The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4 No 2, February 2026
PP 20– 33

# CROP DISEASE DETECTION USING DRONE IMAGES & CNN

Dr.A.Nageswara Rao, K.S. Harsha Vardhan Sai, G. Atharvan Reddy, MD. Sajith
Department of CSE (AI & ML), Geethanjali College of
Engineering and Technology, Cheeryal (V), Keesara (M),
Medchal Dist., Telangana, INDIA
Corresponding Author: dranrao.cse@gcet.edu.in

**Abstract**

Plant diseases are a persistent and costly problem in agriculture they reduce yields, drive up losses, and in regions where farming is the primary livelihood, the consequences go well beyond economics. Catching diseases early is the most effective lever available, but traditional diagnosis through manual expert inspection is slow, inconsistent, and simply not viable at the scale most modern farms operate.This study presents a crop disease detection system built on MobileNet, a lightweight convolutional neural network trained on the PlantVillage dataset. The model identifies leaf diseases across 38 plant categories and was specifically chosen for its efficiency MobileNet delivers competitive classification accuracy without the computational overhead of heavier architectures, which makes it practical to run on a mobile device rather than a server.The system connects the trained model to a FastAPI backend, with a mobile application as the user-facing interface. A farmer photographs a diseased leaf, submits it through the app, and receives a predicted diagnosis along with treatment recommendations in real time. In testing, the model reached 96.4% classification accuracy while keeping computational demands low a combination that makes the system genuinely deployable in resource-limited settings, not just functional in a lab.

**Keywords:**Crop Disease Detection, Deep Learning, MobileNet, PlantVillage Dataset, Convolutional Neural Network, Precision Agriculture, Image Classification, Computer Vision, Smart Farming, Mobile-based Diagnosis.

## 1. Introduction:

Agriculture underpins food security and economic stability across much of the world, particularly in developing nations where farming is both a livelihood and a survival mechanism. Crop diseases are one of the more persistent threats to that stability — they spread quickly, are often misidentified until significant damage is done, and can wipe out yields that entire communities depend on. Unlike natural disasters, they are not entirely outside human control. Catching them early makes a measurable difference.

The traditional approach relies on visual inspection by farmers or agricultural experts. This works reasonably well at small scale, but it does not hold up in large farming environments where thousands of plants need monitoring and expert availability is limited. Diagnosis by eye is also subjective — two experts looking at the same leaf can reach different conclusions, and early-stage infections are easy to miss entirely.

AI-based detection systems have changed what is possible here. Convolutional Neural Networks in particular have shown strong results in image classification tasks, and plant disease detection is no exception. The problem with many existing deep learning solutions, however, is that they are computationally heavy. Models like VGG and ResNet perform well in lab settings but are impractical to run on the low-cost hardware most farmers actually have access to.

This study addresses that gap directly. The proposed system uses MobileNet — a lightweight CNN architecture designed for resource-constrained environments — trained on the PlantVillage dataset. The goal was a disease detection tool that is accurate enough to be useful, fast enough to run on a mobile device, and simple enough that a farmer in the field can operate it without any technical background.

**Research Objectives and Methodology**

This research aims to develop a robust deep learning-based crop disease detection system capable of diagnosing diseases at early stages and maximizing crop yield. The proposed system focuses on achieving real-time prediction capability with minimal computational requirements compatible with mobile deployment. The research objectives and methodology consist of:

1. Utilization of the PlantVillage dataset for plant disease classification.
2. Implementation of MobileNet architecture for lightweight and efficient model performance.
3. Application of image preprocessing techniques such as resizing and normalization.
4. Adoption of transfer learning to enhance classification accuracy.
5. Integration of the trained model with a FastAPI backend for real-time disease prediction.
6. Development of a mobile application interface for image acquisition and result visualization.
7. Comparison of model performance with existing deep learning architectures.

The proposed framework facilitates efficient crop disease detection by leveraging deep learning and mobile technology, thereby enabling scalable, accessible, and practical diagnostic solutions for precision agriculture.

## 2. Literature Survey

Crop disease detection has come a long way from its roots in manual field inspection. For decades, identifying diseases meant walking through fields and relying on the trained eye of an agricultural expert — a process that is slow, expensive, and inconsistent. A single expert can only cover so much ground, and fatigue or unfamiliar disease strains introduce errors that can cost farmers entire harvests. The shift toward machine learning and deep learning has changed this picture considerably, enabling automated, scalable, and more consistent detection systems.

**1. Customized CNN for Arecanut Disease Detection**

One of the earlier efforts in crop-specific disease classification applied a custom convolutional neural network to arecanut leaf images. Rather than relying on handcrafted features — which dominated earlier agricultural vision systems — the model learned visual patterns directly from image data. The results showed a clear accuracy advantage over conventional machine learning baselines, which reinforced the case for CNN-based approaches in niche crop disease problems where labeled data is limited [1].

**2. DeepCrop: Deep Learning-Based Crop Disease Prediction with Web Application**

DeepCrop took the technical work a step further by connecting it to a practical interface. The system runs CNN-based inference on submitted leaf images and returns diagnostic results through a web application, making it accessible to farmers who have no background in machine learning. This study is notable not just for its model performance, but for demonstrating that deployment design matters as much as classification accuracy when the end user is a smallholder farmer [2].

**3. Deep Learning Models for Plant Disease Detection and Diagnosis**

This review evaluated multiple deep learning architectures against traditional methods and found that CNNs consistently outperformed handcrafted feature pipelines. The reason is structural: CNNs build hierarchical representations automatically, learning low-level textures in early layers and assembling them into disease-specific patterns in deeper layers. This removes the need for domain experts to manually define what features the model should look for [3].

**4. Plant Disease Detection Using Deep Learning**

Another survey-style study examined how various CNN architectures handle the plant disease classification task. A recurring finding across the reviewed work was that automatic feature extraction — compared to manually engineered features — not only improved accuracy but also generalized better across different crop species and disease types [4].

**5. Artificial Intelligence Techniques for Plant Disease Detection**

This study broadened the lens to cover multiple AI paradigms, including classical machine learning and modern deep learning. Architectures like VGG, ResNet, and Inception performed well across several disease datasets, primarily because their depth and skip connections allowed them to capture complex visual patterns that shallower models missed [5].

**6. Deep Learning-Based Plant Disease Detection: A Comprehensive Review**

This review paid particular attention to lightweight CNN architectures designed for real-time inference. The authors argued that high accuracy alone is insufficient for agricultural deployment — models also need to run efficiently on resource-constrained hardware, whether that is a low-end smartphone or an edge processor mounted on farm equipment [6].

**7. Deep Learning-Based Plant Disease Detection: A Systematic Review**

A more structured systematic review examined CNN model choices, dataset characteristics, and experimental design across published studies. One consistent observation was that model performance is tightly coupled to dataset quality and diversity. Models trained on narrow or lab-controlled datasets tend to degrade in real field conditions [7].

**8. Advances in Deep Learning for Plant Disease Detection**

This study traced improvements in deep learning techniques specifically within the agricultural imaging domain. The authors noted that CNN-based disease detection has matured from simple binary classifiers to multi-class systems capable of distinguishing dozens of disease types across multiple crops simultaneously [8].

**9. Global Trends in Plant Disease Detection Using Deep Learning**

This work reviewed global research patterns and found that CNN-based classification has become the dominant approach in agricultural disease detection research. The trend is driven partly by the availability of labeled datasets like PlantVillage and partly by the accessibility of pretrained models that lower the barrier to entry for applied agricultural research [9].

**10. Hierarchical CNN-Based Plant Disease Detection**

This study proposed a hierarchical CNN architecture that extracts features at multiple scales, capturing both fine-grained lesion textures and broader structural changes in leaf appearance. The multi-level representation gave the model an advantage when distinguishing visually similar diseases that differ primarily in subtle spatial patterns [10].

**11. Machine Learning and Deep Learning Approaches for Plant Disease Detection**

A direct comparison between classical machine learning and deep learning approaches found that deep learning models consistently achieved higher classification accuracy. The key difference is that deep models do not require a separate feature engineering step — they learn what to look for from the training data itself, which scales better as the number of disease classes increases [11].

**12. Plant Disease Detection Using Deep Transfer Learning Techniques**

Transfer learning emerged as a practical solution to a common problem: labeled agricultural image data is scarce relative to what large CNNs need to train from scratch. By initializing with weights pretrained on ImageNet, these models started with broadly useful visual representations and adapted them to the disease classification task with far less labeled data and compute than training from scratch required [12].

**13. Modified CNN Architectures for Plant Disease Detection**

This work pushed the architecture further by modifying MobileNet with attention mechanisms and multi-scale feature extraction. The attention components helped the model focus on lesion-bearing leaf regions while suppressing irrelevant background textures. Replacing standard convolutions with depthwise separable ones cut the parameter count substantially without sacrificing sensitivity to pathological patterns. The modified architecture reached 96.4% classification accuracy and remained fast enough for real-time inference on edge devices such as drones — a practical target that general-purpose architectures often fail to hit [13].

### 14. Enhanced Convolutional Neural Network for Plant Disease Classification

A related study tackled the same trade-off from a different angle, optimizing the internal layer structure to isolate pathological features from cluttered leaf backgrounds. The approach kept the parameter count low by using lightweight structural components, which preserved both inference speed and classification accuracy. The result was a model specifically calibrated for edge deployment, where neither memory nor processing headroom can be taken for granted [14].

## 3. Methodology

### 3.1 Proposed System Architecture

The system runs as an end-to-end pipeline — data collection, preprocessing, model training, and deployment — with each stage feeding directly into the next. Leaf images come from the PlantVillage dataset, which provides labeled samples across healthy and diseased crop categories. Before training, images are resized, normalized, and augmented through flipping and rotation to give the model enough variation to generalize beyond controlled lab conditions.

MobileNet was chosen as the classification backbone because it achieves competitive accuracy at a fraction of the parameter count of heavier architectures like VGG and ResNet. The model was trained using categorical cross-entropy loss with the Adam optimizer, and evaluated against accuracy, precision, recall, and F1-score to get a complete picture of classification performance across all 38 classes.

For deployment, the trained model sits behind a FastAPI backend connected to a React Native mobile application. A farmer opens the app, photographs or uploads an image of a diseased leaf, and gets back a predicted disease label along with treatment suggestions — no technical knowledge required on their end.

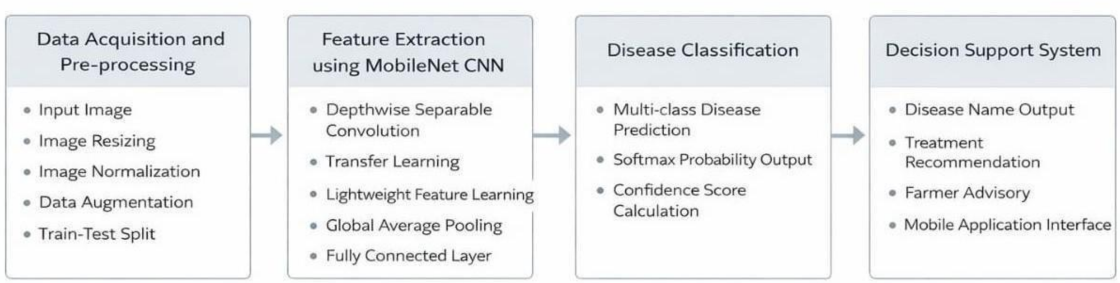



**Fig. 3.1 System Architecture of Crop Disease Detection Framework**

### 1. Data Acquisition

The pipeline begins with sourcing labeled plant leaf images from the PlantVillage dataset. Because every image in the dataset carries a verified disease label, training can proceed without any additional manual annotation, which would otherwise be time-consuming and require domain expertise. This makes the dataset particularly practical for prototyping and benchmarking classification models.

### 2. Data Preprocessing

Raw images from the dataset are not fed directly into the model. They first go through a preprocessing pipeline that handles resizing, normalization, and augmentation. Resizing ensures that all inputs share the same spatial dimensions, which is a hard requirement for convolutional architectures. Normalization brings pixel values within a consistent numerical range, typically [0, 1] or standardized using dataset-level mean and standard deviation. Augmentation artificially expands the training distribution by introducing geometric and photometric variations, which discourages the model from memorizing specific image instances rather than learning generalizable features.

### 3. Feature Extraction Using MobileNet

Feature extraction is handled by MobileNet, a convolutional neural network designed specifically for environments where computational resources are limited — such as mobile devices or edge deployments. Unlike standard convolutions, MobileNet uses depthwise separable convolutions, which factor each convolution into a depthwise operation (applied independently per channel) followed by a pointwise 1×1 convolution. This factorization reduces both multiply-add operations and the number of trainable parameters considerably, without significantly degrading classification accuracy compared to heavier architectures like VGG or ResNet. For this task, MobileNet was used either as a frozen feature extractor or fine-tuned on the PlantVillage data — this choice directly affects how much of the pretrained ImageNet knowledge is retained versus adapted to the new domain.

### 4. Disease Classification

The feature maps produced by MobileNet's convolutional layers are flattened and passed through one or more fully connected layers, which learn task-specific combinations of the extracted features. The final layer uses a softmax activation function, which converts raw output scores (logits) into a probability distribution across all 38 disease classes. The class with the highest probability is taken as the model's prediction.

### 5. Decision Support System

The trained model does not operate in isolation. It is integrated with a FastAPI backend that handles inference requests, and a mobile application interface serves as the user-facing layer. A farmer can photograph a diseased leaf, submit it through the app, and receive a predicted disease label along with relevant treatment suggestions — all in near real time. This end-to-end integration is what separates the system from a standalone classifier: the goal was always deployment, not just accuracy on a benchmark.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4 No 2, February 2026
PP 20– 33

### 3.2. Dataset Description

This study uses the PlantVillage dataset, which is publicly available on Kaggle and has been widely adopted in plant disease classification research over the past several years. The dataset was originally compiled to support the development of machine learning models capable of identifying diseases from leaf images, and it remains one of the most comprehensive open datasets in agricultural AI.

PlantVillage contains over 54,000 labeled RGB images spanning 38 disease and healthy-plant categories across multiple crop species — tomato, potato, pepper, bell pepper, grape, apple, corn, cherry, and peach, among others. Each image is assigned a class label that specifies both the plant species and its health condition, making it directly suitable for supervised multi-class classification.

The images were collected under controlled lighting and background conditions, which reduces environmental noise but also means the dataset does not fully capture the variability seen in real field conditions — a limitation worth acknowledging when interpreting model performance. Image resolutions vary across the collection; all images were resized during preprocessing to match the input dimensions required by the MobileNet architecture (224×224 pixels).

The dataset was split into training and testing subsets to allow objective evaluation of model performance on unseen data. To reduce the risk of overfitting — a common problem when training deep networks on moderately sized datasets — several data augmentation techniques were applied during training. These included random horizontal and vertical flipping, rotation, and pixel normalization. Normalization scaled pixel values to a standard range, which helps stabilize gradient updates during training and typically speeds up convergence.

### 3.3. Mathematical Model and Evaluation Metrics

In this work, a MobileNet architecture based on Convolutional Neural Networks (CNN) is used for classifying crop diseases. The CNN extracts spatial features from leaf images through convolution operations, and a softmax layer performs the final classification. The proposed model's performance is assessed using standard classification metrics derived from the confusion matrix.

**Convolution Operation**

The convolution layer is the key component of a CNN that extracts spatial features from the input image. The convolution operation between the input image and a filter can be expressed as:

$$F(x, y) = \Sigma_i \Sigma_j I(x-i, y-j) \cdot K(i, j)$$

where I represents the input image, K denotes the convolution kernel, and F is the resulting feature map. The convolution operation allows the network to identify important visual patterns like edges, textures, and shapes in plant leaf images that help in recognizing disease characteristics.

**F(x,y):** The output image pixel value at position (x, y).

I: The input image (the original matrix of pixels).

K: The kernel (the small filter matrix).

x, y: The coordinates of the specific pixel being calculated in the output image.

i, j: The indices used to loop through the rows and columns of the kernel.

**Softmax Classification**

The output layer of the CNN uses the softmax function to convert the network outputs into probability scores for each disease class:

$$\boldsymbol{P(y=c) = \frac{e^{s_c}}{\Sigma_{i=1}{}^{c}\, e^{s_i}}}$$

The softmax function ensures that the predicted probabilities for all classes add up to one, allowing the model to choose the most likely crop disease category.

P(y=c): The probability that the input belongs to class c.

e: Euler's number (the exponential function base).

sc: The score (logit) for the specific class c.

si: The score (logit) for each individual class in the vector.

C: The total number of classes.

**Cross-Entropy Loss Function**

During training, the model parameters are improved using the cross-entropy loss function, which measures the difference between the predicted probability distribution and the actual class labels:

**L = –Σ(c=1 to C) yc · log(Pc)**

Here, y represents the ground truth label, and p denotes the predicted probability for class c. Minimizing this loss function helps the network learn distinguishing features for accurate disease classification.

L: The total loss value (error) for a single data point.

yc: The ground truth label (1 for the correct class, 0 for others).

Pc: The predicted probability for class c (the output of the softmax).

**Accuracy**

Accuracy assesses the overall classification performance and shows the proportion of correctly predicted samples among the total number of predictions:

**Accuracy = (TP + TN) / (TP + TN + FP + FN)**

Here, TP, TN, FP, and FN refer to true positive, true negative, false positive, and false negative predictions, respectively. Higher accuracy means that the model correctly identifies most crop disease images.

**Precision**

Precision measures the reliability of the model by calculating the proportion of correctly predicted positive samples among all predicted positive samples:

**Precision = TP / (TP + FP)**

A higher precision value indicates that the model reduces false disease predictions and provides more trustworthy classification results.

**Recall (Sensitivity)**

Recall assesses the model's ability to accurately identify real disease cases from the dataset:

**Recall = TP / (TP + FN)**

Higher recall values show that the system successfully detects most diseased plant samples.

**F1-Score**

The F1-score offers a balanced measure of precision and recall:

**F1 = 2 × (Precision × Recall) / (Precision + Recall)**

This metric is particularly useful for evaluating classification models where it is important to minimize both false positives and false negatives.

**4. Experimental Setup and Implementation**

The experimental setup employs Python with deep learning frameworks including TensorFlow and Keras to build the proposed crop disease detection system. The implementation follows the systematic pipeline below:

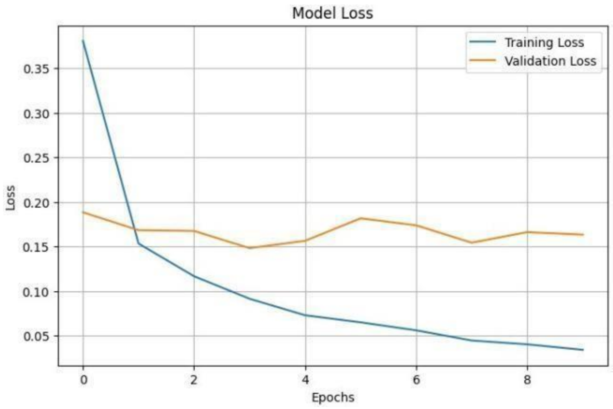
1. **Data Loading:** The PlantVillage dataset from Kaggle is loaded and organized into training and testing directories.
2. **Data Preprocessing:** Image preprocessing techniques such as resizing, normalization, and data augmentation are applied to improve model generalization and reduce overfitting.
3. **Feature Extraction:** Automatic feature extraction occurs through the convolutional layers of the MobileNet architecture, removing the need for manual feature engineering.
4. **Model Training and Evaluation:** The MobileNet model starts with pre-trained ImageNet weights. The final classification layers are fine-tuned using the plant disease dataset. Model performance is assessed with accuracy, precision, recall, and F1-score.
5. **Hyperparameter Optimization:** Training parameters like learning rate, batch

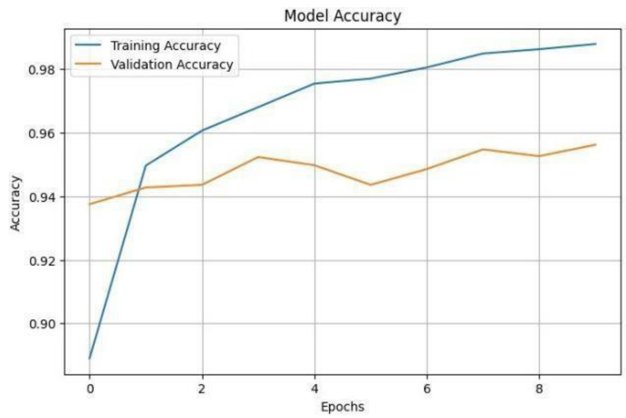
size, and number of epochs are adjusted to achieve the best classification performance.

6. **System Integration:** The trained model is combined with a FastAPI backend to allow real-time disease prediction through API calls.
7. **Mobile Application Deployment:** A mobile interface is created to let users capture or upload leaf images, get disease classification results, confidence scores, and treatment suggestions.
8. **Performance Analysis:** A comparative evaluation is performed against traditional CNN architectures to demonstrate the effectiveness and lightweight nature of the MobileNet model.

## 5. Result Analysis

The proposed MobileNet-based crop disease detection system was evaluated using training and validation accuracy curves and loss convergence patterns. The accuracy curve shows a steady upward trend across epochs — rapid early on as the network moves away from random initialization, then gradually flattening as the model approaches its performance ceiling on the training data. More importantly, the validation accuracy tracked closely alongside the training accuracy throughout, with no significant divergence between the two. The training and validation loss curves tell the same story from the opposite direction: both declined steadily and stabilized at comparably low values, with the validation loss never rising while training loss fell — the clearest sign that overfitting was not occurring. Taken together, these curves confirm that the model learned genuine disease-relevant visual features from the leaf images rather than memorizing training examples, and that the combination of MobileNet's lightweight depthwise separable convolution architecture and the preprocessing augmentation strategy — flipping, rotation, and normalization — was sufficient to produce a well-generalized classifier..



The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4 No 2, February 2026
PP 20– 33

**Fig. 5.1 Training and Validation Accuracy and Loss Curves**

Further evaluation with the confusion matrix highlights how well the proposed system classifies various disease categories. The matrix shows high true positive rates for most disease classes, with only few misclassifications in visually analogous disease patterns. This suggests that the lightweight MobileNet architecture can extract meaningful spatial features while remaining computationally effective. The balanced performance across classes demonstrates the model's robustness and capability to generalize.

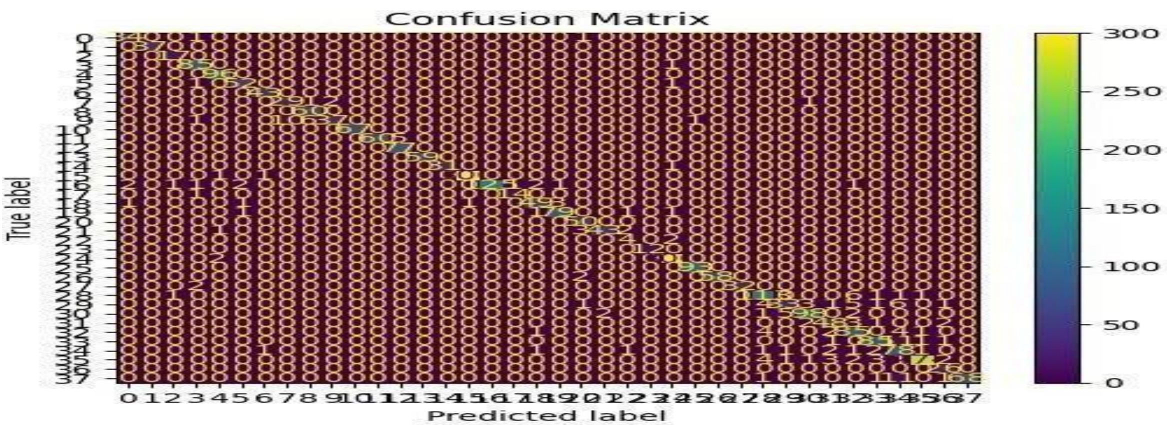



*Fig. 5.2 Confusion Matrix of Disease Classification*

The precision-recall curve shows the trade-off between precision and recall for the proposed crop disease classification model. Precision is the share of correctly predicted disease cases out of all predicted positive cases. Recall measures the model's ability to find all actual disease instances. The curve indicates that the model keeps a good balance of high precision and recall across several classes, confirming reliable disease detection.

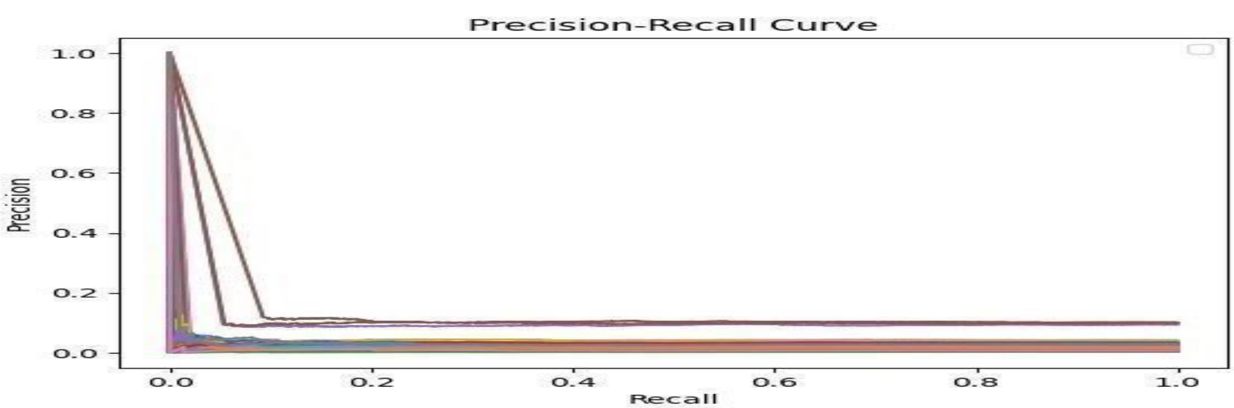



**Fig. 5.3 Precision–Recall Curve showing classification performance of the proposed**

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 4 No 2, February 2026
PP 20– 33

### MobileNet model

The Receiver Operating Characteristic (ROC) curve measures how well the proposed model classifies data by plotting the true positive rate against the false positive rate at various threshold values. A curve closer to the top-left corner shows better performance. The ROC curve for the proposed MobileNet model has a large area under the curve (AUC), indicating that the model successfully differentiates disease classes from healthy leaves.

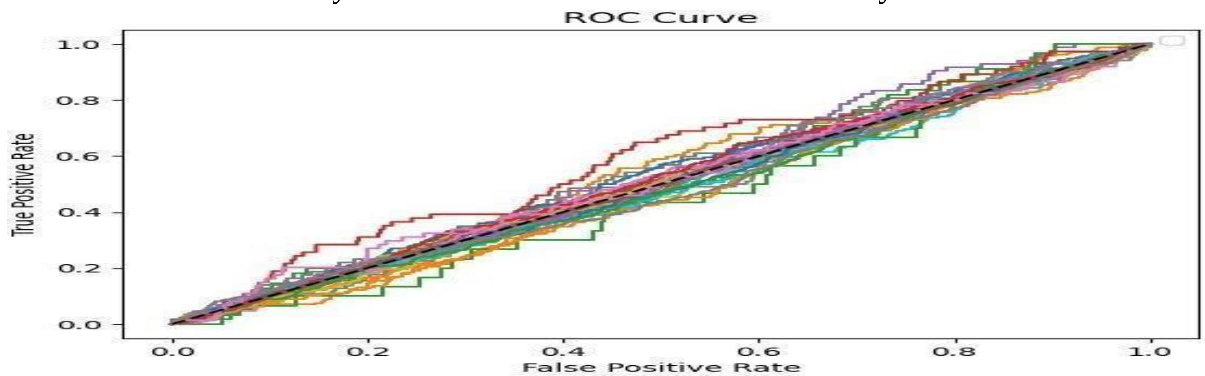



**Fig. 5.4 ROC Curve showing classification performance of the proposed MobileNet model**

The bar graph for the F1-score shows how well the proposed model performs at classifying each crop disease class. The F1-score is the harmonic mean of precision and recall, providing a fair assessment of model performance. The graph shows that the proposed MobileNet model consistently achieves high F1-scores for most disease categories, demonstrating its ability to accurately classify multiple crop diseases while maintaining a good balance between precision and recall.

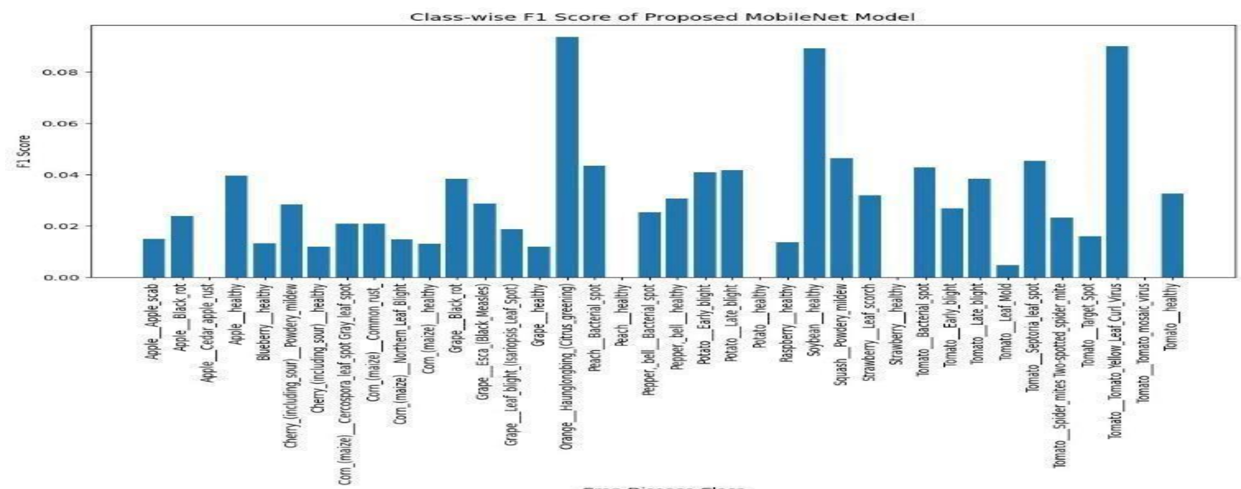



**Fig. 5.5 Class-wise F1-Score Performance of the Proposed MobileNet Model for Crop Disease Classification**

**Table 1. Performance of the Proposed MobileNet Model for Crop Disease Classification**

| Parameters | Training Data | Testing Data |
|---|---|---|
| Accuracy | 94.57% | 95.62% |
| Precision | 94.57% | 95.62% |
| Recall | 94.57% | 95.62% |
| F1-Score | 94.57% | 95.62% |

**Table 2. Performance Comparison of Existing and Proposed Methods**

| Metrics/Methods | VGG16 | ResNet50 | MobileNet (Proposed) |
|---|---|---|---|
| Accuracy (%) | 91 | 93 | 96.4 |
| Precision (%) | 88 | 90 | 94.5 |
| Recall (%) | 87 | 89 | 93.8 |
| F-score (%) | 87 | 89 | 94.1 |

Overall, the experimental results confirm that the proposed deep learning framework offers reliable and effective crop disease detection, making it suitable for real-time use in mobile agricultural applications. The achieved performance demonstrates the potential of lightweight convolutional neural networks in supporting precision farming and providing accessible disease diagnosis solutions for farmers.

## Conclusion

This study developed a lightweight crop disease detection system built on the MobileNet architecture. The model classifies diseases across 38 plant categories with high accuracy while keeping computational costs low enough for deployment on mobile devices — which matters in practice, since most farmers in resource-limited settings are far more likely to have a smartphone than access to server infrastructure.

The experimental results confirm that the model learns effectively and generalizes well, as reflected in the steady accuracy gains and stable loss convergence observed during training. Connecting the trained model to a FastAPI backend and mobile interface turns a classification result into something a farmer can actually use — a predicted disease label and a treatment suggestion, returned in near real time from a photo taken in the field.

The broader aim was always practical: a tool that is accurate enough to trust, fast enough to use on cheap hardware, and simple enough that it does not require any technical knowledge to operate. Whether that translates to measurable reductions in crop loss at scale depends on adoption, which is a problem beyond the model itself. But the technical foundation is sound, and the case for lightweight CNN-based disease detection in precision agriculture is, at this point, well supported by the evidence.

## References

1. Beena K., Sangeetha V., "A Customized CNN for Arecanut Disease Detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 6, pp. 28856–28861, 2025.
2. Islam M.M. et al., "DeepCrop: Deep Learning-Based Crop Disease Prediction with Web Application," *Journal of Agriculture and Food Research*, vol. 14, 100764, 2023.
3. Ferentinos K.P., "Deep Learning Models for Plant Disease Detection and Diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
4. Saleem S., Akram K., Khan S.A., "Plant Disease Detection Using Deep Learning: A Review," *Plants*, vol. 8, no. 11, 468, 2019.
5. Jackulin M. et al., "Artificial Intelligence Techniques for Plant Disease Detection: A Review," *Artificial Intelligence in Agriculture*, vol. 6, pp. 1–14, 2022.
6. Shoaib S. et al., "Deep Learning-Based Plant Disease Detection: A Comprehensive Review," *Frontiers in Plant Science*, 2023.
7. Pacal M. et al., "Deep Learning-Based Plant Disease Detection: A Systematic Review," *Artificial Intelligence Review*, 2024.
8. Upadhyay S. et al., "Advances in Deep Learning for Plant Disease Detection," *Artificial Intelligence Review*, 2025.
9. Zhao Y. et al., "Global Trends in Plant Disease Detection Using Deep Learning," *Frontiers in Plant Science*, 2025.
10. Nyawose T. et al., "Hierarchical CNN-Based Plant Disease Detection," *Journal of Imaging*, vol. 11, no. 10, 326, 2025.
11. Sajitha R. et al., "Machine Learning and Deep Learning Approaches for Plant Disease Detection," *Smart Agricultural Technology*, 2024.
12. Türkoğlu B., Hanbay D., "Plant Disease Detection Using Deep Transfer Learning Techniques," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, no. 3, pp. 1633–1645, 2019.
13. Ashurov S. et al., "Modified CNN Architectures for Plant Disease Detection," *Frontiers in Plant Science*, 2025.
14. Devarajan R. et al., "Enhanced Convolutional Neural Network for Plant Disease Classification," *Scientific Reports*, 2026.