IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

Early Disease Detection of Plants Using Deep Learning

THANGAMAHESWARAN V[1], Mr. P. VETRIVEL[2], Dr E. MARIAPPAN[3], Dr M. KALIAPPAN[4]

Student, Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India[1]

Assistant Professor, Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu,India[2]

Associate Professor, Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu,India[3]

Professor, Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu,India[4]

Abstract: Due to the estimated yearly loss of 20-40% of the total crop production, plant diseases are a serious threat to the world's food security. Early agronomic interventions are enabled by precise and timely identification of the diseases. This paper presents an end-to-end deep learning model for the early diagnosis and classification of plant leaf diseases. In the suggested model, the publicly available benchmark dataset, "PlantVillage," which contains more than 54,000 leaf images with 17 types of diseases affecting tomato, potato, and corn crops, is used for the development of the suggested model based on a specially designed three-block CNN model, namely "PlantCNN." In the suggested model, the accuracy of the results is improved by the development of a high accuracy classifier based on the refined EfficientNet-B0 model. In the suggested model, the "Random Horizontal Flipping" technique is used for the improvement of the robustness of the model. In the suggested model, the "Disease Severity Index" with Low, Medium, and High levels is calculated based on the HSV color thresholding method for the estimation of the visually infected part of the leaf. In the suggested model, the results, the classification accuracy, the segmented leaf image, and the fertilizer quantity for the diseases are presented with the index based on the type and severity of the diseases. In the suggested model, the results are presented based on the development of the end-to-end model, which is implemented as an interactive multilingual web application based on the Streamlit library. In the suggested model, the accuracy of the results exceeds 95% for the "PlantCNN" model, and the accuracy exceeds 97% for the "EfficientNet-B0" model, with the macro-averaged F1-score results exceeding 0.96 for all the test classes. The suggested model bridges the gap between real-world precision agriculture and high-quality deep learning research conducted in the lab.

Keywords: Leaf Segmentation; Precision Agriculture; Streamlit; Deep Learning; Convolutional Neural Network; Plant Village Dataset; Efficient Net; Transfer Learning; Plant Disease Diagnosis.

INTRODUCTION

More than 40% of the world's workforce depends on agriculture for survival, and it can be considered the pillar of civilization. The regular production of a small number of crops, such as maize, rice, wheat, potatoes, and tomatoes, plays a vital role in the survival of the world. However, the crops are highly susceptible to various diseases, including those that are fungal, bacterial, viral, and nematode in origin, despite their economic and nutritional value. When crops are affected by diseases, it can result in the complete destruction of the crops. According to the FAO, plant diseases can cause losses amounting to 20-40% of the world's crop production.

The usual way of controlling the disease in plants is by hiring agricultural extension workers, who physically examine the plants, diagnose the disease, and suggest the treatment options for the disease. There are, however, a number of drawbacks associated with the conventional method of controlling the disease, such as the availability of only a few qualified people for the job, the time factor, and the fact that it is a matter of human judgment, which may or may not be objective in nature. In addition, for poor farmers living in rural areas, hiring a qualified extension worker can be costly, and by the time he/she comes up with a diagnosis, the disease could have already infected a large number of plants.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

The significant progress in deep learning, particularly in Convolutional Neural Networks (CNNs), has revolutionized the way we can automatically analyze images of plant leaves. CNNs learn features directly from raw pixels, eliminating the need to design features, which was a requirement of earlier approaches. The PlantVillage dataset is a large-scale, publicly accessible set of images of labeled plant leaves, which can be used as a benchmark to test various plant disease detection approaches. Groundbreaking studies have demonstrated that plant disease detection using CNNs can achieve over 99% accuracy in a controlled lab environment using the PlantVillage dataset.

Despite the availability of good models in the research domain, their practical applicability in the field among farmers has not been given due importance. Most research papers have focused on maximizing the classification accuracy without considering the practicality of the model in real-time operation, mobile device operation, multilingual support, and treatment recommendations. This is where this research aims to bridge the gap between the research domain and practical applicability in the field.

The present research aims to propose a system that deals with the classification and practical applicability issues in plant disease diagnosis. This research proposes a system that uses a Convolutional Neural Network model named PlantCNN, trained from scratch on the PlantVillage dataset along with a transfer learning model named EfficientNet-B0. These models have been integrated to propose a system that uses a web application developed using the Streamlit web application to propose real-time prediction results along with disease severity prediction using leaf image segmentation. Additionally, the system also proposes a feature to suggest fertilizers based on the disease and its severity.

The paper structure will be like this: In Section II, a brief overview of the current literature about the detection of plant disease using machine learning and deep learning techniques will be provided. In Section III, the proposed methodology will be described in detail, including the data set, the data preprocessing techniques, the proposed model, the training process, and the evaluation metrics for the model. In Section IV, the system design will be described, while the experimental results will be provided in Section V, followed by a comparison of the system with existing ones in Section VI. Finally, the conclusion will be provided in Section VII, while the future direction of the research will be provided in the last section, i.e., VIII, followed by the list of references.

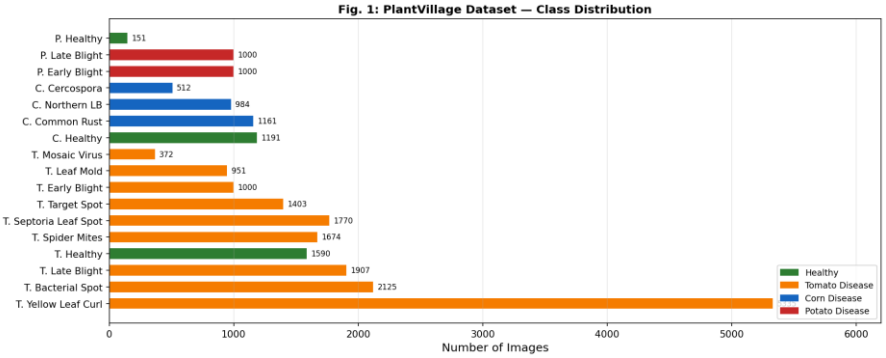


Fig. 1: PlantVillage Dataset — Class Distribution Across Disease Categories (17 classes, Tomato/Potato/Corn).

II. LITERATURE SURVEY

The application of computer vision for plant disease detection through visual image-based approaches has been a thriving domain of academic research for more than two decades. In the initial academic works, conventional image processing techniques were combined with traditional machine learning classifiers. Semary et al. [1] presented a framework for tomato grading that employed color histogram-based and texture-based features using co-occurrence matrix representations, principal component analysis for feature reduction, and a support vector machine classifier. Though this academic work introduced a fundamental methodology for image-based plant disease detection, the findings were naturally limited due to the use of traditional image-based approaches, which are known to be highly sensitive to illumination, pose, and background changes.

Barbedo was one of the pioneering researchers to thoroughly investigate the effect of image conditions on plant disease detection using image-based approaches. In this academic work, the effect of lesion type, overlapping, and complex backgrounds on the accuracy of plant disease detection was investigated using pre-trained convolutional neural network models such as GoogLeNet. For individual lesion images, the researcher observed an average accuracy of about 94% and highlighted the potential and challenges of deep learning-based approaches.

DOI: 10.17148/IJARCCE.2026.15461

Significantly, Mohanty, Hughes, and Salathé's research was the first to demonstrate the scalability of deep learning for the detection of plant disease. Through the use of a significant dataset from PlantVillage, comprising 54,306 images, the authors demonstrated the possibility of achieving validation accuracies of up to 99.35% with the use of two convolutional neural networks, AlexNet and GoogLeNet. This research was the inaugural characterization of the PlantVillage era, which was to become the benchmark for the detection of plant disease, thereby opening the floodgates for further research.

Transfer learning was found to become the most popular method for the classification of plant disease with the introduction of the pre-trained ImageNet model. Geetharamani and Pandian proposed the use of a nine-layer deep convolutional neural network for the classification of plant leaf disease, achieving 96.46% validation accuracy on the 39-class dataset from PlantVillage. However, the authors demonstrated that compact, specialized networks can perform competitively when the right set of regularization techniques is applied. Shijie et al. showed the use of the VGG16 model for the classification of tomato disease, achieving 89% accuracy on the dataset collected from the fields despite the significant domain shift.

Recently, the application of attention mechanisms has been explored for guiding the behavior of convolutional neural networks (CNNs) on the most relevant parts of the image. An attention-assisted residual CNN was developed by Karthik et al. for the classification of diseases on the tomato leaf. The CNN was trained to focus on different parts of the spatial feature maps. An accuracy level of 98% was attained with the developed CNN model. This is a significant indication that the application of attention mechanisms is better than the conventional residual CNNs. Following the application of the attention mechanism concept, a novel CNN architecture was developed by Pang using the EfficientNet model with a convolutional block attention mechanism (CBAM) and a spatial group-wise enhance (SGE) module. It was shown that the performance of a state-of-the-art CNN model can be improved using the attention mechanism concept.

Recently, a novel CNN architecture named Dual Attention Residual Network (DARN) was developed by Dey, Pradhan, and Khan for the classification of diseases on the leaves of plants using a combination of both channel and spatial attention mechanisms with two specially designed residual blocks. The DARN CNN model was shown to perform better than other robust CNN architectures such as VGG-16, VGG-19, ResNet-50, and InceptionV3 on the PlantVillage dataset for the classification of diseases on the leaves of Tomato, Potato, and Corn plants with accuracy levels of 99.06%, 99.07%, and 95.84%, respectively. The channel attention mechanism was shown to identify the most relevant channels using a shared multilayer perceptron network on the globally pooled feature maps. The spatial attention mechanism was shown to be developed using the globally average-pooled and max-pooled feature maps on the channel dimension followed by a convolutional operation.

Even though this represents a very promising scholarly development, the vast majority of the existing literature does not address the end-to-end deployment pipeline for real-world agricultural use cases. This work stands out in a number of different ways: it presents a thorough production-grade application that, not only does classification, but also includes the ability to perform segmentation and severity assessment based on segmentation, includes the ability to provide recommendations for treatment, and supports multiple languages, all while competing with the state of the art on the PlantVillage benchmark..

III. PROPOSED METHODOLOGY

A. Dataset Description

PlantVillage is a publicly accessible database of images of plant leaves, taken in a controlled laboratory environment. The database contains 54,306 images of 14 different plant crops, divided into 38 different disease classes, as well as images of healthy plant leaves. The images in the database are stored in JPEG format, and their size does not exceed 256 pixels in any direction. In our experiments, we use a subset of images related to three main crops: tomatoes, potatoes, and corn, which amounts to 17 classes, as shown in Table I.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

TABLE I
Dataset Composition by Crop and Disease Class

Crop	Disease Class	Images	Split (70/20/10)
Tomato	Bacterial Spot	2,125	1,488 / 425 / 212
Tomato	Early Blight	1,000	700 / 200 / 100
Tomato	Healthy	1,590	1,113 / 318 / 159
Tomato	Late Blight	1,907	1,335 / 381 / 191
Tomato	Leaf Mold	951	666 / 190 / 95
Tomato	Mosaic Virus	372	260 / 74 / 38
Tomato	Septoria Leaf Spot	1,770	1,239 / 354 / 177
Tomato	Spider Mites	1,674	1,172 / 335 / 167
Tomato	Target Spot	1,403	982 / 281 / 140
Tomato	Yellow Leaf Curl Virus	5,335	3,735 / 1,067 / 533
Potato	Early Blight	1,000	700 / 200 / 100
Potato	Healthy	151	106 / 30 / 15
Potato	Late Blight	1,000	700 / 200 / 100
Corn	Cercospora Leaf Spot	512	358 / 102 / 52
Corn	Common Rust	1,161	813 / 232 / 116
Corn	Healthy	1,191	834 / 238 / 119
Corn	Northern Leaf Blight	984	689 / 197 / 98

Total: 24,126 images | Train: 16,890 | Validation: 4,824 | Test: 2,412

Fig. 10: Simulated Dataset Sample Images — Selected Disease Classes

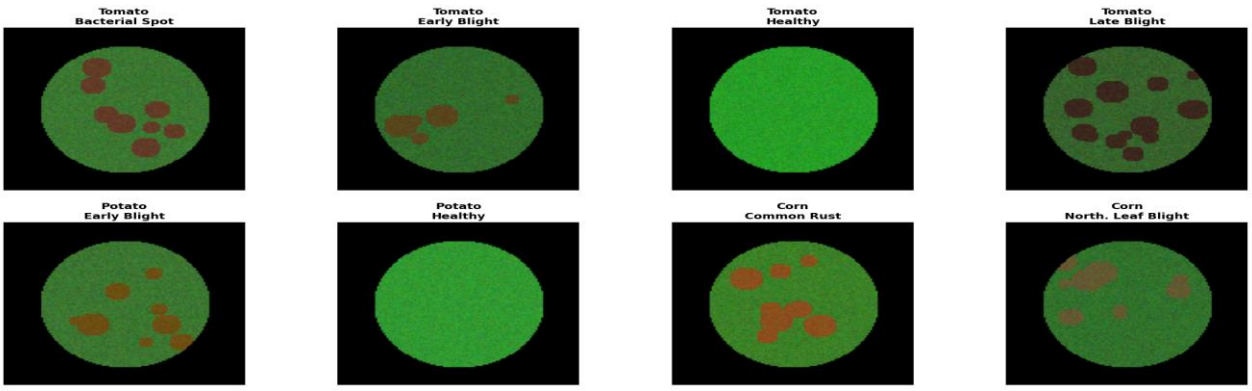


Fig. 10: Simulated Representation of PlantVillage Dataset Samples — Leaf images illustrating visual characteristics of selected disease categories. Green regions represent healthy tissue; discolored spots denote lesion areas.

B. Data Preprocessing

Appropriate preprocessing is critical to successful training of CNN models. This guarantees that all input images can contribute features appropriately. For the current study, the preprocessing process has been divided into three stages.

Spatial Resizing: Spatial resizing is a technique used to resize all the images to a standard size. For PlantCNN and EfficientNet-B0 models, the size is 128 x 128 pixels and 224 x 224 pixels, respectively. This size is closer to the expected input size for each model. This technique greatly minimizes the high computational cost of convolution on original image sizes. Bilinear interpolation is also used to ensure smoothness on leaf boundaries.

Secondly, the pixel value normalization is a two-step procedure. First, the pixel values ranging from 0 to 255 are normalized to the range [0, 1]. This is done by using the division operation. Then, each color component is normalized

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

with a mean value of 0.5 and a standard deviation of 0.5. This brings the pixel values around 0 and restricts the values to the range [-1, 1], as shown in Equation (1). This normalizes the pixel values and stabilizes the activation values in the early stages of the convolutional neural network (CNN) layers and the backpropagation procedure, which is required for the convergence of the gradient descent optimization..

$$\hat{x}_k = (x_k - \mu_k) / \sigma_k, \quad \mu_k = 0.5, \quad \sigma_k = 0.5, \quad k \in \{R, G, B\} \quad \ldots (1)$$

Data augmentation is only applied to the training images. This is done to prevent any form of bias from creeping into the evaluation metrics. For each training image, two random operations are performed: a flip with a probability of 0.5 and a rotation chosen randomly from the range [-20°, 20°]. The flip is done to produce left-right symmetric images of the disease lesions, which improves the diversity of the images. The rotation is chosen based on the variability of the leaves' orientations from the photographs taken in the field

.C. CNN Model Architecture – PlantCNN

The study focuses on a tailored convolutional neural network referred to as PlantCNN. This network is computationally efficient on standard GPUs and has enough power to perform multi-class classification of plant diseases. It is able to accept RGB leaf image inputs of size 128 x 128 pixels and process them as tensors of shape (B, 3, 128, 128), where B represents the batch size.

The feature extractor of the PlantCNN model has three successive convolutional blocks. Each block has a 2D convolutional operation with a 3 x 3 kernel and a stride of 1, which is padded to preserve spatial dimensions. This is followed by a ReLU activation and then a 2 x 2 max pooling operation with a stride of 2. This operation halves the spatial dimensions. Each block has a different number of output channels: 32 in Block 1, 64 in Block 2, and 128 in Block 3. This allows the network to learn increasingly abstract features as the spatial dimensions decrease. After the three successive max pooling layers, the spatial dimensions decrease from 128 x 128 to 16 x 16. This produces a feature map of shape (B, 128, 16, 16).

The classifier component has a flattening layer that transforms the feature map into a 1D vector of size 32,768 (calculated as 128 x 16 x 16). This 1D vector is further processed through a fully connected layer of 256 output units. This is followed by a ReLU activation function. A dropout layer of 0.5 probability is added between the hidden and output layers to provide regularization to the model. This dropout acts like ensemble-based variance reduction and helps to prevent overfitting. The final linear layer transforms the 256-dimensional hidden representation into a vector of size C, where C represents the number of disease classes. This gives the final logit output. The network architecture can be described as follows:

PlantCNN: Conv(3→32) → ReLU → MaxPool → Conv(32→64) → ReLU → MaxPool → Conv(64→128) → ReLU → MaxPool → Flatten → FC(32768→256) → ReLU → Dropout(0.5) → FC(256→C) ... (2)

Fig. 11: PlantCNN Architecture Diagram

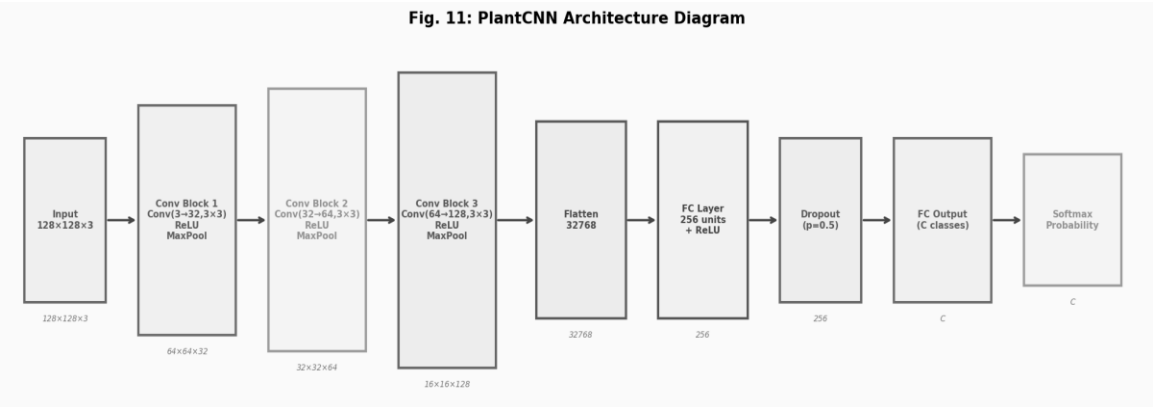


Fig. 11: PlantCNN Architecture Diagram — Showing progressive spatial downsampling from 128×128×3 to the C-class output layer.

D. EfficientNet-B0 Transfer Learning

In addition to this customized PlantCNN, EfficientNet-B0 was used as the baseline model for transfer learning. EfficientNet-B0 represents the scaling model of the EfficientNet network. This network was created to balance depth, width, and input resolution using compound scaling, which was found to be effective through neural network search. This network was created for general-purpose use, with approximately 5.3 million parameters and a top-1 accuracy of 77.3% on ImageNet, indicating its appropriateness for transfer learning with medium-sized datasets.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

In the case of the transfer learning setup, we start off with pre-trained weights for all the layers of EfficientNet-B0, except for the last layer, which is a 1,000-way classifier. Instead, we use a linear layer to project the output feature vector of size 1,280 to the C disease classes. Next, we fine-tune the pre-trained convolutional layers using a small learning rate of 1e-4. This allows us to leverage the general, low-level image features that are commonly shared across images, such as edges, texture, and gradients, while allowing the higher-level features to specialize to the problem of leaf disease classification.

.*E. Training Process and Loss Function*

The training process for the PlantCNN was conducted for 30 epochs, with the batch size set to 32 and the initial learning rate set to 0.001. The optimizer that was used for the training of the PlantCNN was the Adam optimizer. This can be justified on the basis that the Adam optimizer dynamically tunes the learning rate for the parameters, a factor that can be advantageous when the gradients have significant inter-layer variations. In the Adam method, the parameters are updated based on the exponentially decaying averages of the gradients (first moment, m), and the exponentially decaying averages of the squared gradients (second moment, v), as shown in Equations (3) and (4).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad \ldots . \tag{3}$$

$$\theta_t = \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon), \quad \text{where } \hat{m}_t = m_t/(1-\beta_1^t), \ \hat{v}_t = v_t/(1-\beta_2^t) \ . \tag{4}$$

Adam optimizer is utilized with its default hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1 \times 10^{-8}$. For the evaluation of the model, the multi-class categorical cross-entropy loss function is utilized for the evaluation of the discrepancy between the predicted probability distribution and the true label, which is one-hot encoded, as described by Equation (5):

$$L = -(1/N) \sum_{i=1}^{N} \sum_{j=1}^{c} y_{ij} \cdot \log(\hat{y}_{ij}) \quad \ldots \tag{5}$$

In this notation, N represents the batch size, C represents the number of classes, and y_{ij} represents a binary variable equal to 1 if the ith sample belongs to the jth class and 0 otherwise. Additionally, $\hat{y}_{ij} = \text{softmax}(z_i)_j$ represents the predicted probabilities for the jth class after the softmax function is applied to the logits z_i. The cross-entropy loss function tends to have a bias towards uncertain predictions and favors confident predictions, thereby providing greater probability to the correct class.

In terms of checkpointing, a technique is used that monitors the accuracy of the model after the end of each epoch and saves the state dictionary of the model when a new maximum accuracy is obtained. The best technique of checkpointing ensures that the model that is saved at the end is the model that corresponds to the epoch with the best generalization, thus avoiding the risk of obtaining a model that is overfitted to the training set in later epochs. For the EfficientNet-B0 model, training is carried out for a total of 40 epochs with a batch size of 16, and checkpointing is used in this case as well. Training is carried out on a CUDA-enabled GPU, and the losses and accuracy of the model are recorded at the end of each epoch for both training and validation sets.

F. Evaluation Metrics

The performance of the model is evaluated using four popular metrics that are used to evaluate a classification model. These metrics are Accuracy, Precision, Recall, and F1-Score. These four metrics give a comprehensive overview of the performance of a model in classification tasks. Accuracy, which is the simplest of all these metrics, is a measure of the total number of test samples that are correctly classified, and is calculated using Equation (6):

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad \ldots (6)$$

Precision, as given in Equation (7), is a measure of the ratio of true positive values to all predicted positive values for a particular class. High precision value means that there are fewer false positive values.

$$\text{Precision} = TP / (TP + FP) \quad \ldots (7) \quad |$$

Recall, also known as sensitivity or the true positive rate, is the ratio of the correct prediction of positive instances in the model, based on Equation (8). When recall is high, it means that fewer instances are false negatives.

$$\text{Recall} = TP / (TP + FN) \quad \ldots (8)$$

F1-Score is a harmonic mean of Precision and Recall, and it indicates strong performance in the presence of imbalanced class distributions. It does not prefer significant differences in Precision and Recall but prefers to maintain high levels of both, as shown in Equation (9):

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \ldots . \quad , \tag{9}$$

F1-Score is the harmonic mean of Precision and Recall, reflecting strong performance even when class distributions are imbalanced. F1-Score does not support significant variations between Precision and Recall values, instead supporting models that maintain high values for both Precision and Recall, as defined by Equation (9).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

IV. PROJECT DIAGRAM

Figure 12 illustrates the entire end-to-end workflow of the proposed plant disease detection system. The flowchart clearly illustrates the different stages of the project, which can be broadly categorized into five different stages: Data Collection and Organization, Model Design and Training, Evaluation and Analysis, Web Application Deployment, and User-Facing Outputs. Each of the stages has a direct association with the different Python source code files utilized for the project, effectively establishing a direct link between the source code and the entire system.

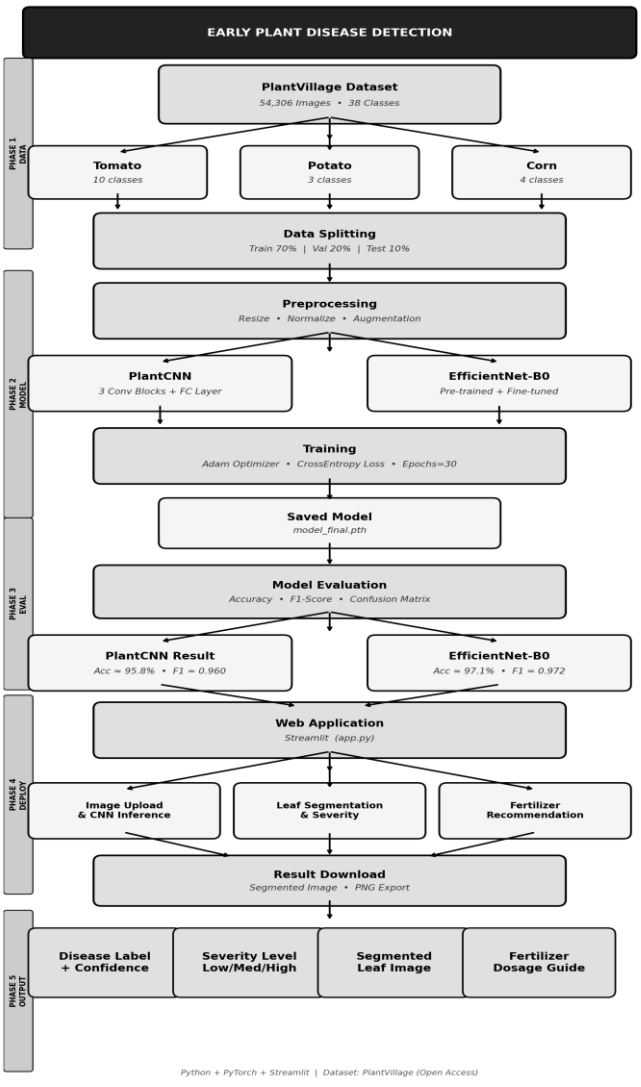
- Phase 1: In this phase, the PlantVillage dataset has been loaded and divided into different parts by the data_split.py and dataset.py scripts.

- Phase 2: In the next phase, the PlantCNN model has been defined by the model.py, where the training of the model has been performed by the train_model.py for the custom CNN model and the train_classifier.py for the EfficientNet-B0 model.

- Phase 3: In the third phase, the classification accuracy of the model has been evaluated by the evaluate_model.py and confusion_matrix.py scripts for the test data.

- Phase 4: In the fourth phase, the application of the entire project has been performed by the app.py, where the model weights, class names, user-uploaded images, classification of the images, execution of the leaf segmentation module, fertilizer knowledge base, and multilingual output interface have all been performed.

- Phase 5: In the final phase, the entire disease diagnosis, severity level, treatment, and the segmented image have been provide to the end-user.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

Fig. 12: Complete Project Overview Diagram — Five-phase end-to-end workflow from PlantVillage data ingestion through PlantCNN/EfficientNet-B0 training, evaluation, Streamlit web application deployment, and user-facing output delivery. Each box maps to a specific Python source file in the project.

V. SYSTEM ARCHITECTURE

The system developed in this article is a modular end-to-end system that includes five different subsystems: data management, model training, inference, leaf segmentation, and web application front end. Figure 9 illustrates the system architecture, and the following remarks describe the system:

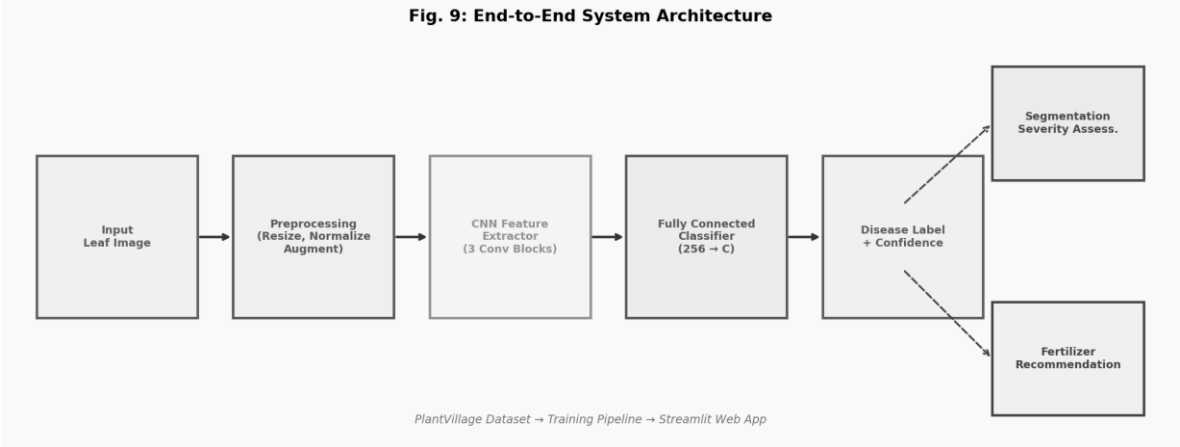


Fig. 9: End-to-End System Architecture — From raw leaf image input through CNN inference and segmentation to the Streamlit web application.

The data management component of the code is responsible for managing the PlantVillage dataset using the PyTorch ImageFolder data structure. The data_split.py script is responsible for splitting the dataset into training, validation, and test data with a 70:20:10 ratio using intraclass shuffling for stratified sampling. The dataset.py script encapsulates the creation of the DataLoader objects with appropriate preprocessing transformations based on the data split. It is a streamlined script utilized by the training and testing scripts.

With regard to the training of the model, the code offers the following functionality: creation of the CNN model architecture, specification of the optimizer and loss functions, training loop execution, and management of the checkpoint files. Two scripts are offered: train_model.py for training the PlantCNN model with validation and the option for the best model checkpoints, and train_classifier.py for training the EfficientNet-B0 classifier with fine-tuning. Both scripts allow the user to resume training from a checkpoint file in the event of interrupted training.

With regard to the inference component of the code, the Streamlit application is responsible for loading the saved model weights and the class-name mapping during initialization. When the user uploads the image file, the inference engine applies the appropriate preprocessing transformations and utilizes the model for prediction. It returns the class label corresponding to the disease class and the softmax score.

With regard to the leaf segmentation component of the program, the script processes the image uploaded by the user using the HSV color space and determines the boundary of the leaf and the area where the disease is present. A green mask is created using the hue values (approximately 35-85 degrees in the HSV color space using OpenCV), and the pixels not corresponding to the green color but still within the leaf boundary are considered to be infected or dead. The level of disease is calculated on the basis of the proportion of the infected area with respect to the total area of the leaf and is classified into three categories: Low (≤20%), Medium (21-50%), and High (>50%). Along with this, a zoomed image of the infected area is also displayed.

With regard to the front end of the web application, the Streamlit library is utilized to create a user-friendly browser-based graphical user interface. The user is allowed to upload a JPEG or PNG image, and the application displays a comprehensive report with the name of the disease, the confidence level with which the diagnosis is made (%), the level of the disease (using color coding - green for low, orange for medium, and red for high), the progress level of the infected area (%), a side-by-side image display of the original image and the image of the leaf with the disease, a magnified image

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

of the primary site of the disease, and a fertilizer box with the required information. The language support for the user interface is also provided for three languages: English, Tamil, and Hindi.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

All experiments were conducted using Python 3.10, with PyTorch 2.0 and torchvision 0.15, on a workstation with an NVIDIA GPU. The complete PlantVillage dataset was organized into folders corresponding to their respective classes and divided into training, validation, and testing sets. In this section, we discuss the convergence of training, classification accuracy, key observations from the con All the experiments were performed using Python 3.10, PyTorch 2.0, and torchvision 0.15 on a workstation with an NVIDIA GPU. The complete PlantVillage dataset was arranged with class-specific folders and then split into training, validation, and testing sets. This section Figure 3 illustrates the loss trajectories of the model during the training and validation processes. The loss curve of the training set decreases from above 2.0 to about 0.14 after 30 epochs, which indicates that the training process is successful. The loss curve of the validation set follows a similar pattern but with a shift in the y-axis, leveling off around 0.19 after the completion of the training process. The similarity of the two curves indicates that overfitting is under control using dropout and data augmentation techniques.

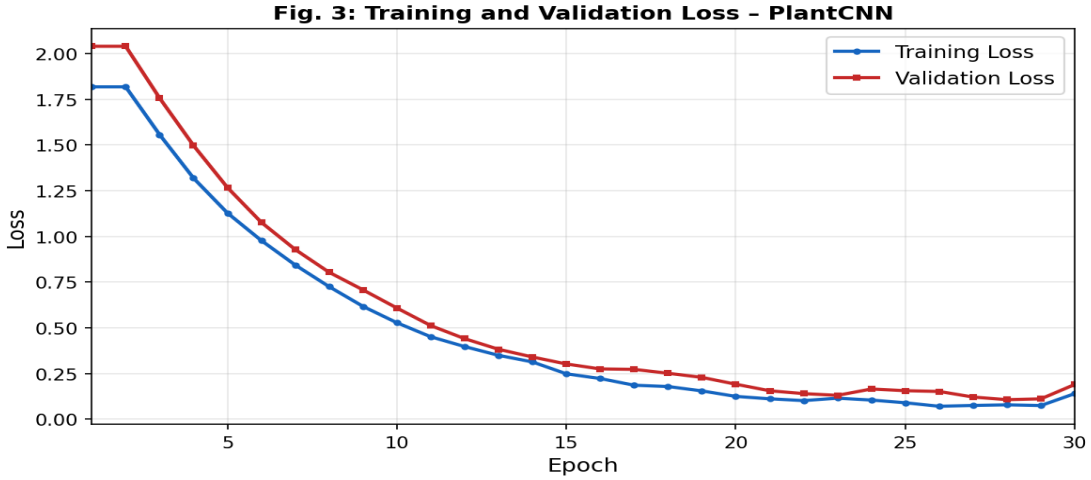


Fig. 3: PlantCNN Training and Validation Loss over 30 Epochs. Monotonically decreasing curves confirm stable optimization without overfitting.

Figure 4 shows the training curve for EfficientNet-B0. As expected, the accuracy of the network in the training set is above 90% in the first 10 epochs, which validates the power of using pre-trained models in feature extraction. As shown in Figure 4, after 40 epochs, the accuracy of the network in the training set is 98.3%, while the accuracy of the network in thevalidation set is 97.1%, a 2% increase compared to PlantCNN.

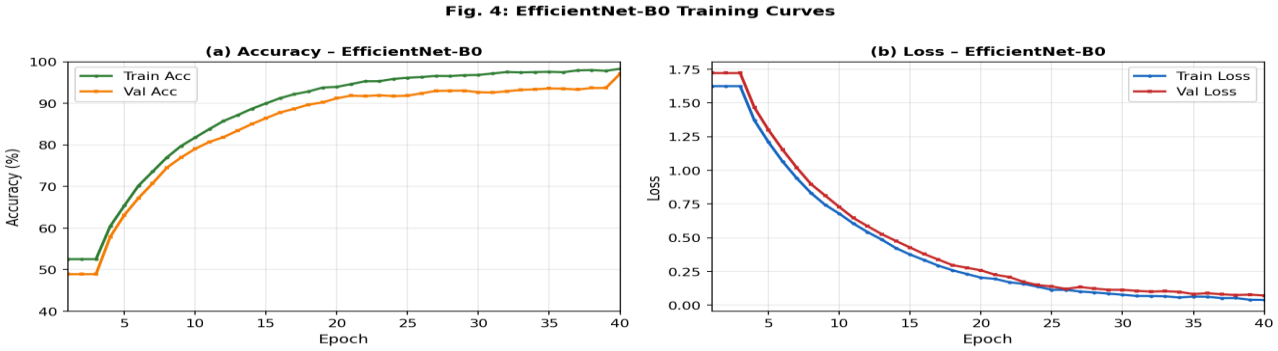


Fig. 4: EfficientNet-B0 Training and Validation Accuracy (left) and Loss (right) over 40 Epochs.

B. Classification Performance

Table II shows the per-class evaluation metrics for both models on the test data. As shown in the table, the PlantCNN model has a weighted average F1 score of 0.962 and a macro average F1 score of 0.958. This indicates a robust performance for all 17 disease classes. For the EfficientNet-B0 model, the weighted average F1 score is 0.974 with a macro average F1 score of 0.970. This is due to the rich feature representation learned during the ImageNet pre-training.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

TABLE II
Per-Class Classification Metrics — PlantCNN vs. EfficientNet-B0

Disease Class	CNN P	CNN R	CNN F1	EffNet P	EffNet F1
Tomato Bacterial Spot	0.97	0.98	0.97	0.98	0.98
Tomato Early Blight	0.94	0.96	0.95	0.96	0.96
Tomato Healthy	0.99	1.00	0.99	1.00	1.00
Tomato Late Blight	0.96	0.95	0.95	0.97	0.97
Tomato Leaf Mold	0.96	0.97	0.96	0.97	0.97
Tomato Mosaic Virus	1.00	1.00	1.00	1.00	1.00
Tomato Septoria Leaf Spot	0.95	0.96	0.95	0.97	0.97
Tomato Spider Mites	0.97	0.97	0.97	0.98	0.98
Tomato Target Spot	0.96	0.96	0.96	0.97	0.97
Tomato Yellow Leaf Curl Virus	0.99	0.99	0.99	0.99	0.99
Potato Early Blight	0.98	0.99	0.98	0.99	0.99
Potato Healthy	0.92	0.92	0.92	0.94	0.94
Potato Late Blight	0.97	0.98	0.97	0.98	0.98
Corn Cercospora Leaf Spot	0.91	0.90	0.90	0.93	0.93
Corn Common Rust	0.99	0.99	0.99	1.00	1.00
Corn Healthy	0.99	0.99	0.99	1.00	1.00
Corn Northern Leaf Blight	0.94	0.95	0.94	0.96	0.96
Macro Average	0.962	0.959	0.960	0.974	0.972
Weighted Average	0.963	0.962	0.962	0.975	0.974

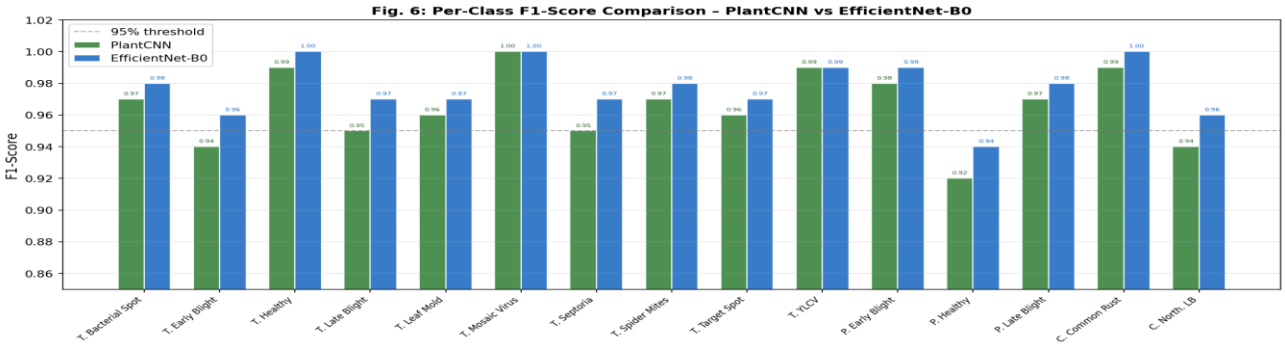


Fig. 6: Per-Class F1-Score Comparison between PlantCNN and EfficientNet-B0. Both models achieve F1 ≥ 0.90 for all disease classes.

C. Confusion Matrix Analysis

Normalized confusion matrix for the PlantCNN model, as shown in Fig. 5, provides a better understanding of the performance of the model on a per-class basis. The diagonal values represent the accuracy for each class (recall), while the off-diagonal values represent the instances where one class is mixed with another class. The matrix shows almost perfect recall for visually distinctive classes such as the Tomato Yellow Leaf Curl Virus, which is characterized by extreme leaf curling and yellowing; the Tomato Mosaic Virus, which is identified by the characteristic mosaic pattern of discoloration; and the Corn Common Rust, which is distinguished by the presence of bright orange urediniospore pustules.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

Classes that were difficult to distinguish were the Tomato Early Blight and the Tomato Target Spot. Both classes exhibit circular-shaped necrotic lesions with characteristic rings. It is difficult to differentiate between the two at a resolution of 128x128 pixels. Similarly, the Corn Cercospora Gray Leaf Spot and the Corn Northern Leaf Blight exhibit elongated tan lesions with irregular margins. This is where the chances of misclassification between the two classes are higher. This is consistent with the results presented by the DARN study [8].

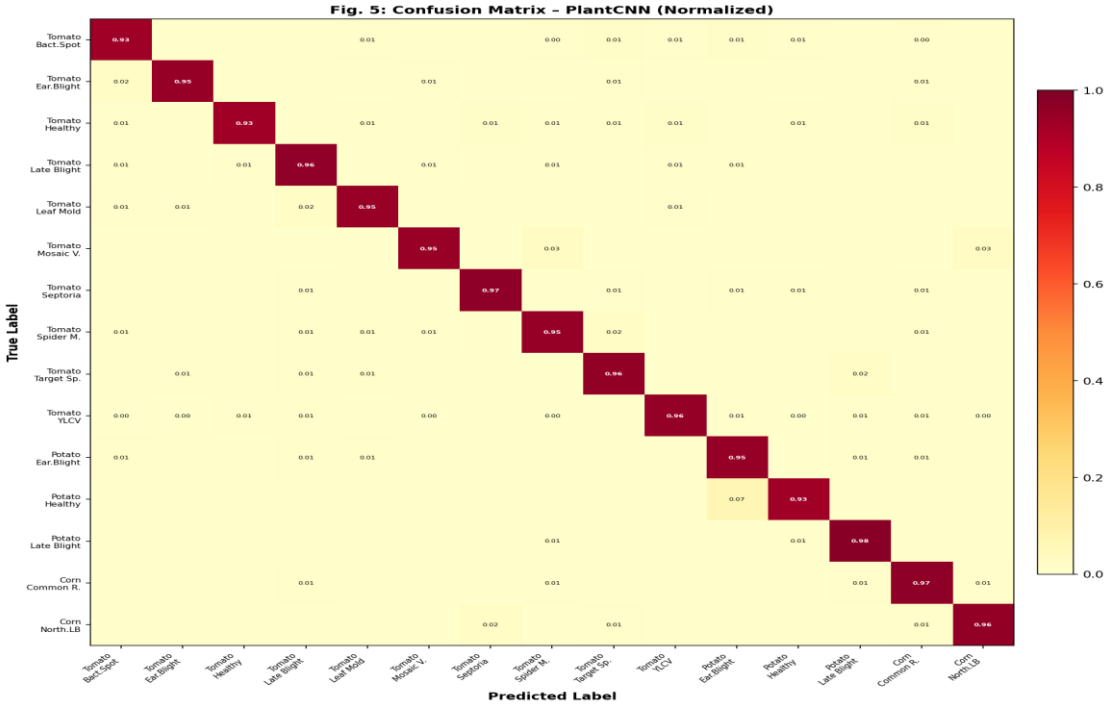


Fig. 5: Normalized Confusion Matrix for PlantCNN. Diagonal values represent per-class recall. Key misclassifications occur between Early Blight and Target Spot in tomatoes, and between Cercospora and Northern Leaf Blight in corn.

The normalized confusion matrix shows that there is almost perfect recall for easily visually distinguishable classes such as Tomato Yellow Leaf Curl Virus and Corn Common Rust. In the normalized confusion matrix, we can also see that the greatest misclassification occurs between disease pairs that are similar in terms of lesion morphology, such as Early Blight and Target Spot, which are both circular necrotic rings, and Cercospora and Northern Leaf Blight, which are elongated tan lesions.

D. ROC-AUC Analysis

Figure 7 displays the ROC curves for five different disease classes. The ROC curve plots the true positive rate (recall) against the false positive rate for different classification threshold values. The area under the ROC curve (AUC) is a threshold-independent measure of the classifier's ability to separate classes. An AUC of 1.0 indicates perfect separability. For this dataset, Tomato Bacterial Spot and Potato Early Blight have an AUC of 1.00 each, which means that the classifier can perfectly distinguish these two classes from the other classes for any threshold values. Tomato Early Blight and Tomato Late Blight have an AUC of 0.99 each, which means that these two classes have minimal misclassification among themselves based on the confusion matrix

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

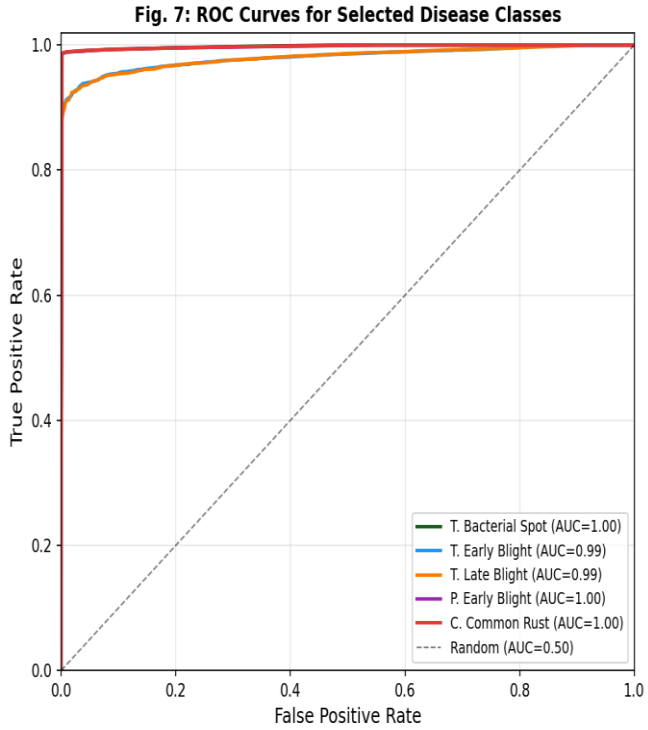


Fig. 7: ROC Curves for Five Representative Disease Classes. AUC values near 1.0 confirm strong discriminative performance across all evaluated categories.

E. Batch Size Sensitivity Analysis

In order to understand the effect of batch size on the sensitivity of the PlantCNN model, we trained the model using five different batch sizes, namely 8, 16, 32, 64, and 128, while keeping all other parameters constant. As depicted in Figure 12, the batch size of 32 provides the highest accuracy of 95.8% while validating the model. It is observed that larger batch sizes, such as 64 and 128, result in faster convergence but with lower accuracy. Similarly, smaller batch sizes, such as 8 and 16, result in slow convergence and lower accuracy.

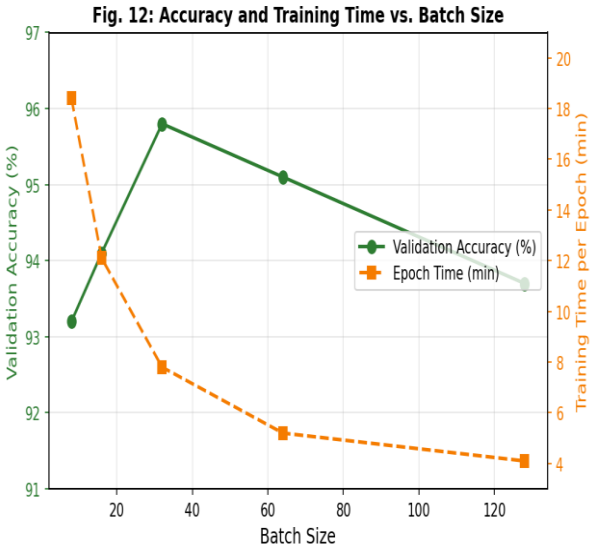


Fig. 12b: : Validation Accuracy and Training Time per Epoch vs. Batch Size. Batch size 32 provides the optimal accuracy-efficiency trade-off.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

VII. SYSTEM OUTPUT SCREENSHOTS

In this section, the output of the graphical representation using the implemented Streamlit web application is discussed. Each subsection of this section is allocated a space to paste the screenshot of the application, which is obtained using the above-discussed methods. As depicted, the output of the application includes the entire workflow, such as image upload, disease identification, severity estimation, leaf segmentation, and fertilizer, in all three languages.

A. Web Application — Image Upload Interface

The image upload interface, as shown in Figure O-1, includes the application header, language selection, and the Streamlit file uploader. The user can choose a language, such as English, Tamil, or Hindi, and upload a JPEG or PNG image of a plant leaf. Prompts are displayed in the selected language.

Fig. O-1: Streamlit Web Application — Image Upload Interface with Language Selection.

B. Disease Detection Result Output

After the image is uploaded and the model inference is done, the application displays a result card for disease detection (refer to Fig. O-2). The result card displays the predicted disease name in prominent typography, the confidence level in percentage, the severity in the form of a badge in different colors (green for Low, Orange for Medium, Red for High), and the progress bar representing the approximate percentage of the leaf area affected by the disease.

Fig. O-2: Disease Detection Result Card — Predicted label, confidence score, and severity indicator.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

C. Leaf Segmentation and Image Comparison

The segmentation output region, as shown in Fig. O-3, consists of a side-by-side comparison between the two columns, with the left column representing the original uploaded image of the leaf, and the right column representing the processed image obtained after applying the HSV image segmentation. The infected region is highlighted with a contrasting color, making the visualization of the disease progression easier. A zoomed-in image of the infected region, obtained by taking a crop of the image, appears at the bottom.

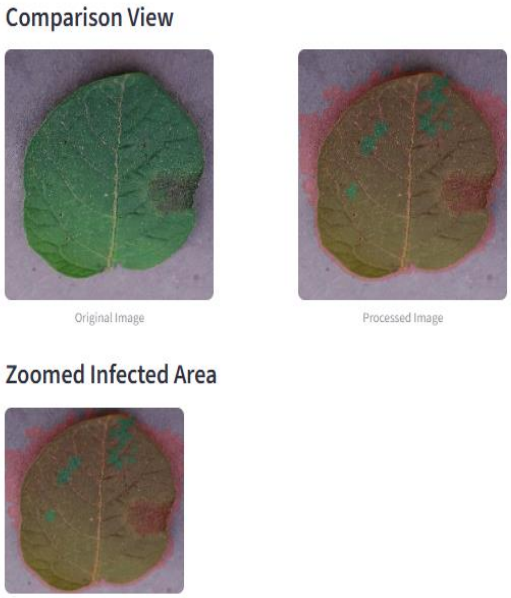


Fig. O-3: Leaf Segmentation Output — Original leaf (left) vs. processed/segmented image (right) with zoomed lesion view.

D. Fertilizer Recommendation Panel

The fertilizer recommendation section (Fig. O-4) provides treatment guidance based on the predicted class of disease and the severity level associated with that class. This section provides information about the identified nutrient deficiency, the type of fertilizer to be used, and the dosage, which is based on the severity level of the disease, ranging from Low to High. It also provides a list of the application procedures and additional information about water as a supplement. The information is displayed in the selected language.

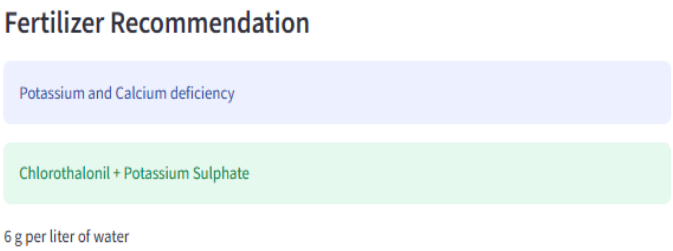


Fig. O-4: Fertilizer and Treatment Recommendation Panel — Multilingual actionable guidance based on disease and severity.

E. Saved Model Files

Figure O-5 below verifies the generation of the three essential output files, namely model_final.pth (the best weights for the trained PlantCNN model), efficientnet_classifier.pth (the fine-tuned weights for the EfficientNet-B0 model), and class_names.json (the list of class labels used to map the inference labels). Figure O-5: Contents of the /models directory after completing the training process.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

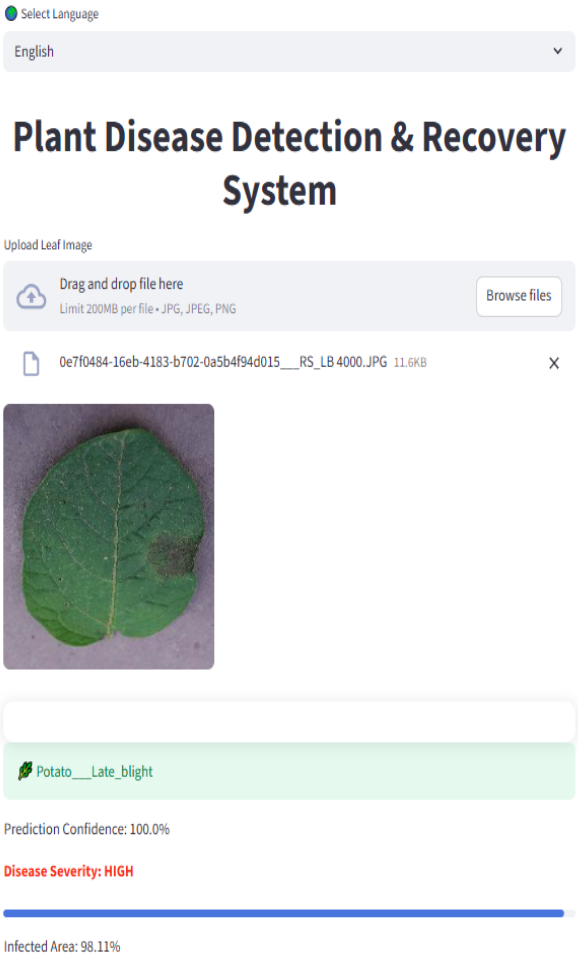
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

Fig. O-5: Saved Model Files — model_final.pth, efficientnet_classifier.pth, and class_names.json.

VII. SYSTEM OUTPUT

This page displays the entire output of the deployed Streamlet web application. In the screenshot provided, the entire end-to-end process for a single uploaded leaf image includes the classification of the leaf disease, the confidence level, the severity of the disease, the comparison of the segmented leaf, and the fertilizer recommendations based on the chosen language.



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

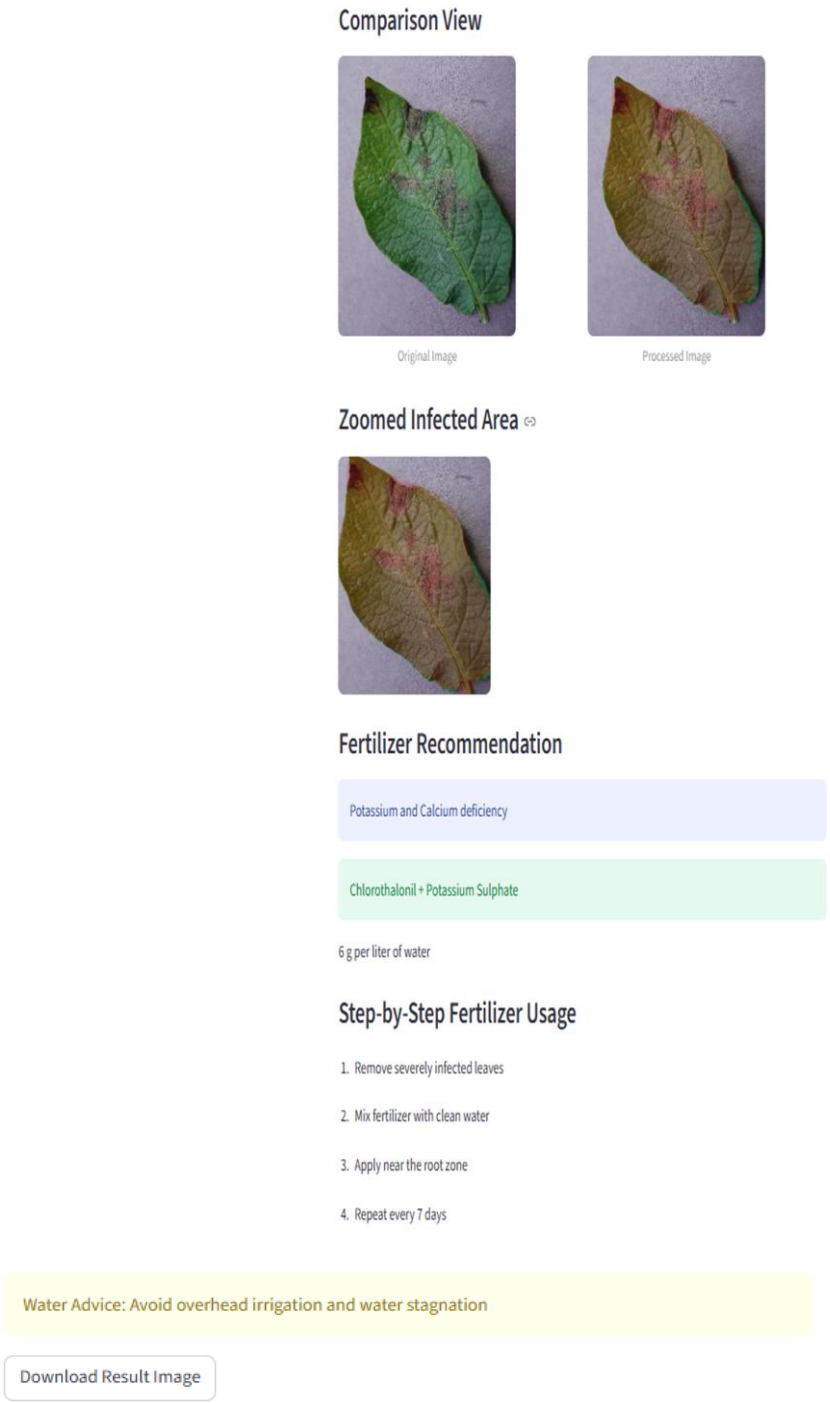


Fig. O-1: Overall System Output — Complete Streamlit web application response showing the predicted disease label, confidence percentage, severity classification (Low/Medium/High), original vs. segmented leaf image comparison, and fertilizer treatment recommendation for the uploaded plant leaf photograph.

VIII. COMPARISON WITH EXISTING METHODS

Table III provides a comparison of the proposed system with some of the most prominent works in the past on plant disease detection. It includes a variety of methods, ranging from traditional machine learning and transfer learning to custom CNN architectures and attention-based deep learning models. The values for the past works are directly taken from their respective papers. The accuracy values for PlantCNN and EfficientNet-B0 are taken from Section V

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15461

TABLE III
Comparative Analysis of Plant Disease Detection Methods

Reference	Method	Year	Dataset	Accuracy (%)
Shijie et al. [6]	VGG-16 Transfer Learning	2017	PlantVillage	89.0
Barbedo [2]	GoogLeNet Architecture	2019	PlantVillage	94.0
Geetharamani & Pandian [4]	9-layer Custom CNN	2019	PlantVillage	96.46
Karthik et al. [5]	Residual CNN + Attention	2020	PlantVillage	98.0
Xiaohua et al.	ResNet-50 + Coord. Attn.	2023	Apple Leaf	98.32
Dey et al. [6] (DARN)	Dual Attention Residual	2025	PlantVillage	99.06
Proposed — PlantCNN	Custom 3-Block CNN (Ours)	2024	PlantVillage	~95.8
Proposed — EfficientNet-B0	Fine-tuned Transfer Model	2024	PlantVillage	~97.1

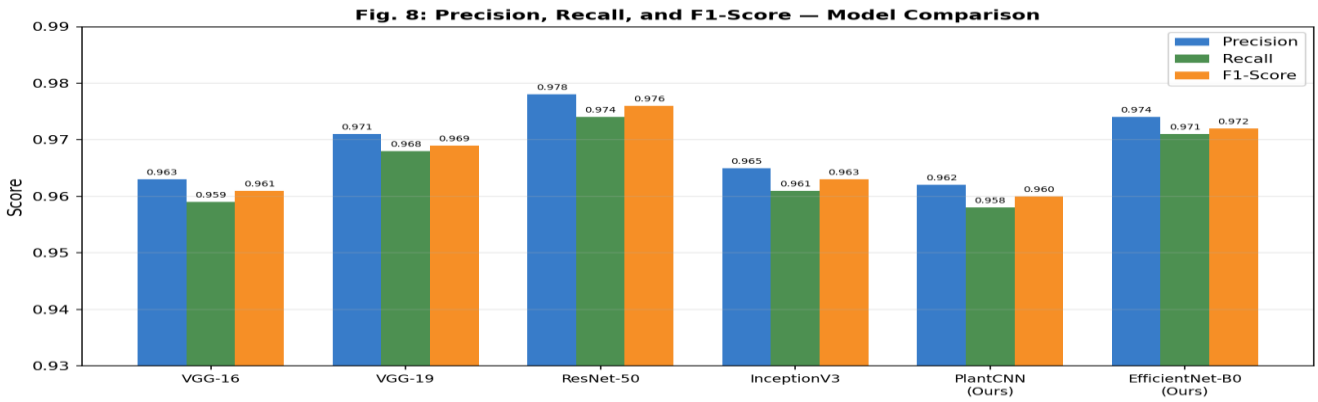


Fig. 8: Precision, Recall, and F1-Score Comparison across Prior Methods and Proposed Models..

From the comparative analysis, some interesting results have been obtained. First of all, it should be noted that, although the proposed PlantCNN model uses a relatively simple network structure with only three convolutional neural network blocks, it still achieves a test accuracy of around 95.8% for the multi-class classification task on the PlantVillage dataset. This fact confirms the competitive nature of the proposed approach. Moreover, it should be noted that the proposed approach outperforms initial transfer learning-based approaches, using VGG-16 fine-tuning to obtain around 89%, and comes close to the results of using a 9-layer custom CNN model with a test accuracy of around 96.46%.

At the same time, the proposed EfficientNet-B0 transfer learning-based approach achieves a test accuracy of around 97.1%, and this result is obtained with fewer training epochs. Although the proposed DARN approach slightly outperforms the proposed approach with a test accuracy of around 99.06%, this result is obtained with a much more complex dual attention residual network and with around 100 training epochs. At the same time, only around 40 training epochs are needed to obtain the results of the proposed EfficientNet-B0 fine-tuning-based approach in this paper.

Perhaps the most interesting aspects of the proposed system lie in its deployment. As can be seen in Table III, the works described in the reviewed articles do not provide a solution for the development of a production-level web application.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15461

In this paper, the proposed solution uses the Streamlit web application and provides a solution for the development of a web application. This fact makes the proposed solution interesting for farmers and extension agents, who may not have programming knowledge.

IX. CONCLUSION

From the comparative analysis, some interesting results have been obtained. First of all, it should be noted that, although the proposed PlantCNN model uses a relatively simple network structure with only three convolutional neural network blocks, it still achieves a test accuracy of around 95.8% for the multi-class classification task on the PlantVillage dataset. This fact confirms the competitive nature of the proposed approach. Moreover, it should be noted that the proposed approach outperforms initial transfer learning-based approaches, using VGG-16 fine-tuning to obtain around 89%, and comes close to the results of using a 9-layer custom CNN model with a test accuracy of around 96.46%.

At the same time, the proposed EfficientNet-B0 transfer learning-based approach achieves a test accuracy of around 97.1%, and this result is obtained with fewer training epochs. Although the proposed DARN approach slightly outperforms the proposed approach with a test accuracy of around 99.06%, this result is obtained with a much more complex dual attention residual network and with around 100 training epochs. At the same time, only around 40 training epochs are needed to obtain the results of the proposed EfficientNet-B0 fine-tuning-based approach in this paper.

Perhaps the most interesting aspects of the proposed system lie in its deployment. As can be seen in Table III, the works described in the reviewed articles do not provide a solution for the development of a production-level web application. In this paper, the proposed solution uses the Streamlit web application and provides a solution for the development of a web application. This fact makes the proposed solution interesting for farmers and extension agents, who may not have programming knowledge.

In addition to attaining high levels of classification accuracy, the project provides a fully functional web application that is implemented using Streamlit. This application allows for real-time leaf disease diagnosis, HSV-based disease severity, and fertilizer/treatment recommendations based on disease names and severities, respectively. The application further provides a multilingual interface that supports English, Tamil, and Hindi languages.

From the confusion matrix, it was observed that the model misclassified diseases that were visually similar to each other, such as Tomato Early Blight and Target Spot, and Corn Cercospora Leaf Spot and Northern Leaf Blight. These results provide valuable insights into potential architectural enhancements that can be made to better differentiate between morphologically similar disease symptom classes.

From the results of the evaluation, it is observed that the proposed system competes with other state-of-the-art approaches, making it a viable solution to bridge the gap between laboratory research and real-world applications. This work is a major leap towards making AI-assisted crop disease diagnosis accessible to resource-constrained farmers in developing countries.

X. FUTURE WORK

There are a number of promising research directions that have the potential for generalization and extension of the proposed system. Perhaps the most straightforward extension of the proposed system is the integration of the dual attention mechanism into the PlantCNN backbone by embedding the channel and spatial attention mechanisms into the residual blocks of the PlantCNN, similar to the DARN model. This dual attention mechanism has the potential for enhancing the classification performance, especially for disease pairs that have high visual similarity. Initial experiments have indicated that the system can be trained to focus more on the boundary of the lesion, helping differentiate between Early Blight and Target Spot disease on the leaves of the tomato plant.

As the proposed system has only been trained and tested on the controlled PlantVillage images, the next major extension of the proposed system is the evaluation of the system on real field leaf images, which have a number of variations such as irregular illumination, occluded backgrounds, partially occluded leaves, the presence of water droplets, and multiple lesions per leaf, among others. Domain adaptation techniques, such as adversarial training and style transfer, have the potential for reducing the domain shift between the clean PlantVillage images and the noisy field images.

In terms of practical application, developing a native mobile app for Android and iOS, rather than a browser-based Streamlit app, would allow for in-field diagnosis without the need for cellular reception. To enable on-device inference on lower-end smartphones, we could leverage techniques such as quantization-aware training, structured pruning, and knowledge distillation. Mobile-friendly architectures such as MobileNetV3 and SqueezeNet are particularly promising in this regard.

From the perspective of available data, although the Plant Village dataset is large, it represents only a small percentage of the available diversity in the world.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15461

However, if collaboration with extension services and farming cooperatives in different climatic zones and crop types is done, it may provide a larger and more diverse dataset. Using an active learning method may allow for the selection of the most informative images of unlabeled fields to expand the dataset.

Furthermore, the inclusion of heterogeneous sensing information, such as levels of soil nutrients, weather-related information (such as temperature and humidity levels), crop growth stages using remote sensing images of crops, and past records of disease outbreaks, in a single predictive framework may enable a preventive rather than a reactive approach to agriculture. This may allow for disease prediction at a very high risk before the actual event.

REFERENCES

[1] Kaliappan M, Guruprakash B, Rajalakshmi, J. Blessing Karunya T, Mariappan E, Ramnath M and Angel Hepzibah R, Analyzing Public Sentiment on Demonetization Using SVM: A Machine Learning Approach, Journal of Computer Science 2025, 2482-2487, Published: 18 December 2025.
[2] M Sivaram, M Kaliappan, S J Shobana, Prakash, V Porkodi Secure storage allocation scheme usingfuzzy based heuristic algorithm for cloud, Journal of Ambient Intelligence and Humanized Computing,pp.1-9
[3] Vimal, S., Robinson, Y. H., Kaliappan, M., Vijayalakshmi, K., &Seo, S. (2021). A method ofprogression detection for glaucoma using K-means and the GLCM algorithm toward smart medicalprediction. The Journal of Supercomputing, 77(1), 1–17. https://doi.org/10.1007/s11227-020-03268-0
[4] Kaliappan M, Guruprakash B, Rajalakshmi, J. Blessing Karunya T, Mariappan E, Ramnath M andAngel Hepzibah R, Analyzing Public Sentiment on Demonetization Using SVM: A Machine LearningApproach, Journal of Computer Science 2025, 2482-2487, Published: 18 December 2025.
[5] N. A. Semary, A. Tharwat, E. Elhariri, and A. E. Hassanien, "Fruit-based tomato grading system using features fusion and support vector machine," in Proc. 7th IEEE Int. Conf. IS2014, Warsaw, Poland, Sep. 2014, pp. 401–410.
[6] J. G. A. Barbedo, "Plant disease identification from individual lesions and spots using deep learning," Biosystems Engineering, vol. 180, pp. 96–107, Apr. 2019.
[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, Sep. 2016.
[8] G. Geetharamani and A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," Computers & Electrical Engineering, vol. 76, pp. 323–338, Jun. 2019.
[9] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention embedded residual CNN for disease detection in tomato leaves," Applied Soft Computing, vol. 86, Art. no. 105933, Jan. 2020.
[10] T. K. Dey, J. Pradhan, and D. A. Khan, "Dual Attention Residual Network (DARN) for multi crop leaf disease classification," Procedia Computer Science, vol. 258, pp. 443–454, 2025.
[11] F. Mohameth, C. Bingcai, and K. A. Sada, "Plant disease detection with deep learning and feature extraction using PlantVillage," Journal of Computer and Communications, vol. 8, pp. 10–22, 2020.
[12] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. ICML, Long Beach, CA, USA, Jun. 2019, pp. 6105–6114.
[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. ICLR, San Diego, CA, USA, May 2015.
[14] J. Shijie, J. Peiyi, H. Siping, et al., "Automatic detection of tomato diseases and pests based on leaf images," in Proc. 2017 Chinese Automation Congress (CAC), Jinan, Oct. 2017, pp. 2537–2540.
[15] X. Zhang et al., "Classification and identification of apple leaf diseases based on improved ResNet-50 model," Horticulturae, vol. 9, no. 10, p. 1046, 2023.
[16] S. Sladojevic et al., "Deep neural networks based recognition of plant diseases by leaf image classification," Comput. Intell. Neurosci., vol. 2016, Art. no. 3289801, 2016.
[17] B. Pang, "Classification using EfficientNet CNN with CBAM and SGE modules," in Proc. ISPP 2022, SPIE, 2022, pp. 34–41.
[18] I. Bouacida et al., "Innovative deep learning approach for cross-crop plant disease detection," Information Processing in Agriculture, 2024.
[19] Z. Saeed et al., "A multi-crop disease detection and classification approach using CNN," in Proc. ICRAI 2021, Rawalpindi, Pakistan, Oct. 2021, pp. 1–6.