

# A Study on the Performance of Hybrid Approach for Image Classification using CNNs and SVM for Plant Disease Detection

<sup>1</sup>Nahid Hasan Ashik

Department of Computer Science and Engineering  
East West University, Dhaka, Bangladesh  
nahidhashik@gmail.com

<sup>2</sup>Rakibul Islam

Department of Electronic & Communication Engineering  
East West University, Dhaka, Bangladesh  
rakibulislamshohan95@gmail.com

<sup>3</sup>Md. Arif Hossain

Department of Computer Science and Engineering  
East West University, Dhaka, Bangladesh  
iammdarifhossain@gmail.com

<sup>4</sup>Toukir Ahammed

Department of Computer Science and Engineering  
East West University, Dhaka, Bangladesh  
toukir.ahammed@ewubd.edu

<sup>5</sup>Md Shakirul Islam

Department of Computer Science and Engineering  
East West University, Dhaka, Bangladesh  
Shhakirul@gmail.com

**Abstract:-** In a country where agriculture is the primary industry, plant diseases can have a wide range of adverse effects on the economy and the management of food resources. The increasing occurrence of plant diseases significantly threatens global food security and plant productivity. Classification is constantly constrained by issues like overfitting and low accuracy as potential diseases manifest on plant leaves. The agricultural sector needs accurate and error-free analysis to distinguish healthy products from defective ones. An effective model of autonomous feature extraction that has been demonstrated to be reasonably effective for detection and classification tasks is deep convolutional neural networks. Deep convolutional neural networks, on the other hand, frequently require a substantial amount of training data, cannot be translated, and require a number of parameters to be specified and adjusted. In this study, we suggested a hybrid deep learning approach for quickly identifying and categorizing different plant leaf diseases. This hybrid system combines support vector machines (SVM), convolutional neural networks (CNN), and both.

To summarize, we extract features through model engineering (ME). To improve feature discrimination and processing speed, support vector machine (SVM) models are used. Using the datasets, 25 leaf image sets of healthy and diseased leaves of tomato, potato, grape, apple, and corn were analyzed. There were roughly 31397 images produced as a result of the classification process. SVM is a key component of hybrid feature selection; this algorithm's goal is to produce the desired output with the fewest features possible. By utilizing the

**mentioned evaluation criteria, a comparative analysis of both techniques is presented.**

**Keywords:-** Plant Disease Detection, Image Processing, CNN, VGG19, Xception, DenseNet201, ResNet15V2, SVM.

## I. INTRODUCTION

Plant diseases are a big global concern that poses substantial threats to agricultural production, food security, and economic stability [10]. Some plant diseases can even be transmitted to humans. As a result of the diverse array of plants that are cultivated all over the world, including food staples such as rice, wheat, maize, potatoes, and soybeans, as well as cash plants such as coffee, cotton, bananas, citrus fruits, and palm oil, efficient management of plant diseases, is essential for maintaining agricultural practices and ensuring that enough food is produced on a global scale.

In recent years, the confluence of Artificial Intelligence (AI), Machine Learning (ML), and Computer Vision has transformed the field of plant disease detection and diagnosis [11]. AI stands for artificial intelligence, and ML and CV stand for machine learning. New doors for conducting precise information analysis based on visual representations of plants have been opened as a result of the incorporation of cameras with high resolution into portable electronic devices and the application of advanced image processing techniques. Researchers and farmers alike are given the ability to recognize illness indicators more quickly, which paves the way for more efficient disease management through the implementation of timely interventions.



Fig 1 Affected Leaves

Figure 1, shows the raw image of affected leaves from our dataset. Agriculture is essential to the functioning of economies all over the world because it helps maintain livelihoods, makes certain that there is sufficient food supply, and meets the nutritional requirements of populations. On the other hand, plant diseases can have a catastrophic effect on agricultural output, leading to significant economic losses as well as the possibility of food shortages [10]. The timely and correct diagnosis of plant diseases, as well as their early detection, are vital for the implementation of focused control measures, the protection of global food production, and the promotion of sustainable agricultural practices.

CNNs, which stand for convolutional neural networks, are a powerful class of deep learning models that have recently emerged as effective tools for image classification tasks such as the identification of plant diseases [4]. CNNs do exceptionally well when it comes to learning complex patterns and features from images, which makes them ideally suited for the task of assessing plant leaf images as accurate indications of the presence of illness. Diverse CNN architectures, such as VGGNet, ResNet, InceptionNet, MobileNet, DenseNet, EfficientNet, and NASNet, have been devised and shown to be effective for a variety of image classification problems [5].

The goal of our research is to provide a hybrid model for identifying plant illnesses. Convolutional Neural Networks (CNNs) are a state-of-the-art technology that we are using, together with a massive dataset of images of genuine plant leaves that were taken in various areas across the world [12]. Among the plants included in the dataset are apple, corn, grape, potato and tomato. In order to guarantee that we gain a global perspective, we collected our dataset from Kaggle of affected and healthy images of plant leaves. We added the SVM classifier at the last layer of CNN feature extraction. By doing this, we aim to be able to develop a hybrid model for identifying plant illnesses that will be helpful to farmers and will help to protect agricultural products.

The CNN model will be trained using an appropriate architecture and cutting-edge methodologies on this massive dataset. Transfer learning will be utilized by leveraging pre-trained models trained on large-scale image datasets such as ImageNet to improve the model's learning capabilities and accelerate convergence [15]. To supplement the dataset, data augmentation techniques such as rotation, scaling, flipping, random cropping, and noise injection will be used, boosting the model's capacity to generalize across different illness presentations and environmental variables [16].

To evaluate the performance of the developed plant disease detection system, comprehensive metrics including accuracy, precision, recall, F1-score, an area under the receiver operating characteristic curve (AUC-ROC), and Matthews correlation coefficient (MC) will be used [17]. To ensure the effectiveness and adaptability of the system on a global scale, comparative analyses will be conducted against state-of-the-art approaches and benchmark datasets from various global regions.

This research project's outcomes have far-reaching implications for automated plant disease detection, providing solutions for early diagnosis and intervention on a global scale. Significant potential exists for the developed system to mitigate crop losses, optimize resource allocation, and contribute to global food security efforts. The research findings will have wide applicability in diverse agricultural contexts around the globe, contributing to the advancement of computer vision and machine learning in agriculture.

#### A. Plant Disease Overview

Plants are essential to the world's food supply because they provide the basis for countless people's sustenance. But a number of diseases that can wreak havoc on plant growth and productivity are a constant threat to the health of plants. This overview focuses on four important plants—apple, corn, grape, potato, and tomato—as well as the illnesses that frequently affect them.

Farmers use a variety of preventive techniques, such as crop rotation, resistant cultivars, and timely application of fungicides or bactericides, to safeguard these priceless plants from diseases [6]. To maintain the health and productivity of these crucial plants for future generations, ongoing research and development aims to improve disease resistance and control strategies. Given that farmers are unaware of hereditary diseases, these methods are time-consuming and extremely expensive for large farms.



*B. Types of Plant Disease in Our Dataset*

We have a dataset of 31397 plant images which contains a total of five plants and 25 classes. Figure 2 shows the images of our 25 classes.

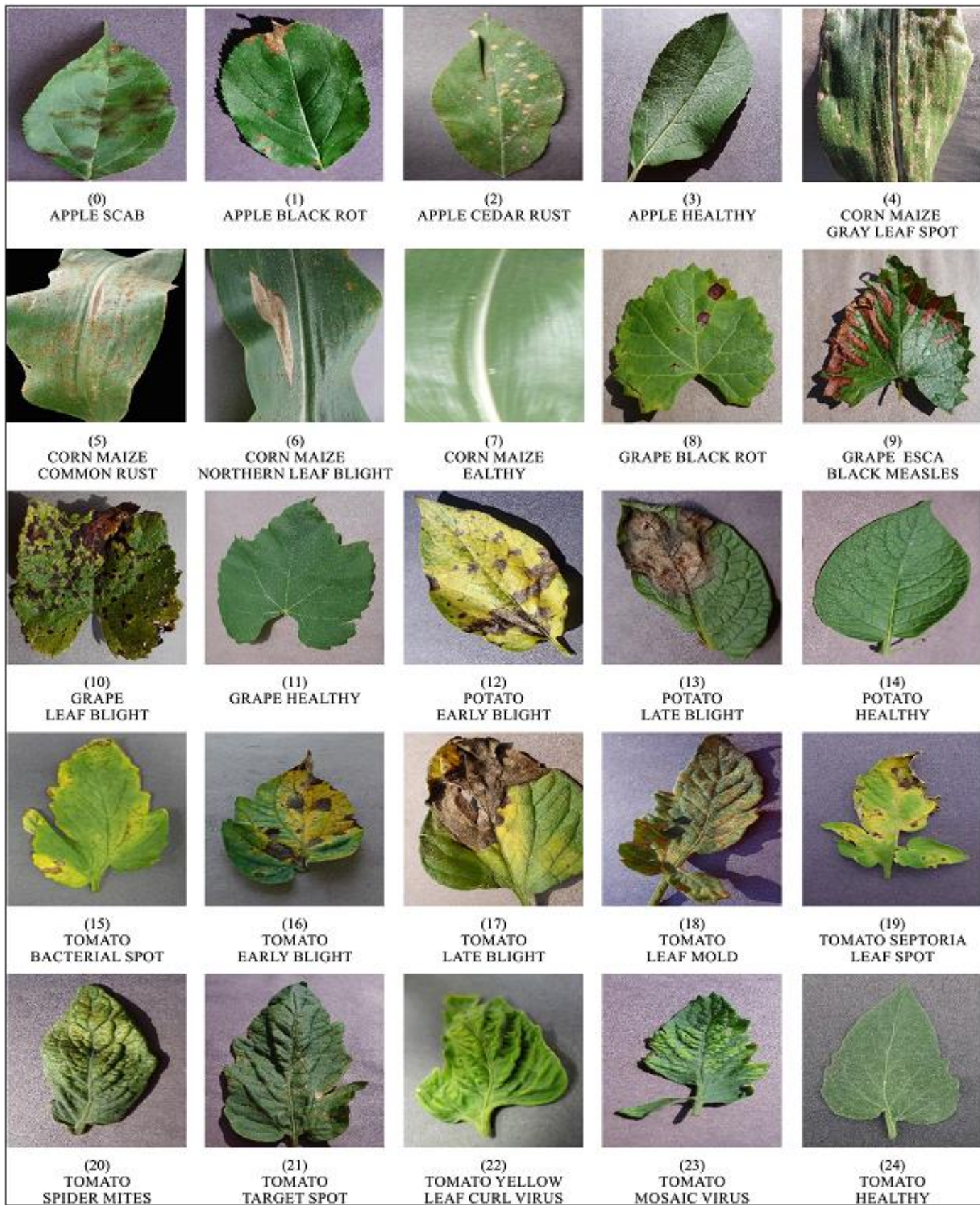


Fig 2 25 Classes of Our Dataset

### ➤ *Apple Disease*

The health and yield of apple trees can be impacted by a number of diseases, just like any other plant. Apple scab, apple black rot, and cedar apple rust are three typical apple diseases that growers frequently run into.

A common and important disease in terms of economic impact is apple scab, which is brought on by the fungus *Venturia inaequalis*. The apple tree's leaves, fruit, and twigs all develop dark, scaly lesions as a result. As a result of these lesions, the tree's capacity for photosynthesizing is diminished, which lowers its vigor and weakens the quality of its fruit. If not properly controlled, apple scab spreads quickly throughout an orchard and thrives in moist environments.[21]

Fruit is the main victim of apple black rot, which is brought on by the fungus *Botryosphaeria obtusa*. It results in dark, sunken lesions that grow over time and eventually cause the apples to shrivel and become mummified. Brown, atypical lesions can appear on the tree's leaves as a result of the disease. Warm, humid weather promotes the development of apple black rot, which must be controlled with appropriate fungicide applications and orchard sanitation.[22]

A complicated life cycle between apple trees and juniper or cedar trees is required for the unique disease known as cedar apple rust. The fungus *Gymnosporangium juniperi-virginianae* is the culprit. As the infection spreads, the disease begins as small, yellow-orange spots on apple leaves that grow into larger, bright-orange lesions with spore structures. The fruit may also be harmed by cedar apple rust, which can result in raised, corky areas and distorted growth. In order to reduce the source of infection, managing cedar apple rust frequently involves cutting down nearby juniper or cedar trees.[23]

### ➤ *Corn Disease*

A common crop in every country, corn is susceptible to a number of diseases that can reduce its growth and yield. *Cercospora* leaf spot, gray leaf spot, and common rust are three diseases that are particularly prevalent in corn plants.

Corn frequently develops the foliar disease *Cercospora* leaf spot, which is brought on by the fungus *Cercospora zae-maydis*. On the leaves, it manifests as small, round to elongated lesions with gray centers and dark brown borders. The spots may combine as the disease worsens, growing into larger, atypical lesions. As a result of the plant's reduced ability to photosynthesize and subsequent premature defoliation brought on by severe infections, grain development may be impacted. *Cercospora* leaf spot spreads more readily in warm, humid weather.[24]

Another important condition affecting corn is gray leaf spot, which is brought on by the fungus *Cercospora zae-maydis*. Long, rectangular lesions on the leaves with distinct gray centers and dark brown borders are its primary symptom. These lesions frequently appear in the spaces between the leaf veins and, in warm, humid weather, they

can spread quickly. Gray leaf spots can make a plant vulnerable, which will reduce photosynthesis and compromise grain fill. Some management techniques used to lessen the effects of this disease include crop rotation, resistant hybrids, and balanced fertilization [25].

Corn is frequently affected by common rust, which is brought on by the fungus *Puccinia sorghi*. It primarily affects the leaves but can also affect other plant parts and manifests as small, circular to elongated, rusty-orange pustules. A large amount of orange spores is released as the pustules rupture. While common rust rarely causes severe damage, heavy infections can reduce photosynthetic capacity and lead to premature defoliation. Though it can still happen in warm, humid weather, the disease is more common in cooler climates. The common rust is typically treated with fungicides and resistant corn hybrids [26].

### ➤ *Grape Disease*

Grapes, which are prized for their use in winemaking and as table fruits, are susceptible to a number of diseases that can negatively affect the health of the vine and the quality of the grapes. Black rot, esca, and leaf blight are three common grape diseases.

A common and harmful disease of grapevines, black rot is brought on by the fungus *Guignardia bidwellii*[27]. It mainly impacts fruit, resulting in brown, cirrhotic lesions with a distinctive black border. The grapes shrivel, mummify, and prematurely drop as the infection worsens. Brown lesions and defoliation can be seen on leaves and shoots when black rot is present. Black rot is more likely to develop and spread during warm, humid weather during the growing season. For effective management of this disease, proper cultural practices, including pruning and canopy management, are necessary [28].

A number of different fungal pathogens, including *Phaeoacremonium chlamydospora*, *Phaeoacremonium* spp., and others, interact to cause the complex grapevine disease known as esca [29]. It shows up as streaks of necrotic decay and discoloration in the wood of the vine, most frequently in the trunk and cordons. Infected vines display a slow decline, diminished vigor, and shoot dieback. Esca is frequently linked to older vineyards and is spread through pruning wounds [30]. Although there is currently no known treatment for esca, disease management requires taking preventive measures like using disease-resistant rootstocks and maintaining strict hygiene [31].

The fungus *Guignardia bidwellii*, a pathogen that also causes black rot, is the source of leaf blight, which affects grape leaves [32]. The potential growth of the lesions could cause defoliation and a reduction in photosynthetic capacity. Leaf blight can be particularly harmful in wet and warm weather, favoring the disease's growth and spread. Cultural practices such as proper pruning, leaf removal, and adequate vineyard airflow can help.



### ➤ *Potato Disease*

One of the most widely grown and consumed crops in the world, the potato is prone to a number of illnesses that may have a significant negative impact on the crop's quality and yield. Early blight and late blight are two prevalent diseases that affect potatoes [34].

Throughout the growing season, potato plants are frequently afflicted by the disease known as early blight, which is brought on by the fungus *Alternaria solani*. The lower leaves usually develop small, concentric ring-shaped lesions that are dark brown in color. The lesions grow larger and take on a distinctive "bullseye" pattern as the disease worsens. When it is warm and humid outside, early blight can spread quickly, causing defoliation and smaller tubers. In order to effectively manage early blight, it is important to rotate crops, apply fungicides on schedule, and follow good cultural practices [35].

Potato crops can suffer significant losses due to the devastating disease known as late blight [35], which is brought on by the water mold *Phytophthora infestans*. It is a serious concern in areas with high humidity because it thrives in cool, damp environments. Early signs of late blight include water-soaked lesions on the leaves, which quickly turn brown and necrotic. When the humidity is high, a white, fuzzy mold may start to grow on the undersides of the diseased leaves. Also susceptible to late blight, potato tubers can rot and become unfit for consumption or storage. To effectively manage late blight, it is essential to strictly follow integrated pest management procedures, which include using resistant potato varieties, routinely scouting for symptoms, and timely fungicide applications [36].

### ➤ *Tomato Disease*

As a well-liked and adaptable crop, tomatoes are vulnerable to a number of diseases that may have a negative effect on their development, yield, and fruit quality. Bacterial spots, early blight, late blight, leaf mold, *Septoria* leaf spots, spider mites (particularly two-spotted spider mites), tomato mosaic viruses, and tomato yellow leaf curl viruses are among the prevalent tomato diseases [38].

Tomato plants are affected by bacterial spots caused by *Xanthomonas* spp. on their leaves, stems, and fruits. It first appears as small, moist lesions that have the potential to enlarge and turn black or dark brown [39]. Bacterial spots can result in fruit blemishes, defoliation, and decreased yield. Examples of management strategies include crop rotation, the use of disease-resistant varieties, and copper-based sprays.

A typical tomato disease is called early blight, which is brought on by the fungus *Alternaria solani*. A large number of brown lesions eventually develop after it first appears as dark concentric rings on the lower leaves. Early blight has the potential to spread quickly, especially in warm, muggy weather. Effective controls for this disease include good sanitation, prompt removal of infected plant material, and fungicide applications [40].

The destructive disease known as late blight [41], which is brought on by *Phytophthora infestans*, affects both fruits and foliage. It causes leaves to develop dark, wet lesions that are frequently covered in a fuzzy, white mold on the undersides. Fruits may quickly defoliate and rot as a result of late blight. Strict sanitation procedures, the removal of infected plants, and the application of fungicides are all part of management.

Tomato foliage is most commonly impacted by the fungus *Cladosporium fulvum*'s leaf mold. In addition to fuzzy, olive-brown mold on the underside, it manifests as yellowish patches on the upper leaf surface. The photosynthetic capacity of the leaves can be decreased, weakening the plants. Leaf mold control requires appropriate spacing, good ventilation, and regular fungicide application [42].

Tomato plant foliage is impacted by the fungus *Septoria Lycopersicon*, which causes *Septoria* leaf spot. It results in tiny, light-colored spots with dark centers on the leaves, which promotes the growth of necrotic lesions. Defoliation and lowered fruit quality can both be caused by *Septoria* leaf spot. The management of this disease requires the use of fungicides and good sanitation practices [43].

Tomato plants can become infected by spider mites, particularly the two-spotted spider mite (*Tetranychus Urticae*). They consume plant sap, which causes the leaves to stipple and eventually turn yellow and necrotic. Invasions that are particularly bad can stunt plant growth and lower yields. To control spider mite populations, miticides, predatory mites, and routine observation are used [44].

The viral disease tomato mosaic virus, which affects tomatoes, is typically transmitted mechanically and through plant material from infected plants. Mosaic leaf patterns, stunted growth, and poor fruit quality are some signs of the disease. To effectively manage this illness, it is crucial to use seeds free of viruses, practice good hygiene, and manage insect vectors.

An additional viral disease that severely harms tomato plants is the tomato yellow leaf curl virus. It causes leaf curling, yellowing, and a decreased fruit yield. Whiteflies are the main vectors of this disease [44]. The management of this virus requires actions like the use of insecticides, the use of resistant varieties, and the control of whiteflies.

### *C. Image Processing, Deep learning and Efficiency*

Our plant disease detection proposed a hybrid approach which is a Deep Learning algorithm and classification based on image processing. For image processing, Deep Learning is the best choice for all. For multiclass classification SVM is one of the best classifiers. By considering all factors of all possible interfaces, Machine Learning (ML) minimizes the error between the predicted and actual outcomes [22].

Artificial Neural Networks (ANNs) that use machine learning (ML) and have features resembling those of the human brain include Deep Learning (DL). Deep Learning (DL) utilizes numerous connected nodes to predict the outcome, much like the human brain does [23]. You must provide a dataset of images to the machine that includes classes in order for deep learning to provide the end-to-end learning concept. With the help of the provided dataset, the machine will be trained to use Deep Learning (DL) models, which aim to identify and predict underlying patterns based on classes [24]. Because its neurons are connected, deep learning has a better capacity for self-learning. In DL algorithms, the neurons are dispersed over a number of layers. Together, connected layers and connected neurons predict an image class and process an image more effectively than other approaches. Deep Learning Algorithms are what we have used as our suggested system in this situation.

#### D. Objective

- **Plant Disease Detection:** The main goal of this project is to develop a system capable of correctly diagnosing a variety of plant diseases. The goal is to improve the accuracy and effectiveness of illness detection procedures via the use of cutting-edge technology.
- **Deep Learning Algorithm Comparison:** The study aims to undertake a comparative examination of several deep learning algorithms with a focus on selecting the best algorithm for data equilibrium using advanced data preparation methods. This comparative analysis attempts to shed light on the benefits and drawbacks of different algorithms in dealing with the particular difficulties of the research.
- **Performance Evaluation of the Proposed Model:** An essential component of this research is a thorough assessment of the performance displayed by the proposed model. This review includes a thorough examination of the model's correctness and effectiveness in comparison to earlier models created for similar goals. The goal is to evaluate the suggested model's originality and development within the existing area.
- **Integration of a Hybrid strategy with SVM:** The study aims to investigate the integration of a hybrid strategy, combining different techniques based on convolutional neural networks (CNNs) with support vector machine (SVM) algorithms. The research intends to enable a discriminating examination of these techniques' specific characteristics, consequently offering insightful information about their respective performances.

#### E. Existing System

The current system has been used with a variety of Deep Learning Models (DL), including ANN, CNN,[6] R-CNN, Faster R-CNN [8], Inception V3[6][8], DenseNet121[6], ResNet[20], and Logistic Regression. It has also been used with Machine Learning Algorithms, including Decision Tree Classifier, Logistic Regression, K-Nearest Neighbor, Naive Bayes Classifier, and so forth. Using balancing techniques, some people can make the data set balanced.

#### F. Proposed System

Our proposed system represents a paradigm shift by harnessing the synergies of both Deep Learning Algorithms and the precision of Support Vector Machine (SVM) to effectively detect plant diseases. To ensure impeccable accuracy, our process entails meticulous data refinement, encompassing the removal of extraneous data instances. This strategic curation encompasses the elimination of images characterized by a singular attribute, wherein they are exclusively healthy or afflicted by a solitary disease.

To further elevate the quality of our dataset, we applied image sharpening techniques, bolstering the clarity of crucial visual cues. Achieving a harmonized dataset proved pivotal; thus, the Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTE-ENN) was instrumental in attaining this equilibrium. Augmentation techniques such as Shear, Zoom, and horizontal flip were employed to introduce diversity and foster robust model generalization.

The ultimate litmus test for our system's prowess came through the rigorous evaluation of esteemed Deep Learning Algorithms. Notable contenders including CNN, VGG16, DenseNet201, Xception, and ResNet50 were subjected to comprehensive testing. This pivotal experiment was conducted on a dataset comprising a diverse array of 25 distinct plant diseases alongside healthy plant instances.

The amalgamation of diverse methodologies, deliberate data curation, and a holistic evaluation approach culminate in a pioneering system poised to redefine the landscape of plant disease detection.

#### G. Proposed Hybrid Approach

Our proposed method for detecting plant diseases involves combining Convolutional Neural Networks (CNNs) to extract image features and Support Vector Machines (SVMs) for accurate classification. By using the strengths of both approaches, we aim to optimize the performance of the disease detection system. This approach combines modern image analysis techniques with advanced classification methods, resulting in improved accuracy and efficacy in identifying plant diseases.

## II. RELATED WORKS

In this study, various powerful deep learning algorithms, such as AlexNet, AlexNetOWTBn, GoogLeNet, OverFeat, and VGG, were effectively used to differentiate between healthy and diseased plants. The algorithms were thoroughly trained on basic leaf images to quickly diagnose and recognize plant diseases with precision. To train the models, a public database containing 87,848 images, 25 different plant species, and 58 distinct classes of [plant, disease] pair combinations, including healthy plants, was used. The VGG convolutional neural network was the most successful model architecture among those that had been trained, with a success rate of 99.53% in the classification of 17,548 plant leaf images that the model had never seen before. [3].

In this study [1], a new dataset and an innovative two-stage architecture for plant disease detection are presented. The dataset includes 79,265 images of leaf diseases, making it the largest dataset of leaf images to date. The two-stage architecture has the ability to learn both local and global features of plant leaves, which enhances its ability to adapt to variations in lighting and background conditions. The results of the experiments indicate that the proposed method is highly effective in detecting plant diseases in real-world situations, with an accuracy rate of 93.67%. The new dataset and proposed method offer valuable resources to the plant disease detection community. They can be utilized to train more precise plant disease detection models, which can help decrease yield losses caused by plant diseases.

Xception, VGG16, ResNet152V2, InceptionResNetV2, DenseNet201, and MobileNetV2 are just a few of the well-known CNN models that were used in this study [2] to analyze four different major Bangladeshi crops. For Corn and Potato, Densenet201 achieves an accuracy of 95.52 and 98.55 percent, respectively; for Rice, it achieves an accuracy of 64.30 percent. When MobileNetV2 is used, wheat can be identified with 98.28% accuracy. For the experiments, images in both color and grayscale are used. Except for Rice, the majority of the crops have seen a significant improvement in disease prediction.

In this study [5], the authors have successfully developed a new and advanced method for identifying plant leaf diseases using a deep convolutional neural network (Deep CNN) technique. The model was trained with a diverse dataset consisting of 39 different classes of plant leaves and background images. To enhance the model's performance, six types of data augmentation techniques were employed, making it stand out from traditional machine-learning methods. The proposed model achieved an outstanding classification accuracy of 96.46%, exceeding the accuracy of conventional methods. Additionally, the model was evaluated for its consistency and reliability, both of which proved to be top-notch. This innovative approach to plant leaf disease identification is precise, reliable, and user-friendly, making it a suitable tool to assist farmers in detecting diseases early and reducing crop losses. The future of plant disease detection looks promising with this new method.

This study [6] has shown that smartphone-assisted crop disease diagnosis is a promising approach for improving food security on a global scale. Through analyzing 54,306 images of both healthy and diseased plant leaves, a deep convolutional neural network was trained in this study. The model achieved an impressive accuracy rate of 99.35% on a separate test set, showcasing the potential effectiveness of this technique. The study suggests that smartphone-assisted crop disease diagnosis could be used to identify a wide range of diseases in a variety of crop species. This could help to reduce crop losses and improve food security for millions of people around the world.

The paper [7] uses machine learning to identify rice blast disease. For the model's training and testing, 300 images from the dataset collected where both healthy and affected rice leaves were applied. The images are first preprocessed by utilizing HSV color conversion, and then image processing is carried out using K-Means clustering. The data was run by an Artificial Neural Network after feature extraction in order to generate predictions. The final results showed a test set accuracy of 90% for healthy and diseased leaves, respectively, and a training set accuracy of 99%.

The final model produced by the authors of the paper [7] has an accuracy of 96.7% for detecting corn/maize leaf diseases. Preprocessing (before converting the binary images with and without noise, converting grayscale was done) and feature extraction were carried out using the PlantVillage dataset's images of corn leaves. The collected information was then used to categorize the various classes of corn leaf images that were infected and healthy. Neural Networks, Support Vector Machine, and Random Forest were used when accuracy was higher than 93%, with Random Forest providing better prediction accuracy.

Deep learning was the chosen method for identifying potato leaf disease in this research paper[8]. The dataset comprised four distinct categories of images, all of which displayed unhealthy potato leaves affected by the disease. The study implemented four primary approaches, including data acquisition, data preprocessing, data augmentation, and image classification. An experiment with 64 batches and 250 epochs is conducted to find diseases in potato leaves. Utilizing the VGG16 and VGG19 convolutional neural networks yields detection results with an accuracy of 91%.

In their paper [9], the authors utilized deep learning to examine a damaged image of a wheat plant. They specifically focused on the spikes and leaves, which were the most severely affected areas. Through deep learning, they trained 10 classes – 9 for affected wheat and 1 for healthy wheat – achieving an accuracy of 97.88% and 98.62%. The results indicated significant improvements of 7.01% and 15.92% for the VGG16 and ResNet50 deep-learning models, respectively.

In this paper the author analyzed a dataset of corn, apple, tomato, potato and rice to detect plant disease using a hybrid algorithm of CNN and Bayesian Optimized SVM, and Random Forest Classifier. LeNet, ShuffleNet, AlexNet, EffNet, and MobileNet were among the five CNN approaches that were examined for the issue at hand by training and testing. With an accuracy of 96.1%, MobileNet outperformed the others. In committees, all networks were combined and voted on using one of three voting methods: majority vote, hybrid feature-based random forest, or Bayesian optimized SVM. Along with providing excellent performance for the suggested task, it was noted that the best committee's total trainable attributes were lower than those of the canonical methods VGG16 and VGG19, which are frequently used for computer vision tasks. The idea of comparing various CNN in terms of performance and time

was finally put forth. ShuffleNet outperformed AlexNet in terms of performance, but AlexNet was able to do so in less time and at a lower 1.53%. [18]

### III. METHODOLOGY

This section uses an architectural diagram to explain how the proposed model works. It helps to give a full understanding of the system's operation. The chapter describes how the dataset collection process works using image sharpening techniques. It also explains how the Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTE-ENN) is used to resample the dataset. Furthermore, the chapter covers the application of image augmentation methods, the use of Convolutional Neural Networks (CNN), and the utilization of VGG16, Xception, and ResNet architectures. Additionally, Support Vector Machine (SVM) algorithms are incorporated into the process.

#### A. Architectural Design

A clear picture of our suggested system will be provided by the architectural design. It will describe how the system works in detail. The architectural design of the system we propose has multiple phases, as shown in Figure 3. The preprocessing stage will be the first thing the dataset goes through. There are techniques for resizing and sharpening during the preprocessing stage. In the following step, the dataset has been balanced by using the SMOTE-ENN approach. A training dataset and a test dataset will then be created from the dataset. Additional zooming, shifting, rotating, etc. will be applied to the train data. The augmented dataset will be used by the algorithms after augmentation. Applied CNN dense layer, conv2D, and maxpooling layer, and with CNN features final layer was added as SVM as a classifier for the hybrid model. Afterward, the test dataset will be used to execute the model and determine its accuracy. Figure 3 depicts the architectural layout of our proposed model.

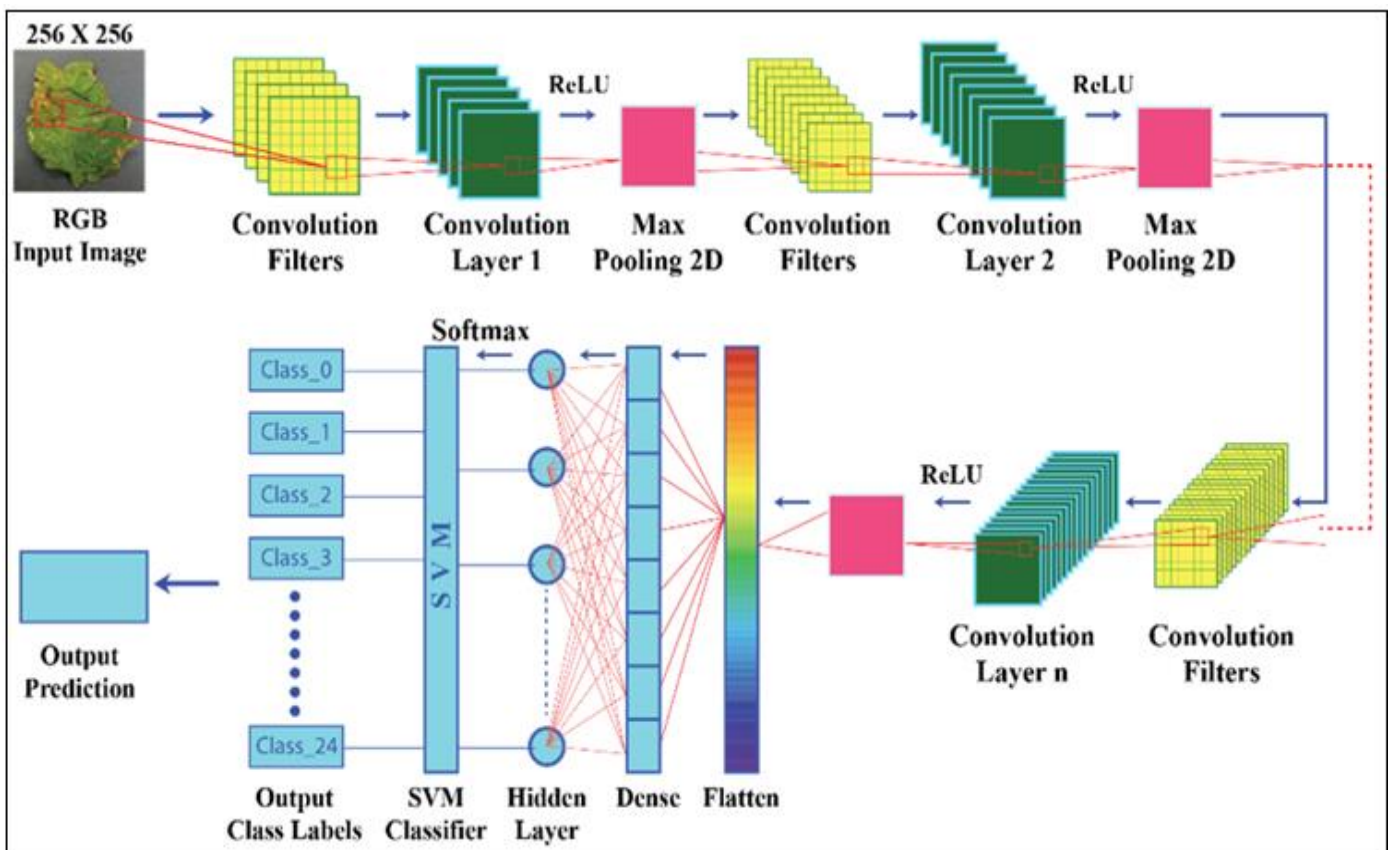


Fig 3 Architectural Design of our Model

#### B. Dataset Collection

Our suggested model makes use of a combined dataset. The www.kaggle.com website was used to collect three parts of the dataset. In addition to disinfected plant data, we obtained 31397 images from 20 diseases. The quantity of pictures in each category is shown in Figure 4.



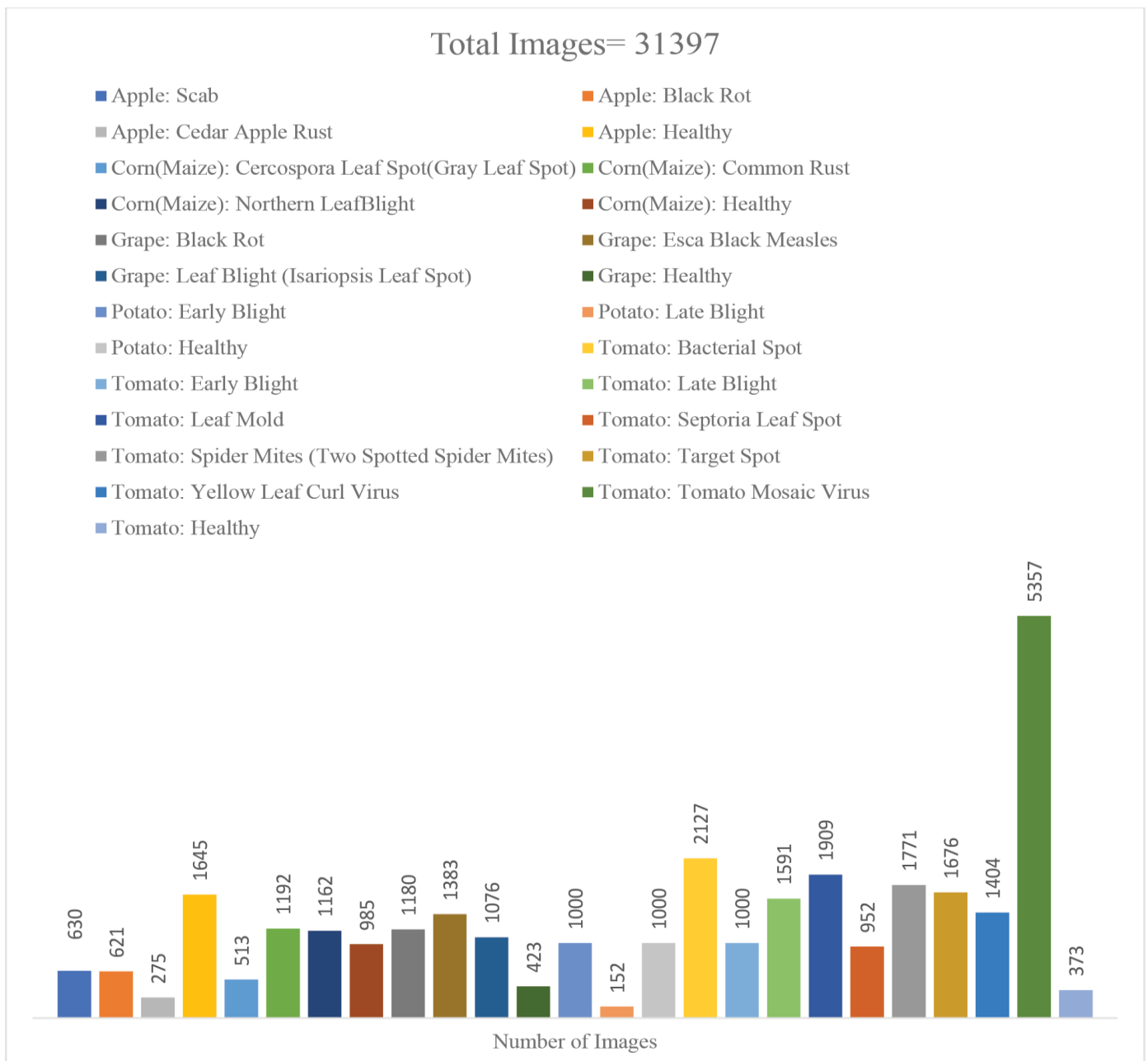


Fig 4 Number of Images in Each Category

C. Pre-Processing Technique

Since there was a lot of noise in the data, we downloaded from Kaggle, we used image sharpening to reduce the noise. With regard to disease names, we leveled the image. There were too many different types of diseases. As a result, we used the SMOTE-ENN resampling method to balance the dataset.

➤ Image Sharpening Technique

We have cleaned up the noise in our dataset using the image smoothing technique. A Gaussian kernel is utilized in this method. The kernel's width and height should be specified, and they should both be odd and positive. For X and Y directions, respectively, we defined sigma X Y. In order to establish identical sigma Y, it is essential to have the value of sigma X. If both values are set to 0, then the kernel size plays a crucial role in determining the values of

both sigma X and sigma Y. Gaussian blur successfully eliminates Gaussian noise [25].

We combined the original image with the image produced by the Gaussian blur using the addweight function. Alpha channel blending is aided by the function addweight. Using alpha blending, background image can be overlaid on top of a foreground image which is ransparent. Transparency usually serves as the fourth channel in an image. Alpha masks are used for masks like this one to let light through [40]

$$Sharp\ image = a * blur\_image + \beta * actual\_image + Y$$

We were able to create a clear image that the algorithm could use after putting these functions into practice.

➤ *SMOTE-ENN Technique*

The SMOTE-ENN technique is a combination of the Synthetic Minority Over-sampling Technique (SMOTE) and Wilson's Edited Nearest Neighbor Rule (ENN). The method was first introduced by Batista et al. (2004) [26]. SMOTE generates new instances of the minority class by creating synthetic examples that lie within the vicinity of existing minority class instances. The goal is to address the sparsity of minority class data. ENN, on the other hand, is used to improve class cluster definition by removing noise from the

dataset. While ENN can potentially enhance classification accuracy, it also runs the risk of misidentifying boundary samples.

During the SMOTE-ENN process, instances are examined, and those whose class labels differ from two out of their three nearest neighbors are eliminated [17]. To provide better context, Figure 5 illustrates the image that results from applying preprocessing procedures.

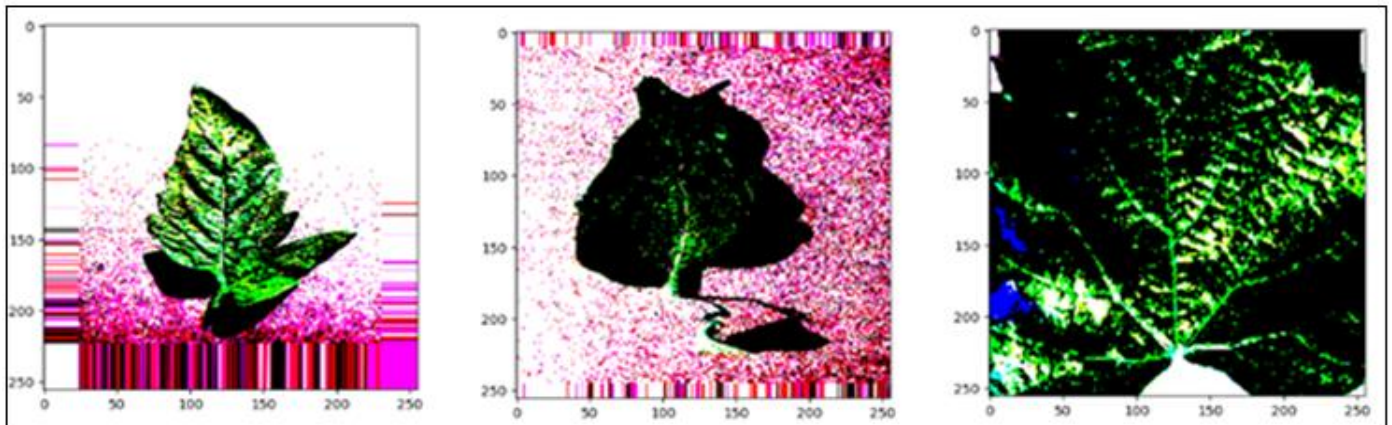


Fig 5 Images after Applying Preprocess Function

D. *CNN Algorithm*

The flow of continuous information along with a feed-forward neural network is called as a "convolutional neural network" (CNN). This is an Artificial Neural Network (ANN) that comes from nature on the basis of the human brain. The brain's architecture is made up of very simple parts and layers that change depending on the complexity of the cells in the visual cortex [19]. These modules are followed by a fully convolutional network, which may have various layers or sometimes one fully linked layer together. The approaches of a deep CNN model are always built by dealing with the various CNN methods. Although many different CNN variant architectures can be broken down into different modules, the most typical CNN components are convolutional and pooling layers, and fully connected layer (FC).

➤ *Convolution Layer*

The convolutional layer's primary function is to extract features, and in order to do this, it learns the informative feature representation from the image that it receives as input. Feature maps in this layer are used to organize the neurons in the convolutional layer.

➤ *Pooling Layer*

The pooling layer in CNN is primarily used to restrict the feature space spatial resolution in order to achieve high positional stability for input distortion [30]. The typical CNN design makes use of mean pooling aggregation layers to distribute an image's average data input to the following layer in the network. The most recent designs, on the other hand, use max-pooling layers, which transfer the highest

amplitude to the layer beneath them in the stack. In its most formal form, max-pooling involves choosing the values that are highest across all receptive fields.

➤ *Connected Layer*

Only a few pooling and convolutional layers are stacked on top of one another during the propagation through to the CNN. By doing this, the high-order generic characteristics map is obtained. The fully linked layer, which was the final layer to be added, was made to translate the results of the calculations made by the layers before it and to calculate the high-level reasoning function.

The following is how a convolutional layer works:

$$F(i, j) = (I * K)(i, j) \sum \sum (i + m, j + n)K(m, n)$$

The size (m n) 2D filter, the input matrix, and the output of the 2D feature map are all denoted by the letters "I," "K," and "F," respectively. I\*K offers the convolutional layer procedure [31] [32].

Figure 5, below shows the approaches of our proposed CNN model as well as the total trainable and non-trainable parameters. Our CNN model has an input size of (256,256,3) and ten classes with a softmax activation function. Five convolution layers and three fully connected layers with the activation function were used. Our chosen pooling layer was max pooling. We used regularizers and dropout in order to avoid overfitting. Figure 6, represents the hidden layer of CNN.

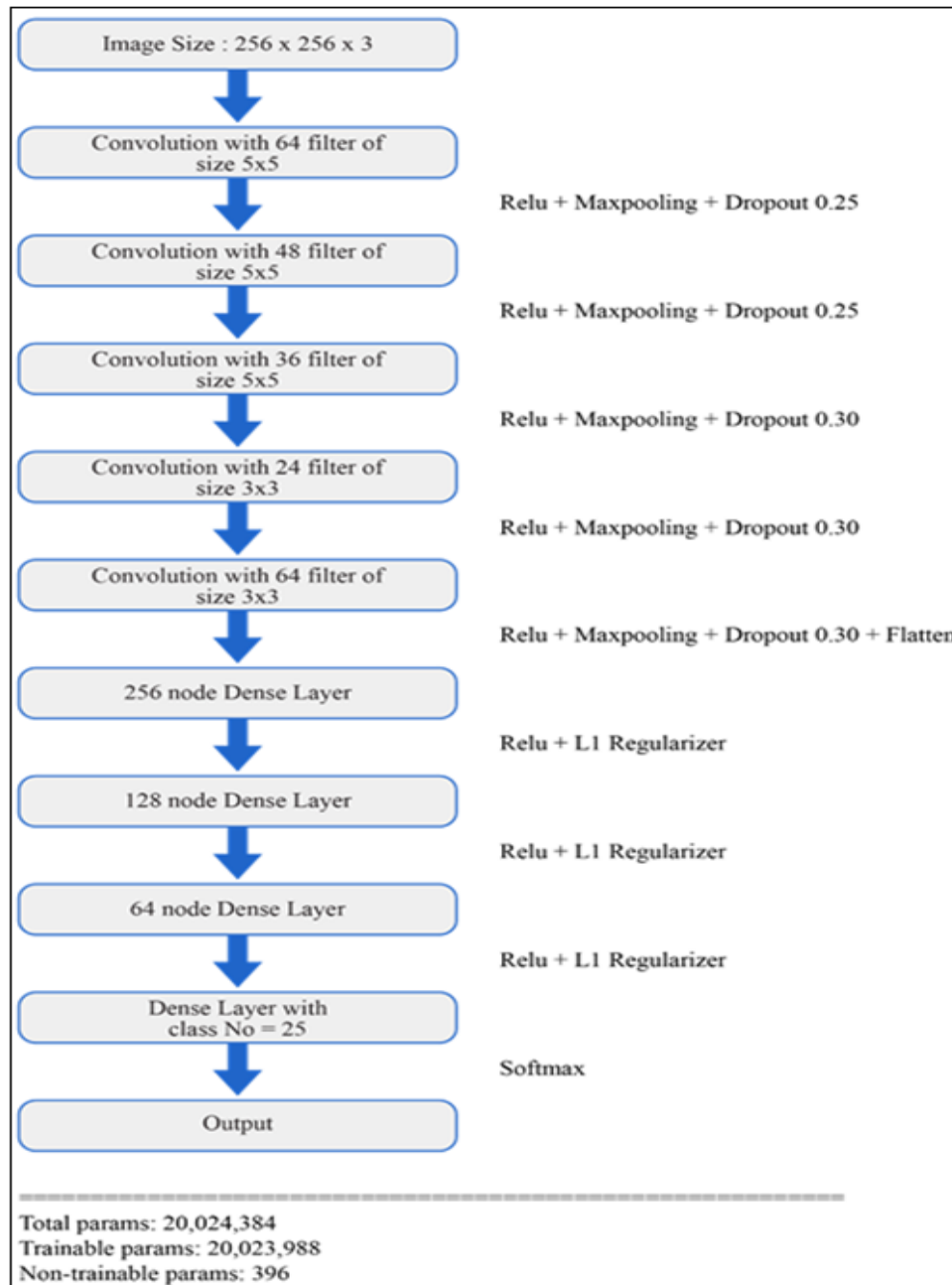


Fig 6 CNN Architectural and Parameter

**E. VGG Algorithm**

The Convolutional Neural Network (CNN) approach known as VGG is a multilayer deep CNN approach. VGG-16 and VGG-19, which have 16 and 19 hidden layers, respectively, are considered "deep" in terms of the number of various layers. All cutting-edge new object recognition models can be linked to the VGG architecture as their source of inspiration. Along with ImageNet, many other tasks and datasets are also outperformed by the VGGNet, which was created as a deep neural network. It is also thought to be one of the most popular designs for image recognition right now [33].

➤ **VGG19 Algorithm**

Conceptually equivalent to the VGG16 model, the VGG19 model, also referred to as VGGNet-19, differs only which supports 19 layers. The words "16" and "19"

referring the total number of weight layers (convolutional layers) in the model. As a result, it can be concluded that VGG19 has more convolutional layers than VGG16. We will delve into the characteristics that set the VGG16 and VGG19 networks apart in more detail in the next section of this essay [36].

This network was given an RGB image that was 256 by 256 in size, which indicates that the matrix was not in the proper shape (256,256,3). The entire training set's average RGB value integrity was subtracted from each pixel as the only pre-processing step. They could cover the entire image with kernels that were size (32), with a length of 1 pixel. Space-filling was used to keep the image's resolution in space. The 22-pixel window was max-pooled with stride 2 [36]. The summary of the current VGG-19 model is shown in Figure 7.



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 256, 256, 3)]	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590080
block3_conv4 (Conv2D)	(None, 64, 64, 256)	590080
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv4 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv4 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0
flatten (Flatten)	(None, 32768)	0
dense (Dense)	(None, 256)	8388864
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense)	(None, 120)	15480
dense_3 (Dense)	(None, 25)	3025
=====		
Total params: 28,464,649		
Trainable params: 20,024,384		
Non-trainable params: 8,440,265		

Fig 7 Existing VGG19 Model Summary

Figure 6 showcases the comprehensive summary of the VGG19 model. The input image size has been set to (36,969,561) and the trainable parameter has been disabled for optimum results. The model boasts 25 layers with Relu and softmax activation functions, and unit sizes of 256 and 128. To ensure minimal overfitting, we have incorporated a Flatten layer and two Dense layers to convert the 2D array to a 1D array. L2 regularizers and dropouts have been implemented as well. The final layer has a class size of 25 and utilizes the softmax activation function. It is noteworthy that the trainable parameter count is lower than that of the original model. Figure 8 has been included to provide you with a visual representation of the modified VGG19 model

```

Model: "Sequential"
-----
Layer (type)                Output Shape                Param #
-----
Sequential (Sequential)    (None, 10)                  0
flatten_1                   (None, 10)                  0
dense_2 (Dense)             (None, 512)                 16777728
dense_5 (Dense)             (None, 256)                 131328
dense_6 (Dense)             (None, 128)                 32896
dense_7 (Dense)             (None, 25)                  3225
-----
Total params: 36,969,561
Trainable params: 26,969,400
Non-trainable params: 10,000,161
    
```

Fig 8 Modified VGG19 Model Summary

*F. Xception Algorithm*

Deep CNN architecture called Xception is built on Depth Wise Separable Convolutions. The idea from "Inception" is carried to the identical conclusion of them in "Extreme Inception," or "Xception" [37]. The Inception model combines deeper convolutional layers with broader convolutional layers. There are two levels in this model, and each level has three convolutional layers. Because the Xception model has one level of a single layer, which differs from the Inception model along with another level. The three parts of the result are separated into three layers by this layer and sent on to the following set of filters What characterizes the Xception model is the Depth-wise Invertible Convolution. Depth-wise and point-wise convolution are used by Xception model in all deep CNN models, space and channels are dispersed [38]. The summary of the current Xception model is shown in Figure 9.

```

Mblock13      pool (MaxPooling2D) (None, 6, 6, 1024) 0 ['block13 sepconv2_bn[0][0]']
batch normalization 3 (BatchNormalization) (None 11, 11, 1024) 4096 ['conv2d 3[0][0]']
add 11 (Add)      (None 11, 11, 1024) 0 ['block13 sepconv2_bn[0][0]']
block13 sepconv1 (SeparableConv2D) (None 11, 11, 1536) 1582080 ['add_11[0][0]']
block13 sepconv2 bn (BatchNormalization) (None 11, 11, 1536) 6144 ['block13 sepconv2[0][0]']
block13 sepconv2 bn (BatchNormalization) (None 11, 11, 1024) 4096 ['block13 sepconv2[0][0]']
Prediction (Dense) (None 10) 20470 ['avg_pool[0][0]']
-----
Total params: 20,221,870
Trainable params: 20,727,890
Non-trainable params: 54,528
    
```

Fig 9 Xception Model Summary

*G. ResNet15V2 Algorithm*

A convolutional neural network (CNN) model from the ResNet family of architectures is called ResNet-15V2. By utilizing skip connections or shortcuts, ResNet—short for Residual Network—was developed to address the difficulty of training deeper networks.

Specific to a ResNet variant with 15 convolutional layers, the ResNet-15V2 model is used. It adheres to the fundamental ResNet building block, called a residual block. A series of convolutional layers, batch normalization, and non-linear activation methods like ReLU (Rectified Linear Unit) is applied to the input in a residual block. The original input is then combined with the residual block's output using a skip connection. Figure 10, shows our model summary.

```

block13 sepconv2 bn (BatchNormalization) (None 11, 11, 1024) 4096 ['block13 sepconv2[0][0]']
conv2d 3 (Conv2D) (None, 6, 6, 1024) 745472 ['add_10[0][0]']
block13 pool (MaxPooling2D) (None, 6, 6, 1024) 0 ['block13 sepconv2_bn[0][0]']
batch normalization 3 (BatchNormalization) (None 11, 11, 1024) 4096 ['conv2d 3[0][0]']
add 11 (Add) (None 11, 11, 1024) 0 ['block13 sepconv2_bn[0][0]']
block13 sepconv1 (SeparableConv2D) (None 11, 11, 1536) 1582080 ['add_11[0][0]']
block13 sepconv2 bn (BatchNormalization) (None 11, 11, 1536) 6144 ['block13 sepconv2[0][0]']
block13 sepconv2 bn (BatchNormalization) (None 11, 11, 1024) 4096 ['block13 sepconv2[0][0]']
Prediction (Dense) (None 10) 20490 ['avg_pool[0][0]']

=====
Total params: 22,881,970
Trainable params: 20,827,442
Non-trainable params: 2,054,528
    
```

Fig 10 ResNet15V2 Model Summary

*H. DenseNet*

A member of the DenseNet architecture family is the deep learning model known as the DenseNet-201. DenseNet, short for Dense Convolutional Network, is renowned for its dense pattern of connectivity, in which every layer is connected to every other layer in a feed-forward manner. This network's distinct connectivity structure allows for efficient feature reuse and gradient flow.

A version of DenseNet with 201 layers is known as the DenseNet-201 model. It is a highly deep and powerful convolutional neural network (CNN) model. DenseNet-201 is appropriate for tasks requiring a large number of trainable parameters and intricate feature representations because it combines the advantages of deep networks with the effective use of parameters. The summary of the current DenseNet model is shown in Figure 11.

```

=====
Total params: 81,419,785
Trainable params: 81,190,729
Non-trainable params: 229,056
    
```

Fig 11 DenseNet Model Summary

*I. SVM Algorithm*

For classification and regression tasks, a well-liked machine learning algorithm called the Support Vector Machine (SVM) is frequently used. It is a member of the family of supervised learning algorithms and is renowned for its ability to handle datasets with both linear and non-linear separability. Finding an ideal hyperplane to divide data points into different classes in the feature space is the basic tenet of SVM. In a scenario involving binary classification, the hyperplane serves as a decision boundary, maximizing the distance between the nearest data points of various classes. The support vectors, also known as these data points, are very important in determining the hyperplane.

that is desired. The ability of SVM to handle datasets with a large number of features is one of its main advantages. Through the discovery of a hyperplane in the reorganized feature space, it successfully overcomes the "curse of dimensionality." Since SVM aims to maximize the margin between classes rather than perfectly fitting the training data, it is also renowned for its resistance to overfitting.

*J. Hybridization*

The hybridization of CNNs and SVMs will involve using the pre-trained CNN as a feature extractor for plant leaf images. The CNN will be fed with plant leaf images from the dataset, and its penultimate layer's output (i.e., the layer before the final classification layer) will be extracted as high-level features.

SVM is a powerful tool for analyzing both numerical and categorical features in a range of different data types. Based on a kernel function, it increases the dimensionality of the input data to allow for more complex decision boundaries. The choice of kernel function, such as a linear, polynomial, or radial basis function, depends on the characteristics of the data and the level of model flexibility

These features will then be used as the input for the SVM classifier, which will be trained to differentiate between healthy and diseased plants. During the training process, the SVM will learn to optimize the hyperplane that best separates the two classes in the feature space.



During the prediction phase, a plant leaf image's features will be extracted using the CNN, and the SVM will classify the sample based on these features. If the output is close to the hyperplane, the plant may be deemed healthy, but if it is farther away, the plant may be classified as diseased.

By combining CNN's feature extraction capabilities with SVM's classification power, we aim to build a reliable and accurate system for detecting plant diseases so that farmers may efficiently monitor their crops.

We first add a sequential layer to our model. After we made a CNN model with a conv2D layer as our dataset contains 2D images, after that we added a maxpooling layer. Then we added a dense layer with 512, 256, and 128 nodes and the activation function are ReLu. This is the normal building of the CNN model. Then we added an SVM classifier layer. As our dataset has 25 classes, we needed to add a softmax activation function with 25 Dense. And we

needed to use Adam as optimizer and squired\_hinge as loss calculation for adding the SVM classification in our model. Finally, our mode is ready with CNN multi-layers with the SVM classifier as the final layer.

#### IV. EXPERIMENTAL RESULT

In this chapter, we have covered how to set up the experimental environment. This includes selecting the appropriate language, environment, libraries, packages, functions, and parameters. We have also discussed the evaluation metrics used to measure the performance of different algorithms. Additionally, we have identified the algorithm that performs the best. Table 1 shows the train and test accuracy of our dataset, with two types of accuracy: one from only the CNN model layer and the other from the CNN layer with the SVM layer. Figure 12 displays the accuracy and loss curve, with blue and red colors representing accuracy, validation accuracy, loss, and validation loss for both graphs.

Table 1 Difference between SVM and non-SVM Accuracy of Our Train and Test Data.

Model			Before Applying SVM		After Applying SVM	
			Train Accuracy	Test Accuracy	Train Accuracy	Test Accuracy
CNN			88.02%	89.77%	88.54%	89.79%
VGG19			89.88%	90.30%	89.97%	90.71%
ResNet15V2			84.02%	86.20%	84.31%	84.89%
DenseNet			89.90%	90.01%	90.12%	91.96%
Xception	80.01%	80.78%	81.02%	82.05%		

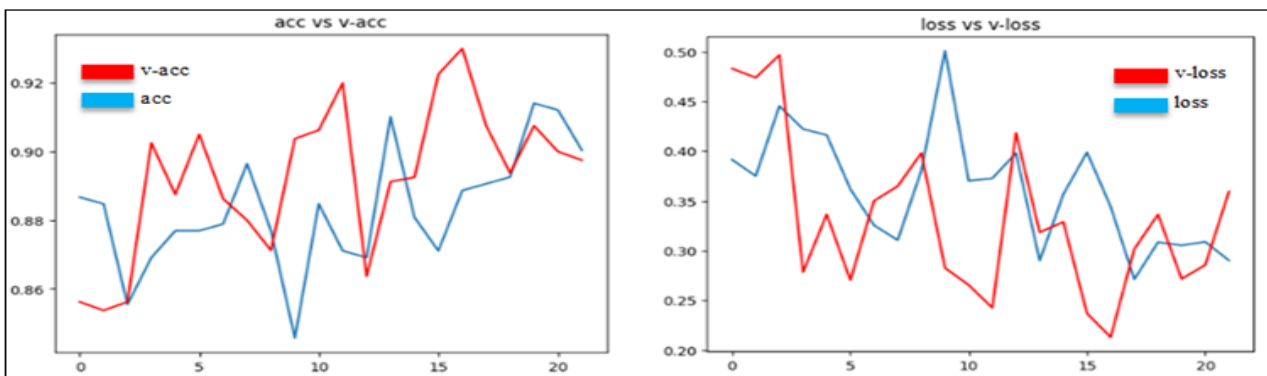


Fig 12 Accuracy and Loss Curve

##### A. Project Setup

Our proposed model was carefully crafted through a meticulous project setup that combined key components and strategic decisions. Here, we provide insight into this setup.

To begin, we partitioned the dataset into an 80% allocation for training and a 20% allocation for testing. Before this, we applied the Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTE-ENN) to ensure balanced representation within the dataset. Figure 13 clearly shows the equitable distribution of each category within the training dataset.

The technical construction of our suggested system unfolded within the Google Colab Pro environment. This environment made it easy to integrate the Python

programming language, alongside the powerful TensorFlow and Keras libraries. We used diverse techniques and tools, including essential components such as dropout layers, regularizers, and model checkpointing. Precision in numerical computations was facilitated by the collaborative prowess of the sklearn library and numpy.

We also made strategic decisions about hyperparameter configuration, carefully tuning epochs and batch size. We used algorithmic frameworks like VGG19, DenseNet, Xception, and ResNet15V2 to define the model's architecture. The Adam optimizer led the model's optimization journey, while the dynamic interplay of ReLu and softmax activation functions contributed to the system's robustness. The training process extended across a span of 50 epochs, marked by a systematic augmentation in batch size from 16 to 32.

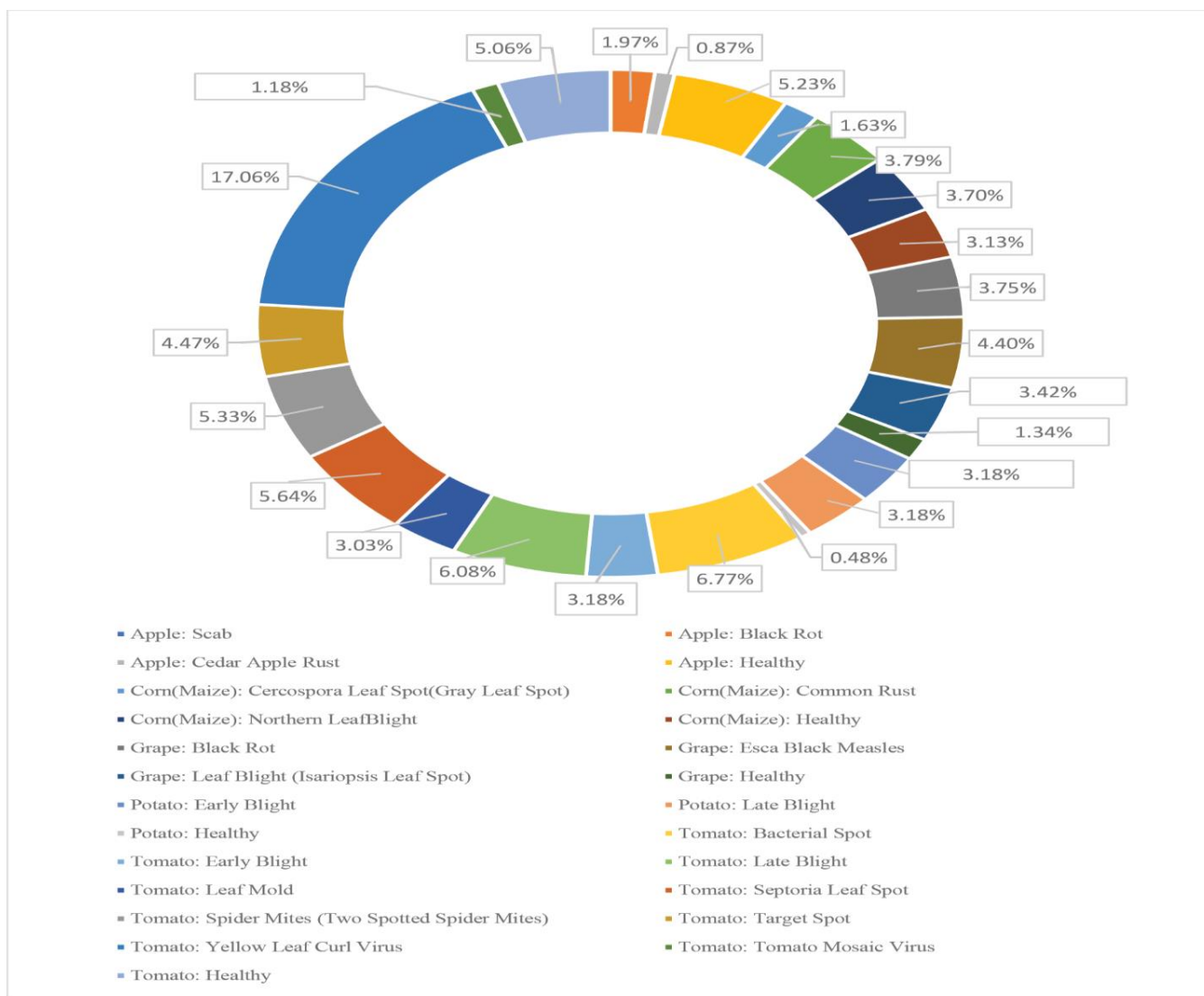


Fig 13 Training Dataset

To offer a comprehensive understanding of disease taxonomy, Table 2 serves as a repository, outlining the categorical labels pertaining to distinct diseases under consideration. This foundational groundwork paved the way for the ensuing stages of evaluation and analysis.

Table 2 Disease Labels of Our Dataset

Categories	Apple___ Apple_scab	Apple___Blac k_rot	Apple___Cedar_a pple_rust	Apple___healthy	Corn_maize___Cercospora _leaf_spot_Gray_leaf_spot
Label	0	1	2	3	4
Categories	Corn_maize___Common _rust_	Corn_maize___ Northern_Leaf _Blight	Corn_maize___he althy	Grape___Black_ rot	Grape___Esca_Black_ Measles
Label	5	6	7	8	9
Categories	Grape___Leaf_ blight_Isariopsis_Leaf_S pot	Grape___ healthy	Potato___Early_ blight	Potato___Late_bli ght	Potato___healthy
Label	10	11	12	13	14
Categories	Tomato___ Bacterial_spot	Tomato___ Early_blight	Tomato___Late_ blight	Tomato___Leaf_ Mold	Tomato___Septoria_ leaf_spot
Label	15	16	17	18	19
Categories	Tomato___Spider_mites _Two_spotted_ spider_mite	Tomato___ Target_Spot	Tomato___ Tomato_Yellow_ Leaf_Curl_Virus	Tomato___ Tomato_mosaic_ virus	Tomato___healthy
Label	20	21	22	23	24

**B. Evaluation Metrics**

Our proposed model's effectiveness will be greatly improved by utilizing key metrics such as accuracy, precision, recall, and F1-Score. We will provide an in-depth explanation of the confusion matrix and its interpretation to emphasize the criticality of these metrics. Properly assessing the performance and efficiency of our model in its intended use is essential.

➤ **Confusion Matrix**

Confusion matrix is a table that represents the performance of a machine-learning classification model. Even though the confusion matrix uses a confusing set of terms, it is relatively easy to understand. In Figure 14, a straightforward confusion matrix is displayed and in table 3 displays the confusion matrix generated from our dataset.

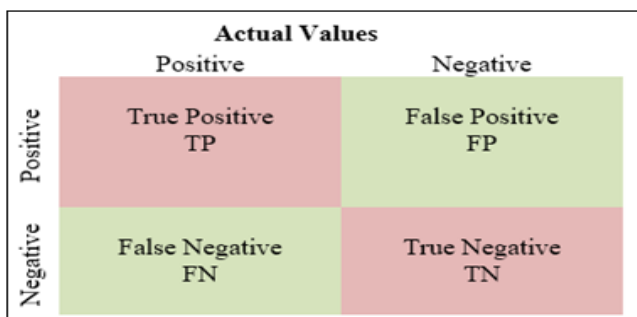


Fig 14 Confusion Matrix

- True Positive (TP): The predicted result is positive and the actual value is also positive (1, 1).
- True Negative (TN): The predicted result is negative, and the actual value is also negative (0, 0).
- False Positive (FP): The predicted result is positive, but the actual value is negative (1, 0).
- False Negative (FN): The predicted result is negative, but the actual value is positive (0, 1).

The calculation of the accuracy, precision, recall, and F1-score values came from the confusion matrix.

➤ **Accuracy:**

Accuracy is the ratio of total right predictions and total predictions. This ratio represents the possibility to get the correct value of an ML model.

$$Accuracy = \frac{Total\ Number\ of\ Right\ Prediction}{Total\ Number\ of\ Prediction}$$

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

➤ **Precision:**

Precision is the ratio of a total number of true positive predictions and a total number of predicted positives.

$$Precision = \frac{Total\ Number\ of\ True\ Positive}{Total\ Number\ of\ Predicted\ Positive}$$

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

➤ **Recall:**

Recall is the ratio of a total number of true positive predictions and a total number of actual positives.

$$Recall = \frac{Total\ Number\ of\ True\ Positive}{Total\ Number\ of\ Actual\ Positive}$$

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

➤ **F1-Score:**

The F1-score takes the harmonic mean of a classifier's precision and recall to create a single statistic [41]

$$F1 - Score = \frac{2 \times TP}{2 \times TP + FP + FN}$$

Table 3 Confusion Matrix

	Precision	Recall	F1-Score	Support
0	0.87	0.85	0.86	94
1	0.82	0.89	0.89	91
2	0.87	0.87	0.86	93
3	0.89	0.85	0.87	89
4	0.84	0.84	0.89	87
5	0.89	0.86	0.69	86
6	0.88	0.89	0.76	89
7	0.85	0.86	0.79	88
8	0.83	0.85	0.81	92
9	0.88	0.85	0.88	87
10	0.88	0.87	0.87	81
11	0.89	0.81	0.86	85
12	0.87	0.89	0.76	89
13	0.86	0.83	0.89	81
14	0.85	0.86	0.62	90
15	0.89	0.87	0.87	92
16	0.91	0.89	0.78	91
17	0.90	0.86	0.82	90
18	0.81	0.76	0.86	90
19	0.82	0.78	0.84	90
20	0.88	0.79	0.87	87
21	0.84	0.68	0.86	81
22	0.87	0.89	0.86	89
23	0.87	0.87	0.87	90
24	0.89	0.64	0.89	90
accuracy			0.91	128
macro avg	0.74	0.75	0.74	128
weighted avg	0.80	0.80	0.80	128

**C. Result Analysis:**

Our system uses a well-balanced dataset and the latest techniques such as SMOTE-ENN, along with advanced models like CNN, VGG19, ResNet15V2, DenseNet, Xception, and a strong hybrid model with SVM classification. We are excited to report that our models have achieved exceptional accuracy scores. CNN, VGG19, ResNet15V2, DenseNet, and Xception each have impressive scores of 90.60%, 91.14%, 89.02%, 91.82%, and 82.82%, respectively. These scores demonstrate the accuracy and effectiveness of our system in correctly classifying instances.



We have included Table 4 and Table 5 for a comprehensive comparison, which outlines the F1-Score, recall, accuracy, and precision metrics. The integration of SVM into our hybrid model resulted in even higher accuracy

compared to our previous models, further highlighting the efficacy of our approach. We are confident that our system can deliver accurate and reliable results.

Table 4 Comparing Table of Our Applied Models

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	90.60	91	98	95
VGG19	91.14	92	98	94
ResNet15V2	89	90	90	89
DenseNet	91.82	92	99	96
Xception	82.69	85	94	90

Table 5 After Adding SVM Classifier with CNN Models

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	93.88	94	98	96
VGG19	94.02	95	98	97
ResNet15V2	87.88	88	97	92
DenseNet	95.82	96	1.0	98
Xception	89.45	89	98	93

## DECISION

In our study, we applied a hybrid model along with CNN and SVM classifier to detect plant disease. We discovered that the DenseNet model, when combined with the SMOTE-ENN and Image Sharpening Technique, yields remarkable results in the detection of Plant diseases. The suggested model can distinguish between the harmed and unaffected Plant plants with greater accuracy. DenseNet achieved 91.82% accuracy above all of our models. The precision is 87%, the recall is 89 and the f1-score is 88. After applying the SVM layer to our model, DenseNet also achieved more accuracy above all of the models which is 92.82%. Our goal was to find the changes in accuracy after the hybridization of our model. We noticed that after adding SVM as the final layer, the accuracy of all our models increased.

## V. CONCLUSION

The detection of plant diseases is a crucial area of study because it affects the livelihood of countless people worldwide. A significant area of study is plant disease detection because plants are an important source of food and for the environment. In order to observe the model evaluation separately, five different CNN models—apple, corn, grape, potato, and tomato—have been applied in this study. By using CNN, VGG19, ResNet152V2, DenseNet201, and Xception respectively, the model without applying SVM images of apple, grape, potato, and tomato datasets was able to achieve the highest accuracy, while ResNet152V2, VGG19, Inception ResNet15V2, and DenseNet201 were able to achieve more accuracy after applying SVM to the images. In comparison to the other six models, DenseNet using SVM has significantly higher accuracy (91.82%) across all plants. In our study, the dataset containing the five main plants with total of 25 class labels was made for plant disease detection. Five CNN architectures are being tested on RGB-colored images, and a comparative analysis can be seen for the major plants for

hybrid algorithm performance instead of displaying only the performance of CNNs across all 25 classes, the study is restricted to individual experiments with the plant species. For the future work of additional experiments, the dataset can be combined and conducted as part of this study's future work, and using the best CNN model and a proposed hybrid model a comparative analysis will be done. Additionally, a mobile application based on the top-performing models could be created to make it simpler for farmers to identify plant diseases.

## FUTURE WORKS

In future work, we would like to implement our model using segmentation and other classifier for hybridization such as Random Forest. A real-life tool such as mobile application can be made for farmers to detect their plant disease and take necessary steps.

## REFERENCES

- [1]. M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry (Basel)*, vol. 11, no. 7, 2019, doi: 10.3390/sym11070939.
- [2]. Disease Detection of Bangladeshi Crops Using Image Processing and Deep Learning – A Comparative Analysis
- [3]. L. Cai, J. Gao, and D. Zhao, "A review of the application of deep learning in medical image classification and segmentation," *Ann. Transl. Med.*, vol. 8, no. 11, pp. 713–713, 2020, doi: 10.21037/atm.2020.02.44
- [4]. Plant Disease Dataset <https://www.kaggle.com/datasets/saroz014/plant-disease>

- [5]. G. Geetharamani and A. P. J., "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Comput. Electr. Eng.*, vol. 76, pp. 323–338, 2019, doi: 10.1016/j.compeleceng.2019.04.011.
- [6]. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, no. September, pp. 1–10, 2016, doi: 10.3389/fpls.2016.01419.
- [7]. M. Adekunle, "Implementation of Improved Machine Learning Techniques for Plant Disease Detection and Classification," *IEEE Access*, vol. 2018-Janua, no. 1, pp. 1–4, 2019.
- [8]. R. A. Sholihati, I. A. Sulistijono, A. Risnumawan, and E. Kusumawati, "Potato Leaf Disease Classification Using Deep Learning Approach," *IES 2020 - Int. Electron. Symp. Role Auton. Intell. Syst. Hum. Life Comf.*, pp. 392–397, 2020, doi: 10.1109/IES50839.2020.9231784.
- [9]. L. Goyal, C. M. Sharma, A. Singh, and P. K. Singh, "Leaf and spike wheat disease detection & classification using an improved deep convolutional architecture," *Informatics Med. Unlocked*, vol. 25, no. April, p. 100642, 2021, doi: 10.1016/j.imu.2021.100642.
- [10]. OERKE, E. C. (2005, December 9). Crop losses to pests. *The Journal of Agricultural Science*, 144(1), 31–43. <https://doi.org/10.1017/s0021859605005708>
- [11]. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018, April). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90. <https://doi.org/10.1016/j.compag.2018.02.016>
- [12]. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016, September 22). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7. <https://doi.org/10.3389/fpls.2016.01419>
- [13]. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017, May 24). Image Net classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- [14]. Simonyan, K., & Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*. <https://doi.org/10.48550/arXiv.1409.1556>
- [15]. Fei-Fei, L., Deng, J., & Li, K. (2010, March 22). ImageNet: Constructing a large-scale image database. *Journal of Vision*, 9(8), 1037–1037. <https://doi.org/10.1167/9.8.1037>
- [16]. Shorten, C., & Khoshgoftaar, T. M. (2019, July 6). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1). <https://doi.org/10.1186/s40537-019-0197-0>
- [17]. Sokolova, M., & Lapalme, G. (2009, July). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- [18]. Ashutosh Kumar Singh, SVN Sreenivasu, U.S.B. K. Mahalaxmi, Himanshu Sharma, Dinesh D. Patil, Evans Asenso, "Hybrid Feature-Based Disease Detection in Plant Leaf Using Convolutional Neural Network, Bayesian Optimized SVM, and Random Forest Classifier", *Journal of Food Quality*, vol. 2022, Article ID 2845320, 16 pages, 2022. <https://doi.org/10.1155/2022/2845320>
- [19]. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015). <https://doi.org/10.1038/nature14539>
- [20]. Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017, doi:10.1016/j.neucom.2017.06.023.
- [21]. "Apple Scab." Penn State Extension, The Pennsylvania State University, [extension.psu.edu/plants/trees-and-shrubs/apple/scab](http://extension.psu.edu/plants/trees-and-shrubs/apple/scab).
- [22]. "Apple Black Rot." University of California Agriculture and Natural Resources, [ucanr.edu/sites/agnr/files/12340.pdf](http://ucanr.edu/sites/agnr/files/12340.pdf).
- [23]. "Cedar Apple Rust". Cornell University Cooperative Extension,
- [24]. "Corn Diseases." University of Minnesota Extension, [www.extension.umn.edu/agriculture/crops/corn/corn-diseases/](http://www.extension.umn.edu/agriculture/crops/corn/corn-diseases/).
- [25]. "Corn Diseases." Iowa State University Extension and Outreach, [www.extension.iastate.edu/crops/corn/diseases/](http://www.extension.iastate.edu/crops/corn/diseases/).
- [26]. "Corn Diseases." Purdue University Extension, [www.extension.purdue.edu/extmedia/ANR-11-001.pdf](http://www.extension.purdue.edu/extmedia/ANR-11-001.pdf).
- [27]. Black Rot: [https://en.wikipedia.org/wiki/Black\\_rot\\_\(grape\)](https://en.wikipedia.org/wiki/Black_rot_(grape))
- [28]. Guignardia bidwellii: [https://en.wikipedia.org/wiki/Guignardia\\_bidwellii](https://en.wikipedia.org/wiki/Guignardia_bidwellii)
- [29]. Martín MT, Cobos R, Martín L, López-Enríquez L. Real-time PCR detection of *Phaeoconiella chlamydospora* and *Phaeoacremonium aleophilum*. *Appl Environ Microbiol*. 2012 Jun;78(11):3985-91. doi: 10.1128/AEM.07360-11. Epub 2012 Mar 23. PMID: 22447605; PMCID: PMC3346403.
- [30]. *Phaeoconiella chlamydospora*: [https://en.wikipedia.org/wiki/Phaeoconiella\\_chlamydospora](https://en.wikipedia.org/wiki/Phaeoconiella_chlamydospora)
- [31]. *Phaeoacremonium* spp.: <https://en.wikipedia.org/wiki/Phaeoacremonium>
- [32]. Grape Leaf Blight [https://en.wikipedia.org/wiki/Leaf\\_blight\\_\(grape\)](https://en.wikipedia.org/wiki/Leaf_blight_(grape))
- [33]. Shattock, R. (2002, August). *Compendium of Potato Diseases*, Second Edition. W.R. Stevenson. *Plant Pathology*, 51(4), 520–520. <https://doi.org/10.1046/j.1365-3059.2002.06934.x>
- [34]. Early Blight and Late Blight of Potato | Integrated Pest Management. (2021, September 7). Early Blight and Late Blight of Potato | Integrated Pest Management. <https://ipm.cahnr.uconn.edu/early-blight-and-late-blight-of-potato/>
- [35]. FRY, W. (2008, February 24). *Phytophthora infestans*: the plant (andRgene) destroyer. *Molecular Plant Pathology*, 9(3), 385–402. <https://doi.org/10.1111/j.1364-3703.2007.00465.x>

- [36]. Hadwiger, L. A., McDonel, H., & Glawe, D. (2015, March 14). Wild Yeast Strains as Prospective Candidates to Induce Resistance Against Potato Late Blight (*Phytophthora infestans*). *American Journal of Potato Research*, 92(3), 379–386. <https://doi.org/10.1007/s12230-015-9443-y>
- [37]. VAUTERIN, L., HOSTE, B., KERSTERS, K., & SWINGS, J. (1995, July 1). Reclassification of *Xanthomonas*. *International Journal of Systematic Bacteriology*, 45(3), 472–489. <https://doi.org/10.1099/00207713-45-3-472>
- [38]. Özer Çalış. (2011, December 12). Genetic analysis of resistance to early blight disease in tomato. *AFRICAN JOURNAL OF BIOTECHNOLOGY*, 10(79). <https://doi.org/10.5897/ajb11.1872>
- [39]. FRY, W. (2008, February 24). *Phytophthora infestans*: the plant (andRgene) destroyer. *Molecular Plant Pathology*, 9(3), 385–402. <https://doi.org/10.1111/j.1364-3703.2007.00465.x>
- [40]. Hartman, G. L. (1992). Black Leaf Mold Development and Its Effect on Tomato Yield. *Plant Disease*, 76(5), 462. <https://doi.org/10.1094/pd-76-0462>
- [41]. Parker, S. K., Nutter, F. W., & Gleason, M. L. (1997, March). Directional Spread of *Septoria* Leaf Spot in Tomato Rows. *Plant Disease*, 81(3), 272–276. <https://doi.org/10.1094/pdis.1997.81.3.272>
- [42]. Two-spotted Spider Mite. (2015, January 13). Center for Agriculture, Food, and the Environment. <https://ag.umass.edu/vegetable/fact-sheets/two-spotted-spider-mite>
- [43]. Kennedy, G. G. (2003, January). Tomato, Pests, Parasitoids, and Predators: Tritrophic Interactions Involving the Genus *Lycopersicon*. *Annual Review of Entomology*, 48(1), 51–72. <https://doi.org/10.1146/annurev.ento.48.091801.112733>
- [44]. Navas-Castillo, J., Sánchez-Campos, S., Noris, E., Louro, D., Accotto, G. P., & Moriones, E. (2000, November 1). Natural recombination between Tomato yellow leaf curl virus-Is and Tomato leaf curl virus. *Journal of General Virology*, 81(11), 2797–2801. <https://doi.org/10.1099/0022-1317-81-11-2797>