Machine Learning system for accurate and reliable detection for plant diseases

Nazim Uddin, Tanbirul Alam, Fahim Yareed, Ahmad Farruk Towsif, Md. Sabbir Ahmed, and Md. Tawhid Anwar

***Abstract*—Crop pests and diseases present substantial threats to global farming, affecting both yield and food security. To overcome these problems, a capable machine learning (ML) system that is specifically designed to detect crop diseases with high precision and dependability is essential. By implementing machine-learning algorithms with deep-learning models, our system can analyze crop images and deliver accurate detection results. In pre-processing, the model uses advanced image processing techniques to collect required features from plant images, such as color, texture, and shape characteristics. The obtained features are then used to train an efficient machine-learning model on a vast dataset of different diseased plant images. The system is intended to precisely identify the overall condition of plants by analyzing the patterns that have been acquired from the image data. Additionally, it will offer clear explanations for its predictions. The system's efficiency was verified through an in-depth testing process, and the results overwhelmingly show its high precision in comparison with traditional methods. This system will be very effective for farmers and agricultural professionals to make better decisions and successfully manage plant health.**

***Index Terms*—Plant disease detection; Machine Learning; Deep Learning; Image Classification; Automated plant diagnosis; Disease Identification.**

I. INTRODUCTION

AGRICULTURAL production is essential for meeting the growing global population's food needs, providing farmers with livelihoods, and supporting economies worldwide. As the global population continues to grow, agricultural production must meet increasing demands for food. Furthermore, stable and productive farming systems are essential for maintaining food security, preventing hunger, and reducing poverty worldwide.

Plants are not only crucial for humans [15], but also for animals as they rely on these plants for food, oxygen, and other needs. The increasing demand for food and agricultural products places immense pressure on farming systems to produce more while maintaining sustainability and environmental balance. However, one of the major threats to agricultural productivity is plant diseases, which can devastate crops, reduce yields, and cause significant financial losses for farmers. Some diseases can spread rapidly, causing epidemics that affect entire regions or countries, resulting in food shortages and market instability. Detecting plant diseases quickly and accurately helps prevent them from spreading, saves resources, and strengthens farms against new problems.

To prevent losses in yield, it is very crucial to detect plant diseases. [11] However, manual observation of plant diseases is extremely challenging, requiring significant labor, expertise in plant diseases, and a considerable amount of time. Due to such circumstances, it has become important to automate and detect diseases early and classify them accordingly. The advancement in technology has enhanced the application of plant disease detection and protection techniques in farming. Recent developments in computing have introduced AI and machine learning, enabling the automatic detection of plant diseases and significantly improving disease management. Farmers will benefit from an automated disease detection system as the results are very precise, and diseases will be detected in a very short amount of time [15]. There are various classification techniques to classify different plant diseases, but due to the introduction of the classification technique, deep learning combined with Neural Network resulted in better classification and accuracy. The Convolutional Neural Network CNN is a commonly used machine learning method in deep neural networks, which demonstrates better performance compared to conventional methods. Further improvements and developments in CNN, including hybrid architectures, have been introduced to enhance both the stability and accuracy of CNN methods [9]. Furthermore, researchers are actively exploring ways to make machine-learning models more transparent and effective in diagnosing rare diseases.

II. PROBLEM STATEMENT

One of the most critical worldwide concerns facing humanity today is food insecurity [6], and one of the major contributing factors to the problem is plant diseases. The manual observation of plants and knowledge of plant diseases requires more time and is very tiresome [10]. As a result, the productivity of farmers is greatly decreased. Therefore, to minimize crop losses, it is critical to identify plant diseases early. These factors make automatic disease detection and classification essential [10]. Our work is also based on this emerging topic. In building a machine learning system for accurate and reliable identification of plant diseases, one of the primary steps lies in the complexity and variability of plant disease symptoms. Of the papers we reviewed, almost all of them used advanced machine learning models, particularly deep learning, and achieved high accuracy by leveraging large datasets. While deep learning models have demonstrated significant potential in achieving high accuracy for disease classification with large datasets, they also present challenges in terms of high computational requirements and the necessity for extensive labeled data. Reducing model complexity by selecting fewer features or focusing on specific disease categories can help streamline the system, but this often leads to a decline in overall accuracy and robustness, especially in real-world agricultural environments where multiple diseases may coexist. Our main goal, therefore, is to create a system that is able to balance the need for accurate and reliable detection with computational efficiency, ensuring that it performs well in dynamic agricultural environments.

III. Research Objective

This project aims at creating an optimized and efficient system for accurate and reliable plant disease detection. The following are the specific objectives for the project:

- Classifying various plant species based on their leaf characteristics and detecting specific diseases by analyzing the leaf conditions.
- Early detection of diseases for minimizing the negative impacts of diseases on crop yield and quality.
- Implementing efficient techniques of disease detection compared to traditional methods, to minimize cost and time consumption.
- Enhancing the detection accuracy of diseases in plants.

IV. Literature Review

A literature review is an analytical assessment of previous studies or published papers conducted on a specific subject. Its purpose is to critically examine the main findings, methodologies, and theoretical frameworks presented in previous studies. As we are working on a Machine Learning system for accurate and reliable detection of plant diseases, from the literature reviews, our main intention is to gain a deep understanding of the current and main context of knowledge, identification of any lacking in the paper, and establish a foundation for future studies.

In this paper [11], Kulkarni et al., along with other authors, focused on detecting plant diseases using image processing techniques and ML. They used the PlantVillage dataset, which contains around 87,000 leaf images of diseased plants. Image processing techniques were used to analyze color and texture features for disease detection. Additionally, data preprocessing and feature extraction were conducted on the images, and back propagation neural network (BPNN) was used for classification, with image processing methods like boundary detection and spot detection assisted in disease detection. The system successfully detected plant diseases, achieving an average accuracy of 93% in detecting plant diseases, with specific accuracy and F1 scores for different plants. However, the high computational demands of a neural network may limit its usability for real-time applications.

In this paper [9], author David et al. reviewed multiple papers that used different ways and approaches to detect diseases of tomato leaves using deep learning methods and emphasized the importance of detecting such diseases early to help farmers. This study used a dataset consisting of multiple tomato leaf images affected by various diseases to test and train deep-learning models. They utilized deep CNN models and techniques, supervised learning techniques, such as classification tree models, and a hybrid CNN-RNN model. Their paper combined the applications of Convolutional Neural Networks and Recurrent Neural Networks for early disease detection of tomato leaf diseases. Their study reveals that deep CNN techniques significantly enhance detection rates. As a result, the hybrid CNN-RNN model was suggested for early identification of diseases, and it demonstrated improved disease detection rates compared to traditional methods, with accuracies ranging from 93% to 99.35%. While the research presents promising results in disease detection, further advancements can be achieved by exploring advanced optimization techniques, expanding the dataset, and addressing the limitations of specific model architectures.

John et al. in this paper [10] studied existing works on developing accurate automated plant disease detection systems for early recognition and treatment of diseases to nurture healthy plants and increase agricultural yield and quality. They also compared and suggested various methodologies for accurately recognizing and classifying different types of plant infections. The commonly used input dataset in various research were images of plant leaves, but other studies have employed datasets gathered through remote sensing techniques for plant image recognition and classification systems, and some have used datasets including satellite spectral data and ground-truth yield sampling to develop disease mapping systems for crops such as carrots. Researchers for identifying and classifying plant diseases have used different machine learning approaches, including Extreme Gradient Boosting Decision Tree, Convolutional Neural Networks, K-Means Clustering, Support Vector Machines, and Random Forest. One study implemented a machine learning method called Extreme Gradient Boosting Decision Tree to classify rice leaf diseases and was able to correctly identify 86.58% of the diseases. Some disease detection methods might not work well on large farms, raising concerns about the practical implementation of these systems in commercial farming operations.

Similarly, Shelar et al. in this paper [15], focused on developing a Recognition Model for disease detection of plants that utilize images of leaves for accurate plant disease detection. The researchers used the PlantVillage dataset, a large collection of around 87,000 RGB images released by CrowdAI for disease classification. After preprocessing and augmenting the data, a Convolutional Neural Network (CNN) based on the VGG-19 architecture was developed. The developed CNN model demonstrated high accuracy, with a success rate of 99.53%. However, the system's overall result may be affected by the quality of the input images. A significant drop in accuracy may occur in situations when the photos are incompatible or of poor quality.

The author of this work, Hasan et al. [8], provided a thorough summary of the current state of deep learning in the diagnosis of plant diseases. It highlights the main barriers and developments in applying deep learning techniques for the early identification and classification of plant diseases. Here, the PlantVillage and RoCoLe databases make up the many publicly accessible datasets that the researchers use to train and assess their models. To increase the precision and efficacy of disease determination, they looked into a number of deep learning architectures, including CNNs (convolutional neural networks) and transfer learning. Furthermore, the study produces good results, such as a 98.42% accuracy percentage for detecting apple leaf diseases using GoogLeNet. Also, it highlights the efficacy of transfer learning by contrasting the performance of different models. However, the paper also identifies numerous challenges, such as segmentation sensitivity, the need for large, labeled datasets, and overfitting issues. Lastly, future work is suggested to focus on improving real-time detection capabilities and handling multiple infections simultaneously.

The author of the paper, Mohanty et al. [4], focused on applying deep learning techniques to identify plant diseases using images. The main issue is the difficulty in the rapid identification of crop diseases, which threatens food security. The researchers used a dataset named Plant Village, containing 54,306 images of unhealthy and healthy plant leaves. They trained a deep CNN model to classify 14 crop species and 26 diseases. The system achieved a result of 99.35%, showcasing the practicality of this approach. The authors demonstrated the potential for mobile device-assisted disease diagnosis worldwide. Furthermore, the model's overall result on images collected in uncontrolled environments was lower, indicating a need for further refinement.

This paper [14], by Sujatha et al., compared the performance outcomes of machine learning and deep learning methods in detecting plant leaf diseases, specifically citrus leaf diseases. The identification of plant diseases is the main topic of the article since it is vital to the agricultural sector in preventing crop damage and financial loss.609 photos of citrus leaves—both healthy and diseased—that were gathered from the Citrus Research Center in Punjab, Pakistan, make up the dataset. The study uses 10-fold cross-validation to evaluate the various machine learning (ML) techniques (SVM, RF, SGD) and deep learning (DL) models (Inception-v3, VGG-16, and VGG-19) for disease categorization. In terms of classification accuracy, DL approaches performed better than ML methods; VGG-16 achieved a maximum accuracy of 89.5%. The study indicates that DL models—of which VGG-16 is the most accurate—perform better on this task than ML models. According to the study, uncertain reasoning and bio-inspired techniques could be used in future research to increase accuracy, particularly with smaller datasets.

Also, the paper [2], by Akhtar et al., compared the performance results of various Machine Learning techniques for detecting and classifying plant diseases from images of diseased leaves. In order to enhance agricultural productivity maximization, the study attempts to increase the accuracy and robustness of automatic plant disease recognition and categorization. The dataset used in this study by the authors includes 40 photos of rose leaves that have been separated into subsets that are healthy and diseased, with conditions like Anthracnose and black spots. Image segmentation, feature extraction (using Statistical Features, DCT, and DWT), and classification using KNN, Naïve Bayes, SVM, Decision Tree, and RNN are the steps in the process. The maximum accuracy of 94.45% was attained by combining the Support Vector Machine (SVM) with the DCT and DWT features. The accuracy of plant disease diagnosis and classification is significantly improved by the suggested procedure, which performs better than alternative approaches. The paper notes the small dataset and recommends that larger datasets and improved image processing and classification methods be used in future research.

The author of this paper, Rashmi et al. [13], presents a machine-learning approach to detect plant diseases using images of unhealthy leaves, addressing the challenge of manual disease detection, which is inefficient for large farms. To segment apple leaf disease from the original images, the authors suggest a computational solution by an automated system that involves machine learning and image processing methods to classify plant diseases. The dataset contains pictures of both healthy and unhealthy leaves sourced from the internet. Their approach consists of capturing leaf images, noise removal with preprocessing steps, and extracting features using Gray level co-occurrence matrix (GLCM), followed by related disease classification through SVM. Well, these are defined as four classes: Class 0 for Alternaria Alternata, Class 1 for Anthracnose, Class 2 for Bacterial Blight, and finally, the healthy leaves that make up Class 3. The leaves were accurately classified into these classes by the SVM. The results demonstrate that the proposed model can accurately diagnose plant diseases, which lays a solid foundation for farmers to solve their practical problems. But the dataset is small, and more work needs to be done in terms of scale, including investigating how well it works on a real farm. The authors also mentioned that the system works well in a controlled environment, even though there is still room to improve performance when scaling.

In this paper [7], Gobalakrishnan et al., along with other authors, discussed various reviews on machine learning-based methods to detect plant diseases from images of unhealthy leaves. They focused on the problem of traditional methods, which require more time and are costly. Moreover, the authors focused on the implementation of machine-learning algorithms for automated disease identification. They categorize plant diseases into infectious (fungal, bacterial, viral) and non-infectious. Additionally, the paper surveys different image-based detection techniques, such as genetic algorithms, wavelet transforms, and convolutional neural networks (CNN). These techniques have shown varied accuracy across different crops, with CNN achieving the highest accuracy of 99.35% for plant disease detection. The findings suggest that while many methods are effective, none can universally detect all diseases.

Future work should focus on creating systems capable of detecting a wider variety of diseases and pests for better real-world applications.

The author of the paper Wang et al. [1], worked on a neural network-based image recognition technique to identify and distinguish plant diseases from images of diseased leaves. Moreover, they have focused on the need for fast and accurate diagnosis of diseases that are hard to differentiate visually. Their used dataset included 185 images of diseased wheat and grapes. The study used methods like neural networks, such as backpropagation (BP), radial basis function (RBF), generalized regression networks (GRNN), and probabilistic neural networks (PNN) to classify diseases. The paper shows how to detect harmful plant disorders such as grape powdery mildew, grape downy mildew, wheat leaf rust, and wheat stripe rust. The outcome showed 100% accuracy in identifying diseases using BP, GRNN, and PNN. At the same time, RBF achieved 97.5% accuracy for wheat and 94.29% for grapes. Finally, they highlighted the efficiency of neural networks in plant disease detection, but there are chances for improvements to handle more complex disease cases and with much broader datasets.

The author of the research, Francis et al. [3], focused on detecting diseases in the leaves of pepper plants by applying enhanced algorithms and methods. They discussed the effectiveness of digital image processing in the identification of plant diseases. The methodology includes capturing leaf images, converting them into HSV color space, and applying segmentation techniques like masking and threshold-based segmentation to identify diseased areas. The study uses a backpropagation neural network (BPNN) to classify the diseases. In addition, two specific diseases (Berry Spot and Quick Wilt) were identified using the system, with successful disease identification based on a calculated damage ratio (DR). Again, the results indicate that this method is effective for detecting pepper plant diseases. However, these techniques could be improved if we focus on testing the system in diverse lighting conditions and identifying more plant diseases.

Akulwar et al. in this paper [5], prioritized the challenges faced by farmers in identifying crop diseases using machine learning techniques to detect crop diseases. So that it becomes easier for farmers to identify crop diseases early and efficiently manage them to optimize farming outcomes. The dataset they used contained over 1,000 images of strawberry and citrus crops. The images were analyzed through various image processing techniques, like feature extraction. The author used Methods like Convolutional Neural Network (CNN) for crop type identification, and machine learning algorithms like Linear Regression for predicting crop yields. The system successfully detected crop diseases such as Gray Mold and Citrus Canker with high accuracy, where the disease Gray Mold detection had an accuracy of 100%. Lastly, the authors suggested the use of more diverse datasets and real-time testing that would help in enhancing the system's accuracy and stability.

The authors of the paper [17], Javidan et al., and Banakar et al. used automatic K-means clustering for image segmentation and multi-class support vector machines to analyze grape leaf diseases. They used a dataset consisting of healthy and unhealthy images of grapes showing various symptoms. It contained 1,309 images of Black Measles, 1,105 images of Black Rot, 1,058 images of Leaf Blight, and 458 images of healthy leaves. The authors implemented a combination of methods, including K-means clustering for image segmentation, RGB, HSV, and Lab color models for feature extraction and classification using multi-class Support Vector Machine (SVM) with Principal Component Analysis (PCA) for dimensionality reduction. The study showed that combining K-means clustering with SVM and PCA for reduction resulted in an accuracy of 98.97%, and without PCA it was 98.7%. The proposed method for this paper was an SVM-based method that showed higher accuracy compared to deep learning models while using minimum data and less processing time. The limitation of this paper was that the models and methods are dependent on the quality and lighting conditions of images, which can affect the overall system's performance. Lastly, the authors suggested that integrating more advanced feature selection techniques will improve accuracy, and future work should be focused on improving the ability to handle images with multiple overlapping diseases.

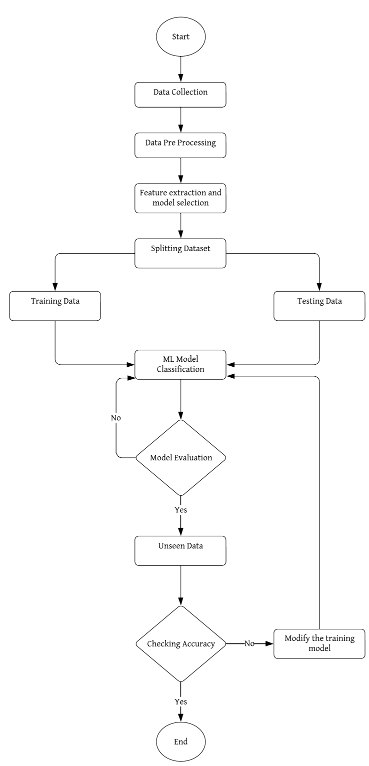
In this paper, Chowdhury et al. [16], prioritized their work on the early detection of plant diseases. For their work, they have chosen Bell Peppers, Tomatoes, and Potatoes. As these plants may have various diseases that can heavily affect the production rate eventually, the authors were trying to detect if these plants have any kind of symptoms of being affected in the near future. For this research or study authors have worked on a pre-collected dataset, which is obtained from Kaggle. However, the dataset is fairly rich in terms of information as it has over 17,400 images of these plants. They divided this dataset into a test set and a training set and used some deep learning models, namely SVM, Logistic Regression, KNN, and CNN. Among all the methods used, CNN provides the highest accuracy rate in terms of testing, which is 85.31%, followed by SVM, Logistic Regression, and KNN, respectively. Though their primary goal was to detect diseases of Bangladeshi plants, they haven't worked with native plants, and their study was limited to only three plants. Otherwise, their study on plant disease was pretty decent and reached.

The author Nguyen et al. in the paper [12] developed a method using hyperspectral imaging and deep learning for early detection of grapevine viral disease to prevent mass spread. The key issue of this research is developing an accurate and scalable method for detecting plant viral diseases like GVCV in their early stage. The early symptoms are difficult to notice manually as they are very vague. A total of 40 images of both healthy and GVCV-infected grapevines were used, and images were taken in four different stages 30,50,70,90 days, respectively. For image capture, hyperspectral imaging was used, SVM and RF were used for Classification, and lastly, 2D and 3D CNN for spatial spectrum features. Wavelengths (900-940nm in NIR) and (400-700nm in VIS) were used for disease detection. In index-wise classification, SVM had the highest accuracy of 90.24%, in pixel-wise classification, the 50-feature model had the highest accuracy of 95.30% on 50 days, and in spatial-spectral classification, feature 3D CNN had a higher extraction rate. The key finding of this paper is certain wavelengths, especially in NIR and VIS ranges, are better and must for detecting plant viral disease. The limitation of this research was a limited dataset of 40 and complicated models like CNN, requiring higher computational power, and the authors suggested larger datasets and more efficient models to be used for real-time applications.

V. Workflow

Our work can be presented in the following steps:

- Firstly, we will collect a diverse dataset of plant images, which will include both healthy and disease-affected plant images with various disease types.
- Secondly, we will be pre-processing the data. In this process, we will remove noise or irrelevant parts of the image. We will normalize the pixel values, resize images to a uniform size, and implement other pre-processing techniques based on the specific requirements.
- Finally, the processed data will go through the model classifying and processing phase. Here, the selected models are applied to generate the desired outcome.



Workflow

VI. Dataset Description and Preparation

For our project, we have explored 7 different datasets. Among them, we have chosen a dataset named “Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network” taken from the Mendeley Data repository.

This dataset is organized into 39 classes, 38 different classes of healthy and diseased plant leaves, and one class of background images without leaves, totaling 61,486 images. Other than the background images without leaves, we have used the rest of the 38 classes of plant leaves. The classes are Apple Scab, Apple black rot, Apple cedar apple rust, Apple healthy, Blueberry healthy, Cherry powdery mildew, Cherry healthy, Corn gray leaf spot, Corn common rust, Corn northern leaf blight, Corn healthy, Grape black rot, Grape black measles, Grape leaf blight, Grape healthy, Orange haunglongbing, Peach bacterial spot, Peach healthy, Pepper bacterial spot, Pepper healthy, Potato early blight, Potato healthy, Potato late blight, Raspberry healthy, Soybean healthy, Squash powdery mildew, Strawberry healthy, Strawberry leaf scorch, Tomato bacterial spot, Tomato early blight, Tomato healthy, Tomato late blight, Tomato leaf mold, Tomato septoria leaf spot, Tomato spider mites two-spotted spider mite, Tomato target spot, Tomato mosaic virus, and Tomato yellow leaf curl virus. The dimension of each image in the dataset is (256x256).

The 38 classes of diseased and healthy plants consist of 59,812 images, and all the images are divided into train (80%), test (10%), and validation (10%). We have used a total of 47,848 images for our model training, 5982 images for testing, and the rest of the 5982 images used to validate the model.

Train Data (80%)	47,848
Test Data (10%)	5,982
Validation Data (10%)	5,982
Total	**59,812**

TABLE I
Distribution of Train, Test, and Validation Data

A. Dataset Sample

Fig. 1. Sample images of the Dataset

B. Image Conversion to arrays

Since our data is in image format, to make it workable for our models to run, we first convert it into an image array with the help of Keras functions. This step is crucial as neural networks operate on numerical data. Therefore, converting images into numerical arrays is essential to enable model training and inference.

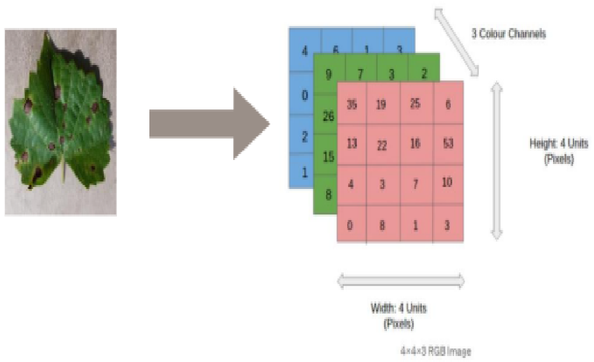


Fig. 2. Converting images to array

C. Data Labeling

To improve the learning algorithm in our dataset, we will provide labels to our data. After converting the images into arrays, the data is labeled by their respective classes. This will help classify the data during model training.

Fig. 3. Labeling Data

D. Resize and Rescale

Models cannot function with a dataset having images of various shapes. As a result, resizing and rescaling are required for a dataset that has images of different shapes or sizes. Additionally, a dataset's time and space complexity is reduced by resizing every image to remove redundant data, which improves model efficiency. For our dataset, in order to adjust the dimension of all the images, we set the dimension based on each model's default configuration to get better performance during the model training phase. The image pixel values within the dataset are also normalized to expedite model training. Rescaling by dividing pixel values by 255 scales them to a range between 0 and 1. Normalization ensures consistent data dimensions and prevents potential bias in the training process.

E. Data Augmentation

Data augmentation is critically important because it enhances machine learning models' robustness, accuracy, and generalization ability. It artificially expands the dataset by generating new, varied samples, reducing the risk of overfitting. By exposing the model to diverse variations of the same data, it learns to generalize better. As a result, it accurately detects diseases in real-world scenarios. For our dataset, we have implemented some augmentation techniques. The explanation of each augmentation technique used is given below.

Horizontal and Vertical Flipping: Horizontal flipping flips images horizontally, and vertical Flipping flips images vertically. Plant leaves are among the many symmetrical things that can appear in flipped orientations in real-world situations. This makes the model robust to horizontal or vertical orientation changes. For example, if a leaf with a disease is photographed from the left, the model can also recognize the disease when the leaf is flipped horizontally.

Rotation Range: It rotates images within a specified range of degrees. This helps the model to effectively handle small rotations that may occur naturally in real-world situations, such as taking images of plants from slightly varied angles. In our case, we are rotating the images by an angle between -20 and +20 degrees.

Zoom Range: It zooms in or out of the image within a specified range. So, if the zoom factor is set to 0.8, the image could be zoomed in by 80% or zoomed out by 80%. In our case, we have used a 0.1 zoom range. It will help the model generalize to images where the orientation of leaves is not perfectly aligned. We used a minimum range as excessive zooming out might reduce resolution, and excessive zooming in could crop important parts of the image.

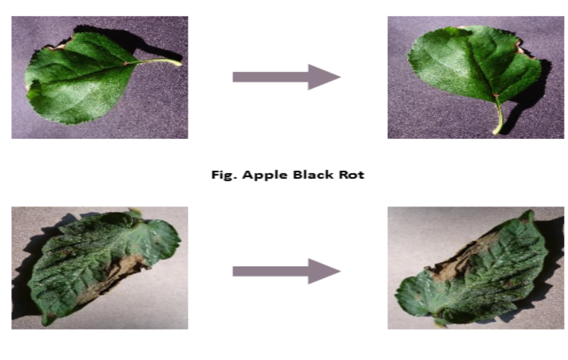


Fig. 4. Sample Augmented Images

VII. Description of the Models

In our project, we have used a total of 5 models. One custom basic CNN model and 4 pre-trained models, Inception v3, MobileNet v2, ViT, and Swin, and conducted a detailed study on the performance and accuracy of each model. The Vision Transformer (ViT) and Swin are two transformer models. The pre-trained models will be transfer learning-based models.

A. Basic CNN

CNNs are fundamental to numerous image classification, object detection, and image segmentation tasks. They excel in feature extraction and pattern recognition tasks through convolutional layers. For our project, we used a custom CNN architecture for plant disease classification using diseased leaf images as shown in Figure 5.

CNN Architecture:

Input Layer:

- Images of size 256×256×3 (height, width, and color channels).

Convolutional Layers:

- First and Second Blocks: It uses 32 and 64 filters of (3x3) for basic feature detection, such as edges, lines, shapes, and textures in leaves. Batch normalization and ReLU activation improve learning. Max pooling downsamples feature maps.
- Third and Fourth Blocks: With 128 and 256 filters of (3x3), respectively, these layers identify complex patterns like leaf spot shapes and color gradients. Max pooling ensures that critical features are retained.

Fully Connected Layers:

- Flatten Layer: It converts 2D spatial information of the feature maps into a single, 1D array, preparing the data in the network for the following fully connected layers.
- Dense Layer: 512 neurons with 20% dropout for overfitting prevention and ReLU activation and L2 regularization (λ=0.001).

Output Layer: A softmax layer with 38 neurons (38 classes).

Activation and Regularization: ReLU for non-linearity, batch normalization for training stability, and dropout to prevent overfitting.

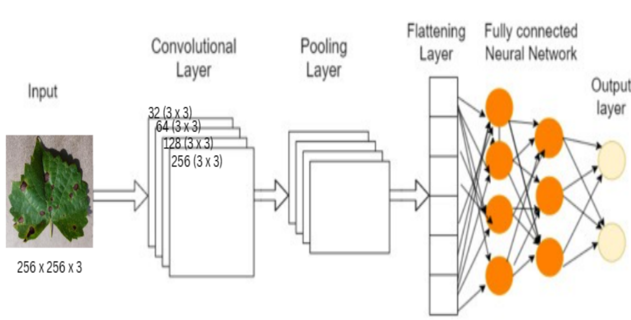


Fig. 5. Basic CNN Architecture

B. Inception v3

Inception v3 is commonly used for image classification and various computer vision tasks. The model enables the network to efficiently capture multi-scale information. We use a pre-trained Inception v3 model for our model classification, like Figure 6.

Inception v3 Architecture:

Base Model:

- Pre-trained weights are used from the ImageNet dataset, which allows it to recognize a wide variety of image patterns.
- During training, the weights of the base model are not updated. This helps retain the generalized features learned during pretraining.

Global Average Pooling: It reduces dimensionality by averaging feature maps across spatial dimensions for each channel. It reduces the computational load of the model.

Output Layer: A softmax layer with 38 neurons predicts probabilities based on the extracted features.

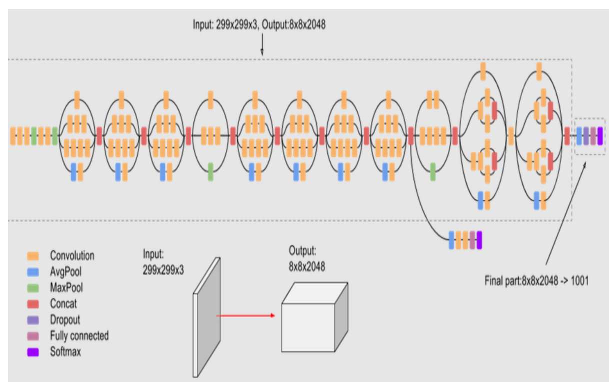


Fig. 6. Inception v3 Architecture

C. MobileNet v2

Similar to Inception v3, MobileNet v2 is also widely used for image classification. It's used in lightweight object detection frameworks for real-time detection tasks. In Figure 7, as we are working on a large dataset, implementing pre-trained MobileNet v2 with fine-tuning can significantly reduce the training time and lower the computational cost.

MobileNet v2 Architecture:

Base Model:

- Pre-trained weights are used from the ImageNet dataset, which allows it to recognize a wide variety of image patterns.
- During training, the weights of the base model are not updated. This helps retain the generalized features learned during pretraining.

- The new layers added will be trained.

Global Average Pooling:

- It reduces dimensionality by averaging feature maps across spatial dimensions for each channel.
- Computational resources are reduced, which allows the model to focus on higher-level features.

Output Layer: 38 neurons with a softmax activation function for multi-class classification (38 classes).

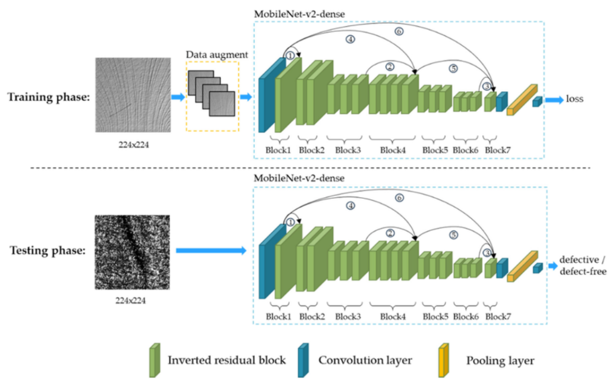


Fig. 7. MobileNet v2 Architecture

D. ViT

Vision Transformer (ViT) is a groundbreaking deep-learning architecture that leverages the power of transformers as shown in Figure 10. It was originally designed to process visual data using natural language. By employing self-attention techniques, ViT divides the image taken as input into patches of consistent size and addresses them as tokens, much like words in a sentence.

For our task, the input images were resized to 224x224 pixels and divided into 16x16 patches. The AdamW optimizer was used during model training, with a warm-up rate to prevent overfitting. A batch size of 32 was selected to optimize GPU utilization without consuming excessive memory. We also used early stopping with a patience of 5 to prevent overfitting and improve generalization.

ViT Architecture:

Image Patch Embedding:

- The input image is split into patches. For example, a 224x224 leaf image can be divided into 16x16 patches.
- Each flattened patch undergoes a learnable linear transformation into a higher-dimensional space. This transformation allows the model to capture richer and more expressive feature representations for each patch.

Positional Encoding:

Positional encoding is added to each patch embedding to preserve spatial information. The transformer encoder then processes the sequence of patches using multi-head self-attention (MHSA) and feedforward neural networks.

Transformer Encoder Layers:

- Multi-head Self-Attention (MHSA): This enables the model to simultaneously focus on various regions of the image, capturing both local and global patterns.
- Feedforward Neural Networks: It consists of two fully connected layers with a non-linear activation function (typically GELU) in between. The fully connected feed-forward networks are applied after self-attention to capture more complex image representations.

Classification Head:

The classification head in ViT uses a CLS token to aggregate global information from all patch embeddings, which is then processed through fully connected layers to predict the class.

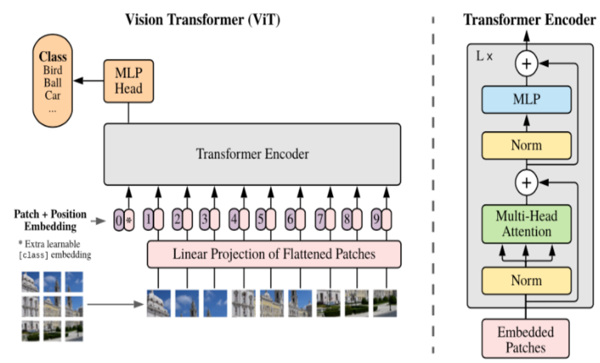


Fig. 8. ViT Architecture

E. Swin

The Shifted Window Transformer (SWIN) is a hierarchical vision transformer specifically designed for large-scale image analysis. It employs a hierarchical architecture with shifted windows, enabling efficient computation and scalability for high-resolution images as displayed in Figure 9. By integrating local and global attention mechanisms, it delivers cutting-edge results across numerous vision applications, including image segmentation.

Swin Architecture:

Unlike the flat structure of ViT, the Swin Transformer processes images at multiple resolutions, effectively capturing multi-scale features. The model employs a window-based multi-head self-attention (W-MSA) mechanism. This approach substantially reduces computational complexity, improving efficiency for large images by decreasing the scaling factor from quadratic to linear with respect to image size.

The initial step in the Swin Transformer involves dividing the input image into a grid of non-overlapping patches, similar to Vision Transformers (ViT). For example, a 224x224 image might be divided into 4x4 windows, each containing 16x16 patches. These patches are then flattened and projected into a higher-dimensional space, becoming the "tokens" for the Transformer. Within each window, standard multi-head self-attention (MHSA) is applied. This involves calculating attention scores for every possible pair of tokens within the window, allowing the model to capture dependencies and the connections that exist between different parts of the image. Following MHSA, the tokens within each window are passed through a series of feedforward layers and normalization layers, enabling the model to learn complex representations and improve its performance. The feature maps are then passed to a supervised head, consisting of fully connected layers and a softmax function, to predict image classes.

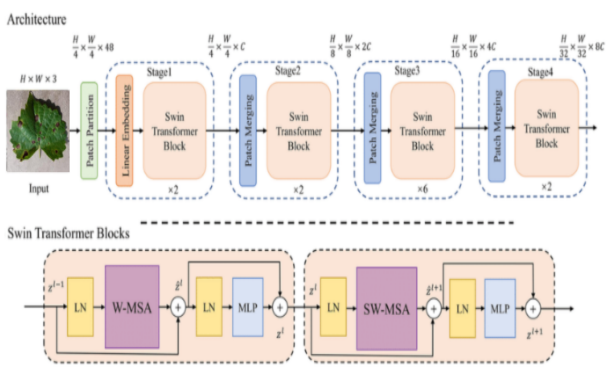


Fig. 9. Swin Architecture

VIII. Result and Discussion

A. Model Tuning

Parameters	Basic CNN	Inception v3	MobileNet v2	ViT	Swin
Input	256×256	299×299	224×224	224×224	224×224
Batch Size	32	32	32	32	32
Channel	3	3	3	3	3
Filter Size	3×3	3×3	3×3	3×3	3×3
Epochs	50	50	50	50	50
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate	0.0001	0.0001	0.0001	0.0001	0.0001
Activation Function	ReLU	ReLU	ReLU	ReLU	ReLU
L2 Regularization	λ=0.001	λ=0.001	λ=0.001	λ=0.001	λ=0.001
Weight Decay Rate	0.0001	0.0001	0.0001	0.0001	0.0001
Dropout Rate	0.2	0.3	0.3	–	–

Fig. 10. Parameters for different Models

All models were trained using standardized parameters to ensure consistency. The input size was set based on each model's default configuration: 256×256 for the basic CNN, 299×299 for Inception v3, and 224×224 for MobileNet v2, ViT, and Swin Transformer. A uniform batch size of 32 was chosen to balance training efficiency and memory usage. All models received 3-channel RGB inputs, and a standard 3×3 convolutional filter was employed to effectively capture fine-grained local spatial features. Each model was trained for 50 epochs using the AdamW optimizer, which combines the benefits of Adam with decoupled weight decay, enhancing generalization. A low learning rate of 0.0001 was selected to ensure stable convergence, particularly important when fine-tuning pretrained models. ReLU was employed as the activation function due to its computational efficiency and ability to mitigate vanishing gradients. To prevent overfitting, L2 regularization with a weight decay factor of λ = 0.001 was applied. Additionally, a dropout rate of 0.2 was used for Basic CNN, 0.3 for InceptionV3 and MobileNetV2, while ViT and Swin did not include explicit dropout, as their transformer-based architectures inherently incorporate regularization techniques such as stochastic depth. This helps reduce overfitting by randomly dropping entire transformer blocks or attention weights during training. The base model was frozen, and only the final layer was trained for both ViT and Swin. As the models were generalizing well, no manual dropout was necessary.

Furthermore, additional training parameters included the use of CrossEntropyLoss as the loss function, a learning rate scheduler to dynamically adjust the learning rate during training, and early stopping to halt training when validation performance stopped improving, thereby reducing the risk of overfitting and saving computational resources.

B. Basic CNN

The Figure 11 illustrates the CNN model's performance, depicting Model Accuracy and Model Loss for training and validation. Over 20 epochs, training accuracy rises from 0.70 to over 0.90, with validation accuracy closely following, indicating strong learning and minimal overfitting. Training and validation losses steadily decrease, reflecting effective error minimization. The model achieved 97.53% training accuracy and 95.03% testing accuracy.

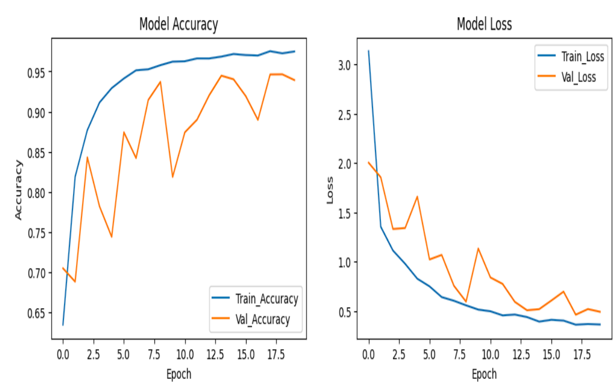


Fig. 11. Accuracy and Loss of Basic CNN

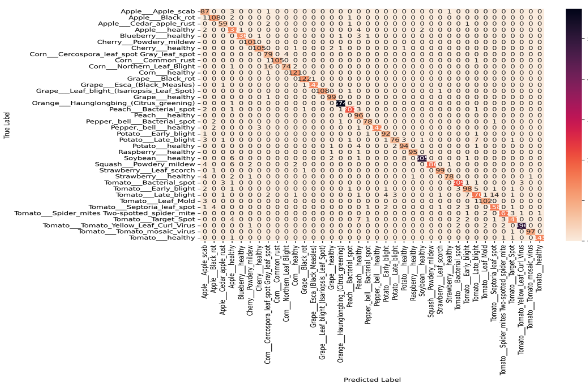


Fig. 12. CNN Confusion Matrix

In the Figure 12, diseases like Apple Rust, Cherry Powdery, and most corn and grape classes were almost perfectly classified, landing near the diagonal. However, subtle symptoms posed challenges: Peach Bacterial Spot was often mistaken for healthy, Pepper Bacterial Spot was misclassified as healthy, and Potato Late Blight was confused with Tomato Late Blight. Even healthy tomato leaves were misidentified as diseased in some cases. Minor confusions also occurred between visually similar diseases, such as mix-ups between potato late and early blight and occasional swaps among tomato diseases.

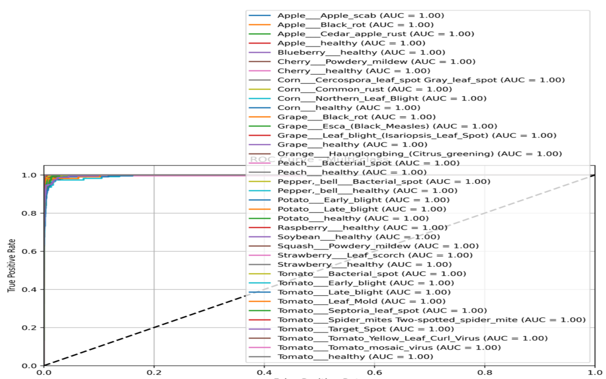


Fig. 13. CNN ROC Curve

In this plot 13, almost all the curves hug the top-left corner, which means the model performs exceptionally well. This is also confirmed by the AUC values—most are 1.00, meaning the model has nearly perfect classification ability for every disease and healthy class, with almost no confusion between them.

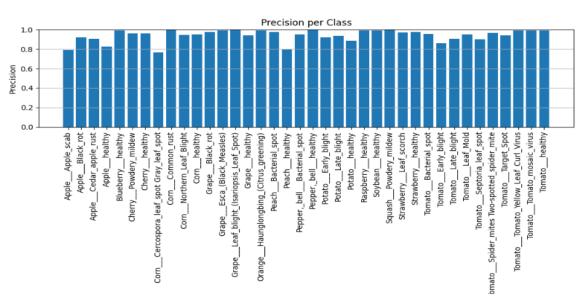


Fig. 14. Precision of CNN

Here, most classes have precision above 0.90, meaning when the model predicts a disease or healthy label, it's correct over 90% of the time—while a few harder cases like Apple scab, Corn Gray Leaf Spot, and Peach Healthy drop into the 0.75–0.85 range, showing they occasionally produce false positives.

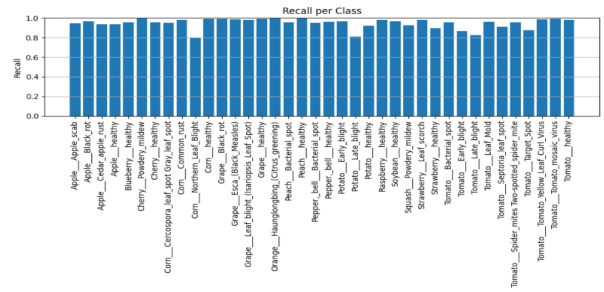


Fig. 15. Recall of CNN

Recall value in almost every class has a bar above 0.90—meaning the CNN detects over 90% of true examples—while a few tougher ones like Corn Northern Leaf Blight, Potato Leaf Blight, and Tomato Late Blight dip lower, showing the model sometimes misses those real cases.

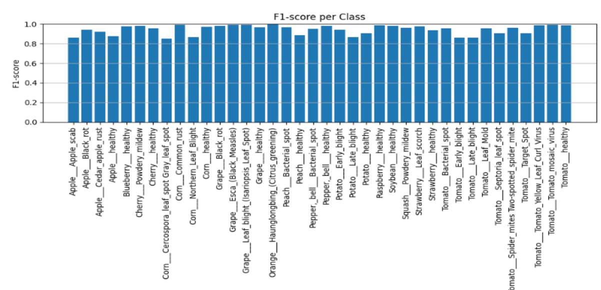


Fig. 16. F1-Score of CNN

In the plot 16, F1 scores above 0.90 (many hitting 1.00), the model both finds and labels those diseases almost flawlessly. A few classes with very subtle or mild symptoms—Apple Scrab, Tomato Early Blight and Tomato Late Blight—dip noticeably lower, highlighting that these particular conditions remain the hardest for the model to detect consistently.

C. Inception v3

The model achieved 92.12% accuracy using early stopping in 22 epochs. The training accuracy increased from 0.80 to over 0.92, demonstrating effective learning. Similarly, validation accuracy showed a steady rise, closely following training accuracy. Both trends indicate strong generalization to unseen data. Overfitting remains minimal, ensuring robust model performance. The model scored 92.12% in training accuracy and 91% in testing accuracy as shown in Figure 17.

Most disease and healthy classes—such as Apple Black Rot, Blueberry healthy, Orange Huanglongbing, Cherry Powdery Mildew, and the various corn, grape, and soybean categories—are correctly identified almost every time, as shown

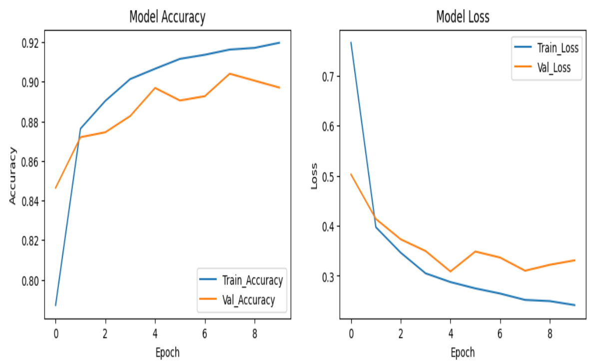


Fig. 17. Accuracy and Loss of Inception v3

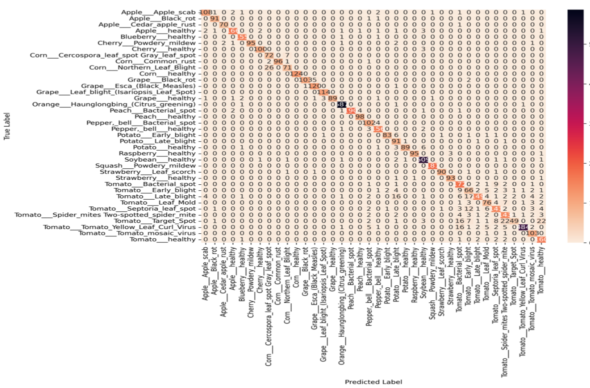


Fig. 18. Inception Confusion Matrix

by the strong diagonal. Some errors occur in milder or visually similar cases: Peach Bacterial Spot is often called "Peach Healthy," some Potato Early and Late Blight samples swap labels, and healthy tomatoes are misread as few other Tomato classes.

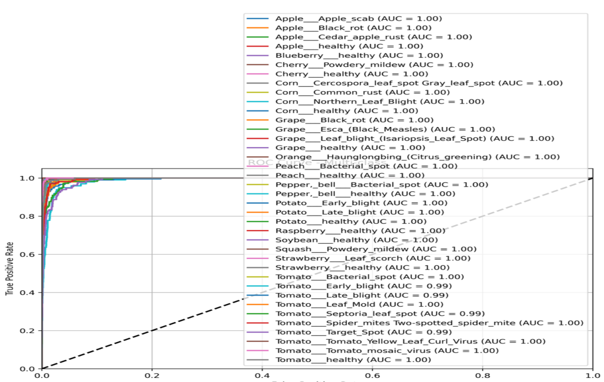


Fig. 19. Inception ROC Curve

In the Figure 19, the AUC values mostly score a perfect 1.00, indicating excellent performance. However, a few classes—like Tomato Early Blight, Tomato Late Blight, and Tomato Septoria Leaf Spot—have AUC values of 0.99, which still reflect strong but slightly less perfect accuracy. Overall, the Inception model performs extremely well in classifying the plant disease dataset, with only minor variations in a few specific classes.

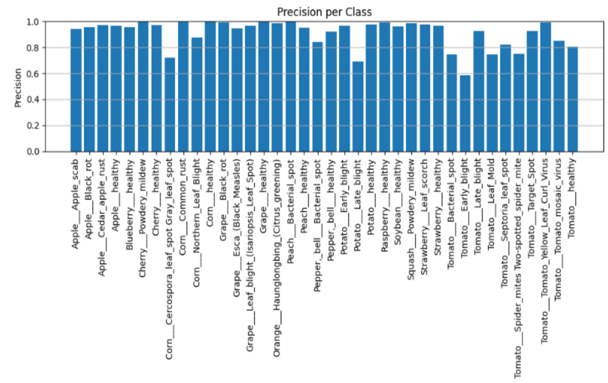


Fig. 20. Precision of Inception

This chart 20 shows nearly every bar sits above 0.90, many close to 1.00. A few tougher cases—like Corn gray leaf spot, Tomato Bacterial Spot, Tomato Early Blight, and Potato late blight—dip lower, meaning the model sometimes mistakes other conditions for those more subtle or similar-looking ones.

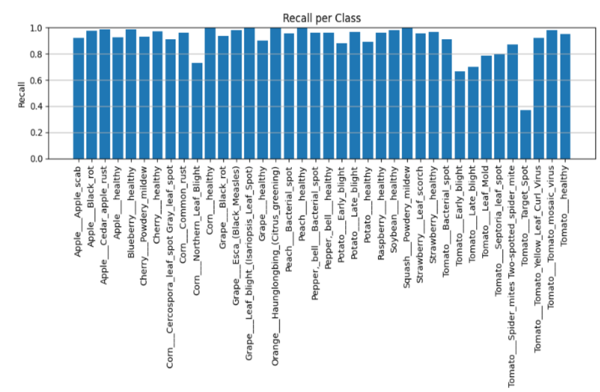


Fig. 21. Recall of Inception

Here, almost every bar hits above 0.90—meaning it catches over 90% of true cases for most classes like Apple black rot, Blueberry Healthy, Grape Esca, and others. A few more challenging conditions dip lower: Corn Northern Leaf is detected about 75% of the time, Tomato early blight roughly 70%, Tomato late blight about 73%, and Tomato Target Spot only around 35%, showing the model still misses many real examples of those specific types.

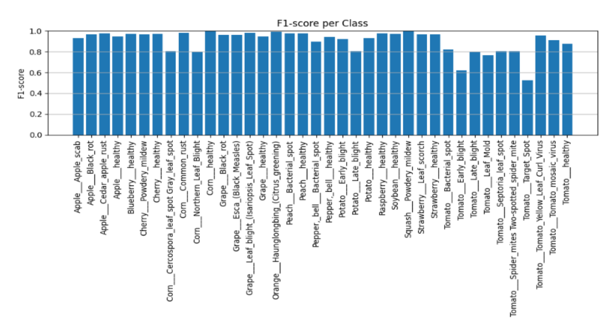


Fig. 22. F1-Score of Inception

This chart shows that nearly all bars rise above 0.90 (many at or near 1.00), indicating the model both detects and labels those diseases almost accurately. A few tougher classes dip noticeably: two Corn classes, Tomato early, and Tomato Target Spot, highlighting that these conditions remain the hardest for the model to catch and classify consistently.

D. MobileNet v2

MobileNet achieved 94.86% accuracy and 94% testing accuracy in Figure 23. Training accuracy steadily increased from 0.86 to over 0.94, showcasing effective learning and improved classification. The Model Loss plot shows a consistent decline from 0.85 to below 0.20, indicating successful error reduction. These trends suggest efficient learning, with increasing accuracy and decreasing loss reflecting strong performance. The model generalizes well to unseen data while maintaining robustness.

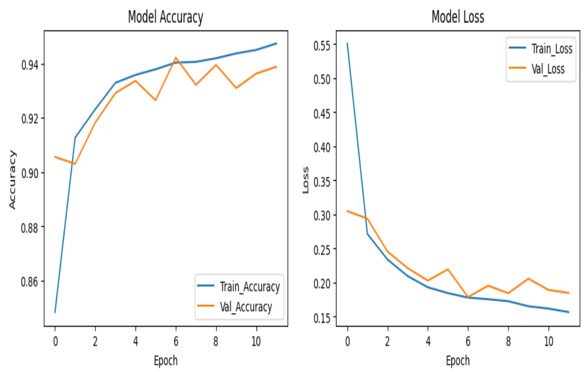


Fig. 23. Accuracy and Loss of MobileNet v2

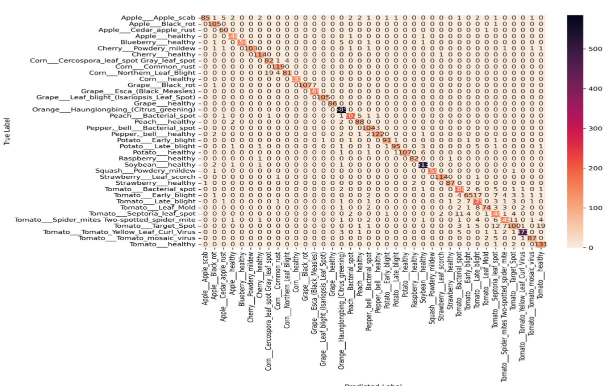


Fig. 24. MobileNet Confusion Matrix

In the Figure 24, most diseases—like Apple Black Rot, Orange Huanglongbing, Cherry Powdery Mildew, and almost every grape and soybean category—land squarely on the diagonal, showing the model's high accuracy. A few mix-ups happen when symptoms look alike or mild: Apple Scab is sometimes called Apple Cedar and Black Rot, Corn Gray Leaf Spot swaps with Leaf Blight, Potato Early and Late Blight confuse each other, and a handful of Tomato Target Spot get labeled as healthy tomatoes. Overall, the model does very well, but still confuses the most similar or subtle cases.

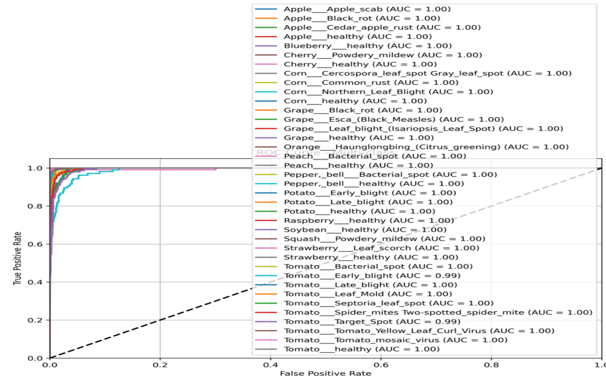


Fig. 25. MobileNet ROC Curve

Here, most classes have an AUC of 1.00, which means the model correctly identifies almost all true cases while making very few false predictions. However, a few tomato-related diseases like Tomato Early Blight and Tomato Target Spot have slightly lower AUCs of 0.99, suggesting a small drop in performance for those specific classes.

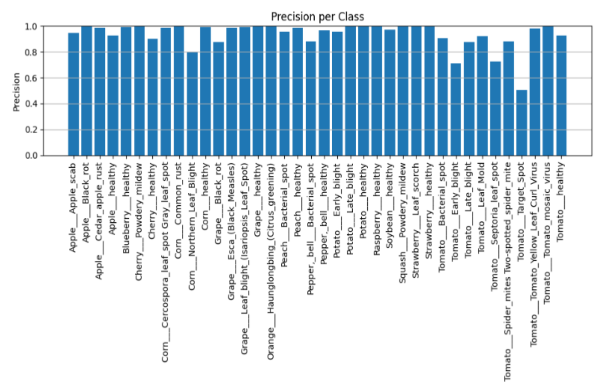


Fig. 26. Precision of MobileNet

In this plot 26, nearly all bars exceed 0.90, many approaching 1.00. However, performance dips slightly for conditions with subtle or overlapping symptoms: Corn Leaf Blight drops to around 0.80, Tomato Early Blight to about 0.75, and Tomato Target Spot falls to roughly 0.50, suggesting these are more frequently confused with other diseases.

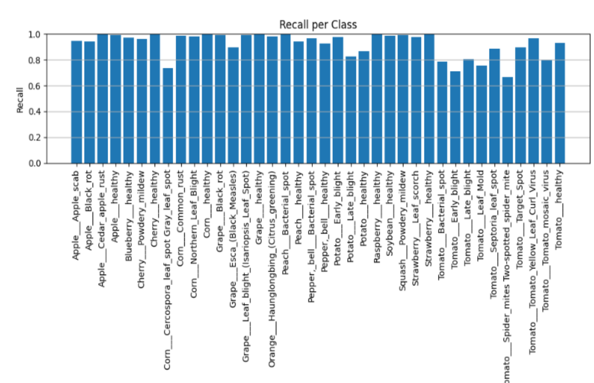


Fig. 27. Recall of MobileNet

In the plot, most bars exceed 0.90, indicating the model accurately detects conditions like Apple Scab, Apple Black Rot, Grape diseases, and Soybean Healthy. However, detection drops for classes with subtle signs—Corn Gray Leaf Spot falls below 0.80, while Tomato Early Blight and Two-Spotted Spider Mite range between 0.60 and 0.75, suggesting MobileNet occasionally fails to catch these actual cases.

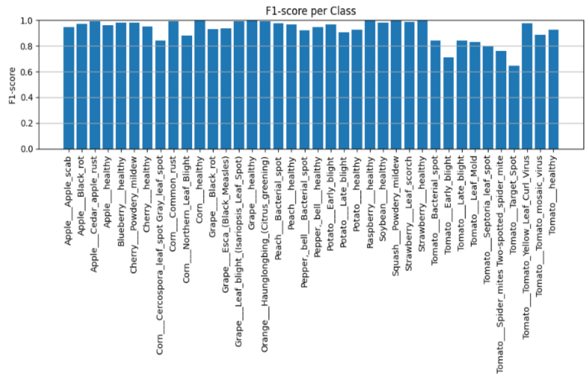


Fig. 28. F1-Score of MobileNet

Here, most classes achieve F1-scores above 0.90, indicating the model reliably detects and classifies those diseases or healthy leaves. However, conditions with subtle or overlapping symptoms are seen in tomato classes ranging between 0.65–0.75, highlighting areas where MobileNet has more difficulty.

E. ViT

In Figure 29, ViT achieved 98.38% training accuracy and 95.45% test accuracy using early stopping in 31 epochs. Training and validation losses steadily decreased from 0.8 to below 0.2, indicating effective error minimization. Training accuracy rose from 75% to over 95%, showcasing strong learning and classification ability. The Validation accuracy followed a consistent upward trend, surpassing 90%.

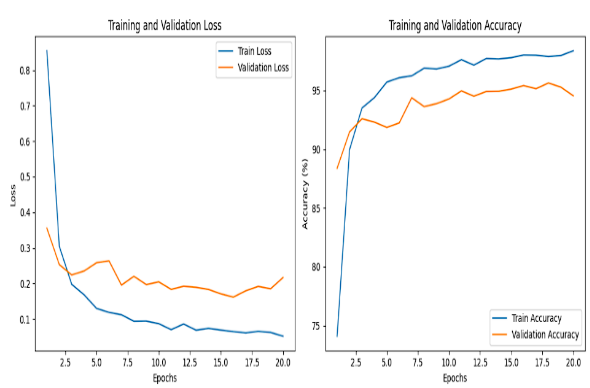


Fig. 29. Accuracy and Loss of ViT

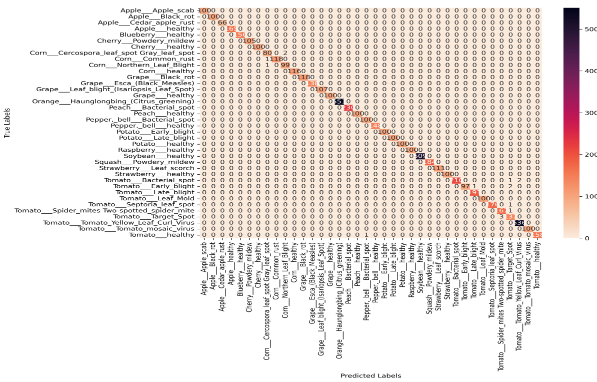


Fig. 30. ViT Confusion Matrix

The ViT confusion matrix in Figure 30 reveals outstanding model performance, with predictions aligning almost perfectly along the diagonal, indicating nearly flawless classification. Diseases such as Apple Scab, Black Rot, Cherry Powdery Mildew, and all Grape and Soybean categories achieve 100% accuracy. Even typically challenging pairs, like Potato Early vs. Late Blight and Strawberry Leaf Scorch vs. Healthy, show minimal or no confusion. Notably, Tomato classes—often harder to distinguish—are also classified with high precision. Overall, ViT delivers exceptional accuracy across all 38 disease and healthy leaf classes, making virtually no errors.

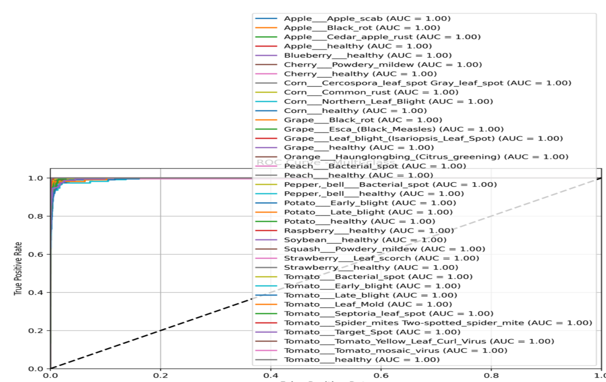


Fig. 31. ViT ROC Curve

In the above figure, nearly all class curves cluster closely in the top-left corner, indicating the model consistently makes accurate predictions with minimal errors. Most classes have an AUC of 1.00, reflecting near-perfect classification performance. This demonstrates that ViT is highly reliable in detecting and distinguishing a wide range of plant diseases and healthy conditions.

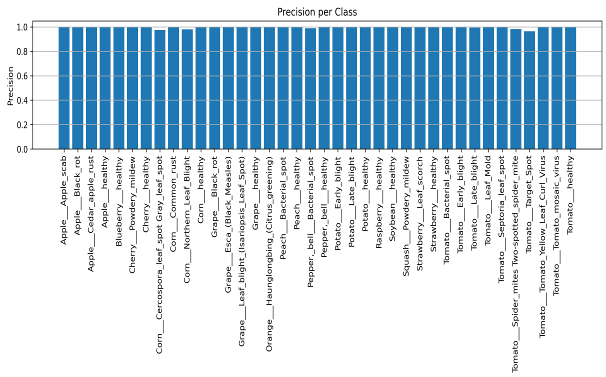


Fig. 32. Precision of ViT

In Figure 32, nearly all bars reach or approach 1.0, indicating that the model only predicts a class when it's highly confident. From Apple diseases and Corn infections to Tomato viruses and healthy leaves, ViT makes very few false positives. This consistently high precision highlights the model's strong reliability in identifying plant conditions.

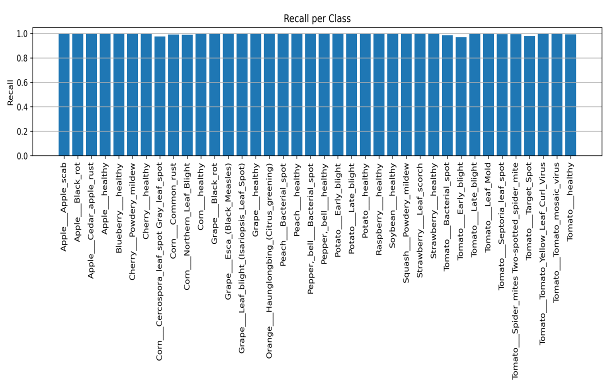


Fig. 33. Recall of ViT

This chart 33 illustrates the ViT model's recall for each class—that is, how effectively it identifies actual cases. Nearly all bars reach or are close to 1.0, indicating the model accurately detects most instances of each plant disease or healthy condition. From Apple and Corn diseases to more challenging Tomato infections, ViT rarely overlooks true cases.

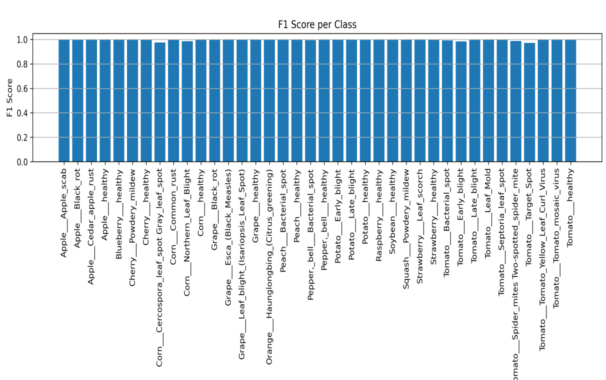


Fig. 34. F1-Score of ViT

The chart above presents the F1-score of the ViT model, where nearly all bars reach or closely approach 1.0, indicating the model is both accurate and consistent in classifying each category. From detecting diseases like Leaf Mold and Bacterial Spot to identifying healthy leaves, ViT delivers near-perfect performance across all 38 classes.

F. Swin

The Swin model achieved 99.66% training accuracy and 99.43% test accuracy in 37 epochs, as shown in Figure 35. Training and validation losses steadily decreased from 0.16 to below 0.06, indicating effective error reduction. Training accuracy improved from 95.5% to over 99.6%, demonstrating strong learning and classification ability. Validation accuracy followed a steady rise, surpassing 96%, showing good generalization. These trends reflect efficient learning with minimal overfitting.

Fig. 35. Accuracy and Loss of Swin

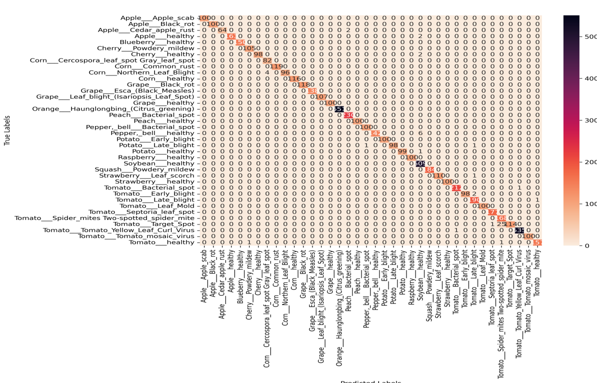


Fig. 36. Swin Confusion Matrix

This model's confusion matrix above shows strong performance, with most predictions falling along the diagonal, indicating accurate classification across nearly all plant disease and healthy classes. Only a few minor errors are observed, such as a few Apple class miss labels and Pepper Bell Healthy labeled as Soyabean Healthy. Overall, Swin achieves high accuracy across all 38 classes with minimal misclassification.

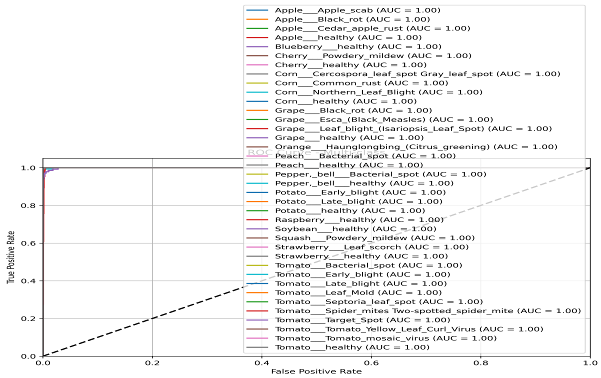


Fig. 37. Swin ROC Curve

In this Figure 37, most curves cluster tightly near the top-left corner, indicating high true positive rates and very few false positives. Each class reaches an AUC of 1.00, signifying perfect classification without errors. This confirms that the Swin Transformer delivers flawless performance across all plant disease and healthy leaf categories in the dataset.

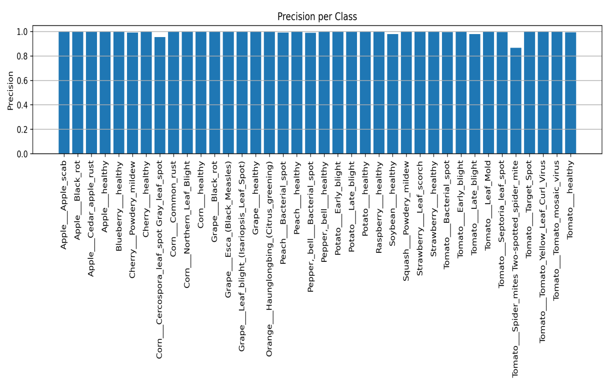


Fig. 38. Precision of Swin

The chart above shows that most bars reach or are very close to 1.0, showing that the model rarely makes incorrect predictions. It excels at identifying diseases like Apple Scab, Corn Rust, and Grape infections. Only a few classes—such as Corn Gray Leaf Spot and Tomato Two-Spotted Spider Mite—exhibit minor drops, indicating occasional confusion with visually similar conditions.

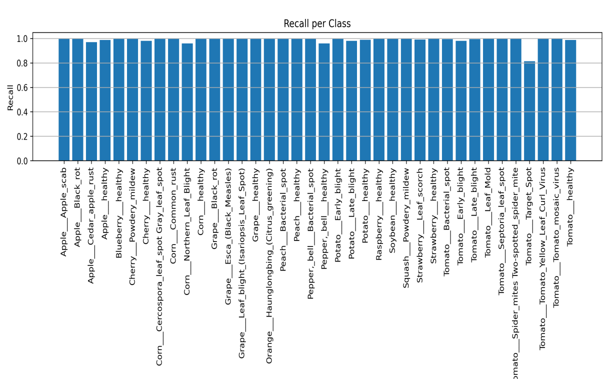


Fig. 39. Recall of Swin

The chart 39 shows that almost all bars are close to 1.0, indicating the model rarely fails to detect actual cases of disease or healthy leaves. It performs exceptionally well across most classes, particularly for Apple, Grape, Potato, and other diseases. A minor case, such as Tomato Target Spot, shows slightly lower recall, meaning the model occasionally misses some true examples. Overall, the Swin model demonstrates strong effectiveness in accurately identifying nearly all plant conditions.

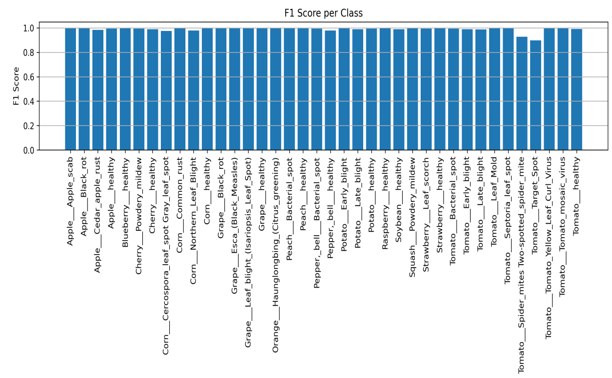


Fig. 40. F1-Score of Swin

Here, almost all bars are near 1.0, showing that the model is both accurate and consistent in predicting the majority of diseases and healthy leaves. Slightly lower scores for Tomato Two-Spotted Spider Mite and Tomato Target Spot, indicating occasional difficulty with these specific cases. Overall, it shows strong and well-balanced results across nearly all plant conditions.

Accuracy values:

Models	Train Accuracy	Test Accuracy	Precision	Recall	F1-score
Basic CNN	97.53%	95.03%	94.5%	93.8%	94.1%
Inception v3	92.12%	91%	97.2%	96.8%	97.0%
MobileNet v2	94.86%	94%	92.8%	93.0%	92.9%
ViT	98.38%	95.45%	95.7%	95.2%	95.4%
Swin	99.66%	99.43%	98.6%	98.8%	98.7%

TABLE II
ACCURACY VALUES OF DIFFERENT MODELS

The performance comparison from the above table demonstrates that transformer-based models, particularly Swin and ViT, outperformed traditional CNN architectures across all evaluation metrics. Swin achieved the highest overall performance, with a test accuracy of 99.43% and an F1-score of 98.7%, indicating exceptional generalization and a strong balance between precision and recall. ViT also performed well, achieving a test accuracy of 95.45% and an F1-score of 95.4%. Among the CNN-based models, Basic CNN showed competitive results with a test accuracy of 95.03% and an F1-score of 94.1%, while MobileNet v2 followed closely behind. Inception v3, despite its deeper architecture, recorded the lowest test accuracy at 91%, though it maintained a high recall of 96.8% and an F1-score of 97%, suggesting it was effective in detecting most true positives but less consistent in overall prediction accuracy. Overall, transformer-based models demonstrated superior performance in plant disease classification.

Accuracy charts:

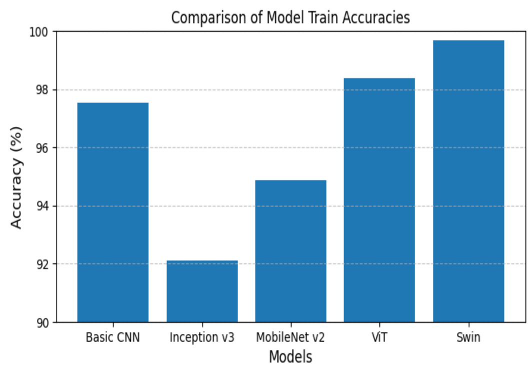


Fig. 41. Model Train Accuracies

This bar chart displays the training accuracy of five models. Swin leads with 99.66% accuracy, followed by ViT at 98.38%. Basic CNN also performs strongly at 97.53%, while MobileNet v2 and Inception v3 trail behind with 94.86% and 92.12% respectively. Overall, Swin and ViT demonstrate the most effective learning during training, whereas Inception v3 shows the slowest learning progress among the group.

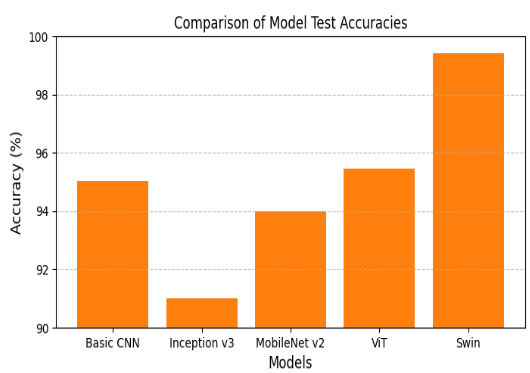


Fig. 42. Model Test Accuracies

The figure above shows the test accuracy of five different models. Swin performs the best with the highest accuracy of 99.43%, followed by ViT at 95.45%. Basic CNN also performs well at 95.03%, while MobileNet v2 scores slightly lower at 94%. Inception v3 has the lowest test accuracy, at 91%. Overall, Swin and ViT show the strongest generalization to new data, making them the most reliable models in this comparison.

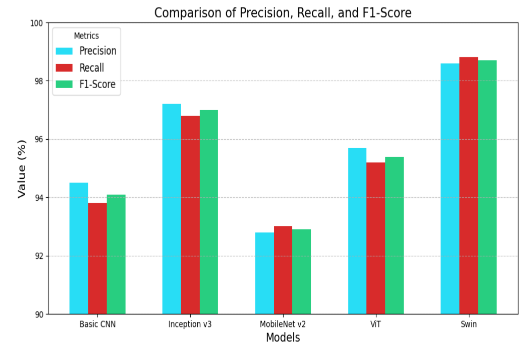


Fig. 43. Precision, Recall and F1-score for different models

This bar chart 43 compares the precision, recall, and F1-score of five models used for plant disease classification. The Swin Transformer outperforms the rest, achieving nearly 99% across all three metrics. Inception v3 follows with scores around 97%, while ViT also performs strongly with values around 95%. Basic CNN shows moderate results between 93.5% to 94.5%, and MobileNet v2 records the lowest scores, just above 93%. These results highlight the Swin Transformer as the most accurate and well-balanced model among the five.

IX. Interpretation Using Explainable AI

We have utilized XAI techniques, specifically the Lime framework, to enhance the interpretability of our machine learning models. This framework enabled us to explain our model's predictions, providing valuable insights into how the model arrived at its decisions (green regions = regions that most drove the model's decision). Moreover, 100% confidence means the model predicted the disease with high probability.

CNN:

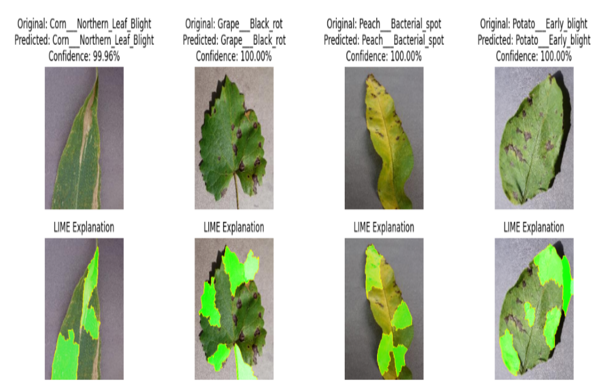


Fig. 44. CNN model lime explanation

Corn Northern Leaf Blight: The model focused and highlighted the elongated lesion region, indicating it correctly detected the characteristic "Northern Leaf Blight lesion."

Grape Black Rot: Even though the confidence score is 100% and it predicted the disease correctly, the green highlights didn't fully concentrate on those necrotic spots. Also, the surrounding areas are highlighted, indicating that the model is also leaning on the leaf's shape and background contrast for its prediction.

Peach Bacterial Spot: The explanation highlighted those brown, water-soaked spots, indicating that the model correctly detected bacterial spot lesions and patterns.

Potato Early Blight: The highlighted super-pixels almost align with the necrotic patches, which are the symptoms of early blight, demonstrating the model's reliance on its prediction.

Inception:

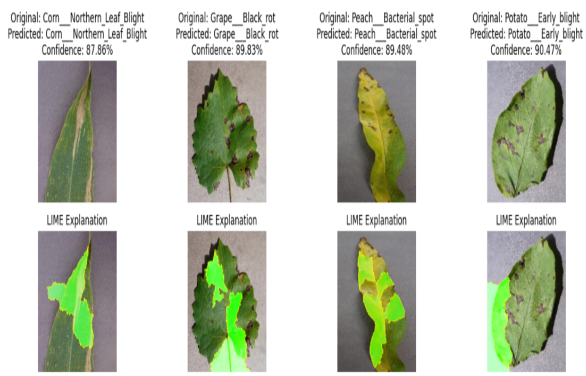


Fig. 45. Inception model lime explanation

Corn Northern Leaf Blight: Rather than exclusively focusing on the elongated lesion region, the model also highlighted a small patch of the background region, indicating the characteristic Northern Leaf Blight lesion for its prediction.

Grape Black Rot: The model didn't highlight those dark necrotic spots correctly. The largest green highlight covers surrounding areas, indicating the model is also leaning on the leaf's shape and background for its prediction rather than homing in on the actual rot lesions themselves.

Peach Bacterial Spot: The irregular, water-soaked brown lesions and their surroundings are almost precisely outlined by the green super-pixels, indicating that the model correctly detected bacterial spot lesions.

Potato Early Blight: The model's prediction heavily favored the background region instead of focusing on the actual rot lesions.

MobileNet:

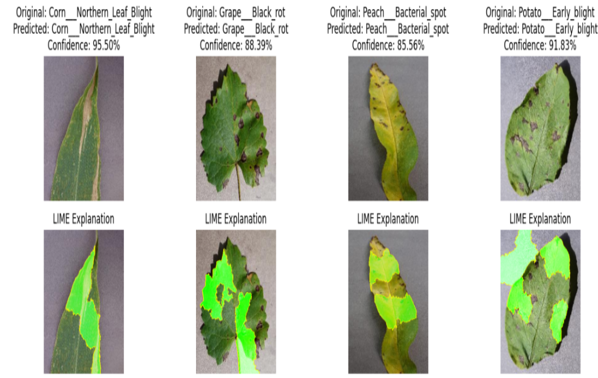


Fig. 46. MobileNet model lime explanation

Corn Northern Leaf Blight: The model focused and highlighted the elongated lesion region, indicating it correctly detected the characteristic Northern Leaf Blight lesion.

Grape Black Rot: The green highlights are distributed over both the dark necrotic spots and large areas of unblemished leaf, indicating that the model is also relying on the leaf's shape, texture, and background contrast for its prediction.

Peach Bacterial Spot: The explanation almost highlighted those brown, water-soaked spots, indicating that the model correctly detected bacterial spot lesions and patterns.

Potato Early Blight: LIME reveals a mix of lesion and non-lesion cues. The dominant green patch covers a large region of healthy green tissue and background for its prediction.

ViT:

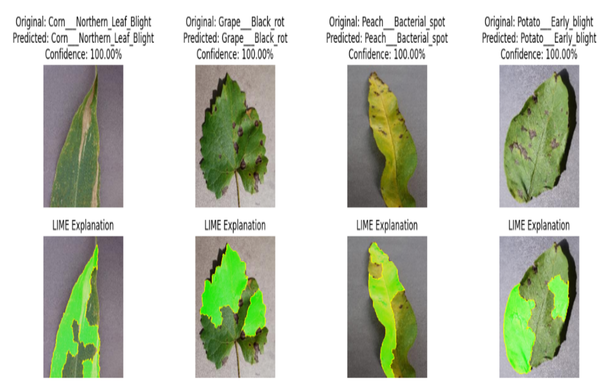


Fig. 47. ViT model lime explanation

Corn Northern Leaf Blight: Lime correctly highlighted the elongated lesion region and also the healthy regions, indicating it relies on both the rot lesions and the shape, color, and texture of the leaf to correctly detect the characteristics of the Northern Leaf Blight lesion.

Grape Black Rot: Rather than relying solely on the lesion itself, the model appears to largely focus on the shape, color, and texture of the leaf for its prediction.

Peach Bacterial Spot: Lime highlighted the brown, water-soaked spots, indicating that the model correctly detected bacterial spot lesions and patterns. It also relies on healthy regions for its prediction.

Potato Early Blight: The highlighted super-pixels almost fully cover the necrotic regions, indicating the symptoms of early blight, demonstrating the model's reliance on its prediction.

Swin:

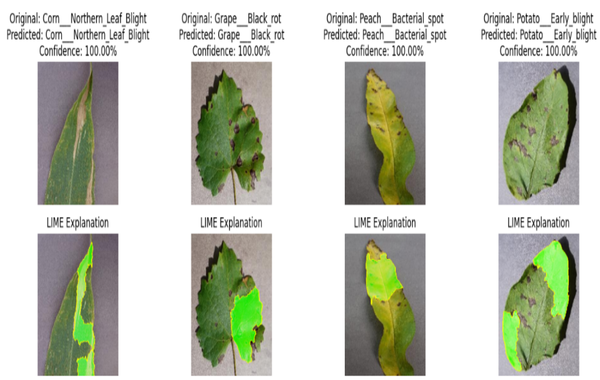


Fig. 48. Swin model lime explanation

Corn Northern Leaf Blight: Lime's precise focus on the elongated lesion region indicates its correct identification of the characteristic Northern Leaf Blight lesion.

Grape Black Rot: The green highlight is almost fully concentrated on those dark necrotic spots. This indicates the model is mostly leaning on the leaf's black rot spots for its prediction.

Peach Bacterial Spot: The explanation almost highlighted the brown, water-soaked spots, indicating that the model correctly detected bacterial spot lesions.

Potato Early Blight: The highlighted super-pixels are divided into two regions to highlight the healthy regions. Both upper and lower regions cover a few rot spots of the leaf, and these lead to its prediction.

Based on the LIME explanations across four test images, Swin is the only model that consistently highlights the diseased pixels in all four test leaves. The highlighted regions are more focused and targeted, and the areas correspond more closely to actual disease symptoms. Swin's explanations highlight the infected areas more accurately, whereas other models tend to either over-highlight masks into only healthy or background regions. It explains a clear connection between what the model sees and what humans would identify as disease features. As a result, Swin offers the most reliable interpretability for plant disease detection.

X. App Implementation

For our project, we developed a web application using Streamlit that offers a simple and interactive interface for predicting plant diseases from leaf images. The interface uses custom HTML and CSS within Streamlit to center elements and style the output for a clean, user-friendly experience. Users can upload leaf images using either a drag-and-drop feature or a file selection dialog. The uploaded image is displayed centrally, and a "Predict" button initiates classification using our pretrained Swin Transformer model. Based on the prediction, the app shows the disease name, a brief description, and relevant preventive measures.

Fig. 49. Image Upload Interface

The Figure 49 displays the initial interface of the Plant Disease Detector web application. Users are prompted to upload an image of a plant leaf by either dragging and dropping a file or browsing their device.

Fig. 50. File Selection Dialog

The screenshot 50 shows the file selection dialog window after the user clicks "Browse files." It demonstrates how users can select an image of a plant leaf from their local system for prediction.

Fig. 51. Image Preview

Here, the selected image is successfully uploaded to the application. The image preview confirms the uploaded file, and a "Predict" button appears, enabling the user to initiate disease classification.

Fig. 52. Image Prediction

After the "Predict" button is clicked, and disease classification is completed, it displays the prediction. The app displays the predicted disease label, along with a brief description and suggested preventive measures as shown in Figure 52.

XI. Conclusion

Whilst machine learning methods to identify plant diseases have shown positive results, they often need complex designs with significant computer resources. [15] This study shows that strategic model selection and optimization can effectively address these limitations. Among the models evaluated, the Swin Transformer consistently outperformed others, as evidenced by comprehensive LIME explanations across four test images. It reliably highlighted diseased regions with high precision, indicating superior localization capability. These findings hold significant practical implications for real-world agricultural deployment. By selecting the Swin Transformer as the optimal model architecture, we can enhance detection accuracy while maintaining computational efficiency. The model's ability to focus on relevant disease features represents effective feature selection in action - the system naturally identifies and prioritizes the most diagnostically important visual cues. This targeted approach not only improves detection performance but also reduces the computational burden typically associated with less focused models. These findings enable practical deployment through scalable AI-driven mobile diagnostic apps, precision agriculture systems, and automated monitoring systems, where Swin's computational efficiency and interpretable results can enhance detection accuracy while reducing implementation costs. Its balance of accuracy, interpretability, and efficiency supports the development of trustworthy diagnostic tools that are widely applicable to real-world agricultural applications. Ultimately, this contributes to improved agricultural outcomes and food security through reliable, interpretable disease detection systems.

Acknowledgment

Firstly, all praise to the Great Allah for whom our project has been completed without any major interruption.

Secondly, to our Supervisor, Md. Tawhid Anwar and Co-Supervisor Md. Sabbir Ahmed sir, for his kind support and advice in our work. He helped us whenever we needed help.

And finally, to our parents without their throughout support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

References

[1] H. Wang, G. Li, Z. Ma, and X. Li, "Application of neural networks to image recognition of plant diseases," in *2012 International conference on systems and informatics (ICSAI2012)*, IEEE, 2012, pp. 2159–2164.

[2] A. Akhtar, A. Khanum, S. A. Khan, and A. Shaukat, "Automated plant disease analysis (apda): Performance comparison of machine learning techniques," in *2013 11th International Conference on Frontiers of Information Technology*, IEEE, 2013, pp. 60–65.

[3] J. Francis, B. Anoop, *et al.*, "Identification of leaf diseases in pepper plants using soft computing techniques," in *2016 conference on emerging devices and smart systems (ICEDSS)*, IEEE, 2016, pp. 168–173.

[4] 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, 2016.

[5] P. Akulwar, "A recommended system for crop disease detection and yield prediction using machine learning approach," *Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries*, pp. 141–163, 2020.

[6] M. E. Chowdhury, A. Khandakar, S. Ahmed, *et al.*, "Design, construction and testing of iot based automated indoor vertical hydroponics farming test-bed in qatar," *Sensors*, vol. 20, no. 19, p. 5637, 2020.

[7] N. Gobalakrishnan, K. Pradeep, C. Raman, L. J. Ali, and M. Gopinath, “A systematic review on image processing and machine learning techniques for detecting plant diseases,” in *2020 international conference on communication and signal processing (ICCSP)*, IEEE, 2020, pp. 0465–0468.

[8] R. I. Hasan, S. M. Yusuf, and L. Alzubaidi, “Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion,” *Plants*, vol. 9, no. 10, p. 1302, 2020.

[9] H. E. David, K. Ramalakshmi, H. Gunasekaran, and R. Venkatesan, “Literature review of disease detection in tomato leaf using deep learning techniques,” in *2021 7th International conference on advanced computing and communication systems (ICACCS)*, IEEE, vol. 1, 2021, pp. 274–278.

[10] S. John and A. L. Rose, “Machine learning techniques in plant disease detection and classification-a state of the art.,” *INMATEH-Agricultural Engineering*, vol. 65, no. 3, 2021.

[11] P. Kulkarni, A. Karwande, T. Kolhe, S. Kamble, A. Joshi, and M. Wyawahare, “Plant disease detection using image processing and machine learning,” *arXiv preprint arXiv:2106.10698*, 2021.

[12] C. Nguyen, V. Sagan, M. Maimaitiyiming, M. Maimaitijiang, S. Bhadra, and M. T. Kwasniewski, “Early detection of plant viral disease using hyperspectral imaging and deep learning,” *Sensors*, vol. 21, no. 3, p. 742, 2021.

[13] N. Rashmi, C. Shetty, *et al.*, “A machine learning technique for identification of plant diseases in leaves,” in *2021 6th international conference on inventive computation technologies (ICICT)*, IEEE, 2021, pp. 481–484.

[14] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, “Performance of deep learning vs machine learning in plant leaf disease detection,” *Microprocessors and Microsystems*, vol. 80, p. 103 615, 2021.

[15] N. Shelar, S. Shinde, S. Sawant, S. Dhumal, and K. Fakir, “Plant disease detection using cnn,” in *ITM Web of Conferences*, EDP Sciences, vol. 44, 2022, p. 03 049.

[16] M. J. U. Chowdhury, Z. I. Mou, R. Afrin, and S. Kibria, “Plant leaf disease detection and classification using deep learning: A review and a proposed system on bangladeshâ€™ s perspective,” *International Journal of Science and Business*, vol. 28, no. 1, pp. 193–204, 2023.

[17] S. Javidan, A. Banakar, K. Vakilian, and Y. Ampatzidis, *Diagnosis of grape leaf diseases using automatic k-means clustering and machine learning. smart agric technol 3: 100081*, 2023.