

# Plant Disease Classification using Deep Learning for Agricultural Applications

<sup>1</sup>K.R.Jansi, <sup>2</sup>A.L.Amutha, <sup>3</sup>A.Bhavani Shankar, <sup>4</sup>Adesh JP and <sup>5</sup>Krishna Kant Pandey

<sup>1</sup>DEPARTMENT OF COMPUTING TECHNOLOGY

<sup>2,3,4,5</sup>DEPARTMENT OF COMPUTATIONAL INTELLIGENCE

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM University, SRM Nagar, Kattankulathur -  
603 203 Chengalpattu District, Tamil Nadu, India

**Abstract** The role of agriculture in human life is crucial. Plant diseases seriously compromise food safety and agricultural productivity. Conventional procedures can diagnose and determine plant diseases, but they can be laborious and draining. Due to their complex nature and difficulties in detection, plant diseases can seriously injure plants. We can use in-depth research to classify plant diseases, address issues with traditional approaches, detect maladies early, and increase dependability. Through precise diagnosis, planning, and application of preventive measures against plant disease, this research aims to improve agriculture and ensure food security. This will support global efforts to produce food in an environmentally friendly way. This research utilizes Convolutional Neural Networks (CNN) based pre-trained models for categorizing plant disease symptoms. The novelty of this work lies in its ability to classify multiple plant diseases simultaneously using CNN, whereas currently available technologies for plant disease classification are limited to predicting one disease at a time. The proposed work focuses on fine-tuning the hyperparameters of CNN and DenseNet169 models. We train the deep learning models on the Plant Village dataset. After training, we thoroughly test the models to validate the results, using a 10% sample from the Plant Village dataset that contains images of both healthy and diseased plants. The proposed system achieved 99.61% accuracy using DenseNet169 architecture when identifying and classifying pepper bell, potato and tomato diseases, demonstrating the effectiveness of deep learning in plant disease classification.

**Keywords:** DenseNet169, fine-tuning of hyperparameters, trained systems, prediction and evaluation, metrics, matrix.

## 1. Introduction

Recent years have seen a dramatic increase in agricultural difficulties due to a combination of factors, including sudden climatic shifts and a deficiency of crop immunity. This destroys crops on a large scale, lowers civilization, and causes farmers to lose money. As per the Food and Agricultural Organization of the United Nations (FAO) [1], 45% of food that is wasted globally is attributed to plant diseases. This circumstance cost the global economy around 230 billion dollars in lost revenue. Thus, plant diseases represent a significant threat to the world's food security, and early identification is essential to prevent disastrous crop loss.

Identification and treatment of plant diseases have become extremely difficult due to the rapid expansion of the diversity of conditions and the acceptable knowledge of farmers [2]. Conventional methods can determine and classify plant diseases, but they can be challenging and costly. The use of deep learning algorithms for autonomous disease classification has grown in popularity in recent years [3]. Deep learning involves training artificial neural networks to learn from vast amounts of data. One class of deep learning technique that has proven particularly useful in image recognition applications, such as the diagnosis of plant diseases, is “Convolutional Neural Networks” (CNNs). One technique to categorize ailments into distinct groups is to train CNN on massive sample sets of pictures of healthy and ailing leaves. CNN training can identify fresh leaf images based on the presence or absence of specific ailments. There are several advantages to implementing deep learning techniques for plant disease classification, such as handling large and complex sample sets and achieving precise ailment identification and categorization. Deep learning-based plant disease classification possesses a mandate to improve farming practices and consumer security while revolutionizing plant disease research. Researchers widely employ deep learning techniques like CNN [4], RNN [5], and LSTM [6] to detect bacterial spots. CNN's strong performance on image recognition tasks has led to the creation of accurate plant disease classification systems.

Researchers have recommended several techniques to accurately detect and classify plant infections. Some use traditional image processing techniques that incorporate hand-crafted—that is, manual—feature extraction and segmentation. Johannes et al. [7] created a novel image reconstruction technique that combines hot spot detection and measurement techniques to address the issue of disease identification via nature's surroundings. The study focused on the initial detection of three distinct ailments of crops that are common in the European Union: brown spot, rust, and scab. During the period of 2014 to 2016, nearly 3,500 samples and photos from Germany and Spain were analyzed using seven mobile phones. Mondal et al. [8] presented an efficient technique to detect and classify the presence of Yellow Vein Mosaic Virus disease in okra leaf with the joint use of image processing, K-means and Naive Bayesian classifier. The recommended technique was experimented with 79 standard diseased and non-diseased okra leaf images and achieved 87% success rate. Tejonidhi et al. [9] used Bhattacharya's similarity calculation to compare sick paddy plants to healthy ones. The authors focused on two diseases, namely leaf blast and leaf blight. First, the leaves are classified into healthy and diseased ones. The authors used Bhattacharya's similarity calculation method for finding similarity in the histogram of a test image or sample image with respect to a clinically proven healthy image (standard image). During the training phase, 100 sample images of healthy and diseased leaves are used to obtain standard values, which represent the respective types based on which type of test leaf is detected. In [10], the authors adapted a different kind of segmentation called indices-based histogram for getting the plant disease attributes based on the leaf images. The dataset used was collected from Plant Village, Kaggle, and Mendeley datasets with different plant leaf images that include different shape, margin, and texture features for identifying the disease. The collected dataset was sliced with train and test data classifiers and processed. Segmentation performance using the indices-based histogram approach resulted in 92.06% accuracy. In [11], the authors used image processing techniques to detect plant leaf diseases. The objective of their work was to implement image analysis and classification techniques for the detection of leaf diseases and classification. The recommended framework consists of four parts. They were: (1) image preprocessing; (2) segmentation of the leaf using K-means clustering to determine the diseased areas; (3) feature extraction; and (4) classification of diseases. Texture

features were extracted using statistical Gray-Level Co-Occurrence Matrix (GLCM) features and classification was done using Support Vector Machines (SVM).

With the advancement of machine learning techniques, few authors have used machine learning to identify and classify diseased plants from healthy ones. Ernest et al. suggested an automatic vision-based [12] diagnosis of Banana Black Sigatoka (BBS) disease and Banana Bacterial Wilt (BBW) disease using computer vision-based techniques. The researchers used seven classifiers to classify the banana leaves as either healthy or diseased. Out of the seven classifiers, extremely randomized trees performed best in identifying the diseases, achieving 0.96 area under the curve for BBW and 0.91 for BBS. Krithika et al. [13] suggested a method based on K-means clustering and support vector machines to detect diseases present in the leaves of salad cucumber. The most common diseases that are present in salad cucumbers are *Alternaria* leaf blight, bacterial wilt, cucumber green mottle mosaic, leaf miner, leaf spot, Cucumber Mosaic Virus (CMV) disease and so on. Iniyan et al. [14] suggested machine learning techniques, especially support vector machines and artificial neural networks, to detect plant diseases. Their research work concluded with the pros and cons of every method in context with input parameters (Crop type). The authors suggested a series of procedures to detect whether a leaf is sick or healthy. These steps include data cleansing, feature extraction, classifier training, and classification. A classification of pomegranates was proposed in [15] by combining a supervised learning model with Support Vector Machine. Colour and texture were employed as characteristics to segment the area where the defect is most apparent. A Support Vector Machine classifier performed the categorization in this case. The authors [16] suggested a tea leaf disease recognizer (TLDR) to recognize diseases of the tea leaf. In TLDR, at first the image of the tea leaf was cropped, resized and converted to its threshold value in the image processing. Then the feature extraction method was applied and Neural Network Ensemble (NNE) was used for pattern recognition. The extracted features were passed to the ANN along with the disease type and the ANN was trained. When a new image was uploaded into the system, the most suitable match was found and the disease was returned. After going through the testing process, 91% accuracy was found. Some researchers have used pattern recognition methods to detect plant diseases. Pattern recognition methods [17] for cotton leaf disease detection used snake segmentation; in this case, Hu's moments served as a defining feature. With an active contour model to reduce vitality within the infected region, the BPNN classifier handled a wide variety of classification issues. The BPNN classifier correctly classified 85.52% of cases on average.

In recent years, with the successful application of deep learning models represented by CNN, several plant disease detection methods based on deep learning have been applied in real agricultural practice. Barbedo et al. [18] conducted a study to understand the impact of sample set size and variety on the effectiveness of deep learning and transfer learning for flora ailment classification. The study used an image sample containing samples of 12 distinct floral species, each displaying different conditions and ailments. Ferentinos et al. [19] trained a deep learning system by using a publicly available sample consisting of 87,848 photos of 25 distinct Flora species, separated into 58 states of health and illness. It has been demonstrated that the top-performing systems are successful in determining the appropriate [flora, disease] pairings. Mohanty and others [20] used a special set of 54,306 pictures of healthy and infected plants that were being watched to use a deep neural network CNN to find 26 different diseases in 14 different crops. Additionally, J. Anita and colleagues [21] carried out comparison research using the CNN

method and multiple pretrained transfer learning systems for plant disease classification using the FloraVillage sample set. Three families—FR-CNN, R-CNN and SSD—were examined by Akila et al. [22] to identify a variety of ailments, particularly in complex environments. E.C. Too et al. [23] used 38 sample types containing the health and ailments of 14 Flora species to compare various transfer learning systems, including VGG\_16, Inception\_V4, DenseNet\_121 and ResNet\_Xception. Atila and colleagues [24] recommended the EfficientNet system for plant disease classification and conducted a comparison with other advanced transfer learning systems previously studied. They did this using the FloraVillage sample set. Sladojevic et al. [25] successfully developed a CNN system that identified 13 distinct floral ailments in otherwise healthy leaves. The authors [26] implemented many deep learning architectures like Resnet50, Xception, MobileNet, ShuffleNet and Densenet121\_Xception for classifying healthy and diseased tomato leaves. Nine types of diseased leaves, including healthy tomato leaves, are classified. These five deep network structures with different learning rates were compared in the experiment. The best recognition accuracy of Densenet\_Xception is 97.10%. Additionally, Singh et al. [27] classified the image into five groups using DenseNet169, a pre-trained system, to identify and categorize the ailment of leaves. They trained and assessed the system using a sample base containing 21,397 photos of various configurations of cassava leaves.

The rationale behind this study is that all previous research in the field has focused on detecting diseases in individual plants. There is currently no established model for identifying diseases in plants of any type. However, it is important to perform the same action. The aim of this project is to construct a model capable of identifying diseases in various types of plants. The objective of this study is to identify and categorize diseases in various types of plants by utilizing the CNN and DenseNet169 transfer learning models. To optimize the results, the DenseNet169 system has eight additional high-quality layers. The values of the error rate, cost function, and performance array indicate the system's capacity. Here, we provide an overview of the structures for each individual component. The second part of the document focuses on the methodology. Sections 3 and 4 of the document analyse the findings and significance of the research.

## **2. Plant disease classification using CNN and DenseNet 169**

The proposed method classifies the healthy leaves from the diseased ones that belong to any variety of plants using CNN and DenseNet 169. Figure 1 illustrates the overall architecture of the proposed method. As shown in the figure, the first and foremost step is image acquisition, followed by preprocessing, which involves resizing, rotation, normalization and image enhancement. Following preprocessing, we divide the process of plant disease classification using deep learning methods like CNN and DenseNet 169 into two phases: training and testing. During the training phase, we train the CNN model using 80% of the training dataset. Once the model is trained, during the testing phase, the model is loaded with images from the testing set to check whether the images are classified correctly or not. On top of this CNN model, by incorporating transfer learning, we built the DenseNet 169 model to detect plant diseases and classify them. The performance of the CNN model and DenseNet 169 model is evaluated using accuracy, precision, recall and F1 score.

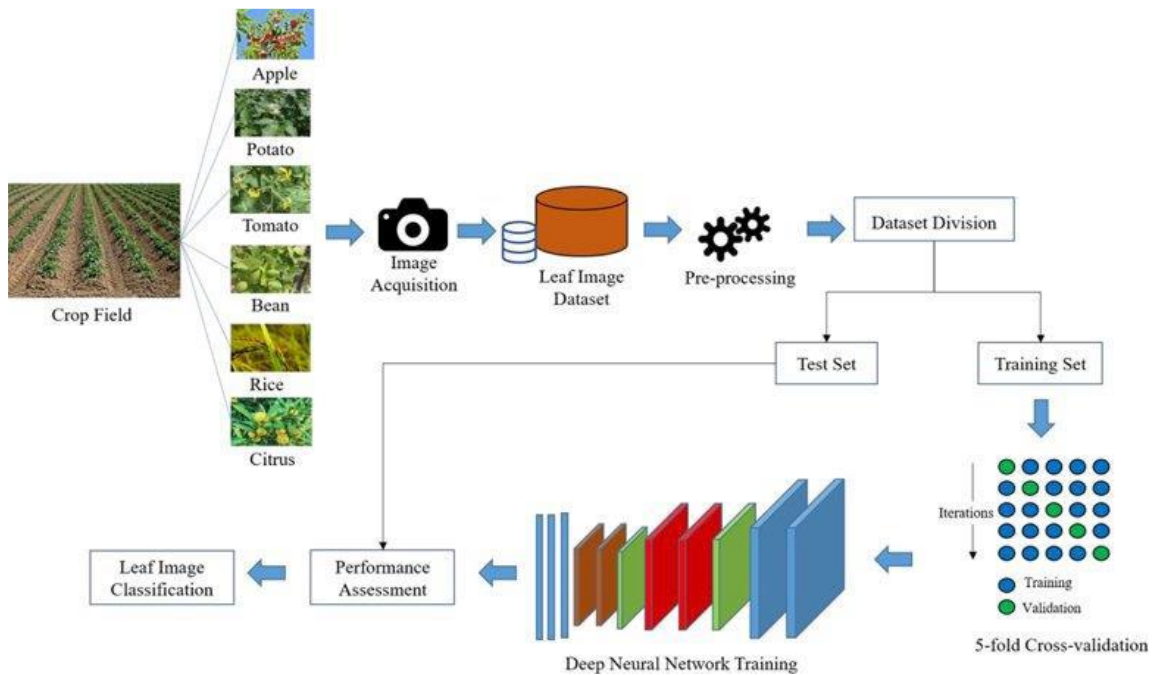


Figure 1: Overall Architecture of Proposed Plant Disease Classification Model

## 2.1 Plant disease classification using CNN

Convolutional neural networks (CNNs) have revolutionized the field of image analysis by providing an automated solution for complex image processing, classification, and segmentation tasks. CNNs use multiple layers of neurons that perform convolution operations, which allow them to extract and identify specific features in images. CNNs are extremely valuable for a variety of applications, as they can use these features to classify or segment images. Here, we use CNNs to detect diseases in a variety of plants. To train a model for detecting plant diseases using a CNN, several steps are involved. First, the dataset of plant images needs to be preprocessed. We preprocess the images by resizing and cropping them, and then convert them into an array format that CNN can utilize. We apply the same preprocessing steps to the test image as well. The CNN should train on a dataset that contains hundreds of images of various plant diseases. These images serve as test cases for the program and help the model learn to identify the specific features that indicate the presence of a particular disease. Several layers, including dense, dropout, activation, flatten, convolution2D, and maxpooling2D, make up the CNN network. Each of these layers performs a specific function, such as reducing the size of the input image or extracting important features from the image. After successfully training on the dataset, the model can identify various plant diseases in new images. We compare the test image with the learned model to detect the presence of any specific disease. By following these steps, a CNN can accurately identify plant diseases, providing a valuable tool for farmers and others involved in agriculture. Figure 2 shows the steps involved in CNN working.

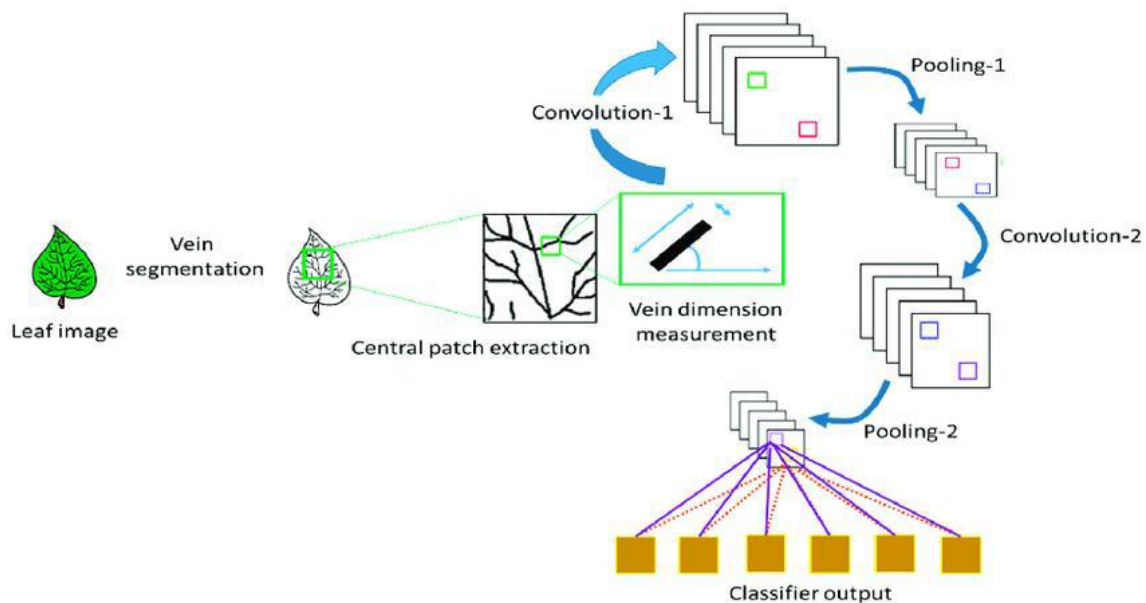


Figure 2: Architecture of Proposed CNN Model

- **Convolution:** We feed images here and produce an activation map by scanning the images several pixels at a time using a filter.
- **Pooling:** After feature maps are extracted, we try to reduce the size of the feature maps, that is, compress them, which is down sampling.
- **Flattening:** We flatten all the feature maps after pooling them. Here, we consider the 2D pixels and convert them into 1D, then feed them to our dense, fully connected artificial neural network and train the model.
- **Padding:** The technique preserves the spatial dimensions of the input image following convolution operations on a feature map. Padding involves adding extra pixels around the border of the input feature map before convolution.
- **Normalization:** The term "normalization" refers to the process of removing the mean and subtracting the standard deviation. Often, it is crucial that the input features center around zero and have the same order and variance. Images, for example, have their data scaled so that the values range from 0 to 1 by dividing each pixel's value by 255.
- **Rectified Linear Unit (ReLU):** A Rectified Linear Unit is used as a non-linear activation function. A ReLU says if the value is less than zero, round it up to zero.
- **Strides:** If the input has width and height and the kernel iterates over each pixel in turn, the resulting output will also have width and height. The output channel/feature map size can be reduced by half by using a stride-two convolution instead of a stride-one convolution, in which the convolutional kernel takes strides of one pixel. Since the kernel only generates a single, aggregated output for each stride, the maximum value that could be produced as input with width  $w$ , height  $h$ , and depth 3 with padding would be width  $w/2$ , height  $h/2$ , and depth 1.
- **Batch Normalization:** Training times can be cut in half and accuracy can be improved by a magnitude when using batch normalization to stabilize the predictions made by a network. By removing the average batch's activations and dividing the standard variation of the batch's activations. However, even after normalizing the input, some activations may still be larger than average, rendering the network less stable. By applying a modification called batch normalization,

we can keep the return confidence interval around 1 and the mean output close to 0. It's important to note that the way batch normalization functions vary between the training and inference phases.

- **Input shape:** In the context of deep learning, the input shape refers to the size and dimensions of the data that is fed into the model for training. For images, the input shape typically consists of the number of pixels in the image and the number of color channels.
- **Output shape:** The output shape of a deep learning model refers to the size and dimensions of the data that is produced by the model after processing the input data. For image classification tasks, the output shape typically consists of the number of classes or categories that the model is trained to recognize.
- **Epochs:** In deep learning, an epoch refers to a single iteration through the entire training dataset. The number of epochs is a hyperparameter that is set by the user and determines how many times the model will be trained on the entire dataset. The layers utilised in the suggested system are listed in Table 1.

Table 1. Trainable and Non-trainable Parameters

Layer	Output Shape	Parameters (#)
Sequential	(32,256,256,3)	0
Conv2D	(32,254,254,32)	896
Max_pooling2d	(32,127,127,32)	0
Conv2D_1	(32,125,125,64)	18496
Max_pooling2d_1	(32,62,62,64)	0
Conv2D_2	(32,60,60,64)	36928
Max_pooling2d_2	(32,30,30,64)	0
Conv2D_3	(32,28,28,64)	36928
Max_pooling2d_3	(32,14,14,64)	0
Conv2D_4	(32,12,12,64)	36928
Max_pooling2d_4	(32,6,6,64)	0
Conv2D_5	(32,4,4,64)	36928
Max_pooling2D_5	(32,2,2,64)	0
Flatten	(32,256)	0
Dense	(32, 64)	16448
Dense_1	(32, 21)	1885

## 2.2 Plant disease classification using DenseNet 169

DenseNet169, a pre-trained transfer learning system, is built on top of the "CNN" model created from the plant village dataset. Rather than creating a deep learning model from scratch, we applied the transfer learning technique to achieve better and more refined outcomes. Figure 3 illustrates the desired framework for this research. The next sections provide a full explanation of the technique.

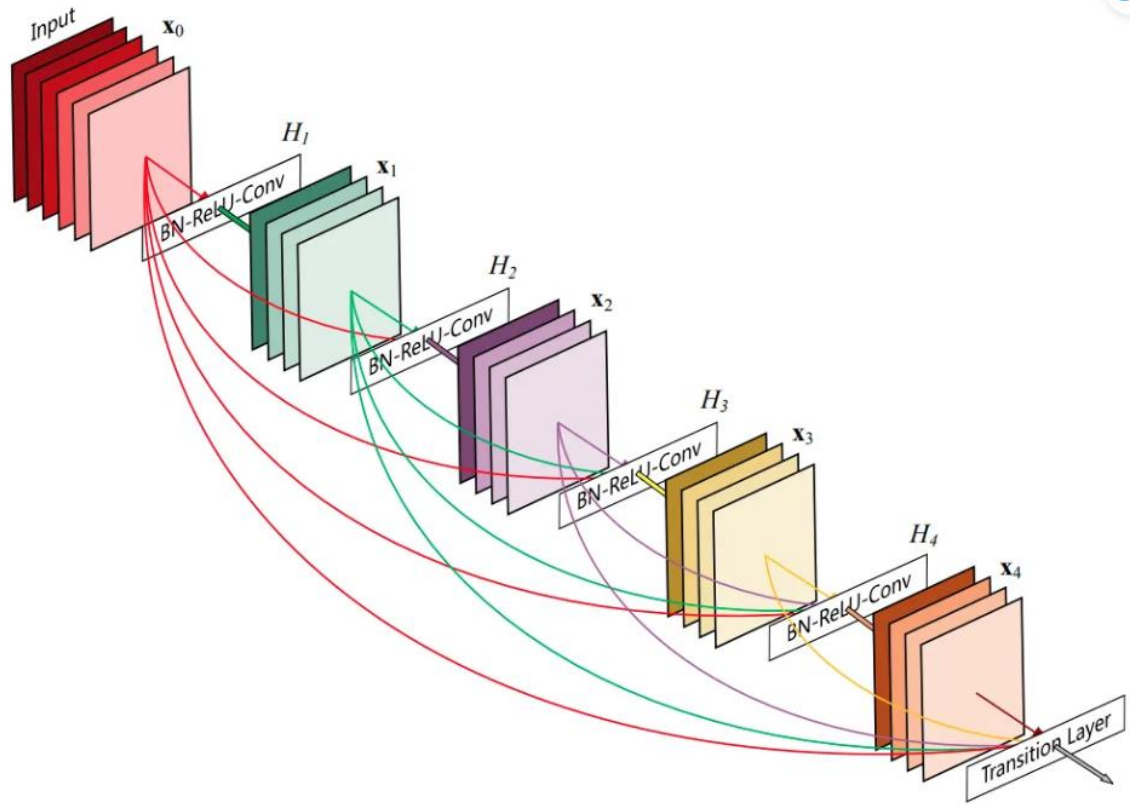


Figure 3: Architecture of Proposed DenseNet 169 Model

### 2.2.1 Fine Tuning of Layers

"Transfer learning" is called "finetuning" when a previously trained system is retrained using a fresh, lesser sample set. Based on the updated sample, the already trained system can more accurately and efficiently complete the new task by adjusting a few parameters. Finetuning can be used if the current assignment and the newly acquired sample are both comparable in size and identical to the prior training assignment for the system that was previously trained. Finetuning can significantly reduce the period and processing power required to train a system when compared to commencing from the beginning. As finetuning techniques, this study used Conv2D\_layers, Global\_Average\_of\_Max\_Pooling, Batch\_Normalisation\_of\_layers, Flattening\_layer, Drop-out\_k\_layers, and Dense\_y\_layers. An integral component of a CNN, the Conv2D\_layer applies a convolution process to the source image or characteristic map. To acquire regional characteristics from the source image, a series of programmable filters or kernels are utilised in the Conv2D\_layer. The sample map is a 3D tensor containing characteristics that were taken from the source image and is an attribute of the Conv2D\_layer. Before the distinctive map created by the convolutional layer is forwarded to the entire layer of connections for classification, Global\_Average\_Pooling ("GAVP") is employed to reduce its size. The provided map's length and breadth are combined in the "GAVP" to yield an identical value for each of the channels. Consequently, the dimensionality of the plot of features decreases to "1\*1\*M", where 'M' represents number of channels. Regression or classification can be performed instantly on the output by feeding it into a fully integrated system. "GAVP" is superior to other approaches like maximum pooling in many ways. Since the network must learn multiple representation strategies, it is first less disruptive. Second, by lowering the



number of system parameters, overfitting is less likely to occur and performance can be enhanced. Lastly, regardless of where an object is in the image, “GAVP” always produces the same result because it is translation independent. Batch normalisation is the process of normalising the activation of layers by transforming and scaling the results to have a set of averages and different units. In order to normalise performance, batch normalisation determines the "standard deviation" and "mean" of performance benchmarks over the collection of samples during training. This assists in keeping values to not turn excessively large or small, that could hinder learning or cause the network to become stranded on the best possible outcomes. The benefits of batch normalisation include decreased precision of the initial weight value, enhanced stability, and increased training efficiency. Furthermore, batch normalisation can be applied consistently to lessen overfitting and enhance the performance of the system. Regular dropout techniques are used by deep learning neural networks to avoid overfitting. Each training cycle, a portion of the input/output elements are divided by nil during the release process, preventing the network from being swamped or overloaded by an input/output unit. Among the many benefits of the dropout layer are improved generalisation, reduced noise thresholds and increased resistance to variations in the input sample. As a form of collective learning, dropout can also be viewed because of the numerous "optimised" networks that is integrated into the system after being taught during learning. Dense\_Layers are used for regression and hierarchical functions. They are also referred to as “Dense”. Each nerve within the dense\_layer is connected to all the remaining nerves in the layer below it. The thick layer's inputs are feature vectors, or matrices, and its outputs are activation vectors. Activation is determined by using a non-linear activation function after the inputs have been weighted. Gradient descent can be used to learn the weight and slope of layers during training and vice versa. One hyperparameter that is frequently adjusted to enhance system performance is the thick layer's neuronal density. Adding more neurons to the system increases their potential to acquire information about the representation of the incoming information, but they also increase the risk of overfitting and poor learning. The layers utilised in the suggested system are listed in Table 2.

Table 2. Trainable and Non-trainable Parameters

Layer	Output Shape	Parameters (#)
Sequential	(32,256,256,3)	0
Conv2D	(32,254,254,32)	896
Max_pooling2d	(32,127,127,32)	0
Conv2D_1	(32,125,125,64)	18496
Max_pooling2d_1	(32,62,62,64)	0
Conv2D_2	(32,60,60,64)	36928
Max_pooling2d_2	(32,30,30,64)	0
Conv2D_3	(32,28,28,64)	36928
Max_pooling2d_3	(32,14,14,64)	0
Conv2D_4	(32,12,12,64)	36928
Max_pooling2d_4	(32,6,6,64)	0
Conv2D_5	(32,4,4,64)	36928
Max_pooling2D_5	(32,2,2,64)	0
Flatten	(32,256)	0
Dense	(32, 64)	16448
Dense_1	(32, 21)	1885

### 3. Experimental Results

#### 3.1 Dataset Description

Plant disease classification is carried by using two different data sets. The first data set has 15 categories, whereas the second has 38. Multiple photographs of each plant are included in both database8s. This work concludes on the village plant dataset, which includes 38 types of plants. Figure 4 shows the sample healthy and diseased leaves from the dataset.

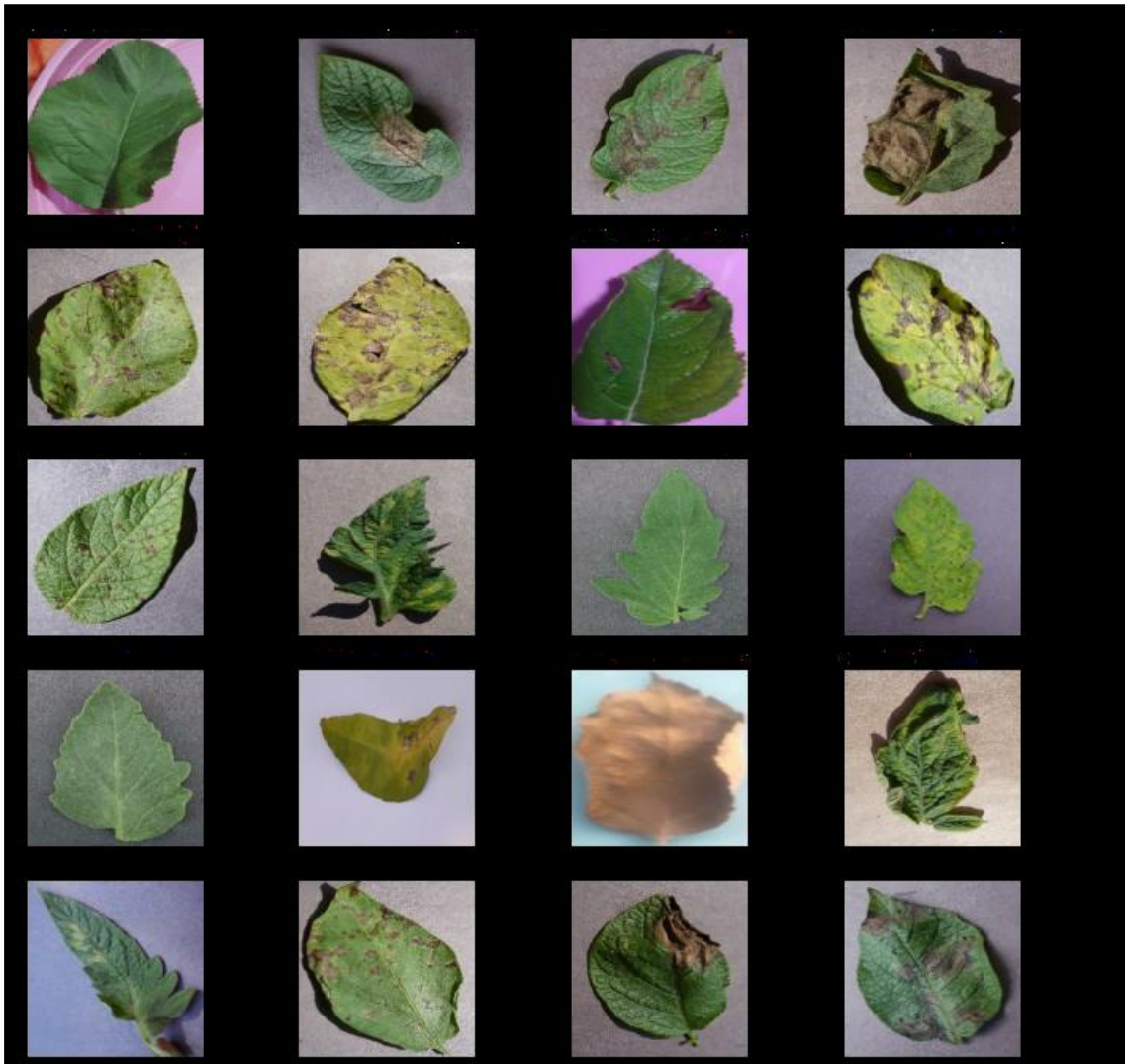


Figure 4: Sample healthy and diseased leaves from the dataset

This dataset contains approximately 6237 images which are further classified into 4865 for training images and 1372 for testing images, which are then classified into 21 classes as shown in Table 3. Table 3 depicts the variety of plant diseases with their description.

Table 3: Variety of plant diseases and their description

<b>S. No</b>	<b>Plant Disease Name</b>	<b>Description</b>
1	Early Blight	The plant which is at starting stage of the disease.
2	Late Blight	The plant which is at ending stage of the disease.
3	Healthy	The plant which is healthy and have no symptoms of disease
4	Apple leaf blotch	Apple leaf contains dark circular green patches which is caused by bacteria.
5	Apple Scab leaves	Apple scab is a common fungal disease that affects apple trees. It causes dark, scabby lesions to form on the leaves, which can ultimately cause the leaves to yellow, wither, and drop prematurely
6	Apple rot leaves	Apple rot is a disease that affects apple trees, causing the leaves to rot and decay. This disease is typically caused by fungal pathogens that can infect the leaves, as well as the fruit and bark of the tree.
7	Citrus scab	affects citrus trees, including oranges, lemons, limes, grapefruits, and tangerines. The disease is caused by the fungus <i>Elsinoe fawcettii</i> , which infects the leaves, fruit, and twigs of the tree
8	Citrus Canker	is a bacterial disease that affects citrus trees. It causes characteristic lesions on the leaves, stems, and fruit of infected trees.
9	Citrus tristeza	Citrus tristeza is a viral disease that affects citrus trees and is caused by the Citrus tristeza virus (CTV).
10	Citrus greening	is a bacterial disease that affects citrus trees, including oranges, lemons, limes, grapefruits, and tangerines. The disease is caused by the bacteria <i>Candidatus Liberibacter</i> .

11	Rice bacterial leaf blight	is a bacterial disease that affects rice plants. Symptoms of the disease include water-soaked, yellowish-green, or grayish-white spots on the leaves, which can turn brown and necrotic as the disease progresses.
12	Rice brown	Symptoms of the disease include circular to oval-shaped lesions on the leaves that are brown or reddish-brown in color& yellow halos around them
13	Rice leaf smut	Symptoms of the disease include elongated, gray-green, or yellow green lesions on the leaves that can expand and merge over time. Infected leaves may also become twisted or distorted.
14	Tomato bacterial spot	Symptoms of the disease include small, water-soaked lesions on the leaves, stems, and fruit that can turn brown or black and necrotic. Infected fruit may also exhibit scarring or cracking.
15	Tomato leaf mold	Symptoms of the disease include yellowing and wilting of the lower leaves, followed by the appearance of velvety, grayish-brown mold on the upper surface of the leaves.
16	Tomato Septoria leaf spot	Symptoms of the disease include small, circular lesions on the leaves that are grayish brown in color and have a yellow halo around them.
17	Tomato spider mites	Symptoms of the infestation include yellowing or bronzing of the leaves, webbing on the undersides of the leaves, and a decline in plant health.
18	Tomato target spot	Symptoms of the disease include circular lesions on the leaves that are grayish brown in color with concentric rings or a "target" pattern
19	Tomato mosaic virus	Symptoms of the disease include mosaic-like patterns of yellowing and discoloration on the leaves, as well as stunted growth, distorted fruit, and reduced yields.
20	Bean angular leaf spot	Bean angular leaf spot is a bacterial disease that affects bean plants. Symptoms of the disease include

		angular, water-soaked lesions on the leaves that can turn brown and necrotic over time.
21	Bean rust	Bean rust is a fungal disease that affects bean plants. Symptoms of the disease include reddish-brown pustules on the undersides of the leaves that can release spores.

### 3.2 Results of Preprocessing

The captured images cannot be directly inputted into the Convolutional Neural Network. Therefore, the images in the dataset are transformed into arrays of size 256 x 256, where each block represents the pixel value. The obtained images may contain noise and other distortions that are suboptimal for image processing. The images have been preprocessed. Preprocessing encompasses several approaches, including resizing, rescaling, rotation, normalization, and image enhancement, which are employed to boost the efficiency of the model. Figure 5 displays the outcome of the preprocessing procedure.

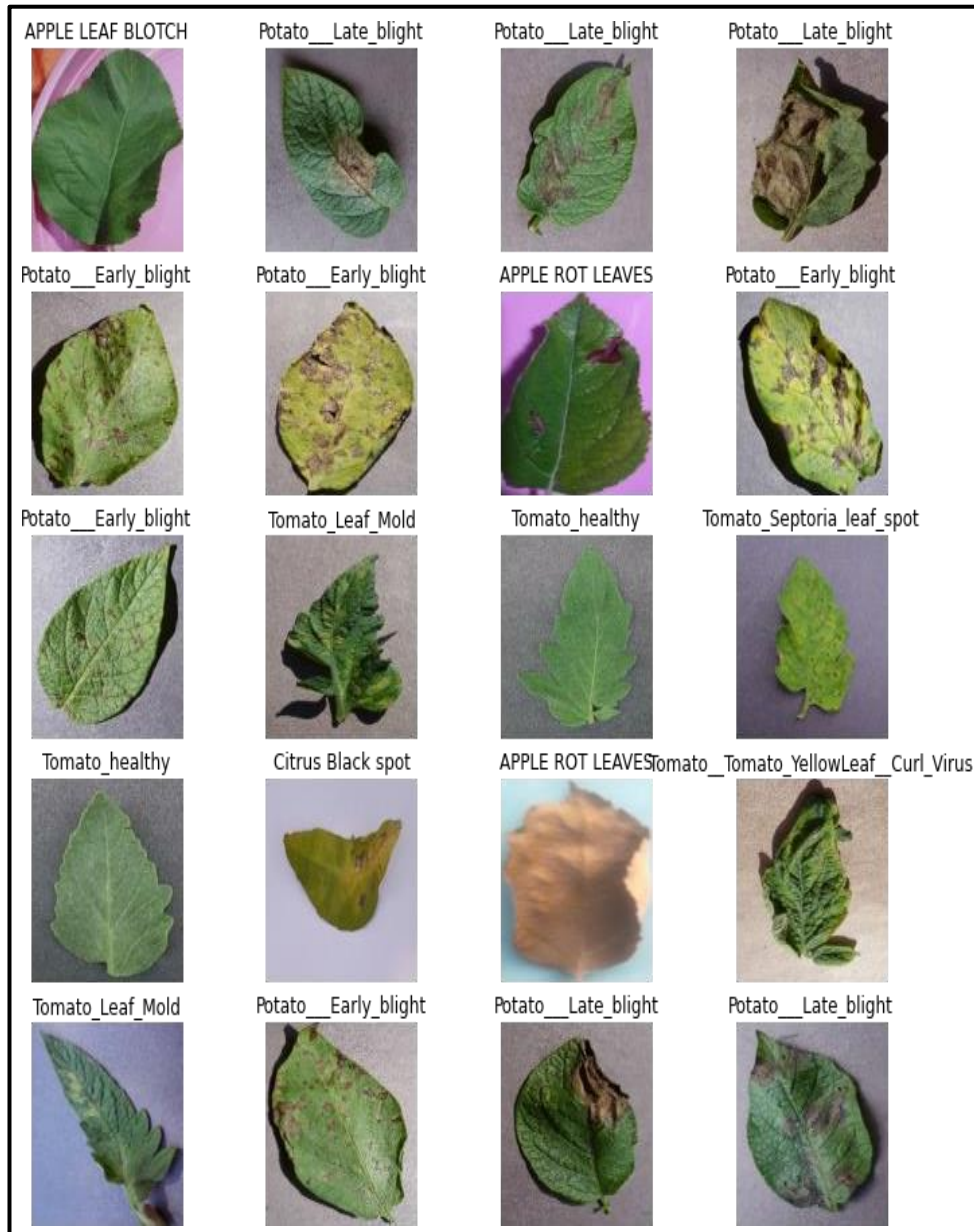


Figure 5: Result of preprocessing

### 3.3 Results using the CNN model

After preprocessing, the images in the training set are used to train the CNN model. The training process involves iteratively feeding batches of input data into the model and adjusting the model's weights and biases in response to the errors between the model's predictions and the ground truth labels. This is done using an optimization algorithm called stochastic gradient descent. After the images are processed into proper format and divided into segments, the next feature extraction is a crucial process. In Convolutional Neural Networks, this is accomplished with the aid of activation function and then max-pooling layers. This method helps in reducing the size of data from the dataset. It also helps in building the model with minimal machine effort and improves the learning speed in the machine learning process. Once the model is trained, it is evaluated on a validation dataset to measure its performance. The metrics used for evaluation

include accuracy, precision, recall and F1-score. The training versus validation loss graph provides a clear and insightful view of how much loss occurs to the model during the training and validation stages. It is a dynamic representation that showcases the loss of the model as it trains on the training dataset and evaluate on the validation dataset. Figure 6 shows the training versus validation accuracy and loss graph of selected plant disease. In the training and validation loss graph, number of epochs forms the x-axis and loss forms the y-axis. In the training and validation accuracy graph, number of epochs forms the x-axis and accuracy forms the y-axis.

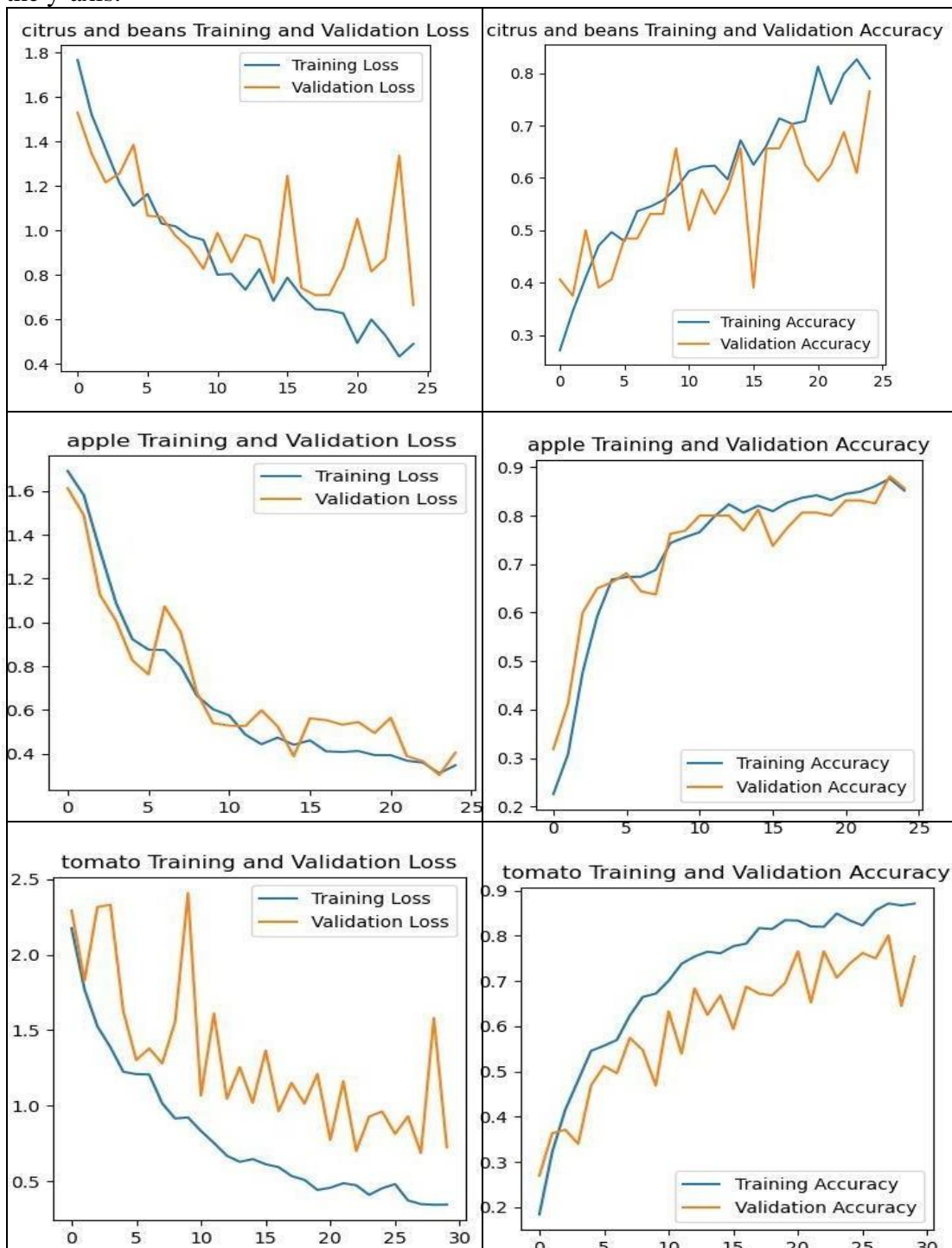


Figure 6: Training versus validation loss and accuracy graphs of CNN model



Finally, the model is tested on a separate, unseen test dataset to assess its generalization performance. The performance of the model on the test dataset is used to determine its effectiveness in classifying plant diseases. The accuracy of the built model is observed to be 96.83%. Figure 7 shows the final result of plant disease classification using CNN model.

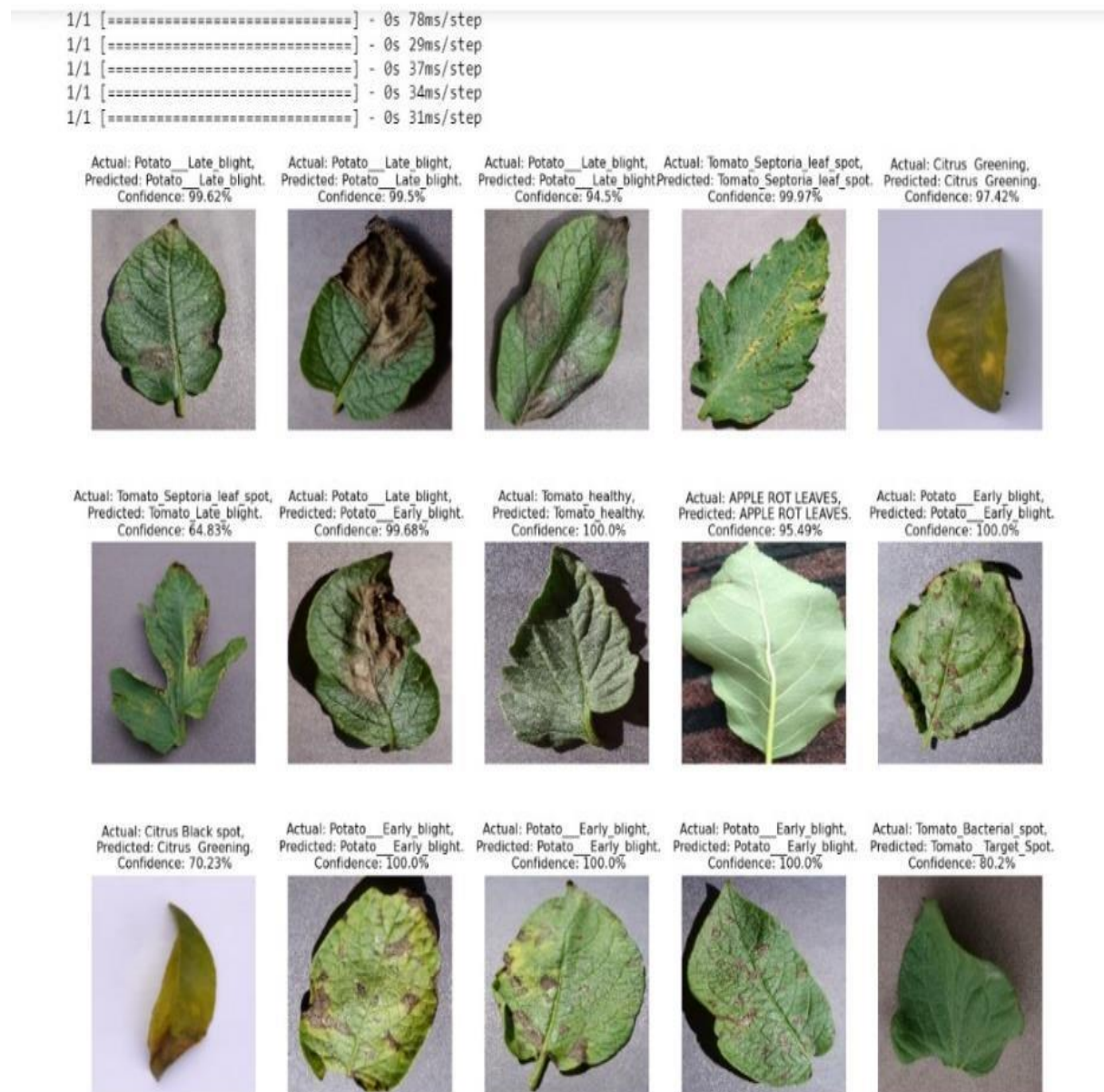


Figure 7: Final result of plant disease classification using CNN model



### 3.4 Results using DenseNet 169 model

By applying fine-tuning layers to the input sample set, it becomes possible to diagnose and categorize different diseases in various types of plants. The DenseNet169 transfer learning system yields 21 unique classes through this approach. The proposed system is executed for a total of 50 iterations. The results are calculated based on the parameters of the Cost\_Function\_Values and Error\_Rate\_Values. Figure 8 illustrates the Cost\_Function observed during the training and testing of the Dense-Net169 system with finetuning layers. The initial cost function of training for the DenseNet169 transfer learning system starts at a high value and gradually decreases over time without enduring significant fluctuations, as observed from the cost function graphs. Eventually, it reaches a value of 0.0278. The validation Cost\_Function, which ultimately amounts to 0.0836, remains exceedingly small but exhibits increasingly noticeable changes over time. Figure 8 illustrates the Error\_Rate achieved during testing and training for the DenseNet169 system, which is implemented using finetuning layers.

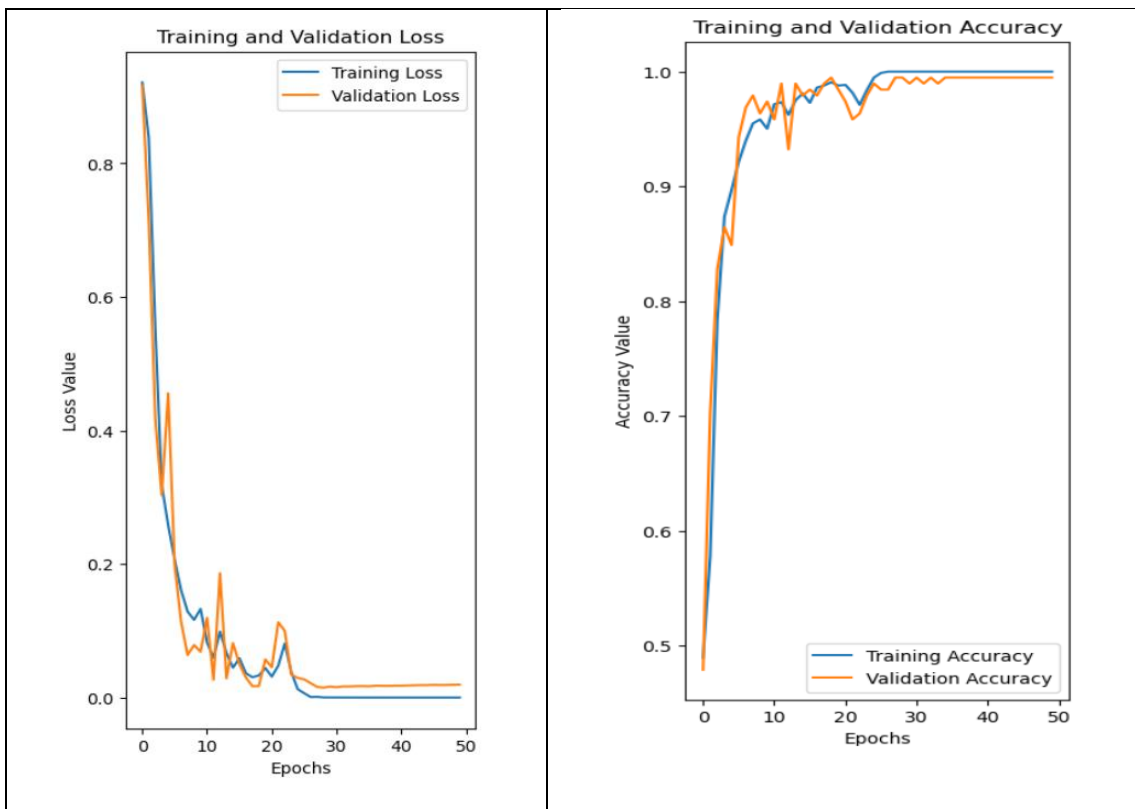


Figure 8: Training versus validation loss and accuracy graphs of DenseNet 169 model

After training the model, the next step is to test the DenseNet 169 model with testing dataset. What happens here is that the DenseNet 169 model when given a healthy or diseased leaf image and actual label as the input shows the predicted label as the output as shown in figure 9.

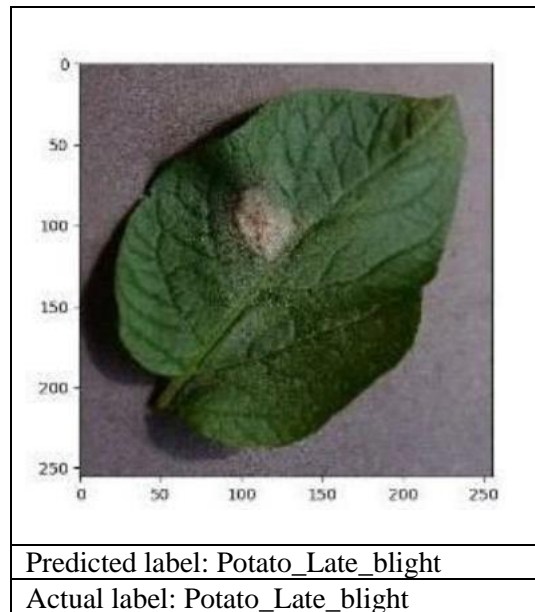


Figure 9: Result of DenseNet 169 model

The accuracy of the built model is observed to be 99.61%. The results of the proposed DenseNet 169 model is compared to the existing model. Table 4 shows the comparison with the existing work.

Reference No	Classifiers used	Accuracy
S.P.Mohanty [5]	AlexNet and GoogleNet	99.35%
K.P.Ferentinos [7]	GoogleNet	99.53%
M.Akila [10]	VGG-FCN-VD16	97.95%
E.C.Too [11]	DenseNet121	99.75%
Ü. Atila [12]	B4, B5 EfficientNet System	99.97%
S. Sladojevic [13]	Deep “CNN”	96.3%
B. Chandra Kar-mokar [14]	NCL Technique	91%
R.Singh [16]	Efficient-NetB3	94%
V.Anand [19]	UNet and“CNN”	97.61%
H.Hong [17]	Dense-Net_Xception	97.10%
Sai.Reddy [18]	Basic Dense-Net	97%
Proposed work	DenseNet 169 model	99.61%

#### 4. Conclusion and Future work

The proposed model utilizes a Convolutional Neural Network (CNN) and DenseNet 169 transfer learning system. It incorporates several computer vision techniques, such as converting RGB to gray, to extract meaningful features from the leaf samples. These properties collectively aid in the categorization of different plant leaf diseases. The objective of this project is to present a comprehensive examination of the subject matter by exploring various methodologies employed in the analysis of leaf images for the purpose of diagnosing plant diseases. The study utilizes data from many sources to tackle substantial obstacles in the application of machine learning and deep learning for the diagnosis of plant diseases. To summarize, the domain of plant disease identification employing deep learning approaches is a swiftly progressing discipline with abundant prospects for future investigation and advancement. The results of this experiment can provide valuable insights for the creation of more accurate diagnostic instruments in the future. These tools have the potential to enhance the identification and control of plant diseases, hence promoting more efficient and environmentally-friendly agricultural methods. Potential future research on this model might investigate the application of fusion approaches to extract more meaningful features and assess its performance on additional leaf samples from different datasets. One may create predictive models by utilizing feature selection and extraction techniques from the plant village dataset. These models can aid in the early detection of plant diseases.

#### References

1. FAO. (2022). WORLD FOOD AND AGRICULTURE STATISTICAL YEARBOOK 2022. FAO.
2. Demilie, W.B. Plant disease detection and classification techniques: a comparative study of the performances. *J Big Data* **11**, 5 (2024). <https://doi.org/10.1186/s40537-023-00863-9>
3. Liu, J., Wang, X. Plant diseases and pests detection based on deep learning: a review. *Plant Methods* **17**, 22 (2021). <https://doi.org/10.1186/s13007-021-00722-9>
4. Guerrero-Ibañez A, Reyes-Muñoz A. Monitoring Tomato Leaf Disease through Convolutional Neural Networks. *Electronics*. 2023; 12(1):229. <https://doi.org/10.3390/electronics12010229>.
5. Gnanasaravanan, S., Tharani, B., & Sahu, M. (2021). Long Short-Term Memory Recurrent Neural Networks for Plant disease Identification. *Plant pathology*, 3(2).
6. Türkoğlu, Muammer & Hanbay, Davut & Sengur, Abdulkadir. (2022). Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. *Journal of Ambient Intelligence and Humanized Computing*. 13. 10.1007/s12652-019-01591-w.
7. A. Johannes et al., "Automatic Flora ailment diagnosis using mobile capture devices, applied on a wheat use case," *Comput. Electron. Agric.*, vol. 138, pp. 200–209, 2017.
8. Mondal, D., Chakraborty, A., Kole, D. K., & Majumder, D. D. (2015). *Detection and classification technique of Yellow Vein Mosaic Virus disease in okra leaf*

- images using leaf vein extraction and Naive Bayesian classifier. 2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI).* doi:10.1109/icscti.2015.7489626
9. Tejonidhi, M. & Nanjesh, B.R. & Math, Jagadeesh & D'Sa, Ashwin. (2016). Plant disease analysis using histogram matching based on Bhattacharya's distance calculation. 1546-1549. 10.1109/ICEEOT.2016.7754943.
  10. Pushpa BR, Shree Hari AV, Adarsh Ashok, "Diseased Leaf Segmentation from Complex Background Using Indices Based Histogram" in the Proceedings of the 6th International Conference on Communication and Electronics Systems (ICCES-2021).
  11. R.Meena Prakash, G.P.Saraswathy, G.Ramalakshmi, K.H.Mangaleswari, T.Kaviya, "Detection of Leaf Diseases and Classification using Digital Image Processing" in the proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS).
  12. Owomugisha, Godliver & Quinn, John & Mwebaze, Ernest & Lwasa, James. (2019). Automated Vision-Based Diagnosis of Banana Bacterial Wilt Disease and Black Sigatoka Disease in the proceedings of 1st International conference on the use of mobile ICT in AfricaAt: 9-10th December 2014, Stellenbosch, South Africa
  13. P. Krithika and S. Veni, "Leaf disease detection on cucumber leaves using multiclass Support Vector Machine," *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2017, pp. 1276-1281, doi: 10.1109/WiSPNET.2017.8299969.
  14. Iniyan, S. Jebakumar, R. Mangalraj, P. Mohit, Mayank and Nanda, Aroop," Plant Disease Identification and Detection Using Support Vector Machines and Artificial Neural Networks", in *Advances in intelligent systems and computing*.2020.pp15-27. Springer, Singapore.
  15. Mangena, Venu & Thanh, Dang & Khamparia, Aditya & Pande, Sagar & Malik, Rahul & Gupta, Deepak. (2021). Recognition and Classification of Pomegranate Leaves Diseases by Image Processing and Machine Learning Techniques. *Computers, Materials and Continua*. 66. 2939–2955. 10.32604/cmc.2021.012466.
  16. B. ChandraKarmokar, M. Samawat Ullah, M. Kibria Siddiquee, and K. Md. Rokibul Alam,"Tea leaf ailments recognition using neural network ensemble," *Int. J. Comput. Appl.*, vol. 114, no. 17, pp. 27–30, 2015.
  17. Rothe, Prashant & Kshirsagar, R.. (2015). Cotton leaf disease identification using pattern recognition techniques. 1-6. 10.1109/PERVASIVE.2015.7086983.
  18. J. G. A. Barbedo, "Impact of sample set size and variety on the effectiveness of DL and transfer learning for Flora ailment classification," *Comput. Electron. Agric.*, vol. 153, pp. 46–53, 2018.
  19. K. P. Ferentinos, "DL systems for Flora ailment detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018.
  20. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using DL for image-based Flora ailment detection," *Front. Flora Sci.*, vol. 7, p. 1419, 2016.
  21. A. Kp and J. Anitha, "Flora ailment classification using DL," in *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*, 2021.

22. M. Akila and P. Deepan, "Detection and Classification of Flora Leaf Ailments by using DL Algorithm," INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) ICONNECT, vol. 6, 2018.
23. E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of finetuning DL systems for Flora ailment identification," Comput. Electron. Agric., vol. 161, pp. 272– 279, 2019.
24. Atila, Umit & Ucar, Murat & Akyol, Kemal & Uçar, Emine. (2021). Plant leaf disease classification using EfficientNet deep learning model. Ecological Informatics. 61. 101182. 10.1016/j.ecoinf.2020.101182.
25. S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural net- works based recognition of Flora ailments by leaf image classification," Comput. Intell. Neurosci., vol. 2016, p. 3289801, 2016.
26. R. Singh, A. Sharma, V. Anand, and R. Gupta, "Impact of EfficientNetB3 for stratification of Tomato Leaves Ailment," in 2022 6th International Conference on Electronics, Communication and Aerospace Technology, 2022.
27. R. Singh, A. Sharma, N. Sharma, and R. Gupta, "Automatic detection of cassava leaf ailment using Transfer Learning System," in 2022 6th International Conference on Electronics, Communication and Aerospace Technology, 2022.