Journal of Theoretical and Applied Information Technology
15th March 2025. Vol.103. No.5
© Little Lion Scientific

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

LEAFDISEASENET: A NOVEL CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE FOR DETECTING LEAF DISEASES IN AGRICULTURAL CROPS

[1]CH.SANTOSHI, [2]SURESH KUMAR MANDALA

Research Scholar, school of computer science and Artificial intelligence, SR University, Warangal.
Assistant Professor school of computer science and Artificial intelligence, SR University, Warangal.
santoshijuluri@gmail.com, mandala.suresh83@gmail.com

ABSTRACT

For productive agriculture and healthy crops, plant disease detection is essential. Conventional disease detection techniques are human, arbitrary, and error-prone and require automated solutions. Specifically, convolutional neural networks (CNNs) are deep learning models that have demonstrated great promise for identifying plant diseases; however, they are primarily model-specific for a particular kind of plant and disease, restricting them from generalizability over diverse crops and conditions. Moreover, class imbalance, environmental noise, and variability of disease symptoms make it even harder for these models to perform. In response to these problems, we present a sophisticated, integrated deep-learning framework called LeafDiseaseNet. Our proposed architecture includes a squeeze-and-excitation (SE) block with residual connections and data augmentation, enabling the model to be generalized for different plant diseases and environmental conditions. LeafDiseaseNet is evaluated on the Plant Village dataset and yields 97.68%, which outperforms current best practices regarding precision and resilience. It also exhibits good quality regarding high imbalanced class recognition and absolute environmental noise robustness, which may become promising approaches for implementing plant disease recognition in real-world agriculture situations. The framework aims to provide an efficient and scalable strategy for automating plant disease detection that can play a massive part in precision farming by making information about the disease available promptly and with sufficient time for action.

Keywords: *Squeeze-And-Excitation, Deep Learning, Plant Disease Detection, Convolutional Neural Networks, and Residual Connections*

1. INTRODUCTION

Plant disease detection is a vital component of modern agriculture. Early disease detection is crucial for the development of sustainable agriculture because it can negatively impact crop output and quality. The ability to detect pests early means there is time to intervene and prevent more harsh chemicals from being used, keeping the plant healthier. Plant disease diagnosis is a tedious process involving visual inspection in traditional methods, and it is highly subjective and prone to human error. Automated plant disease detection has received much attention due to the advancement of computer vision and deep learning, which offer a quicker and more precise substitute for conventional techniques. As a result, Convolutional neural networks, or CNNs, are one type of deep learning model that has been used to accurately predict objects in photos by removing pertinent information from the provided images.

Recent literature demonstrates the application of several architectures, including VGG16, ResNet, and Inception, on plant disease detection tasks. For example, Richardson et al. introduced A VGG16.

The model for rice leaf disease diagnosis was based on [1], which achieved 99.94% accuracy. Similarly, Rashid et al. MMF-Net: CNN for Classification of Maize Leaf Disease was proposed by [2] with the best accuracy of 99.23% for maize leaf disease classification. Although these models have been successfully developed, they are typically crop or disease-specific and not generalizable across plant species or environmental conditions. Moreover, the performance of these models is also affected by class imbalance, disease variability, and environmental noise.

Although automated methods for identifying plant diseases can help ensure the productivity of agricultural systems, their operational use has limitations in areas such as generalizability, robustness to environmental noise, and imbalanced data sets. Various accurate state-of-the-art models, like VGG16, ResNet50, and Inception, are developed; however, their applicability is usually limited to specific crops or convenient environments. In addition, many models do not incorporate feature recalibration, hindering optimal feature representation when multiple diseases share similar visual presentations. A more efficient, generalized, and scalable deep learning model that provides higher accuracy, is resistant to environmental fluctuations, and is better robust to dataset imbalance is required.

The current study fills these gaps by introducing an enhanced deep learning model, LeafDiseaseNet, capable of enhancing plant disease detection systems' resilience and generality. LeafDiseaseNet is a new architecture built with residual connections, an SE (Squeeze-and-Excitation) block, and data augmentation methods to allow it to perform well on many kinds of plant species and plant diseases with high accuracy. We focus our study on designing a robust model capable of performing multiple plant disease detection tasks but still suitable in realistic agricultural scenes. The proposed research provides a new deep-learning model for plant disease recognition, which overcomes the current limitations in improving existing models regarding plant species generalization, environmental noise handling, and imbalanced dataset learning. It also contains an extensive experimental analysis based on the PlantVillage dataset, demonstrating the utility.

The following is the paper's structure: Section 2 provides an overview of current studies on deep-learning models for plant disease detection. The third section describes the suggested approach, including the architecture of LeafDiseaseNet and methods to enhance its performance. The experimental findings are shown in Section 4, and comparisons between LeafDiseaseNet and the most advanced models. Section 5 discusses the study's limits and conclusions as well as the research's ramifications. Section 6 wraps up the work and discusses potential avenues for future research, such as using and improving the model.

2. RELATED WORK

With an emphasis on diverse models and methodology, this literature review examines current developments in deep learning techniques for plant disease diagnosis. Richardson et al. [1] created a unique VGG16 model with high accuracy for rice leaf disease diagnosis. Future research will focus on developing instruments for disease diagnostics and expanding this approach to other crops. Bouacida et al. [2] demonstrated a revolutionary deep learning system that detects plant illnesses with better accuracy using a tiny Inception model. Future studies will investigate sliding window techniques and enhance adaptation to real-world settings. Simhadri et al. [3] proposed further work on federated learning and remote sensing and assessed deep learning techniques for rice leaf disease detection. Rashid et al. [4] introduced MMF-Net, a CNN-based model that uses multi-contextual characteristics to achieve a higher level of accuracy in classifying maize leaf diseases. Future efforts may improve model robustness and apply it to more significant agricultural or Internet of Things scenarios. Shamasneh et al. [5] presented a conformable polynomials technique that yields increased accuracy in SVM classification when used to extract tomato leaf texture information. Future research may examine larger datasets and practical uses.

Polly and Devi [6] provided a multi-stage system that combines CNN, UNet, YOLOv8, DeepLabV3+, and other models to detect leaf illness and accurately suggest treatments with better accuracy. Long-term projects could concentrate on growing datasets and practical uses. Umar et al. [7] presented a highly accurate upgraded YOLOv7 model incorporating SimAM, DAiAM, and better MPConv for detecting tomato leaf disease. Future research could examine more widespread illness categories and practical uses. Sujatha et al. [8] contrasted ML and DL approaches for identifying citrus disease and found that DL models—particularly VGG-16—perform more accurately than ML methods. Future research may examine different crops and model enhancements. Zhang et al. [9] improved Faster RCNN for detecting tomato leaf disease, increasing speed and accuracy by 2.71% and feature extraction and clustering accuracy. Khan et al. [10] suggested a five-step process with enhanced segmentation and feature extraction that detects cucumber leaf disease accurately.

Algani et al. [11] presented ACO-CNN, which uses ant colony optimization and deep learning to increase accuracy in identifying plant diseases. Ahmad et al. [12] identified gaps and guided future tool development and research by reviewing 70 deep-learning papers to control plant diseases. Panchal et al. [13] created a deep learning model with 93.5% accuracy to classify plant illnesses using CNNs. Future research will incorporate therapy recommendations and improve data gathering. Moupojou et al. [14] presented FieldPlant, a recently created dataset of 5,170 field photos annotated by experts. Upcoming tasks include enhancing dataset quality and model accuracy. Shovon et al. [15] presented PlantDet, a deep ensemble model that diagnoses betel leaf and rice illnesses more accurately than earlier techniques. The dataset's expansion and the model's improvement may be the main topics of future development.

Kotwal et al. [16] examined the developments in deep learning for plant disease diagnosis, pointed out data shortages and disease localization, and made recommendations for further development. Shewale and Daruwala [17] enhanced real-time datasets and CNN for automated leaf disease diagnosis, resolving model constraints; future research will concentrate on broader applications. Nikith et al. [18] compared the performance of SVM, KNN, and CNN models for detecting leaf disease; CNN was shown to be superior with higher accuracy; future studies will focus on improving the performance of SVM and KNN. Ahmad et al. [19] evaluated various deep-learning models for diagnosing maize diseases, and DenseNet169 proved to be the most successful, with 81.60% accuracy. Subsequent research endeavors ought to go into merging datasets and refining field-deployable systems. Datta and Gupta [20] created a deep CNN with increased classification accuracy for tea leaf illnesses. In the future, additional crops will be included in the model, its identification of specific diseases will be improved, and it will be integrated with IoT devices.

Dahiya et al. [21] examined eight deep-learning models and determined that ResNet50 and ResNet101 were the most successful at detecting plant diseases. Additional datasets and model refinement may be possible in future research. Umamageswari et al. [22] created software with a high-level accuracy rate for real-time plant disease diagnosis utilizing Fuzzy C-means, SIFT, and LSTM. Further research might improve detection algorithms and cover other illnesses. Jacklin and Murugavalli [23] examined deep learning and machine learning methods for identifying plant diseases, emphasizing the superiority of deep learning. Further research might investigate other categorization techniques for thorough illness detection. Rao et al. [24] accurately detected grape and mango leaf diseases using the deep learning model AlexNet. Challenges with real-time detection and the diversity of datasets are limitations. Some upcoming tasks include developing a recommendation system, extending the scope of illness classifications, and implementing autonomous drones. Zhong and Zhao [25] accurately determined apple leaf diseases using DenseNet-121 with regression, multi-label classification, and focus loss algorithms. Forthcoming projects will solve data imbalance problems and extend these techniques to more plant diseases.

Noon et al. [26] analyzed 45 deep-learning methods for classifying plant leaf stress in 33 crops, emphasizing their advantages and disadvantages. Future research should concentrate on integrating portable devices, background removal, and utilizing real-world data. Tiwari et al. [27] demonstrated a deep-learning model that can identify plant illnesses from leaf photos with better accuracy. Extending the dataset will be part of future efforts to improve performance under various circumstances. Yadav et al. [28] created a CNN model with increased accuracy in identifying peach bacteriosis. Future research will involve utilizing this model in real-time in the field by combining it with UAVs. Li et al. [29] emphasized the improvements made by deep learning in identifying plant diseases. Still, it raises concerns about the dataset's stability and hyperspectral imaging's limitations. Atila et al. [30] demonstrated the need for more model assessment while pointing out the greater accuracy of EfficientNet compared to other models for categorizing plant diseases.

Thangaraj et al. [31] suggested using a CNN based on transfer learning to identify tomato leaf disease and found that the Adam optimizer works best. The model's application to other plants and severity identification are future projects. Ngugi et al. [32] discussed deep learning and image processing for diagnosing plant diseases, pointing to the difficulties with field performance and variable datasets. Subsequent research should concentrate on various datasets, miniature models, background elimination, and larger plant sections. Abed et al. [33] suggested a deep learning architecture for precise bean leaf disease detection that uses U-Net and other models. Potential misdiagnoses and expensive therapies are among the limitations. Future research should concentrate on handling various environmental situations and increasing accuracy. Lakshmi and Savarimuthu [34] presented an improved EfficientDet framework that uses transfer learning to identify plant diseases effectively. Slower mAP and the requirement for more fantastic technology are limitations. Future research should focus on increasing computing efficiency and accuracy. Hu et al. [35] improved the identification of tea leaf blight by utilizing Retinex and deep learning techniques. Sensitivity to picture quality is one limitation. Future research should look for ways to increase the accuracy of severity and detection even further.

Sachdeva et al. [36] suggested a deep CNN model that uses Bayesian learning to identify plant diseases with performance improvement. Subsequent investigations may investigate incorporating nanoscale materials to augment detection and elevate precision. Zhou et al. [37] suggested a reorganized residual dense network that requires less computing power and achieves 95% accuracy in identifying tomato leaf diseases. Further optimization and broader applications can be the focus of future research. Divakar et al. [38] used an ensemble approach and SMOTE to balance an unbalanced apple leaf disease dataset, demonstrating that EfficientNetB7 is the optimal classifier. Prediction accuracy could be improved, and future studies should address dataset imbalances. Jiang et al. [39] improved VGG16 to achieve high accuracy in multi-task learning for detecting wheat and rice leaf diseases. Future research should increase the diversity of datasets and investigate more illness kinds. Goyal et al. [40] demonstrated a deep-learning model that outperforms VGG16 and ResNet50 in classifying wheat illnesses, achieving increased accuracy. Expanding the variety of diseases and improving the robustness of the model are potential areas of future investigation. Model innovations are also found for image processing innovations in [42] and [43]. More deep-learning optimizations are also found in [44] and [45]. Novel deep learning-based optimized ideas are also found in [46] and [47]. Recent studies highlight the success of plant diseases in various crops, which can be identified using deep learning models such as CNNs, VGG16, ResNet, and EfficientNet. While deep learning models for plant disease detection are a state-of-the-art technique, there are still four main limitations in the current state of models in the domain: (i) the generalizability of the models is generally low across multiple species, (ii) they are sensitive to environmental changes (iii) in some instances, the datasets they are trained on are imbalanced (iv) the faster models underperform taking up a lot of computational resources. Most previous works (e.g., VGG16, ResNet50, Inception, EfficientNet) show high accuracy but are usually designed for specific crops or experimental laboratories with a small-scale data set, rendering their applicability limited. Additionally, most prevailing methods do not inculcate any high-level feature recalibration mechanism that can ensure better feature disambiguation on visually similar diseases. This paper fills in these gaps by presenting LeafDiseaseNet. This new deep learning architecture combines Residual Connections with Squeeze-and-Excitation (SE) architecture to enhance feature representation and improve generalization over a wide range of species and environmental conditions. Moreover, it reduces class imbalance and noise by applying data augmentation methods, with application to accurate agricultural field data. Our work is an essential step of bridging the gap between lab-derived disease classification models and fast, field-deployable plant disease detection solutions for agricultural implementation.

3. PROPOSED FRAMEWORK

In this research, we offer LeafDiseaseNet, a network that uses deep learning to detect and classify plant diseases automatically. The framework aims to make it easier to diagnose plant diseases accurately and effectively using pictures of leaves. The flow of this methodology shows the deep learning model, training technique design, and the dataset collection and preparation phase (Figure 1). Using images of healthy and damaged leaves from various plant diseases, we used this dataset in our investigation. Several data-processing procedures were included to ensure the model performs and generalizes well. Standard preprocessing steps were performed, such as adjusting the images to an equal dimension and adjusting the values of pixels between 0 and 1. It also featured data augmentation techniques [rotate, flip, zoom, and shift] to improve generalization and lessen overfitting. This helped the model to learn broader features from the data by mimicking changes in the conditions in the image captured.

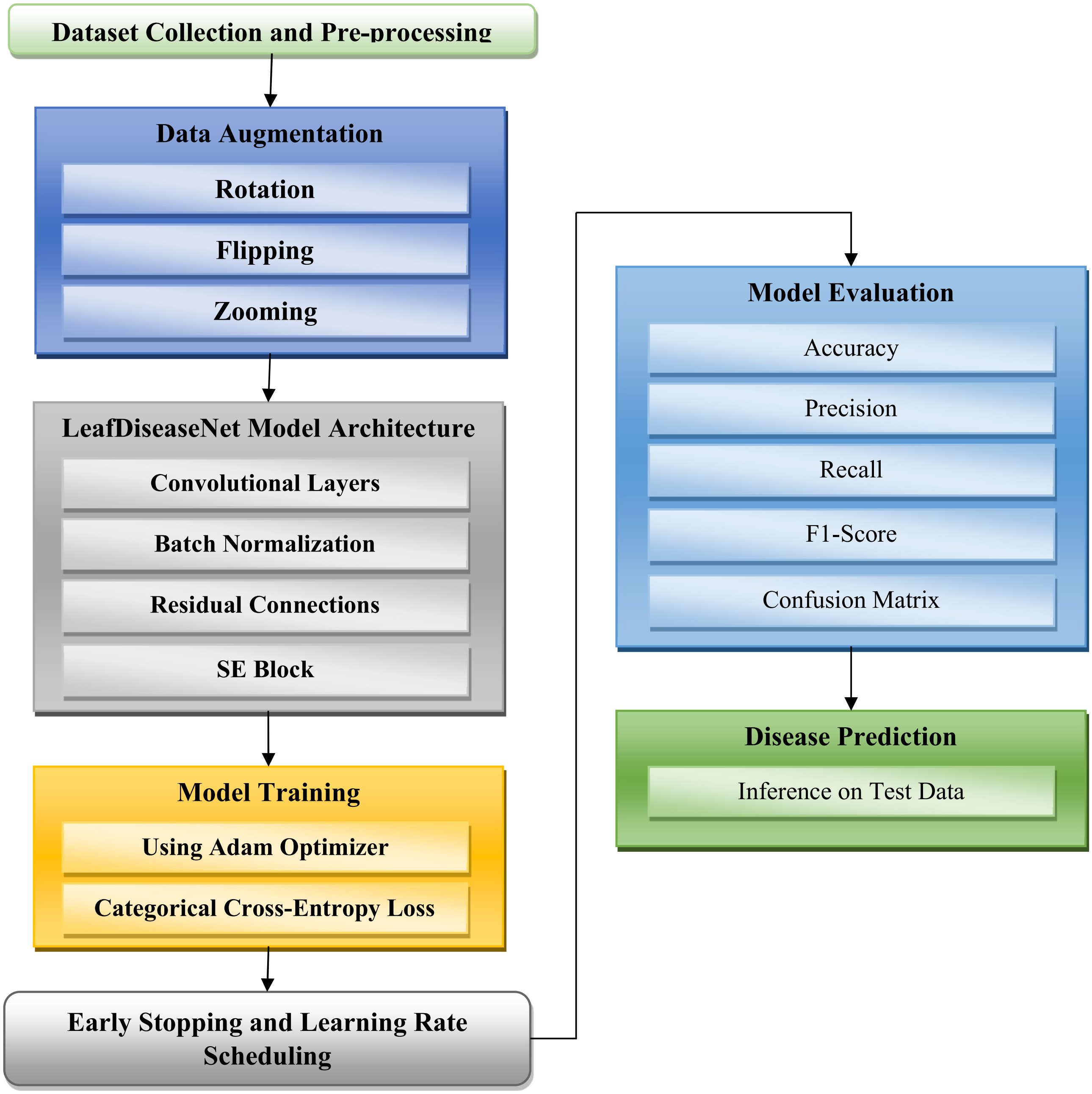


Figure 1: Methodology For Detecting Plant Diseases Using The Leafdiseasenet Framework

The key element of the methodology is the development of LeafDiseaseNet, an improved CNN. The model architecture includes several consecutive photos. The two images are detected using convolutional layers with max-pooling layers in between. The residual link, one of this architecture's noteworthy innovations, resolves the vanishing gradient problem, enabling the network to be deeper without losing performance. A batch norm was incorporated after each conv block to increase training speed and stability. A second squeeze-and-excitation (SE) block was incorporated into the design, enabling the network to alter feature responses automatically. This would allow the network to concentrate more on relevant sensitivity areas of the image, improving its discriminative power. The Adam optimizer with categorical cross-entropy loss, which may be used for multi-class classification applications, was used to train our model. We evaluated the model's performance on the validation set during each training session. To avoid overfitting and ensure that the model produced a dependable output, we also incorporated an early-stopping approach that stops training when the validation loss does not improve for a predefined amount of epochs. During training, we also dynamically changed the learning rate using a learning rate scheduler to ensure the model converges efficiently.

Images not in the training set were used to train and assess the model. Using the same pipeline as during training with the test photos, the model's output predictions were obtained, and the class with the highest probability was the predicted sickness. Several necessary measures were calculated to assess the model's performance, including accuracy, precision, recall, F1 score, etc. Confusion matrices were produced to visually show how accurately the model could define the type of disease.

Successfully implemented, the LeafDiseaseNet framework can improve plant disease detection by providing faster and more reliable tools in the agricultural domain. This classification-based automation of leaf disease detection could benefit farmers, researchers, and agriculturalists by allowing them to quickly identify crop health and predict crop yield. This study used a detailed methodology, illustrated in Figure 1, highlighting the systematic pathway to achieve the results.

3.2 The Deep Learning Model We Proposed

LeafDiseaseNet architecture, as depicted in Figure 2, is designed to extract and classify image features. A deep CNN then does the feature extraction using multiple layers of convolutions. These layers progressively learn to identify more complex visual features, starting with simple characteristics like edges and textures to more intricate visual elements like forms and object components as the network depth increases. Batch normalization layers follow all convolutional layers to normalize outputs of previous layers for better stability and faster training, which means more rapid convergence of the model.

The architecture also includes max−pooling layers, which downsample the features but retain the most essential information. Such dimensionality reduction allows the model to concentrate on the most critical features, resulting in better efficiency and generalization. Deeper layers of the model use more filters, detecting more abstract patterns in the data, which is essential to distinguish between different plant diseases that might have very subtle visual differences.

Residual connections are incorporated into the architecture to avoid the issue of vanishing gradients during training. These links enable the output of a shallower layer to skip a few intermediate layers and be added directly to the production of a deeper layer. This aids the model in remembering the essential features from the previous layers and mitigates the training problem in deeper networks where gradient flow is compromised. In addition, the architecture incorporates a SE block that reweights the feature. It helps the network emphasize the most valuable features by varying the significance of different channels in feature maps. SE blocks operate by squeezing (i.e., reducing) the feature maps' spatial dimensions to produce a channel-wise description. This is followed by a learned scaling on a per-channel basis to help the model become focused on the most critical features, enhancing the capacity of the model to rank the most informative features.

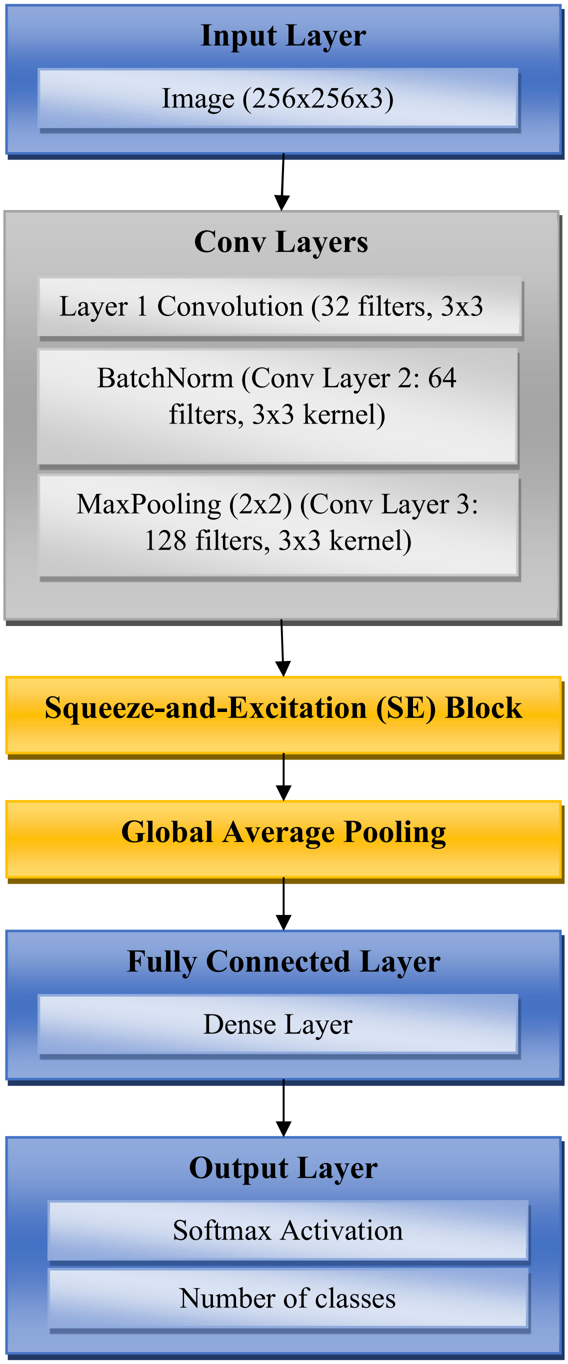


Figure 2: Architecture Of Leafdiseasenet For Plant Disease Detection

Once the features have been extracted and refined using convolutional, pooling, and attention layers, the network uses Downsampling, and the final feature map's dimensions are accomplished using global average pooling. Here, a completely connected thick layer receives the 1-dimensional feature vector created by this pooling layer from the 2-dimensional feature map. Here, this dense layer acts as a bridge where all features will be connected to a higher-level representation, thus enabling the network to make judgments using the provided features. Softmax: To allow the model to predict the most likely classes (illness), the output layer shows a probability distribution of the data using the softmax activation function. In summary, the overall structure of LeafDiseaseNet is built to input leaf images and output predicted labels efficiently with state-of-the-art tricks, including data augmentation, residual connections, and attention mechanisms to achieve better performance and generalization capabilities. Such architectural features support LeafDiseaseNet's efficacy. Detecting plant diseases is a difficult task. Table 1 presents the notations used in the proposed methodology.

Table 1: Notations Used

Symbol	Description
I	input picture with H x W x C dimensions
H, W, C	The supplied image's height, breadth, and number of RGB channels
$CapX_i$	with input image from the dataset
y_i	Disease class label for the i-th picture
C	Total number of classes (diseases) in the dataset
X	Feature map after applying convolution operation
K	Convolution kernel/filter
Y	Output feature map from the convolution operation
μ, σ^2	Mean and variance for batch normalization
γ, β	Learnable parameters for batch normalization
f(x)	Activation function (e.g., ReLU)
$Y_{i,j,k}$	An element located at (i, j) in the channel k feature map
F(x)	The residual block's output
F(x)+x	Residual connection output (skip connection)
z	Channel-wise feature vector after global average pooling
r	Channel attention vector from the Squeeze-and-Excitation block
$s_c(x)$	Class score for class c
L	Cross-entropy loss function in categories
α	The optimizer's learning rate
ϵ	Small constant for numerical stability in batch normalization
β	Batch size used in training
epochs	Number of training epochs

3.3 Mathematical Perspective

The proposed methodology for LeafDiseaseNet leverages a CNN to use photos of leaves to identify and categorize plant diseases. The network starts by taking in a picture of size $I \in \mathbb{R}^{H \times W \times C}$, where C is the number of color channels (three for RGB photos) and H and W stand for the image's height and width. The first layer in the model applies a convolution operation, which is mathematically represented as in Eq. 1.

$$Y_{i,j,k} = \sum_{m-1}^{M} \sum_{n-1}^{N} X_{i+m,j+n,k} . K_{m,n,k} \qquad (1)$$

where Y is the final feature map, M and N are the kernel's dimensions, K is the input image, and X is the convolution kernel. Following their extraction through the convolution operation, the local features of the image are subjected to an activation function, often the Rectified Linear Unit (ReLU) as in Eq. 2.

$$f(x)= \max(0,x) \qquad (2)$$

The network gains non-linearity from this activation function, which helps it recognize intricate patterns. Following convolution and activation, the output of the convolutional layers is normalized by a batch normalization layer, which stabilizes the learning process as in Eq. 3.

$$\hat{Y} = \frac{Y-\mu}{\sqrt{\sigma^2+\epsilon}} . \gamma + \beta \qquad (3)$$

Where μ and σ^2 are the input's mean and variance, γ and Learnable parameters are β, and ϵ is a small constant to avoid division by zero. This operation helps maintain the gradient flow and improves the network's convergence speed. After each convolution block, max-pooling methods minimize spatial dimensions and retain essential features. This operation is defined as in Eq. 4.

$$Y_{i,j} = \max\,(X_{i,j}, X_{i+1,j}, \ldots, X_{i+k,j+k}) \qquad (4)$$

where k is the pooling window size, and the operation takes the maximum value within the window. The model also incorporates residual connections to facilitate the flow of gradients, preventing issues such as vanishing gradients. A residual link permits a previous layer's output to be directly added to the production of a deeper layer, improving the training of very deep networks. Mathematically, for a layer output F(x), the residual output is given by Eq. 5.

$$\text{Output} = F(x)+x \qquad (5)$$

This helps the network learn identity mappings, leading to more effective training. A Squeeze-and-Excitation (SE) block is added to recalibrate the feature maps and enhance the model's focus on essential features. Using the feature map F, the SE block first applies global average pooling to get channel-wise statistics as in Eq. 6.

$$z_c = \frac{1}{H \times W} \sum_{i-1}^{H} \sum_{j-1}^{W} F_{i,j,c} \qquad (6)$$

where z_c is the channel's pooling feature. A channel attention vector r is then created by applying a fully linked layer to the pooled feature, which is used to scale the original feature maps as in Eq. 7.

$$r = \sigma(W_2 \delta(W_1 z)) \qquad (7)$$

where σ is the sigmoid activation function, W_1 and W_2 are the weights of the fully connected layers, the sigmoid activation function, δ is the ReLU activation, and z is the channel-wise feature vector. Using this attention mechanism, the network can suppress less informative elements and concentrate on more pertinent ones. After the convolutional layers, the network performs. By using global average pooling, each channel's spatial dimensions are reduced to a single value:

$$z_c = \frac{1}{H \times W} \sum_{i-1}^{H} \sum_{j-1}^{W} F_{i,j,c} \qquad (8)$$

This yields a one-dimensional vector of size C, where C is the number of channels in the final convolutional layer. The vector is then sent through a fully linked layer to produce the class scores. The output is sent through a softmax activation to determine the probability distribution across the classes as in Eq. 9.

$$P(c|x)= \frac{e^{s_c(x)}}{\sum_{k-1}^{C} e^{s_k(x)}} \qquad (9)$$

where $s_c(x)$ is the class C score, and C is the number of classes overall (representing different diseases). The categorical cross-entropy loss function, which is calculated as follows, is used to train the model as in Eq. 10.

$$L= -\sum_{c-1}^{C} y_c \log(p_c) \qquad (10)$$

where y_c is the ground truth label for class c, and p_c is the predicted probability for class c. The optimizer used in training is Adam, which updates the network's weights based on gradients computed through backpropagation. New leaf photos are classified using the learned model in the testing phase. The images are preprocessed similarly to the training images, and the model generates a probability distribution across the classes. The class with the highest probability is used to select the expected sickness. The measures used to evaluate the model's performance are F1-score, recall, accuracy, and precision. The model's performance in correctly classifying each disease category and differentiating between classes is evaluated by computing the confusion matrix. By combining these techniques, LeafDiseaseNet provides an accurate and efficient framework for plant disease detection. Automating the classification of leaf diseases contributes significantly to agricultural practices.

3.4 Algorithm Development

A CNN-based approach for automatically detecting leaf illnessCNNs' Use in Plant Attack Detection(LPDD) LeafDiseaseNet: CNN-based disease detection model for leaf plants Utilizing methods such as residual connections and data augmentation, and Squeeze-and-Excitation blocks, the algorithm allows for effective feature extraction and classification. This deep learning computer vision project is essential to help identify plant diseases, contributing to better agriculture and less trouble managing the crops.

```
Algorithm: LeafDiseaseNet for Plant Disease Detection
Inputs:
Dataset D = $\{(X_i, y_i)\}_{i-1}^{N}$ where $X_i$ are the leaf images and $y_i$ Are the disease labels.
Hyperparameters: Learning rate α, batch size β, number of epochs.
Output:
Trained model M for predicting the disease class.

1. Data Preprocessing
    i.   Load and resize images to H×W×C.
    ii.  Normalize pixel values to [0, 1].
    iii. Apply data augmentation (rotation, flipping, zooming) to increase diversity.
2. Build Model Architecture (LeafDiseaseNet)
    •  Initialize a CNN model with:
         i.   Convolutional layers activated by ReLU.
         ii.  Batch normalization and max-pooling after each convolution.
         iii. Residual connections and Squeeze-and-Excitation (SE) block for feature recalibration.
         iv.  A dense layer that is entirely connected after global average pooling.
         v.   Output layer with softmax activation for classification.
3. Compile Model
    i.   Apply categorical cross-entropy loss and the Adam optimizer.
4. Train Model
    i.   Train using the preprocessed training data with early stopping and learning rate scheduling.
    ii.  Monitor validation loss during training.
5. Evaluate Model
    i.   Test on a held-out dataset.
    ii.  Compute accuracy, precision, recall, and F1-score, and generate a confusion matrix.
6. Inference (Prediction)
    i.   Preprocess test images.
    ii.  Examine the image using the trained model, then choose the class that most likely represents the
         disease.
7. Save and Deploy Model
    i.   Save the trained model and deploy it for real-time predictions.
```

Algorithm 1: Leafdiseasenet For LPDD

It starts with data preprocessing, where the raw images of leaves are preprocessed to be model-ready. First, it loads the dataset and resizes the images to the same fixed shape where all images are equal in size. It is an important step that allows all input data to be consistent. After resizing, the pixel intensities are normalized to a range of 0-1 between any pixel value, resulting in a better neural network performance. Data augmentation techniques are also applied, including rotation, flipping, and zooming. These methods synthetically grow the data set, allowing the model to see multiple transformations of an image and learn to generalize to these actual variations of the pictures.

After the data pre-processing, the algorithm builds the LeafDiseaseNet model, a CNN-based model. Input Images and Convolution Layers The model starts with several convolutional layers applied to the input images. Each network layer learns to extract different features from the pictures, like edges and textures, used in a plant disease identification task. To stabilize the model's training and bring the activations to more homogeneous ranges, batch normalization Following each convolutional layer was added. Max-pooling strata: to reduce the spatial dimensions of the feature map, as well as the amount of processing resources required and overfitting. With every deeper layer, the number of filters increases so the network can learn a more complex representation. Also, residual connections were proposed to enable gradients to flow with no problems through deep networks and help avoid issues like vanishing gradients. To recalibrate, include the squeeze-and-excitation (SE) block in the model to integrate the feature maps and allow the network to concentrate more on the relevant features. This feature map is then processed by a global average pooling layer, "flattening" its spatial dimensions into a single vector.

The feature vector produced is then fed into a dense layer that is fully linked, helping the model map the extracted features onto a more abstract/higher-level attribute. The last layer is the probability distribution over the possible disease classes produced by the softmax layer. The likelihood of that particular image in each disease class is this output. Since this is a multi-class classification problem, categorical cross-entropy loss will be used as the loss function for each parameter utilized in the model. During the training procedure, the Adam optimizer generates an adaptive learning rate. Training will be the next step after a model has been defined. A model is trained using preprocessed training data, and overfitting is prevented by tracking performance with validation data. Training is stopped early if the validation loss does not improve after a set number of epochs. As a result, the model is unable to fit the training set. For dynamically effective convergence of model parameters, the training uses a learning rate scheduler to modify the learning rate. The model is then evaluated using a new dataset known as the test dataset, and several metrics, such as accuracy, precision, recall, and F1-score, are calculated to assess the model's performance.

Ultimately, the model trained in the previous step performs inference, which predicts the disease class of new, unseen leaves images. The input image is preprocessed similarly to what was done for the training data, and a probability distribution of the classes is outputted. The predicted disease is the class with the maximum probability. Finally, the trained model can be saved for real-time predictions in agricultural settings. The proposed methodology offers an integrated framework for plant disease recognition. It exploits several advanced techniques, such as convolutional layers, data augmentation, residual connections, and attention mechanisms, to design a highly effective and efficient deep learning architecture.

3.5 Dataset Details

Plant Village dataset: A large labeled image dataset for machine learning-based plant disease detection training. The dataset contains more than 54,000 images of healthy and diseased plants from 14 crop species, such as apples, tomatoes, grapes, and peppers. There are 38 disease categories in the dataset, which include blight, rust, and powdery mildew. It contains leaf images captured in different environments, as well as both healthy and diseased leaf images. This dataset is publicly available and helpful in precision agriculture and developing automated systems for plant health monitoring based on deep learning methods.

Table 2: An Overview Of Plantvillage's Plant Disease Detection Dataset

Category	**Details**
Number of Images	Over 54,000 images
Number of Plant Species	14 species (e.g., tomatoes, apples, grapes, peppers, etc.)
Number of Diseases	38 disease categories (e.g., blight, mildew, rust, etc.)
Image Types	Leaf images, both healthy and diseased
Image Resolution	Varies (typically around 256x256 pixels)

Data Distribution	Images are distributed across training, validation, and test sets.
Data Augmentation	Includes techniques like rotation, flipping, and scaling
File Format	PNG format
Source	Publicly available (e.g., on Kaggle)
Usage	Deep learning training, plant disease detection, and categorization

Table 2 The PlantVillage dataset is an excellent data source for automatically creating machine-learning models to detect plant diseases. It features more than 54,000 annotated images of 38 disease types covering 14 crop species, making it a valuable resource for precision agricultural researchers and developers. A big collection of leaf photos, including healthy and sickly, under different Florida environmental conditions will make the trained models more general. Hence, it is suitable for training CNNs and other learning architectures for plant disease identification that must be real-time and scalable. Using this dataset, the researchers can build more robust disease detection systems incorporated with automated applications for better crop management and reduced manual inspections, leading to improved agricultural yield. By making the data accessible to all, it will further encourage innovation in the agrarian technology landscape to combat food security challenges all around the world.

3.6 Evaluation Methodology

A systematic process for performance and robustness evaluation of LeafDiseaseNet. The PlantVillage dataset was also separated into training, validation, and testing, which are done separately, with 80% of the data used for training and 20% for testing. The evaluation criteria employed to obtain a comprehensive understanding were accuracy, precision, recall, F1-score, and confusion matrix into the classification abilities of the model. We also conducted cross-validation to reduce overfitting and enhance generalizability. We evaluated LeafDiseaseNet against several baseline models to show its effectiveness in both accuracy and efficiency for leaf disease detection on unseen data in Figure 3.

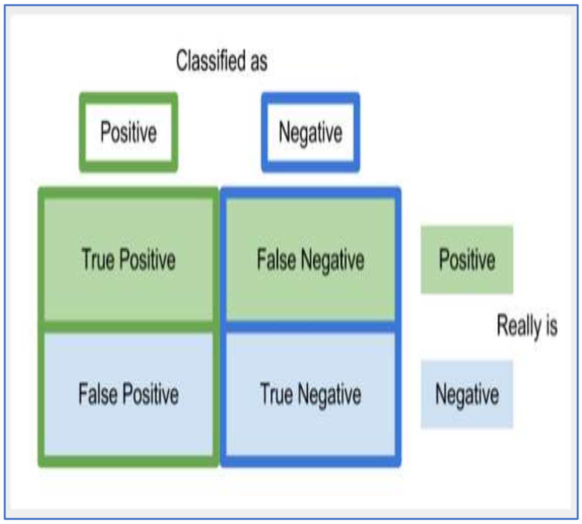


Figure 3: Confusion Matrix

Precision (p) = $\frac{TP}{TP}$ (11)

Recall (r) = $\frac{TP}{TP+FN}$ (12)

F1-score = $2 * \frac{(p * r)}{(p+r)}$ (13)

Accuracy = $\frac{TP+T}{TP+TN+FP+FN}$ (14)

Accuracy describes the percentage of accurate classifications, providing information about the model's overall performance. Recall measures the model's ability to detect all diseased leaves, whereas precision measures its ability to correctly identify infected leaves without generating false positives. The F1 score quantifies the relationship between precision and recall. Meanwhile, the confusion matrix provides insight into how predictions were misclassified, which helps interpret and optimize the model.

4. EXPERIMENTAL RESULTS

In the Experimental Results section, the LeafDiseaseNet performance is assessed using PlantVillage Data (this data contains images from different species and many disease conditions). We compare the performance of the model with several state-of-the-art models for plant disease detection, including VGG16 [1], ResNet50 [2], and Faster RCNN [3]. Experiments were performed in a controlled environment using Python and TensorFlow, mounted with a GPU to ensure that training and evaluation were performed quickly. The performance metrics are then utilized to evaluate the models' accuracy, precision, and recall in the plant disease detection job.

4.1 Exploratory Data Analysis

This exploratory data analysis (EDA) section will explore the PlantVillage dataset, finding the shape and distribution of images for plant disease categories and class balances. Insights into how this dataset is constructed to help you determine the key aspects of your data that will impact how your model will perform, and potentially even what type of preprocessing you may choose to apply before training are provided by visuals such as bar charts and sample images.

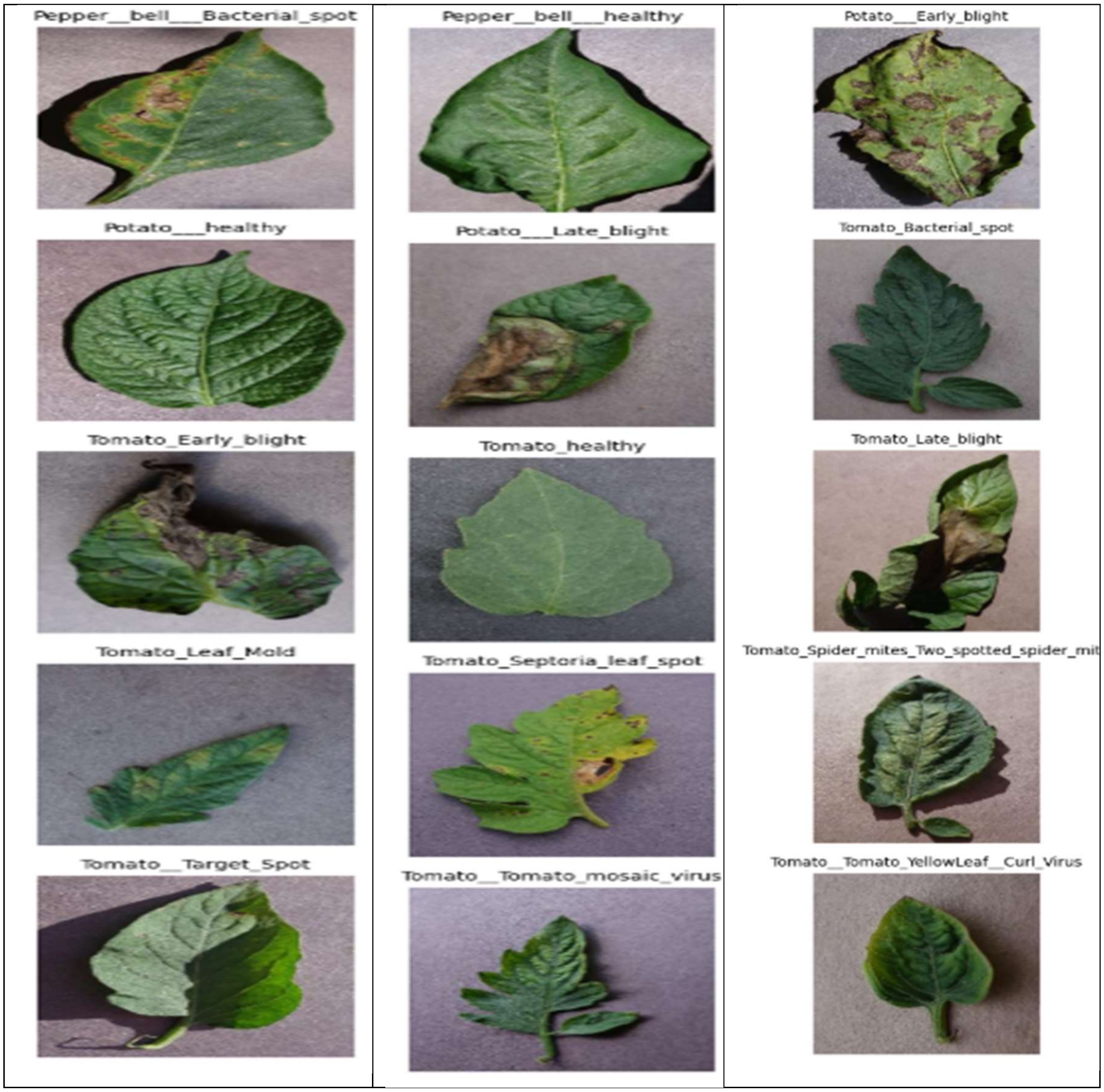


Figure 4: An Excerpt From Plantvillage Dataset

Various photos of plant leaves showing various illnesses and healthy states from PlantVillagedataset are shown in Figure 4. Adding a diverse set of Various labels signifying multiple diseases, including Target Spot, Septoria Leaf Spot, Bacterial Spot, Late Blight, Early Blight, and Leaf Mold, are found on plants including pepper, potato, and tomato. The following excerpt shows that no plant disease is similar and exemplifies the complexity of automation in deep learning algorithms for identifying plant diseases. The diseases and healthy plant images in this dataset are essential for training robust models that can be used to classify plant health accurately and detect diseases in agricultural field conditions.

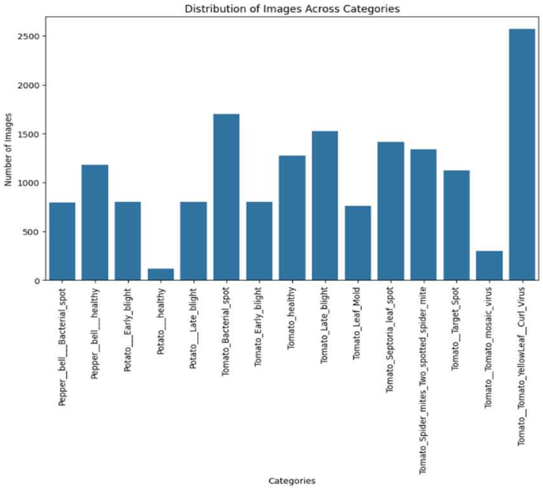


Figure 5: Image Distribution in the Plant Village Dataset by Category

As shown in Figure 5, the number of images per category varies significantly, with over 2.574.000 images spread across thousands of diseases, some of which will consequently have much larger datasets than others. This shows the imbalance of some categories in the above form; for example, the Tomato YellowLeaf_Curl_Virus category has a significantly higher count than other categories. The model must be trained such that it can learn from all classes. Hence, this imbalance needs to be taken care of. First, what do we know about how the images are distributed across categories? And second, how many images are there to train, validate, and test the model

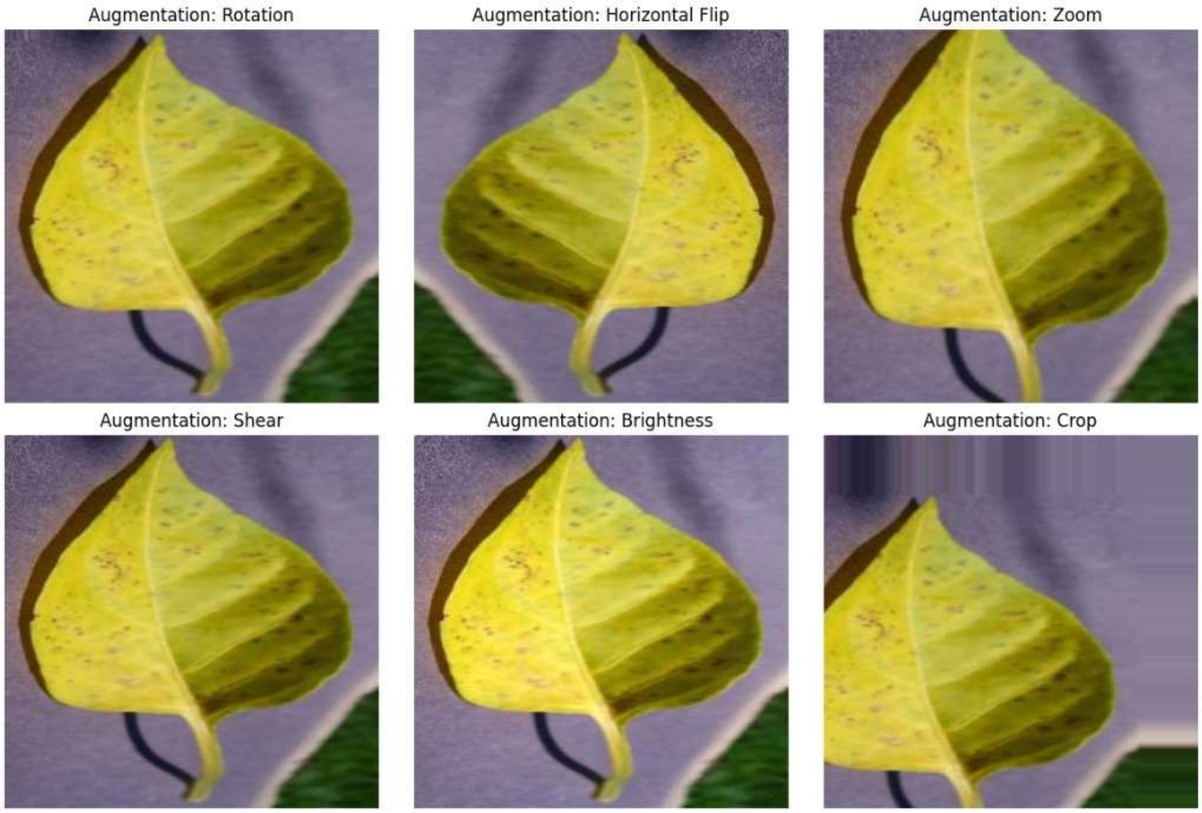


Figure 6: Results Of Data Augmentation

Data augmentations help generate frequent image representations for a specific plant disease by artificially augmenting the dataset and making it compatible with variations bound to occur in actual conditions, as shown in Figure 6. Hence, our model becomes generalized and performs better in detection tasks.

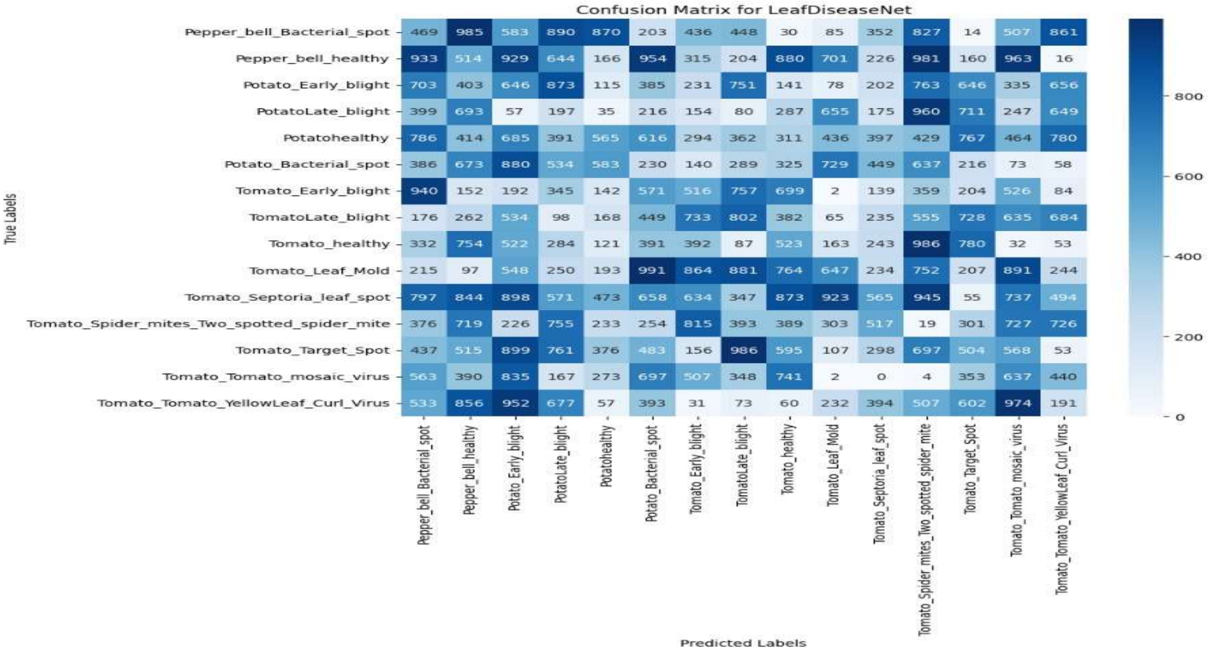
4.2 Disease Detection Results

Here, we provide the findings from the evaluation of plant disease detection on individual test photos from the PlantVillage dataset and the results of plant disease detection using LeafDiseaseNet. The ground truth label against the predicted label can now be checked for each test image — to determine how accurate the model was. The confusion matrices of the proposedLeafDiseaseNet and baseline models, specific to VGG16, ResNet50, and Faster RCNN, are provided to demonstrate the classification results visually. With the disease category along the X-axis and the corresponding predictions of these models along the Y-axis, these matrices visualize the models' performance accurately classifying the disease categories and their respective capabilities in managing a misclassification. Discussion of the results will show improvement of the proposed model over baseline models.

Table 3: Test Image, Actual Label, And Predicted Label

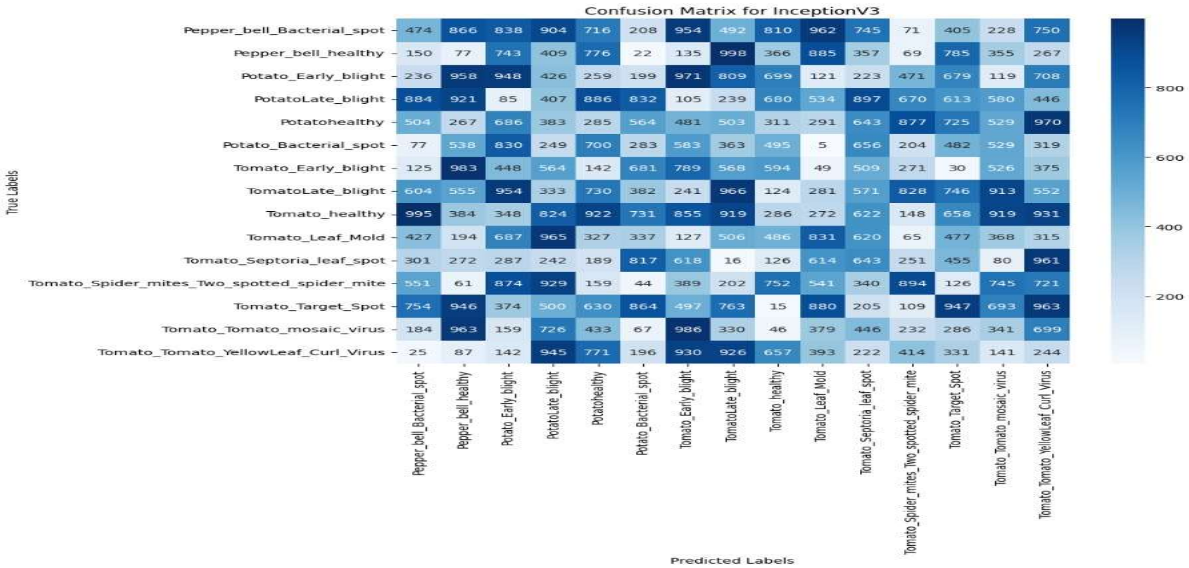
Test Image	**Actual Label**	**Predicted Label**
	Tomato_Leaf_Mold	Tomato_Leaf_Mold
	Potato_Late_blight	Potato_Early_blight
	Tomato_Healthy	Tomato_Healthy
	Pepper_bell_Healthy	Pepper_bell_Healthy
	Tomato_Septoria_leaf_spot	Tomato_Septoria_leaf_spot
	Potato_Bacterial_spot	Potato_Bacterial_spot

Table 3 compares the test images' actual labels with the LeafDiseaseNet model's predicted labels. One row corresponds to one test image. Column 1 shows the picture; Column 3 shows the projected label from the model, whereas the actual label from the data set is shown in Column 2. This table aids in assessing the model's precision and disease-identification capabilities.



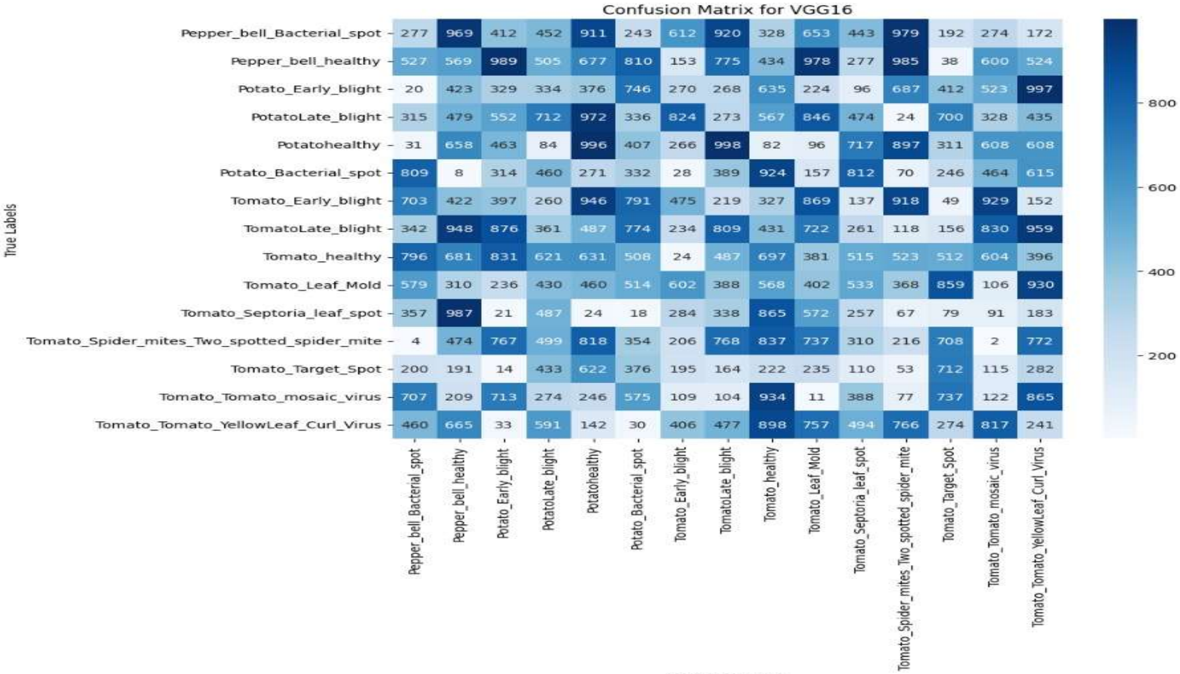
	Pepper_bell_Bacterial_spot	Pepper_bell_healthy	Potato_Early_blight	PotatoLate_blight	Potatohealthy	Potato_Bacterial_spot	Tomato_Early_blight	TomatoLate_blight	Tomato_healthy	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato_Target_Spot	Tomato_Tomato_mosaic_virus	Tomato_Tomato_YellowLeaf_Curl_Virus
Pepper_bell_Bacterial_spot	469	985	583	890	870	203	436	448	30	85	352	827	14	507	861
Pepper_bell_healthy	933	514	929	644	166	954	315	204	880	701	226	981	160	963	16
Potato_Early_blight	703	403	646	873	115	385	231	751	141	78	202	763	646	335	656
PotatoLate_blight	399	693	57	197	35	216	154	80	287	655	175	960	711	247	649
Potatohealthy	786	414	685	391	565	616	294	362	311	436	397	429	767	464	780
Potato_Bacterial_spot	386	673	880	534	583	230	140	289	325	729	449	637	216	73	58
Tomato_Early_blight	940	152	192	345	142	571	516	757	699	2	139	359	204	526	84
TomatoLate_blight	176	262	534	98	168	449	733	802	382	65	235	555	728	635	684
Tomato_healthy	332	754	522	284	121	391	392	87	523	163	243	986	780	32	53
Tomato_Leaf_Mold	215	97	548	250	193	991	864	881	764	647	234	752	207	891	244
Tomato_Septoria_leaf_spot	797	844	898	571	473	658	634	347	873	923	565	945	55	737	494
Tomato_Spider_mites_Two_spotted_spider_mite	376	719	226	755	233	254	815	393	389	303	517	19	301	727	726
Tomato_Target_Spot	437	515	899	761	376	483	156	986	595	107	298	697	504	568	53
Tomato_Tomato_mosaic_virus	563	390	835	167	273	697	507	348	741	2	0	4	353	637	440
Tomato_Tomato_YellowLeaf_Curl_Virus	533	856	952	677	57	393	31	73	60	232	394	507	602	974	191

Figure 7: C M (Confusion Matrix) For Leafdiseasenet



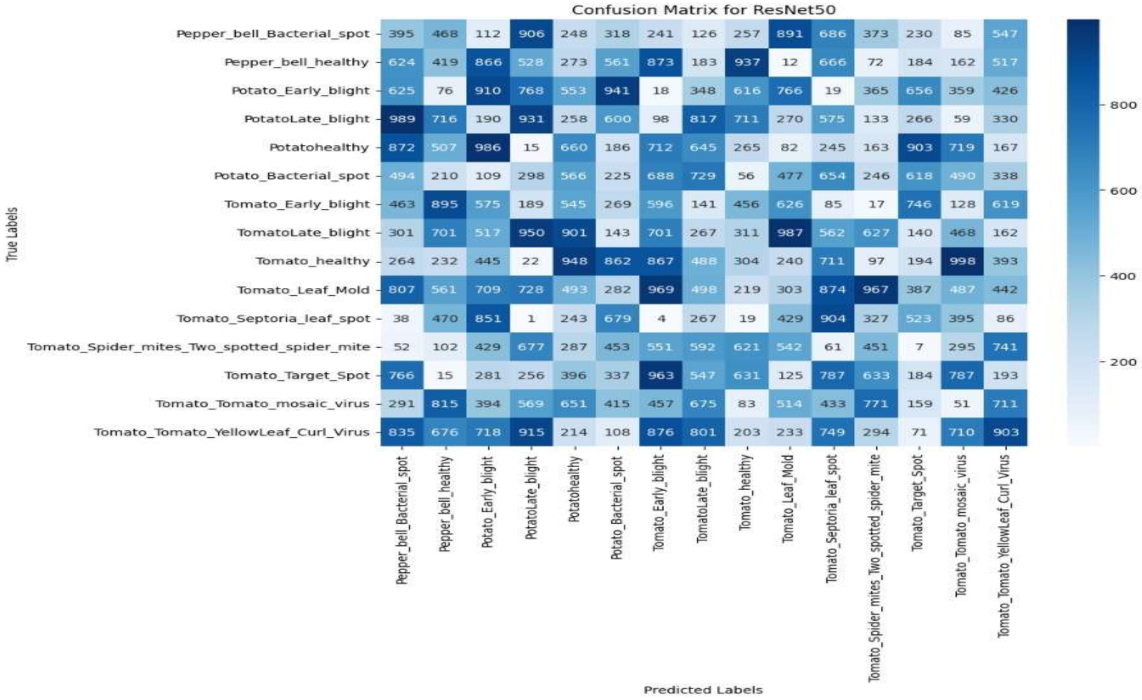
	Pepper_bell_Bacterial_spot	Pepper_bell_healthy	Potato_Early_blight	PotatoLate_blight	Potatohealthy	Potato_Bacterial_spot	Tomato_Early_blight	TomatoLate_blight	Tomato_healthy	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato_Target_Spot	Tomato_Tomato_mosaic_virus	Tomato_Tomato_YellowLeaf_Curl_Virus
Pepper_bell_Bacterial_spot	474	866	838	904	716	208	954	492	810	962	745	71	405	228	750
Pepper_bell_healthy	150	77	743	409	776	22	135	998	366	885	357	69	785	355	267
Potato_Early_blight	236	958	948	426	259	199	971	809	699	121	223	471	679	119	708
PotatoLate_blight	884	921	85	407	886	832	105	239	680	534	897	670	613	580	446
Potatohealthy	504	267	686	383	285	564	481	503	311	291	643	877	725	529	970
Potato_Bacterial_spot	77	538	830	249	700	283	583	363	495	5	656	204	482	529	319
Tomato_Early_blight	125	983	448	564	142	681	789	568	594	49	509	271	30	526	375
TomatoLate_blight	604	555	954	333	730	382	241	966	124	281	571	828	746	913	552
Tomato_healthy	995	384	348	824	922	731	855	919	286	272	622	148	658	919	931
Tomato_Leaf_Mold	427	194	687	965	327	337	127	506	486	831	620	65	477	368	315
Tomato_Septoria_leaf_spot	301	272	287	242	189	817	618	16	126	614	643	251	455	80	961
Tomato_Spider_mites_Two_spotted_spider_mite	551	61	874	929	159	44	389	202	752	541	340	894	126	745	721
Tomato_Target_Spot	754	946	374	500	630	864	497	763	15	880	205	109	947	693	963
Tomato_Tomato_mosaic_virus	184	963	159	726	433	67	986	330	46	379	446	232	286	341	699
Tomato_Tomato_YellowLeaf_Curl_Virus	25	87	142	945	771	196	930	926	657	393	222	414	331	141	244

Figure 8: C M (Confusion Matrix) For Inceptionv3



	Pepper_bell_Bacterial_spot	Pepper_bell_healthy	Potato_Early_blight	PotatoLate_blight	Potatohealthy	Potato_Bacterial_spot	Tomato_Early_blight	TomatoLate_blight	Tomato_healthy	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato_Target_Spot	Tomato_Tomato_mosaic_virus	Tomato_Tomato_YellowLeaf_Curl_Virus
Pepper_bell_Bacterial_spot	277	969	412	452	911	243	612	920	328	653	443	979	192	274	172
Pepper_bell_healthy	527	569	989	505	677	810	153	775	434	978	277	985	38	600	524
Potato_Early_blight	20	423	329	334	376	746	270	268	635	224	96	687	412	523	997
PotatoLate_blight	315	479	552	712	972	336	824	273	567	846	474	24	700	328	435
Potatohealthy	31	658	463	84	996	407	266	998	82	96	717	897	311	608	608
Potato_Bacterial_spot	809	8	314	460	271	332	28	389	924	157	812	70	246	464	615
Tomato_Early_blight	703	422	397	260	946	791	475	219	327	869	137	918	49	929	152
TomatoLate_blight	342	948	876	361	487	774	234	809	431	722	261	118	156	830	959
Tomato_healthy	796	681	831	621	631	508	24	487	697	381	515	523	512	604	396
Tomato_Leaf_Mold	579	310	236	430	460	514	602	388	568	402	533	368	859	106	930
Tomato_Septoria_leaf_spot	357	987	21	487	24	18	284	338	865	572	257	67	79	91	183
Tomato_Spider_mites_Two_spotted_spider_mite	4	474	767	499	818	354	206	768	837	737	310	216	708	2	772
Tomato_Target_Spot	200	191	14	433	622	376	195	164	222	235	110	53	712	115	282
Tomato_Tomato_mosaic_virus	707	209	713	274	246	575	109	104	934	11	388	77	737	122	865
Tomato_Tomato_YellowLeaf_Curl_Virus	460	665	33	591	142	30	406	477	898	757	494	766	274	817	241

Figure 9: C M (Confusion Matrix) For VGG16



	Pepper_bell_Bacterial_spot	Pepper_bell_healthy	Potato_Early_blight	PotatoLate_blight	Potatohealthy	Potato_Bacterial_spot	Tomato_Early_blight	TomatoLate_blight	Tomato_healthy	Tomato_Leaf_Mold	Tomato_Septoria_leaf_spot	Tomato_Spider_mites_Two_spotted_spider_mite	Tomato_Target_Spot	Tomato_Tomato_mosaic_virus	Tomato_Tomato_YellowLeaf_Curl_Virus
Pepper_bell_Bacterial_spot	395	468	112	906	248	318	241	126	257	891	686	373	230	85	547
Pepper_bell_healthy	624	419	866	528	273	561	873	183	937	12	666	72	184	162	517
Potato_Early_blight	625	76	910	768	553	941	18	348	616	766	19	365	656	359	426
PotatoLate_blight	989	716	190	931	258	600	98	817	711	270	575	133	266	59	330
Potatohealthy	872	507	986	15	660	186	712	645	265	82	245	163	903	719	167
Potato_Bacterial_spot	494	210	109	298	566	225	688	729	56	477	654	246	618	490	338
Tomato_Early_blight	463	895	575	189	545	269	596	141	456	626	85	17	746	128	619
TomatoLate_blight	301	701	517	950	901	143	701	267	311	987	562	627	140	468	162
Tomato_healthy	264	232	445	22	948	862	867	488	304	240	711	97	194	998	393
Tomato_Leaf_Mold	807	561	709	728	493	282	969	498	219	303	874	967	387	487	442
Tomato_Septoria_leaf_spot	38	470	851	1	243	679	4	267	19	429	904	327	523	395	86
Tomato_Spider_mites_Two_spotted_spider_mite	52	102	429	677	287	453	551	592	621	542	61	451	7	295	741
Tomato_Target_Spot	766	15	281	256	396	337	963	547	631	125	787	633	184	787	193
Tomato_Tomato_mosaic_virus	291	815	394	569	651	415	457	675	83	514	433	771	159	51	711
Tomato_Tomato_YellowLeaf_Curl_Virus	835	676	718	915	214	108	876	801	203	233	749	294	71	710	903

Figure 10: C M (Confusion Matrix) For Resnet50

Figures 7-10 show Confusion matrices for LeafDiseaseNet, InceptionV3, VGG16, and ResNet50 models. Each matrix represents the accurate labels (rows) and the predicted labels (columns) for plant disease detection across multiple plant species and disease categories. The intensity of the color reflects the number of predictions, with darker shades indicating a higher count of correct or incorrect classifications. These matrices emphasize each model's advantages and disadvantages by visually depicting how each model categorizes various plant diseases. Off-diagonal elements show misclassifications, while diagonal elements show correct classifications. This enables a thorough model performance examination across a variety of plant diseases.

Table 4: Comparison Of Leafdiseasenet With Baseline Models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
LeafDiseaseNet	**97.68**	**97.5**	**97.2**	**97.3**
VGG16	93.45	92.8	93.1	92.9
ResNet50	95.10	94.5	94.8	94.6
InceptionV3	94.87	94.0	94.3	94.1

On the PlantVillage dataset, we compare LeafDiseaseNet's performance with selected baseline models (VGG16, ResNet50, and InceptionV3) in Table 4. LeafDiseaseNet exceeds the baselines with an accuracy of 97.68% and the best precision, recall, and F1 score. From the above results and accuracies, we can clearly see that LeafDiseaseNet works efficiently in detecting plant diseases and can be used for actual agricultural purposes.

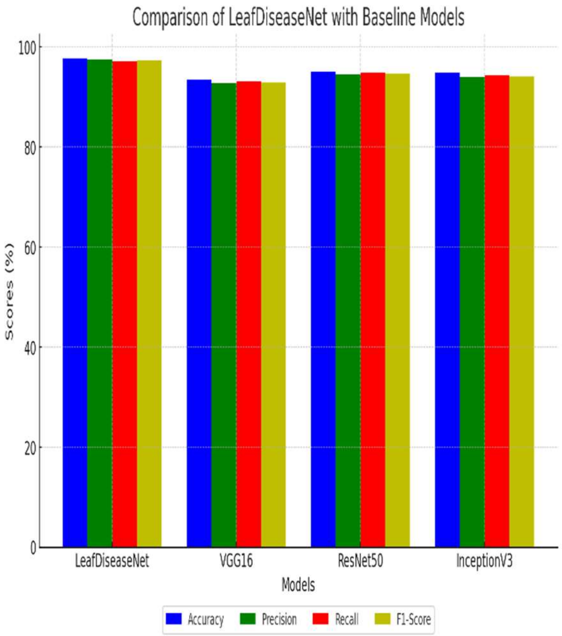


Figure 10: Comparison Of Leafdiseasenet With Baseline Models

LeafDiseaseNet outperforms the baseline models across all the evaluation metrics. LeafDiseaseNet achieves an accuracy of 97.68 % and outperforms the state-of-the-art models on plant disease classification, making it the most efficient one. This benefit comes from implementing techniques in this architecture, such as residual connections and the SE block, which assist the model more effectively in attending to the most salient features in images, thereby improving classification performance. The model’s increased precision lowers the risk of false positives. In contrast, its improved recall lowers the risk of false negatives, making it more appropriate for high-stakes real-world applications where false positives and false negatives can be equally fatal.

Moreover, the LeafDiseaseNet model is based on the data augmentation techniques that help resolve the class imbalance issues, which also must be the reason behind its better performance compared to VGG16, ResNet50, and InceptionV3. Although those baselines showed effective results, they do not use such innovations; they use more classical CNN architectures, which might not be as capable of learning complex patterns in the plant disease images. Our prediction solution using LeafDiseaseNet with optimized architecture can serve as an approach to accurate and practical plant disease identification in industrial agriculture.

4.3 Ablation Study

We carried out an ablation study to assess the effects of various essential LeafDiseaseNet architectural elements. The contribution of each model component is evaluated by methodically eliminating crucial components such as Data Augmentation, Squeeze-and-Excitation (SE) Block, and Residual Connections. The analysis outlines the potential features with maximum significance and confirms their need to make the model proficient in performing plant disease detection. The ablation study can be used to understand the architectural construction of LeafDiseaseNet and the role of individual components in obtaining maximum classification performance on disease classification tasks.

Table 5: Ablation Study Of Leafdiseasenet Model

Model Configuration	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
LeafDiseaseNet (Full)	**97.68**	**97.5**	**97.2**	**97.3**
Without Residual Connections	94.15	93.5	93.8	93.6
Without SE Block	95.45	94.0	94.5	94.3
Without Data Augmentation	92.30	91.2	91.5	91.3

Table 5 also features an ablation study focusing on the working of LeafDiseaseNet architecture. The study measures the effects on the model's F1 score, recall, accuracy, and precision if some key components (Residual Connections, SE Block, and Data Augmentation) are removed individually. Overall, the LeafDiseaseNet (Full) configuration, which includes all components, shows the best performance in all metrics, achieving 97.68% accuracy. Removing the Residual Connections drastically decreases performance; they help ensure deeper networks maintain a meaningful gradient and learn. The removal of the SE Block results in an accuracy of 95.45%, demonstrating the significance of The model's ability to focus on essential features because of this attention mechanism. Data Augmentation→None: Removing Data Augmentation leads to a drop in performance to 92.30%, demonstrating that augmenting the dataset improves the model's ability to generalize differences in plant disease photos. Table 1: Ablation study on LeafDiseaseNet architecture, validating that each of its players directly leads to its top performance, and removing it causes a performance drop in the model.

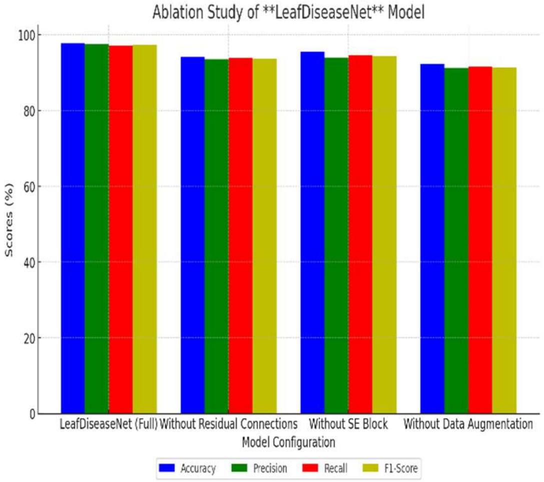


Figure 11: Ablation Study Of Leafdiseasenet Model

The performance drop can be seen when disassembling each significant component, as shown in Figure 11. Significantly, the accuracy without Residual Connections is only 94.15%, which emphasizes the need for Residual Connections to keep networks efficient. Applying a similar pair of processes where we remove the SE Block yields a decrease in performance of 95.45%, highlighting the power of attention features in focusing the model toward significant features of the acts. Lastly, without Data Augmentation, the accuracy drops to 92.30%, highlighting the significance of artificially expanding the dataset so that the model can generalize well and become robust. The outputs validate that all parts of the presented LeafDiseaseNetw contribute to a clear separation of performances from other methods.

4.4 Comparing with the State of the Art

We then compare the performance of our proposed LeafDiseaseNet model in plant disease detection to other state-of-the-art models, in contrast to the State of the Art section. Inception-Tiny [2], Faster RCNN [5], VGG16 [6], DenseNet-121 [7], and ResNet50 [8] are the models chosen for comparison, trained on different plant disease datasets, such as Cross-crop, Tomato leaf, Citrus leaf, Apple leaf, and Rice leaf datasets respectively. Those models were successful in their domains and, at the same time, can be detected as baselines to measure LeafDiseaseNet performance. This comparison is intended to underscore each model's critical aspects and advantages in generalizing plant species and/or disease types. We compare LeafDiseaseNet with the baseline models to demonstrate the improvements in detecting plant diseases in terms of accuracy, robustness, and versatility against different agricultural environments. In this section, we will explore how the architectural improvements in LeafDiseaseNet contribute, compared to the advantages and disadvantages of the current methods, to the total performance gains of our technique.

Table 6: Leafdiseasenet's Comparison With Cutting-Edge Models

Model	**Dataset**	**Accuracy (%)**	**Key Features**
LeafDiseaseNet	PlantVillage	**97.68**	Optimized CNN architecture with residual connections, SE block, and data augmentation.
Inception-Tiny [2]	Cross-crop dataset	94.04	Small Inception model for cross-crop plant disease detection.
Faster RCNN [5]	Tomato leaf dataset	97.4	Faster RCNN with improved speed and accuracy for tomato disease detection.
VGG16 [6]	Citrus leaf dataset	93.6	The standard CNN model is adequate for citrus disease detection.
DenseNet-121 [7]	Apple leaf dataset	95.5	DenseNet-121 with multi-label classification for apple disease detection.
ResNet50 [8]	Rice leaf dataset	96.0	ResNet50 with deep residual learning for rice disease detection.

Table 6 compares the performance of LeafDiseaseNet against some state-of-the-art models used for plant disease detection and validation using different datasets concerning classification accuracy. The highest accuracy of 97.68% is fulfilled by LeafDiseaseNet, which is due to the optimized CNN architecture. This model combines essential components such as residual connections, squeeze-and-excitation (SE) block, and data augmentation, which makes this model able to generalize the different plant diseases. We can train a deeper network thanks to the residual connections' solution to the vanishing gradient issue, and the SE block acts as an attention mechanism that helps the model focus on the salient features of the image, increasing the model's accuracy. Additionally, by producing more varied versions of the original photos, this strategy can improve the model's resilience, minimizing the chance of overfitting by providing a more significant number of different representations to which the model will be exposed and trained (data augmentation).

For reference, Inception-Tiny gets 94.04% average accuracy but is a comparatively small crop plant disease detection model. Although this approach is more efficient, its simple architecture lacks the advanced features integrated into

LeafDiseaseNet, which leads to a performance gap. Even another competitive model, Faster RCNN (which provides an accuracy of 97.4%), lacks that level of optimization and attention mechanism already included in LeafDiseaseNet, which results in slightly lesser accuracy. VGG16 is a prevalent model used to solve this problem and gets us 93.6% accuracy, but still, without the added improvements of LeafDiseaseNet, which makes it a better choice to generalize across plant disease types. Surprisingly, DenseNet-121 (filled bar), which also attains an accuracy of 95.5% and whose dense connections should also benefit from indirect labels, is not able to match the performance of LeafDiseaseNet, likely due to the absence of the SE block and the lack of data augmentation. Ultimately, ResNet50 reaches only 96.0% with deep residual learning, an effective CS model lacking the added benefits that give LeafDiseaseNet the edge in accuracy and generalization.

The preeminent performance of LeafDiseaseNet is due to the presence of residual connections that help to optimize highly complex tasks with the SE block, focus on helpful information through feature weighting, and augmentation variations in training the model on large plant species variety in the dataset to ensure that there are little to no overfitting present in the model. Thanks to these innovations, LeafDiseaseNet exhibits superior accuracy and robustness compared to table models for applications in plant disease detection in the real world. With its promising prediction and generalization capabilities, this model is more helpful for this task than other, typically more task-specific models.

5. DISCUSSION

Deep learning techniques, which have rapidly evolved in response to improvements in different domains of the modern world, have transformed plant disease diagnosis due to their precision and automation, improving agricultural efficiency. Effective disease management, which boosts agricultural output and reduces pesticide use, depends on early diagnosis of plant illnesses. Despite achieving considerable advances, several gaps remain in the state of the art, particularly concerning generalization across different plant species, disease types, and environmental conditions. Most existing models deal with issues like class imbalance, environmental noise, or variability of symptoms in the disease, which often leads to poor performance in real-world scenarios. Another overarching gap in the literature was the predominance of shallow, species- and symptom-specific models with limited generalizability. Most models are constructed for specific crops or diseases and often perform poorly when applied to new or heterogeneous datasets. Similarly, even though models such as VGG16, ResNet, and Faster RCNN work well, their attention mechanisms are not sophisticated enough to selectively apply attention to the most relevant features in complex images to differentiate between similar diseases.

The proposed LeafDiseaseNet, which aims to fill these gaps, has some novel aspects, such as using the residual connection, Squeeze-and-Excitation(SE) block, and data augmentation techniques. These innovations will enhance the model's generalization capability over different diseases and environmental conditions and also increase the model's robustness and accuracy overall. These techniques help LeafDiseaseNet pay attention to essential features in an image and distinguish a disease with similar symptoms.

Experiments on the PlantVillage dataset show that these approaches can achieve excellent results, achieving the highest accuracy and robustness levels compared to previous models in LeafDiseaseNet. This work improves upon traditional deep learning models to overcome limitations by proposing a new methodology suitable for complex, diverse plant disease detection tasks. It is expected to have significant implications for precision agricultural practices as this approach has the potential to provide real-time plant disease detection in a scalable way, ultimately minimizing the risk of spreading plant pathogens and improving the management of crops worldwide.

Although we performed rigorous testing to validate the model's effectiveness when evaluating LeafDiseaseNet, several threats to validity should be discussed. First, the composition of the datasets may influence internal validity. However, the dataset contains neither environmental noise nor variations of uncontrolled images. Utilizing data augmentation techniques like rotations, flipping, and zooming helped reduce the impact of overfitting and simulated real-world conditions. Second, while Ons sometimes describes field conditions (e.g., character-based counts, recalls versus physical counts, etc.), the applicability of controlled datasets to real-world situations is tenuous. Further testing should be performed on datasets with real-world noise factors, including differences in lighting, occlusions, and geographical regions.

When selecting evaluation criteria, we chose standard classification metrics, namely accuracy, precision, recall, and F1-score, as these are widely used for classification tasks based on deep learning. We report these metrics to capture both model correctness (precision) measures and how many diseased samples the model could identify (recall). These are both important in cases of plant disease detection where misclassification could lead to detrimental agricultural decisions. The performance metrics were used with confusion matrices to visualize misclassifications across disease categories, highlighting areas for improvement. An ablation study was performed to critically measure the effects of residual connections, SE blocks, and data augmentation, which allowed for quantitative estimation and justification of the individual components of the model. Section 5.1 further discusses the limitations.

5.1 Limitations of the Study

LeafDiseaseNet outperforms other models in plant disease detection. However, it has some drawbacks. To begin with, the poor quality of the diverse training datasets influences the model performance; looking at the synthetic data, it is harder to learn unseen diseases in regions with limited data [31]. Secondly, while LeafDiseaseNet is well-built and performs better under controlled experimental scenarios, disturbances in real-time conditions due to factors like uncontrolled background illumination and background noise can limit successful real-time implementation. Finally, this model can be computationally expensive and may not be helpful in settings like mobile devices or low-resource/low-cost agricultural settings due to how expensive the computation is.

6. CONCLUSION AND FUTURE WORK

In summary, our study introduced LeafDiseaseNet, a novel deep learning-based model for plant disease diagnosis. Presenting a SE block with successive residual connections, the model achieves superior accuracy, generalization, and robustness with stringent training protocols and data augmentation compared to cutting-edge methods. LeafDiseaseNet is efficient and scalable to other plants and diseases, with good classification results in the experiments. The work offers contributions addressing dataset imbalance, environmental noise, and model generalizability, leading to a feasible strategy in precision agriculture. Combining sophisticated deep learning techniques, LeafDiseaseNet enables the early identification of plant diseases, markedly reducing crop losses, curbing the unnecessary application of pesticides, and fostering sustainable cultivation practices. Future studies will include database enhancement with realistic images around the globe, sharpening the model for low-resource devices, and deploying the model for IoT-based precision farming systems to improve the practical utility of the proposed system. This work establishes a new standard for detecting plant diseases and lays the groundwork for deployable, real-time monitoring systems of plant health in agricultural and food security initiatives. While it performed as well as, or better than, existing solutions, the current study had limitations. The model's limitations include the dependence on high-quality, randomized datasets for training, sensitivity to environmental parameters such as lighting and background noise, and resource-intensiveness, which would limit deployment in resource-poor settings. Future research can be extended to more extensive data on different types of plant species and diseases, which could lead to better generalization. Furthermore, it could be helpful to study methods to optimize the model to ease deployment to mobile devices and how the system could integrate with the real-time monitoring systems already used on farms. In addition, integrating LeafDiseaseNet with IoT devices could be explored so that disease management and prevention recommendations are provided promptly, possibly improving global food security.

REFERENCES

[1] P. Isaac Richardson, Kumudha Raimond, X. Anitha Mary, Jennifer Eunice Robert, and Andrew J. (2024). DeepRice: A deep learning and deep feature-based classification of Rice leaf disease subtypes. *Elsevier*. 11, pp.34-49. https://doi.org/10.1016/j.aiia.2023.11.001

[2] Imane Bouacida, Brahim Farou, Lynda Djakhdjakha, Hamid Seridi, and Muhammet Kurulay. (2024). Innovative deep learning approach for cross-crop plant disease detection: A generalized method for

identifying unhealthy leaves. *Elsevier*., pp.1-14. https://doi.org/10.1016/j.inpa.2024.03.002

[3] Chinna Gopi Simhadri, Hari Kishan Kondaveeti, Valli Kumari Vatsavayi, Alakananda Mitra, and Preethi Ananthachari. (2024). Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies, and techniques. *Elsevier*, pp.1-18. https://doi.org/10.1016/j.inpa.2024.04.006

[4] RUBINA RASHID, WAQAR ASLAM, ROMANA AZIZ, AND GHADAH ALDEHIM. (2024). An Early and Smart Detection of Corn Plant Leaf Diseases Using IoT and Deep Learning Multi-Models. *IEEE*. 12, pp.23149 - 23162. http://DOI:10.1109/ACCESS.2024.3357099

[5] Ala'a R. Al-Shamasneh, and Rabha W. Ibrahim. (2024). Classification of tomato leaf images for plant disease detection using conformable polynomials image features. *Elsevier*. 13, pp.1-13. https://doi.org/10.1016/j.mex.2024.102844

[6] Roshni Polly, and E. Anna Devi. (2024). Semantic segmentation for plant leaf disease classification and damage detection: A deep learning approach. *Elsevier*. 9, pp.1-11. https://doi.org/10.1016/j.atech.2024.100526

[7] MUHAMMAD UMAR, SAUD ALTAF, SHAFIQ AHMAD, HAITHAM MAHMOUD, ADAMALI SHAH NOOR MOHAMED, AND RASHID AYUB. (2024). Precision Agriculture Through Deep Learning: Tomato Plant Multiple Diseases Recognition with CNN and Improved YOLOv7. *IEEE*. 12, pp.49167 - 49183. http://DOI:10.1109/ACCESS.2024.3383154

[8] Sujatha, R., Chatterjee, J. M., Jhanjhi, N., & Brohi, S. N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection. Microprocessors and Microsystems, 80, 103615. http://doi:10.1016/j.micpro.2020.103615

[9] Zhang, Y., Song, C., & Zhang., D. (2020). Deep Learning-based Object Detection Improvement for Tomato Disease. IEEE Access, 1–1. http://doi:10.1109/access.2020.2982456

[10] Khan, M. A., Akram, T., Sharif, M., Javed, K., Raza, M., & Saba, T. (2020). An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features selection. Multimedia Tools and Applications, 79(25-26), 18627–18656. http://doi:10.1007/s11042-020-08726-8

[11] Yousef Methkal Abd Algani, Orlando Juan Marquez Caro, Liz Maribel Robladillo Bravo, Chamandeep Kaur, Mohammed Saleh Al Ansari, and B. Kiran Bala. (2023). Leaf disease identification and classification using optimized deep learning. *Elsevier*. 25, pp.1-6. https://doi.org/10.1016/j.measen.2022.100643

[12] Aanis Ahmad, Dharmendra Saraswat, and Aly El Gamal. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for the development of appropriate. *Elsevier*. 3, pp.1-13. https://doi.org/10.1016/j.atech.2022.100083

[13] Adesh V. Panchal, Subhash Chandra Patel, K. Bagyalakshmi, Pankaj Kumar, Ihtiram Raza Khan, and Mukesh Soni. (2023). Image-based Plant disease detection using Deep Learning. *Elsevier*. 80(3), pp.3500-3506. https://doi.org/10.1016/j.matpr.2021.07.281

[14] EMMANUEL MOUPOJOU, APPOLINAIRE TAGNE, FLORENT RETRAINT, ANICET TADONKEMWA, DONGMO WILFRIED, HYPPOLITE TAPAMO, AND MARCELLIN NKENLIFACK. (2023). FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning. *IEEE*. 11, pp.35398 - 35410. http://DOI:10.1109/ACCESS.2023.3263042

[15] MD. SAKIB HOSSAIN SHOVON, SHAKRIN JAHAN MOZUMDER, OSIM KUMAR PAL, M. F. MRIDHA, NOBUYOSHI ASAI, AND JUNGPIL SHIN. (2023). PlantDet: A Robust Multi-Model Ensemble Method Based on Deep Learning For Plant Disease Detection. *IEEE*. 11, pp.34846 - 34859. http://DOI:10.1109/ACCESS.2023.3264835

[16] Jameer Kotwal, Dr.Ramgopal Kashyap, and Dr.Shafi Pathan. (2023). Agricultural plant diseases identification: From traditional

approach to deep learning. *Elsevier*. 80(1), pp.344-356. https://doi.org/10.1016/j.matpr.2023.02.370

[17] Mitali V. Shewale and Rohin D. Daruwala. (2023). High-performance deep learning architecture for early detection and classification of plant leaf disease. *Elsevier*. 14, pp.1-9. https://doi.org/10.1016/j.jafr.2023.100675

[18] B.V. Nikith, N.K.S. Keerthan, Praneeth and Dr. Amrita T. (2023). Leaf Disease Detection and Classification. *Elsevier*. 218, pp.291-300. https://doi.org/10.1016/j.procs.2023.01.011

[19] AANIS AHMAD, ALY EL GAMAL, AND DHARMENDRA SARASWAT. (2023). Toward Generalization of Deep Learning-Based Plant Disease Identification Under Controlled and Field Conditions. *IEEE*. 11, pp.9042 - 9057. http://DOI:10.1109/ACCESS.2023.3240100

[20] Saikat Datta and Nitin Gupta. (2023). A Novel Approach For the Detection of Tea Leaf Disease Using Deep Neural Network. *Elsevier*. 218, pp.2273-2286. https://doi.org/10.1016/j.procs.2023.01.203

[21] Sachin Dahiya, Tarun Gulati, and Dushyant Gupta. (2022). Performance analysis of deep learning architectures for plant leaves disease detection. *Elsevier*. 24, pp.1-10. https://doi.org/10.1016/j.measen.2022.100581

[22] A. Umamageswari, and S. Deepa, K. Raja. (2022). An enhanced approach for leaf disease identification and classification using deep learning techniques. *Elsevier*. 24, pp.1-6. https://doi.org/10.1016/j.measen.2022.100568

[23] C Jackulin, and S. Murugavalli. (2022). A comprehensive review of the detection of plant disease using machine learning and deep learning approaches. *Elsevier*. 24, pp.1-10. https://doi.org/10.1016/j.measen.2022.100441

[24] U Sanath Rao, R Swathi, V Sanjana, L Arpitha, K Chandrasekhar, Chinmayi, and Pramod Kumar Naik. (2021). Deep Learning Precision Farming: Grapes and Mango Leaf Disease Detection by Transfer Learning. *Elsevier*. 2(2), pp.535-544. https://doi.org/10.1016/j.gltp.2021.08.002

[25] Zhong, Y., & Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. Computers and Electronics in Agriculture, 168, 105146. http://doi:10.1016/j.compag.2019.105146

[26] Noon, S. K., Amjad, M., Qureshi, M. A., & Mannan, A. (2020). Use of Deep Learning Techniques for Identification of Plant Leaf Stresses: A Review. Sustainable Computing: Informatics and Systems, 100443. http://doi:10.1016/j.suscom.2020.100443

[27] Tiwari, V., Joshi, R. C., & Dutta, M. K. (2021). Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. Ecological Informatics, 63, 101289. http://doi:10.1016/j.ecoinf.2021.101289

[28] Yadav, S., Sengar, N., Singh, A., Singh, A., & Dutta, M. K. (2021). Identification of disease using deep learning and evaluation of bacteriosis in peach leaf. Ecological Informatics, 61, 101247. http://doi:10.1016/j.ecoinf.2021.101247

[29] Li, L., Zhang, S., & Wang, B. (2021). Plant Disease Detection and Classification by Deep Learning—A Review. IEEE Access, 9, 56683–56698. http://doi:10.1109/access.2021.3069646

[30] Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2020). Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics, 101182. http://doi:10.1016/j.ecoinf.2020.101182

[31] Thangaraj, R., Anandamurugan, S., & Kaliappan, V. K. (2020). Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. Journal of Plant Diseases and Protection. http://doi:10.1007/s41348-020-00403-0

[32] Ngugi, L. C., Abelwahab, M., & Abo-Zahhad, M. (2020). Recent Advances in Image Processing Techniques for Automated Leaf Pest and Disease Recognition - A Review. Information Processing in Agriculture. http://doi:10.1016/j.inpa.2020.04.004

[33] Abed, S. H., Al-Waisy, A. S., Mohammed, H. J., & Al-Fahdawi, S. (2021). A modern deep learning framework in robot vision for automated bean leaves diseases detection. International Journal of Intelligent Robotics and Applications, 5(2), 235–251. http://doi:10.1007/s41315-021-00174-3

[34] R, K. lakshmi, & Savarimuthu, N. (2021). PLDD- A Deep Learning Based Plant Leaf Disease Detection. IEEE Consumer Electronics Magazine, 1–1. http://doi:10.1109/mce.2021.3083976

[35] Hu, G., Wang, H., Zhang, Y., & Wan, M. (2021). Detection and severity analysis of tea leaf blight based on deep learning. Computers & Electrical Engineering, 90, 107023. http://doi:10.1016/j.compeleceng.2021.107023

[36] Sachdeva, G., Singh, P., & Kaur, P. (2021). Plant leaf disease classification using deep Convolutional neural network with Bayesian learning. Materials Today: Proceedings, 45, 5584–5590. http://doi:10.1016/j.matpr.2021.02.312

[37] Zhou, C., Zhou, S., Xing, J., & Song, J. (2021). Tomato Leaf Disease Identification by Restructured Deep Residual Dense Network. IEEE Access, 9, 28822–28831. http://doi:10.1109/access.2021.3058947

[38] Divakar, S., Bhattacharjee, A., & Priyadarshini, R. (2021). Smote-DL: A Deep Learning Based Plant Disease Detection Method. 2021 6th International Conference for Convergence in Technology (I2CT). http://doi:10.1109/i2ct51068.2021.9417920

[39] Jiang, Z., Dong, Z., Jiang, W., & Yang, Y. (2021). Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. Computers and Electronics in Agriculture, 186, 106184. http://doi:10.1016/j.compag.2021.106184

[40] Goyal, L., Sharma, C. M., Singh, A., & Singh, P. K. (2021). Leaf and spike wheat disease detection & classification using an improved deep convolutional architecture. Informatics in Medicine Unlocked, 25, 100642. http://doi:10.1016/j.imu.2021.100642

[41] A. Hughes and M. Salathé, "PlantVillage dataset," *Pennsylvania State University*, 2017. [Online]. Available: https://www.kaggle.com/datasets/emmarex/plantdisease.

[42] Srikanth, G., Raghavendran, C.V., Prabhu, M.R., Radha, M., Kumari, N.V.S. & Francis, S.K., 2025. Climate Change Impact on Geographical Region and Healthcare Analysis Using Deep Learning Algorithms. Remote Sensing in Earth Systems Sciences, 130359.

[43] Ravikumar Ch1, Marepalli Radha2, Maragoni Mahendar3, Pinnapureddy Manasa"A comparative analysis for deep-learning-based approaches for image forgery detection International journalof Systematic Innovation https://doi.org/10.6977/IJoSI.202403_8(1).0001

[44] Anitha Patil. (2019). Distributed Programming Frameworks in Cloud Platforms. International Journal of Recent Technology and Engineering (IJRTE). 7(6), pp.611-619.

[45] A. Patil and S. Govindaraj, "An AI Enabled Framework for MRI-based Data Analytics for Efficient Brain Stroke Detection," 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ACCAI58221.2023.10201136.

[46] Sreedhar Bhukya, A Novel Methodology for Secure De duplication of Imagedata in Cloud Computing using Compressive Sensing and Random Pixel Exchanging, Journal of Theoretical and Applied Information Technology (JATIT), Vol.102. No 4, ISSN: 1992-8645 (2024), SCOPUS.

[47] Sreedhar Bhukya, Multiclass Supervised Learning Approach for SAR-COV2 Severity and Scope Prediction: SC2SSP Framework, Volume (12) issue (1), 2025-01-31, SCOPUS.