving only 90.00% on the validation and test sets. This consistent trend across evaluations highlights the superior effectiveness of deeper and more modern architectures such as Xception and MobileNetV2 for detecting honeydew symptoms, whereas VGG16 may lack the representational power required to capture subtle distinctions in leaf disease features.

Table 7: Comparison of training, validation, and test accuracies on Honeydew vs. healthy leaves

Model	Training Accuracy (%)	Validation Accuracy (%)	Test Accuracy (%)
Xception	97.85	97.50	97.50
VGG16	90.65	90.00	90.00
ResNet152V2	97.00	96.50	96.50
InceptionResNetV2	96.42	95.00	95.00
DenseNet201	97.00	96.00	96.00
MobileNetV2	97.62	97.00	97.00

3.2 Overall Accuracy

Table 8 summarizes the overall training and test accuracies alongside the highest and lowest F1-scores achieved by six CNN architectures across different mango leaf disease classifications. All models demonstrated strong training accuracies above 97%, with MobileNetV2 and DenseNet201 leading at 99.86% and 99.82% respectively, indicating excellent learning capabilities. Correspondingly, these two models also achieved the highest test accuracy of 99.25%, highlighting their superior generalization performance on unseen data.

Table 8: Comparison of Training, Test Accuracy, Highest and Lowest F1-scores for Different CNN Architectures

Model	Training Accuracy (%)	Test Accuracy (%)	Highest F1-score	Lowest F1-score
Xception	98.89	98.00	1.00 (Canker)	0.93 (Honeydew)
VGG16	97.77	96.00	1.00 (Canker)	0.85 (Die Back)
ResNet152V2	98.46	98.00	1.00 (Canker)	0.93 (Honeydew)
InceptionResNetV2	98.73	97.50	1.00 (Canker)	0.93 (Healthy)
DenseNet201	99.82	99.25	1.00 (Multiple)	0.96 (Gall Midge)
MobileNetV2	99.86	99.25	1.00 (Multiple)	0.96 (Mildew)

In terms of *F*1-scores, which balance precision and recall, all models reached a perfect score of 1.00 for detecting bacterial canker, with DenseNet201 and MobileNetV2 achieving this peak for multiple disease classes, reflecting their robustness in accurately identifying various leaf diseases. The lowest F1-scores varied by model and disease, with VGG16 showing the largest drop at 0.85 for dieback disease, suggesting limited effectiveness in detecting this condition. Other models maintained relatively high minimum F1-scores, with the lowest being 0.93 for Xception, ResNet152V2, and InceptionResNetV2 (notably for honeydew and healthy leaf classifications), and 0.96 for DenseNet201 (gall midge) and MobileNetV2 (powdery mildew), indicating consistent and reliable performance across disease categories. Overall, these results emphasize that DenseNet201 and MobileNetV2 not only provide the best overall accuracies but also maintain strong, balanced classification performance across diverse mango leaf diseases, outperforming older architectures like VGG16.

3.3 ROC Curve Analysis

To comprehensively assess model performance, Receiver Operating Characteristic (ROC) curves and their corresponding Area Under the Curve (AUC) values were examined. The AUC metric evaluates the balance between sensitivity (true positive rate) and specificity (true negative rate), offering a nuanced measure of each model’s ability to discriminate between diseased and healthy leaves. Among the evaluated architectures, DenseNet201 and MobileNetV2 demonstrated the highest AUC values, further confirming their superior classification capabilities. The consistently elevated AUC scores observed across all models reflect strong discriminative power, even when differentiating cases with subtle or overlapping disease features. This analysis highlights the robustness and reliability of the CNN models in accurately identifying diverse mango leaf diseases, supporting their potential application in real-world disease detection scenarios (see Figure 1).

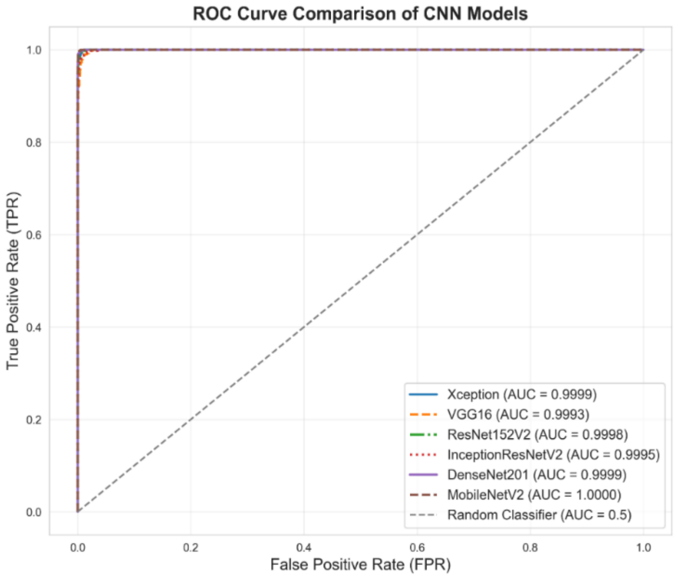

Figure 1: Comparing ROC curves with AUC values for all models

4. Discussion

This study aimed to perform a comparative analysis of various CNN architectures for the detection and classification of mango leaf diseases. The results revealed that DenseNet201 and MobileNetV2 were the most effective CNN architectures, with DenseNet201 achieving the highest accuracy and MobileNetV2 providing an optimal balance between performance and efficiency [24]. MobileNetV2 demonstrated exceptional generalization across most categories, achieving almost 100.00% accuracy in multiple cases, outperforming ResNet and VGG16, which suggests that its lightweight structure and efficient feature extraction make it particularly suitable for this task. Similar findings were reported by Prabu & Chelliah (2022) who found MobileNetV2 model to be highly effective for mango leaf disease detection [25]. This result does not align with Sagar & Jacob (2021) who found ResNet50 outperforms InceptionV3, MobileNet and DenseNet169, achieving accuracy of 98.20% [9]. DenseNet201 consistently ranked highest in overall accuracy, likely due to its densely connected layers that facilitate effective feature propagation [26,27]. ResNet152V2 and Xception also exhibited strong classification performance but required significantly higher computational resources [27]. In contrast, VGG16 showed the lowest performance particularly struggling to distinguish Gall Midge from Healthy samples where it achieved only 81.50% test accuracy [28]. This finding aligns with prior research, such as the study Saleem et al. (2020) which found that VGG16 exhibited lower validation accuracy and higher validation loss compared to other CNN architectures, making it less suitable for plant disease classification [29]. Similarly, a research Wang et al. (2023) found that VGG16 performed the worst among several tested models demonstrating lower precision, recall and F1-score while also converging at a slower rate [30]. These studies suggest that deeper networks with residual or dense connections such as DenseNet201 and ResNet, tend to outperform traditional architectures like VGG16 in plant disease classification [31–34]. In the classification of mango leaf diseases, certain diseases are more challenging to classify due to their visual similarities with healthy leaves or other diseases while others are easier based on their distinct features. Gall Midge, for instance, proved to be the most challenging disease to classify [35]. The physical damage caused by Gall Midge such as small irregular spots and deformations may not always be distinct enough for a model to differentiate [36]. This can lead to lower classification accuracy when distinguishing it from healthy samples or other diseases with similar features.

Disease-specific variation was also evident. Cutting Weevil and Bacterial Canker achieved high classification accuracy due to their distinct and visible symptoms, while Gall Midge remained the most challenging to classify because its subtle symptoms often resemble healthy leaves or less severe conditions. This observation reinforces the importance of symptom clarity in deep learning–based disease detection and suggests that subtle diseases may require higher-resolution images, larger datasets, or hybrid approaches that combine visual and non-visual data for improved recognition. Despite the promising results, several limitations should be acknowledged. The dataset consisted of 4,000 images collected under controlled conditions, which may not fully represent the diversity of field environments. Factors such as lighting variation, background noise, leaf orientation, and overlapping symptoms were not addressed, potentially limiting generalizability. In addition, the study relied solely on image-based data, without incorporating other modalities such as temporal or multispectral information, which could improve robustness. In terms of real-world applicability, MobileNetV2 stands out as a practical solution for deployment on mobile devices or edge systems due to its lightweight architecture and reduced computational requirements [25]. Such models are particularly relevant in low-resource agricultural settings where access to high-end hardware is limited. However, further validation using larger, heterogeneous, and field-collected datasets is essential before large-scale implementation. Practical deployment will also require attention to model interpretability, resilience to environmental variability, and integration into user-friendly tools for farmers.

5. Conclusion

This study evaluated the potential of CNN architectures for accurate and efficient detection of mango leaf diseases. DenseNet201 achieved the highest accuracy (98.10%) and proved to be the most reliable for automated classification. MobileNetV2 showed a good balance between accuracy and computational efficiency and is suitable for deployment in low-resource environments. Disease-specific performance varied as Cutting Weevil and Bacterial Canker were classified with the highest accuracy, while Gall Midge remained the most difficult to identify. These findings validate the effectiveness of CNNs in agricultural disease detection and highlight the need to align model capabilities with practical field conditions. This research provides a pathway for scalable and real-time monitoring that supports sustainable and technology-driven crop management.

Abbreviations list

CNN: Convolutional Neural Network, ROC: Receiver Operating Characteristic, AUC: Area Under the Curve, Xception: Extreme Inception, VGG16: Visual Geometry Group 16-layer network, ResNet152V2: Residual Network 152-layer Version 2, InceptionResNetV2: Inception-ResNet Version 2, DenseNet201: Densely Connected Convolutional Network with 201 layers, MobileNetV2: Mobile Network Version 2.

Authors' contributions: Rasna Begum was responsible for data collection and preprocessing. Yasir Arafat conducted the model implementation and analysis. Md. Saifur Rahman performed the statistical evaluation and interpretation of results. Md. Kaderi Kibria conceived and designed the study, supervised the research process, and prepared the manuscript. All authors reviewed and approved the final manuscript.

Acknowledgement: We would like to acknowledge the Department of Statistics, Hajee Mohammad Danesh Science and Technology University, Dinajpur, for their unconditional support. The author acknowledges the reviewer's contribution in enhancing the clarity and accuracy of the manuscript.

Data Availability Statement: The data used in this study are available within the manuscript.

Conflict of Interest: The authors declare no conflict of interest related to this study.

Funding: This study did not receive any specific funding.

References

[1] FAO (2023). Major Tropical Fruits Market Review. 2023, 1–33.

2] Malik, M. T., Ammar, M. S., Ullah, H., Mohar, T. A. (2020). Mango Diseases and Their Management. 2020, 9.

[3] Mugure, C, M. (2012). Economic Assessment of Losses Due to Fruit Fly Infestation in Mango and the Willingness to Pay for an Integrated Pest Management Package in Embu District, Kenya. עלון הנוטע 2012, 1–82.

[4] Ploetz, R. C. (2018). Integrated Disease Management in Mango Cultivation. 2018, 459–510, doi:10.19103/as.2017.0026.19.

[5] Mohanty, S. P., Hughes, D. P., Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Front. Plant Sci. 2016, 7, doi:10.3389/fpls.2016.01419.

[6] Konstantinos P. Ferentinos (2018). Deep Learning Models for Plant Disease Detection and Diagnosis. Comput. Electron. Agric. 2018.

[7] Yadav, A., Thakur, U., Saxena, R., Pal, V., Bhateja, V., Lin, J. C. W. (2022). AFD-Net: Apple Foliar Disease Multi Classification Using Deep Learning on Plant Pathology Dataset. Plant Soil 2022, 477, 595–611, doi:10.1007/s11104-022-05407-3.

[8] Arun Pandian, J., Geetharamani, G., Annette, B. (2029). Data Augmentation on Plant Leaf Disease Image Dataset Using Image Manipulation and Deep Learning Techniques. Proc. 2019 IEEE 9th Int. Conf. Adv. Comput. IACC 2019, 199–204, doi:10.1109/IACC48062.2019.8971580.

[9] Sagar, A., Jacob, D. (2020). On Using Transfer Learning For Plant Disease Detection. bioRxiv 2021, 2020.05.22.110957.

[10] Sholihati, R. A., Sulistijono, I. A., Risnumawan, A., Kusumawati, E. (2020). Potato Leaf Disease Classification Using Deep Learning Approach. IES 2020 - Int. Electron. Symp. Role Auton. Intell. Syst. Hum. Life Comf. 2020, 392–397, doi:10.1109/IES50839.2020.9231784.

[11] Genaev, M. A., Skolotneva, E. S., Gultyaeva, E. I., Orlova, E. A., Bechtold, N. P., Afonnikov,

D. A. (2021). Image-Based Wheat Fungi Diseases Identification by Deep Learning. Plants 2021, 10, doi:10.3390/plants10081500.

[12] Turkoglu, M., Yanikoğlu, B., Hanbay, D. (2022). Plant DiseaseNet: Convolutional Neural Network Ensemble for Plant Disease and Pest Detection. Signal, Image Video Process. 2022, 16, 301–309, doi:10.1007/s11760-021-01909-2.

[13] Durmus, H., Gunes, E. O., Kirci, M. (2017). Disease Detection on the Leaves of the Tomato Plants by Using Deep Learning. 2017 6th Int. Conf. Agro-Geoinformatics, Agro-Geoinformatics 2017 2017, doi:10.1109/Agro-Geoinformatics.2017.8047016.

[14] Adekunle, I. M. (2019). Implementation of Improved Machine Learning Techniques for Plant Disease Detection and Classification. IEEE Access 2019, 2018-January, 1–4.

[15] Ali, S., Ibrahim, M., Ahmed, S. I., Md. Nadim, Mizanur, M. R., Shejunti, M. M., Jabid, T. (2022). MangoLeaf BD Dataset. Mendeley Data 2022.

[16] Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. 2015.

[17] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016, 2016-December, 770–778, doi:10.1109/CVPR.2016.90.

[18] Sane, P., Agrawal, R. Pixel (2017). Normalization from Numeric Data as Input to Neural Networks: For Machine Learning and Image Processing. Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017 2017, 2018-January, 2221–2225, doi:10.1109/WiSPNET.2017.8300154.

[19] Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017, 2017-January, 1800–1807, doi:10.1109/CVPR.2017.195.

[20] He K, Zhang X, Ren S, Sun J. (2016). Identity Mappings in Deep Residual Networks. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 2016, 9909 LNCS, V.

[21] Längkvist, M., Karlsson, L., Loutfi, A. (2014). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Pattern Recognit. Lett. 2014, 42, 11–24.

[22] Huang, G., Liu, Z., Maaten, L. van der (1978). Densely Connected Convolutional Networks. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 1978, 39, 1442–1446.

[23] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2018, 4510–4520, doi:10.1109/CVPR.2018.00474.

[24] Moin, N. B., Islam, N., Sultana, S., Chhoa, L. A., Ruhul Kabir Howlader, S. M., Ripon, S. H. (2022). Disease Detection of Bangladeshi Crops Using Image Processing and Deep Learning - A Comparative Analysis. 2022 2nd Int. Conf. Intell. Technol. CONIT 2022, doi:10.1109/CONIT55038.2022.9847715.

[25] Prabu, M., Chelliah, B. J. (2022). Mango Leaf Disease Identification and Classification Using a CNN Architecture Optimized by Crossover-Based Levy Flight Distribution Algorithm. Neural Comput. Appl. 2022, 34, 7311–7324, doi:10.1007/s00521-021-06726-9.

[26] Diraco, G., Rescio, G., Leone, A. (2025). Radar-Based Activity Recognition in Strictly Privacy-Sensitive Settings Through Deep Feature Learning. Biomimetics 2025, 10, doi:10.3390/biomimetics10040243.

[27] Salim, F., Saeed, F., Basurra, S., Qasem, S. N., Al-Hadhrami, T. (2023). DenseNet-201 and Xception Pre-Trained Deep Learning Models for Fruit Recognition. Electron. 2023, 12, doi:10.3390/electronics12143132.

[28] Ulfa, S., Kusuma, H., Sardjono, T. A. (2024). Automating Embryo Quality Classification in IVF by Comparing the Performance of ResNet50, VGG16, and InceptionV3. 2024 Beyond Technol. Summit Informatics Int. Conf. BTS-I2C 2024, 292–297, doi:10.1109/BTS-I2C63534.2024.10941949.

[29] Saleem, M. H., Potgieter, J., Arif, K. M. (2020). Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers. Plants 2020, 9, 1–17, doi:10.3390/plants9101319.

[30] Wang, B., Zhang, C., Li, Y., Cao, C., Huang, D., Gong, Y. (2023). An Ultra-Lightweight Efficient Network for Image-Based Plant Disease and Pest Infection Detection. Precis. Agric. 2023, 24, 1836–1861, doi:10.1007/s11119-023-10020-0.

[31] Atila, Ü., Uçar, M., Akyol, K., Uçar, E. (2021). Plant Leaf Disease Classification Using EfficientNet Deep Learning Model. Ecol. Inform. 2021, 61, doi:10.1016/j.ecoinf.2020.101182.

[32] Ngugi, H. N., Akinyelu, A. A., Ezugwu, A. E. (2024). Machine Learning and Deep Learning for Crop Disease Diagnosis: Performance Analysis and Review. Agronomy 2024, 14, doi:10.3390/agronomy14123001.

[33] Paul, N., Sunil, G. C., Horvath, D., Sun, X. (2024). Deep Learning for Plant Stress Detection: A Comprehensive Review of Technologies, Challenges, and Future Directions. Comput. Electron. Agric. 2025, 229, doi:10.1016/j.compag.2024.109734.

[34] Balaji, G. N., Parthasarathy, G., Kovendan, A. K. P., Jha, A. (2025). Using Deep Learning for Plant Disease Detection and Classification. Nat. Environ. Pollut. Technol. 2025, 24, doi:10.46488/NEPT.2025.v24i02.B4260.

[35] Dorchin, N., Shachar, E., Friedman, A. L. L., Bronstein, O. (2021). Reclassification of Gall Midges (Diptera: Cecidomyiidae: Cecidomyiini) from Amaranthaceae, with Description of Ten New Species Based on an Integrative Taxonomic Study. Insects 2021, 12, doi:10.3390/insects12121126.

[36] Heath, J. J., Stireman, J. O. (2021). Dissecting the Association between a Gall Midge, Asteromyia Carbonifera, and Its Symbiotic Fungus, Botryosphaeria Dothidea. Entomol. Exp. Appl. 2010, 137, 36–49, doi:10.1111/j.1570-7458.2010.01040.x.