An Improved CNN Model for Identifying Tomato Leaf Diseases

Debabrat Bharali[1], Dr.Kanak Chandra Bora[2,*] Dr.Rashel Sarkar[3]

[1]Department of Computer Science,University of Science and Technology,Meghalaya

[2]Associate professor,Department of Computer Science, University of Science and Technology,Meghalaya

[3]Associate professor,Department of Computer Science and Engineering, Royal Global University,Guwahati,Assam

*Correspondence: dbharali15@gmail.com

Received 20 March 2024; Revised 25 April 2024; Accepted 25 April 2024

Available online 25 April 2024 at www.atlas-tjes.org, doi: 10.22545/2024/00253

Abstract: *A major loss in gross domestic product, quantity and quality of products produced, as well as tomato production, is caused by diseases in tomato leaves due to which farmers have a difficult time in controlling and monitoring the health of tomato leaves, one of which is leaf disease. In this paper, we developed an Improved CNN by using data augmentation techniques to identify the seven classes(blight ,leaf curl, leaf miner ,Alteneria, leaf spot, cutwork infected, healthy) of tomato leaf diseases. Using 27807 trainable parameters, the improved CNN obtains the maximum training accuracy of ninety nine point nine eight percent(99.98%) and validating accuracy of ninety eight point four percent(98.4%) .With fewer parameters, the Improved CNN can more accurately determine the type of illness of tomato leaf.Our Improved CNN model also determine the type of illness of tomato leaf when tested with the images of diseased tomato leaf collected from the internet sources(plant village dataset). Using 152850 trainable parameters the Improved CNN obtain the maximum training accuracy of 99.68% and validation accuracy of 89%.*

Keywords: Diseased tomato leaf, Improved CNN, Accuracy, plant village dataset.

1 Introduction

The cultivation of tomatoes is a vital part of the agricultural economy. One of the primary reasons restricting tomato cultivation is disease. The most common disease in the world is bacterial sickness one that lowers productivity, is highly detrimental to the health of tomato leaves, and results in significant economic loss for the agricultural sector. To improve the quantity and quality of tomato leaves, protection from disease is essential. Fungi disease causes severe damage to leaves. To choose the best course of action and prevent the spread of the diseases, it is therefore very beneficial to provide early detection and diagnosis of diseases. Farmers may occasionally lack the resources or knowledge to consult professionals. Identification of tomato

leaf disease is more difficult visually, less precise, and only possible in a few places. Deep learning techniques will be applied in the proposed system to automatically identify images of leaves using "CNN". Deep learning-based tomato leaf disease detection has various advantages for sustainable agriculture.
1. Early Detection and Prevention: Deep learning algorithms are capable of precisely identifying a number of diseases in tomato plants before any outward signs manifest. By removing diseased plants or providing targeted therapies in a timely manner, early identification enables farmers to stop the spread of diseases to healthy plants. By taking preventive action, the demand for broad-spectrum chemical pesticides is decreased, supporting environmentally friendly agricultural methods.
2. Precision Agriculture: Deep learning algorithms allow farmers to apply treatments only where they are required by accurately recognizing the kind and severity of diseases affecting tomato plants. By minimizing the use of agrochemicals, this focused strategy lowers pollution to the environment and maintains the health of the soil. Additionally, it maximizes the use of resources, resulting in long-term cost savings for farmers.
3. Optimized Crop Management: Large volumes of data about environmental factors, plant health, and disease prevalence can be analyzed by deep learning algorithms. Farmers can decide on crop management techniques, such as crop rotation, nutrient application, and irrigation schedule, by combining this information. By making these processes more efficient, we may increase crop output and resilience while using fewer inputs, which supports sustainable agriculture.
4. Crop Monitoring and Yield Prediction: Farmers can monitor the advancement of diseases throughout the growing season by employing deep learning to continuously scan tomato plants for disease indicators. Farmers can lower economic losses and increase overall farm profitability by modifying their harvesting schedules and marketing tactics in response to yield losses linked to certain diseases.

5. Providing Knowledge to Farmers: Deep learning models can be used to provide decision assistance tools that farmers can access via websites or mobile apps. These technologies give farmers access to vital knowledge and experience, empowering them to make well-informed decisions that support sustainable agricultural practices. They do this by offering real-time disease diagnosis and management advice.

The primary contributions of this study are as follows:

- To developed an Improved CNN by using data augmentation techniques to identify the seven classes (blight, leaf curl, leaf miner ,Alternaria, leaf spot, cutwork infected, healthy) of tomato leaf diseases
- The Improved CNN is faster and requires fewer parameters to identify the illness of tomato leaf.
- The performance of the Improved CNN is assessed using photos of diseased tomato leaf to verify its resilience. The Improved CNN performs better in terms of training and verifying accuracy.

2 Review of the Existing Work

[1] The features of the input images are extracted in this study using the CNN algorithm so as to differentiate between the healthy and diseased leaves of different plants. Finding the most relevant class for the photographs in the dataset is made easier with the help of the obtained features. The suggested approach, with a training accuracy rate of over 94.5% to identify the image class, according to the authors' observations.[2] In this paper, they modified CNN to improve accuracy with fewer trainable parameters and less computation time. They assessed the proposed model's classification performance against several machine learning and deep learning methods for potato blight. Modified CNN achieved a training accuracy of 97%, the recommended model outperformed the competition in 183 seconds of training time.[3] In this work, plants are identified by the recognition of their leaves and blooms using convolutional neural

networks (CNNs). Using photos of the leaves, flowers, or both, this study investigates how well CNN recognizes different types of plants. The Folio Leaf and Flower Recognition datasets, which are both publicly available, have been employed for training and verifying. According to experimental results, using petal photos alone results in the highest training accuracy for plant identification, which is 98%, when compared to photographs of flowers alone or both.[4] In order to identify plant leaf diseases, Deep CNN based model was presented in this study. The Deep CNN is trained on an accessible dataset that comprises 39 various classes of plant petals. The six distinct data augmentation techniques that were used were "image flipping", "gamma correction", "noise injection","PCA", "color augmentation", "rotation", and "scaling". It has been observed that using "data augmentation" can enhance model's interpretation. The model was trained using various "training epochs", "batch sizes", and "dropout rates". The deep CNN achieves 96.46% using validation data.[5]In this paper they develop CNN for the classification of plant petals photos that will enable users to identify different kinds of medicinal plants . The public can benefit from this research by learning to identify five different kinds of therapeutic plants, such as spinach Duri, Dadap Serep, moringa, and Javanese ginseng achieve training accuracy of 86%[6]They have created an automated method for classifying medicinal plants in order to speed up the identification of helpful plant species. Ten Bangladeshi medicinal plants are included in an updated dataset that was compiled from different regions of the country, along with a few photos that were taken from various online sources. The 71.3% accuracy rate from an additional 3570 photographs after 34123 images were utilized for the training procedure [7] This work proposes a unique CNN architecture to classify lady finger plant leaf pictures into three groups: burned, damaged, and healthy leaves. Of the 1088 photos in the collection, 457 depict healthy (i.e., pest-free) lady finger plant leaves, 509 include disease- and pest-infected leaves, and 122 feature burned leaves from overfertilization. The Tiruvannamalai area of Tamil Nadu's agricultural fields are where the photos were taken. The suggested CNN architecture achieved 96% classification accuracy.[8] In order to identify and diagnose leaf illnesses, this study uses a convolution neural network to classify photos. The suggested method's primary goal is to use a neural network to treat tomato, corn, and apple leaf diseases. There are eight layers in the suggested convolutional neural network model, 3 maximum pool sheets and five blocks of CNN. The proposed system's training accuracy for three different types of leaf picture types varies from 96% to 98%, indicating the neural network approach's feasibility.[9]This research develops an autonomous approach for recognizing leaf illness in tomato leaves using Deep CNN. In this paper, they used a plant village data set. Forty percent of the photos were utilized for testing and sixty percent were used for training. The suggested DCNN model achieved a 97% accuracy rate for the testing set. [10] In this paper, they suggest an automatic plant recognition methodology that recognizes plant species based on their leaves. A deep CNN is applied to acquire good precision. Pre-processing photos, characteristics removal, and identification are the main parts of recognition that are taken into review. The suggested CNN categorizing plant properties, including leaf classification, using unseen sheet including convolution, maximum pooling, dropout, and fully connected layers. With minimum of losses and a 97% training accuracy rate, the model identifies the correct category of an unfamiliar plant by learning about the characteristics of the Swedish leaf datum.[11] This work identifies leaf diseases in various mango plant species using an automated deep learning-based method. Five different leaf diseases have been detected from 1200 photos of both healthy and sick mango leaves: anthracnose, Alternaria petal spots, petal gall, petal webber, and mango petal burn. With an accuracy of 96.67%, the proposed CNN model accurately diagnoses leaf diseases in mango plants, suggesting that real-time model applications are possible.[12] This paper presents CNN with a minor sheet, which reduces the computational load. various augmentation techniques are employed to create extra samples, increasing the "training data", without actually taking more photos. "Apple leaf scab", "black rot", and "cedar rust" are among the diseases that the CNN model is trained to identify using the publicly available,"PlantVillage" datum. The comprehensive experimental findings show that the suggested model is good for diagnosing diseases in "apple petals", achieving 98% classification accuracy. [13]Tomato illnesses are appropriately characterized and classified using the Convolutional Neural Network (CNN). Lastly, CNN extracts several attributes from photos, including edges, textures, and colors. Three thousand photos of tomato petals with 9 distinct diseases and strong petal make the dataset used for the entire project. 97% of the forecast by the suggested model is perfect, according to the results.[14] This research examines the AlexNet model for the purpose of rapidly and correctly detecting petals in maize plants. To verify the result, they used online datum. This datum consists 2 parts of diseases maize petals: petal-spot-based diseases and common rust-based illnesses. There are 929 pictures in the later group and 1363 pictures in the former. One of CNN's biggest advantages is to recognize automatically extract characteristics by simply processing rare images. The suggested model training accuracy is 99.16%.[15] Three common rice diseases are mostly examined in this paper: brown spot and leaf smut are caused by" fungi", "bacterial petal blight" is caused by "bacteria", and both are caused by bacteria. A fully connected CNN-based training set including 4000 image samples of both healthy and diseased rice leaves was utilized to train the suggested method in order to identify the three rice illnesses. The fully connected CNN that has been proposed is a fast and effective technique that attains a 97.7% training accuracy rate on the dataset, according to the investigation's findings. [16] The purpose of this paper is to employ the recently proposed CNN model, which uses customized CNN, to categorize tomato plant petals into ten different classes. The publicly available "Plant Village datum", which has sixteen thousand photos of both strong and sick tomato petals, was used to do this. The 150 x 150 dimensions of the tomato leaf photos were reduced through preprocessing. The suggested model consists of 4 CNN sheets followed by a maximum pool sheet. Following an 80:20 divided datum into training and verifying datum, the model was trained on 105 epochs of images of tomato leaves, with a training accuracy of 98.19%.[17] DCNN was proposed as a tool for the disorder diagnosis of 4 cucumber illnesses: "anthracnose", "downy mildew", "powdery mildew", and "target petals spots". The disorder photos were divided using field-captured photographs of cucumber leaves. The likelihood of overfitting was decreased by increasing the size of the datum produced by the divided disorder photos through the use of data augmentation techniques. At 93.4% training accuracy, the DCNN produced good recognition results. [18] This study's objective is to identify which CNN model is most useful for categorizing and classifying different plant leaf species. Ten different species of medicinal leaves, each falling into two categories, are used in this study to test the feasibility of the basic CNN model, which is comprised of four convolution layers to achieve a training accuracy of 96.88%.[19] Utilizing photos of petals gathered in an uncontrollable setting, the deep learning concept was applied in this work to diagnose and forecast the disease condition of cotton plants. In this work, an enhanced Deep CNN-based model is utilized to address the issue of disease detection &classification in cotton plants. 2293 photos of plants and cotton leaves were used in the database to train the models. Plant illness coalition and their similar classification were added in the data, along with 4various classes of petals. With cotton plants, the model achieved 97.98% training accuracy in categorizing leaves and plant illnesses.[20] This study classifies photos of petals using CNN to create a plant illness recognition model. The suggested model is completed using Keras by adjusting rectified linear unit (Relu) functions, dropout levels, training and testing combinations, and epoch counts. For 14 plants with 38 pathologies (including healthy ones), they have developed a convolutional neural network using 200 photos per class from the 44,016 images in the Plant Village dataset. Plant leaf diseases were automatically diagnosed by this study, with a verifying accuracy of 98%.[21] In this paper, they develop a DCNN for the recognition and categorizing of grape ailments using Red, Green, and Blue photos. The suggested model makes use of a photo collection of "grape plant" from the "Plant Village dataset". The generated model's distinctive feature is that it's a brand-new CNN classification model with 99.34% training accuracy.[22] In this paper, they used CNN as a machine learning tool to deliver results for the input of leaf images into the temporary diagnosis findings. 838 images were used for 4 classes. The average results demonstrate that 73% accuracy may be attained with CNN for the identification of plant pests and diseases in Kenaf plants.[23] In this paper, piper plants were chosen for this study. The Deep CNN method is used for feature extraction method for all plants, including piper plants like piper nigrum and mulesa, as well as cherry, apple, and tomato Comparing the proposed hyperparameter-tuned random forest classifier to existing machine learning algorithms like SVM, naïve Bayes, and logistic regression, the experimental findings demonstrate that the suggested classifier performs better, with a training accuracy value of 0.94 for all plants and training accuracy value 0.88 for piper plant. The following Table 1 shows the methodology, dataset name, and experimental result (training and validation accuracy) obtained by the authors in the review of the existing work section.

Table 1: Summary of the Review of existing work

Authors	Methods	Dataset	Result
[1]Deepalakshmi P,etal(2021)	CNN	Plant Village	94.5%(validation accuracy)
[2]Al-Adhaileh,etal(2023)	CNN	Plant Village	97%(training accuracy)
[3]FatihahSahidan, Nurul, et al(2019).	CNN	Folio Leaf dataset and the Flower Recognition	98% (training accuracy)
[4] Geetharamani, G., & J.Arun Pandian. (2019)	CNN	Plant Village Dataset	96.46% (validation accuracy)
[10] Bisen, D (2021).	CNN	Swedish leaf dataset	97%(training accuracy)
[16] Baser.P,et al(2023).	CNN	Plant Village	98%(training accuracy)
[20]Khan,S.,&Narvekar, M..(2020).	CNN	Plant Village	97.2%(training accuracy)
[21]Math,R.M.,& Dharwadkar, N. V. (2022).	CNN	Plant Village	99.3%(training accuracy)
[23] Pravin,A.,C.Deepa.(2022)	CNN	Piper plants	88%(Training accuracy

3 Methodology

In this section, the following Figure 1 shows the proposed flowchart of our Improved Convolution Neural Network. It describes the procedures needed to meet the training and validation accuracy that we accomplish by preprocessing augmented images of tomato leaves that were collected from the field and applying augmentation techniques to them. Based on the highest accuracy, we can identify the name of the diseased tomato leaf.

Figure 1: Flowchart of Improved Convolutional Neural Network.

Step 1: In the field, images of sick tomato leaf are captured. The seven classes—blight, Alternaria, healthy, leaf miner, leaf curl, leaf spot, and cutwork infected leaf—are labeled on images of diseased tomato petal according to their names.

Step 2: An increase in dataset size is achieved by augmenting images of tomato leaf diseases. Image augmentation is a technique that uses different adjustments to the original image to produce multiple modified versions of the same image. An easy and quick approach to enhance images is to use the Keras **ImageDataGenerator class**. Real-time data augmentation is the primary advantage of utilizing the Keras ImageDataGenerator class. Various techniques used in doing augmenting the images of tomato leaf are: **“rotation”, “width Shifting”, “height Shifting” ,”rescale”, “zoom” , “horizontal flip”, “vertical flip”, “brightness range”, “fill mode”.**

1. **rotation range**: To rotate images at random, use the revolving range parameter, which takes a number in degrees (0-180). Images of tomato disease leaf is rotated at random within the range of 5.
2. **width Shifting and Height Shifting**: To translate images vertically or horizontally at random, use the breath shifting and height shifting ranges (as a percentage of the overall breath or height).We have taken

the width shifting as 0.1. Here the images of diseased tomato leaf is shifted horizontally and vertically by a maximum of 10 percent of the total width of the images either left or right

3. **Rescale**: Rescale is the value by which the data will be multiplied prior to further processing. Our original image had RGB coefficients ranging from 0-255, however, at a typical learning rate, our models couldn't handle these values since they are too high. Consequently, we scale to target values between 0 and 1 using a factor of 1/255.

4. **zoom**: To arbitrarily zoom inside images, use zoom_range.A zoom of 0.2 is used. It scales up or down by 20% relative to the original size of the image during training.

5. **Horizontal Flip**: To flip half of images of tomato leaf horizontally at random, use the horizontal flip function and set as true

6. **Vertical Flip:** To flip half of images of tomato leaf vertically at random, use the vertical flip function and set as true

7. **Brightness range**: It modifies the image's brightness at random. In the ImageDataGenerator class, brightness can be adjusted using the brightness range option. The function selects a brightness shift value from a range provided by a list of two float values. The image is brighter when the value is above 1.0 and darker when it is less than 1.0. Here the brightness range is set to [0.2,1.0].

8. **Fill mode**: “When the image is rotated, certain pixels will move outside of it”, producing voids that need to be filled in. There are several ways to fill this, including using a fixed value or the values of the closest pixels. The default option for this is "nearest," which merely substitutes the closest pixel values for the empty space. This is defined in the fill mode argument.

Step 3: After using augmentation techniques preprocessing the augmented images of tomato leaf diseases is necessary to feed those images into the neural network and create an Improved Convolutional Neural Network. First, we need to preprocess the training images of illnesses of tomato leaves. To perform training images we have to use keras library. First, we have to select the directory where the data is stored. labels is set to as **“inferred”**. It means that the labels are generalized from the directory structure. **label mode** is set to **“categorical”.** It means that the labels are encoded as categorical vectors since it is more than one class. Class name is set to None. It means that Only valid if labels is "inferred". **Color mode** is set to **“ rgb”.**It means that the images of tomato leaf converted to 3 channels. Default size of batch size is 32.To resize the image of the tomato leaf it is set as (100,100)(height, width).**Shuffle** is set to true. It means that at the time of feeding the images of tomato leaf into the model for training shuffle the entire class and pass it because it will reduce the biasness of the model. It is optional. **Interpolation** is used to resize the images. By Default it is set to bilinear. The **following links** means whether to visit the subdirectories or not**. By default, it is set to false.** Similarly to perform the validation image preprocessing we have to use the keras library.

Step 4 Two sets of data are created from the preprocessed augmented images of tomato leaf diseases: training and validation, and testing. Ten percent we used for testing tomato leaf images, twenty percent for validating disease tomato petal photos, and seventy percent for training disease tomato petal images.

Step 5: Finally we trained the Improved CNN model by using the augmented images of diseased tomato leaf collected from the field and calculated the performance of the model by using a confusion matrix. Our model achieved training accuracy of 99.98% and able to identify the images of diseased tomato

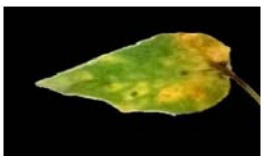
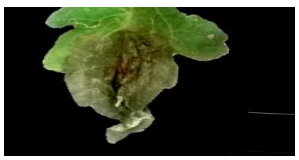
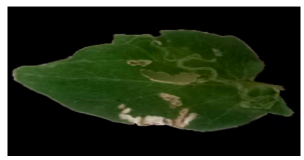
3.1 Dataset description

The images of diseased tomato leaf that were collected from the field served as the dataset for this study. The following Table 2 shows the labeling of each class of diseased tomato leaf along with the image count of each class of diseased tomato leaf collected from the field.

Table 2 shows the class labels of diseased tomato leaves collected from field along with number of images

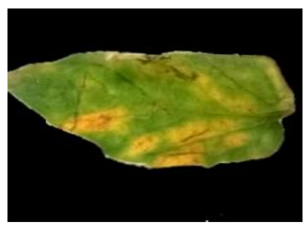
Class Labels	Number Of Images
1.Alteneria	584
2.Blight	1541
3.Cutwork infected	735
4 Healthy	151
5.Leaf Curl	1676
6.Leaf miner	2155
7.Leaf spot	862

The following Figure 2 shows the representative images of diseases in tomato leaf collected from the field. These images served as the dataset for our study and were incorporated into our Improved CNN model.


(a) Alternaria leaf **(b)blight** **(c) Cutwork infected** **(d)Healthy**





(e) Leaf Curl **(f) Leaf miner** **(g)Yellowing and leaf spot**

Figure 2: Sample images of (a)Alternaria (b)blight (c)Cutwork infected (d)Healthy (e) leaf curl (f) leaf miner (g) yellowing and leaf spot disease. These leaves are collected from the field.

3.2 Block Diagram of Improved CNN Model

In this section, we describe the block diagram of our Improved CNN model. The following Figure 3 shows that in our Improved CNN model, we have used four convolutional layers. They are conv1,conv2,conv3, and conv4, and describe the steps that are needed to be done in our Improved CNN model.

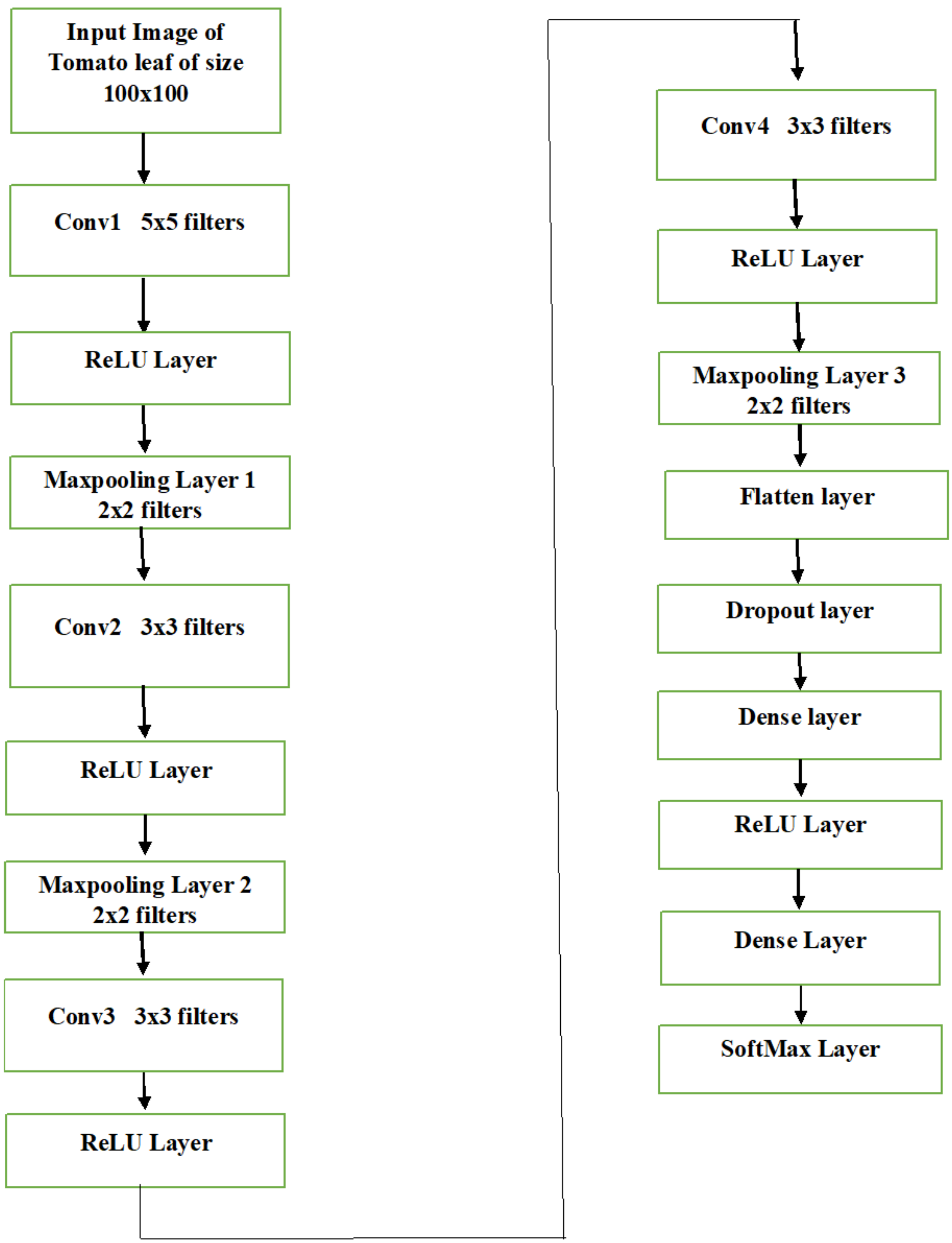


Figure 3: Block diagram of Improved Convolutional Neural Network.

Four Convolution layers are used to construct the Improved CNN. Typically, there are no learnable parameters in the pooling, flattening, activation, and drop-out layers. The only layers that have learnable parameters are the output, dense, and convolutional layers. Generally, parameters are weights. The strength of the connectedness is indicated by weights. weights are learned during training.

Step 1: The size of the image of tomato leaf illnesses that we are first going to look at is 100X100X3, where 100 and 100 stand for width and height, respectively, and 3 for RGB values. The pixel intensity at each of these values is defined by a value between 0 and 255.

Steps2. The first CNN block uses eight filters in our upgraded CNN. We then employ a 5*5 filter size following the convolutional layer, so the output shape and input shape will remain the same because the padding is set to the same to maintain the input's spatial dimensions. Here, the enhanced CNN uses the formula:

output shape=input_shape+floor(2*x-1/2)-x+1.

This formula results in an output shape of (100,100,8). The output shape stays the same if we apply a ReLU activation function element-by-element to the convolutional layer's output. Every input element is subjected to the following operation by the ReLU function: Relu is equal to max(0,x). The ReLU activation function yields an output form that is 100×100 because it does not alter the size of its input.The following Figure 4 shows the ReLU layer that introduces the non-linearity into the Network. All the negative values are changed to 0 because ReLU is equal to max(0,x).

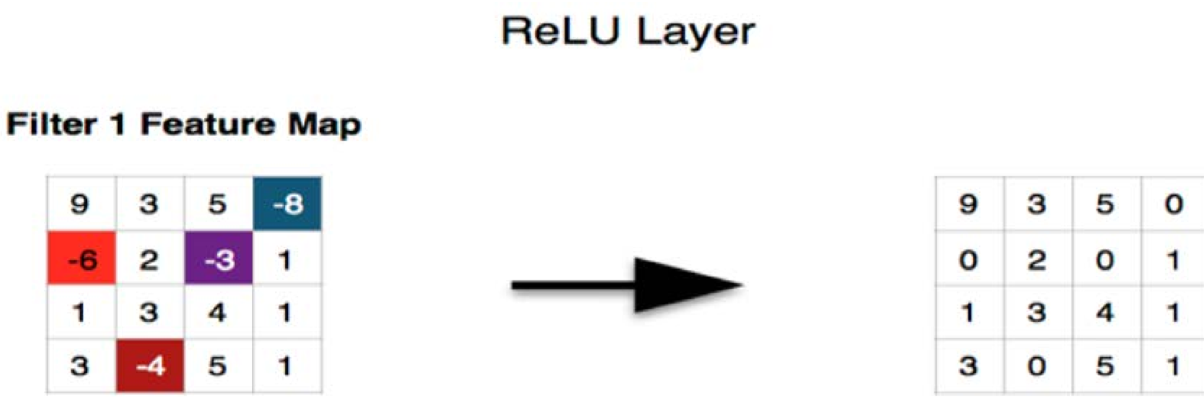


Figure 4: ReLU Layer.

We use the Max Pooling Layer in the CNN's first block to lower each feature map's dimensionality. The final form of the Improved convolutional Neural Network layer is 100,100, which is the input shape of the tomato leaf disease that we are addressing here. It will become 50,50 after max pooling.

Step 3: The output of the Maxpooling layer will be used as input in the second CNN block which consists of(50,50,8) where 50 and 50 stands for the width and height. The second CNN block uses eight filters in our upgraded convolutional neural network. We then employ a 3*3 filter size following the convolutional layer, so the output shape and input shape will remain the same because the padding is set to the same to maintain the input's spatial dimensions. Here, the enhanced CNN uses the formula:

output shape=input_shape+2*floor(x-1/2)-x+1.

This formula results an output shape of (50,50,8). The output shape stays the same if we apply a ReLU activation function element-by-element to the convolutional layer's output. Every input element is subjected to the following operation by the ReLU function: Relu is equal to max(0,x). The ReLU activation function yields an output form that is 50×50 because it does not alter the size of its input. We use the Max Pooling Layer in the CNN's second block to lower each feature map's dimensionality. The final form of the Improved convolutional layer is 50,50 which is the input shape of tomato leaf disease that we are addressing here. It will become (25,25,16) after max pooling

Step 4: The output of Maxpooling layer will be used as input in the third CNN block which consists of (25,25) where 25 and 25 stands for the width and height.The second CNN block uses sixteen filters in our upgraded convolutional neural network. We then employ a 3*3 filter size following the convolutional layer, so the output shape and input shape will remain the same because the padding is set to the same to maintain the input's spatial dimensions. Here, the enhanced CNN uses the formula:

output shape=input_shape+2*floor(x-1/2)-x+1.

This formula results an output shape of (25,25,16). The output shape stays the same if we apply a ReLU activation function element-by-element to the convolutional layer's output. Every input element is subjected to the following operation by the ReLU function: Relu is equal to max(0,x). The ReLU activation function yields an output form that is 25×25 because it does not alter the size of its input. In the third block, we have not applied the max pooling because feature maps are down-sampled using max-pooling, which extracts the maximum value from each pooling zone. While lowering the number of parameters through downsampling helps minimize overfitting and lower computing costs, it can also result in the loss of spatial information. In certain instances, maintaining spatial information becomes increasingly important for accurate feature representation, particularly in deeper layers of the network. Consequently, if max-pooling is skipped in subsequent blocks, more precise spatial information may be retained.

Step 5: The final form of the third Enhanced layer will become as input in the fourth CNN block which consists of (25,25,8) image size. The fourth CNN block uses sixteen filters in our upgraded convolutional neural network. We then employ a 3*3 filter size following the convolutional layer, so the output shape and input shape will remain the same because padding is set to the same to maintain the input's spatial dimensions. Here, the enhanced Convolutional Neural Network uses the formula:

output shape=input_shape+2*floor(x-1/2)-x+1.

This formula results an output shape of (25,25,16). The output shape stays the same if we apply a ReLU activation function element-by-element to the convolutional layer's output. Every input element is subjected to the following operation by the ReLU function: Relu is equal to max(0,x). The ReLU activation function yields an output form that is 25×25 because it does not alter the size of its input. We use the Max Pooling Layer in the CNN's fourth block to lower each feature map's dimensionality. The final form of the improved convolutional layer is 25,25 which is the input shape of tomato leaf disease that we are addressing here. It will become (12,12,16) after max pooling.

Step 6: Flattening will be used to turn every n*n matrix from pooled feature maps into a single-dimensional matrix once all four convolution layers have been completed . In this instance, flatten is set to 2304 because it was maxpooling in previous layers(12,12,16)of the final pooled feature, which is now 12x12x16=2304

Step 7. In the next we have use Droput(0.2) to prevent overfitting. A model performs poorly on unknown data when it overfits—that is, when it learns to memorize the training set rather than drawing conclusions from it.

Step 8: : Ten neutrons are used for the fully connected in the output layers. Because the number 10 indicates the dimensions of the output space, the layer will provide an output of size 10. All neurons in this layer are connected to all neurons in the previous layer by applying the activation function to the layer's output.A completely linked neural network is produced as a result.

Step 9: Output space, or dimensionality, of this layer is implied to be 7 by the Dense(7) layer. During training, the weights of the connections between this layer and the input layer or preceding layer will be discovered.

Step 10. : For a multi-class classification problem, we build an Improved CNN with seven classes after the dense layer. The neural network's output layer uses the activation function called softmax to produce class probabilities. The model may generate a probability distribution over the seven classes for each input.

Following the block diagram of the Improved CNN model, Table 3 shows the procedures that were included in our model and displays the number of parameters (weights) that are generated in our

Improved CNN model using the augmented images of diseased tomato leaves that were collected from the field.

Table 3: Number of parameters generated in Improved CNN using the images of tomato leaf collected from the field

Layer	**Input size of image**	**Filter**	**Parameter**
Input image of tomato leaf	100x100	-	-
Convolution 1	Input image	5x5/8	608
Convolution 2	Convolution 1	3x3/8	584
Convolution 3	Convolution 2	3x3/16	1168
Convolution 4	Convolution 3	3x3/16	2320
Maxpooling	Convolution 4	2x2	-
Flattened	Maxpooling	2x2	-
Dense(10)	Flattened	-	23050
Dense(7)	Dense(10)	-	77
Total parameter			27807

Step 1: First we take an input image of a Tomato leaf of size 100x100.
Step 2: The input image of the Tomato leaf is passed through the first convolution 1 which consists of filter sizes 5x5 and 8 filters. The parameter generated in Convolution 1 is 608.
Step 3: The output of Convolution1 is taken as an input in the convolution2 which consists of a filter size of 3x3 and 8 filters. The parameter generated in Convolution 2 is 584
Step 4: The output of Convolution2 is taken as an input in the convolution3 which consists of a filter size of 3x3 and 16 filters. The parameter generated in Convolution 3 is 1168
Step 5: The output of Convolution3 is taken as an input in the convolution4 which consists of a filter size of 3x3 and 16 filters. The parameter generated in Convolution 3 is 2320.
Step 6. The final output of Convolutional 4 is taken as an input in the maxpooling which consists of a kernel size of 2x2.
Step 7: Flattened will be used as input in Dense(10) where 10 is the number of neuron and the parameters generated is 23050.
Step 8: The output dense(10) will be used as input in Dense(7) to classify the seven classes of diseased tomatoes and the parameter generated is 77.

Therefore, we can say that the Improved CNN consists of 27807 trainable parameters and reduced the computational time.

The following Table 4 describes the overall summary of our Improved CNN model and layers of CNN as well as the output shape of the images of the diseased tomato leaf with less number of trainable parameters

Table 4. Summary of the Improved CNN and the total number of Trainable Parameters of images of tomato leaf collected from the field environment

Layer	Output shape	Parameters
conv2d (Conv2D)	(None, 100, 100, 8)	608
activation (Activation)	(None, 100, 100, 8)	0
max_pooling2d (MaxPooling2D)	None, 50, 50, 8)	0
conv2d_1 (Conv2D)	(None, 50, 50, 8)	584
Activation_1 (Activation)	(None, 50, 50, 8)	0
max_pooling2d _1(MaxPooling2D	(None, 25, 25, 8)	0
conv2d_2 (Conv2D)	None, 25, 25, 16)	1168
activation_2 (Activation)	(None, 25, 25, 16)	0
conv2d_3 (Conv2D)	(None, 25, 25, 16)	2320
activation_3 (Activation)	(None, 25, 25, 16)	0
max_pooling2d _2(MaxPooling2D	(None, 12, 12, 16)	0
flatten (Flatten)	(None, 2304)	0
dropout (Dropout)	(None, 2304)	0
dense (Dense)	(None,10)	23050
activation_4 (Activation)	(None,10)	0
dense_1 (Dense)	(None,7)	77
activation_5 (Activation)	(None,7)	0

===

Total params: 27807 (108.62 KB)
Trainable params: 27807 (108.62 KB)
Non-trainable params: 0 (0.00 Byte)

To test if the Improved CNN model can correctly identify the names of the diseases with fewer parameters and less computational effort, we collected the dataset from the plant village dataset(internet sources). The following Table 5 shows the dataset of class labeling of diseased tomato leaf along with the number of images.

Table 5: shows the class labels of diseased tomato leaves collected from Plant Village Dataset along with number of images

Class labels	Number of images
1.Tomato Bacterial Spot	1702
2.Tomato Early Blight	1902
3.Tomato Healthy	1926
4.Tomato late blight	1851
5.Tomato Septoria leaf spot	1745
6. Tomato Spider mites	1741
7.Tomato leaf mold	1882
8.Tomato target spot	1827
9.Tomato mosaic virus	1790
10.Tomato leaf yellow leaf curl virus	1961

The following Figure 5 shows the representative images of diseased tomato leaf collected from the plant village dataset (https://www.kaggle.com/datasets/emmarex/plantdisease).

(a)Tomato bacterial spot

(b)Tomato Early blight

(c) Healthy

(d) late blight

(e)Septoria leaf spot

(f) spider mites

(g)leaf spot

(h)target spot

(i)mosaic virus

(j)leaf curl virus

Figure 5: Sample Images of (a)Tomato bacterial spot (b)Tomato Early blight (c) Healthy (d) late blight (e)Septoria leaf spot (f) spider mites (g)leaf spot (h)target spot (i)mosaic virus (j)leaf curl virus that has been collected from the plant village dataset.

The following Table 6 shows the procedures that were included in our model and displays the number of parameters (weights) that are generated in our Improved CNN model using the images of diseased tomato leaves that were collected from the plant village dataset (internet sources).

Table 6 below shows the total number of parameters generated in Improved CNN of images of tomato leaf collected from the plant village dataset

Layer	Input size of image	Filter	Parameter
Input image of tomato leaf	100x100	-	-
Convolution 1	Input image	5x5/8	608
Convolution 2	Convolution 1	3x3/8	584
Convolution 3	Convolution 2	3x3/16	1168
Convolution 4	Convolution 3	3x3/16	2320
Maxpooling	Convolution 4	2x2	-
Flattened	Maxpooling	2x2	-
Dense(64)	Flattened	-	147520
Dense(10)	Dense(64)	-	650
Total parameter			152850

Step 1: First we take an input image of a Tomato leaf of size 100x100.
Step 2: The input image of the Tomato leaf is passed through the first convolution 1 which consists of filter size 5x5 and 8 filters. The parameter generated in Convolution 1 is 608.
Step 3: The output of Convolution1 is taken as an input in the convolution2 which consists of a filter size of 3x3 and 8 filters. The parameter generated in Convolution 2 is 584
Step 4: The output of Convolution2 is taken as an input in the convolution3 which consists of a filter size of 3x3 and 16 filters. The parameter generated in Convolution 3 is 1168
Step 5: The output of Convolution3 is taken as an input in the convolution4 which consists of a filter size of 3x3 and 16 filters. The parameter generated in Convolution 3 is 2320.
Step 6. The final output of Convolutional 4 is taken as an input in the maxpooling which consists of a kernel size of 2x2.
Step 7: Flattened will be used as input in Dense(64) where 64 is the number of neuron and parameters generated is 147520.
Step 8: The output dense(10) will be used as input in Dense(10) to classify the ten classes of diseased tomatoes and the parameter generated is 650.
Therefore, we can say that our Improved CNN consists of 152850 trainable parameters which is comparatively less and reduced computational time.

4 Results and Discussions

We utilize the confusion matrix to show the Improved CNN class-wise performance to assess its performance. Confusion matrix is a performance evaluation tool representing the accuracy of a classification model. In a confusion matrix, the labels predicted by the model are represented by the predicted class, and the actual class represents the true labels of the data. Here the horizontal axis is represented by the predicated class of each class label of diseased tomato leaf and the vertical axis is represented by the actual class of each class of diseased tomato leaf. For every class of diseased tomato leaf, we compute the true positive, false positive, true negative, and false negative using the Confusion Matrix. The following Figure 6 shows the heat map or confusion matrix of each class of diseased tomato leaf starting from class 0 to class 6 according to the name of the diseases shown in Table 2 from Alternaria to leaf spot. So here class 0 represents the Alternaria and class 6 represents the leaf spot. In the actual class, we are observing that the disease is present in the tomato leaf but in the predicated class it matches the actual class our Improved CNN model is predicting whether the disease is healthy or unhealthy. If it is unhealthy, it will identify the name of the disease based on the high accuracy of our Model.

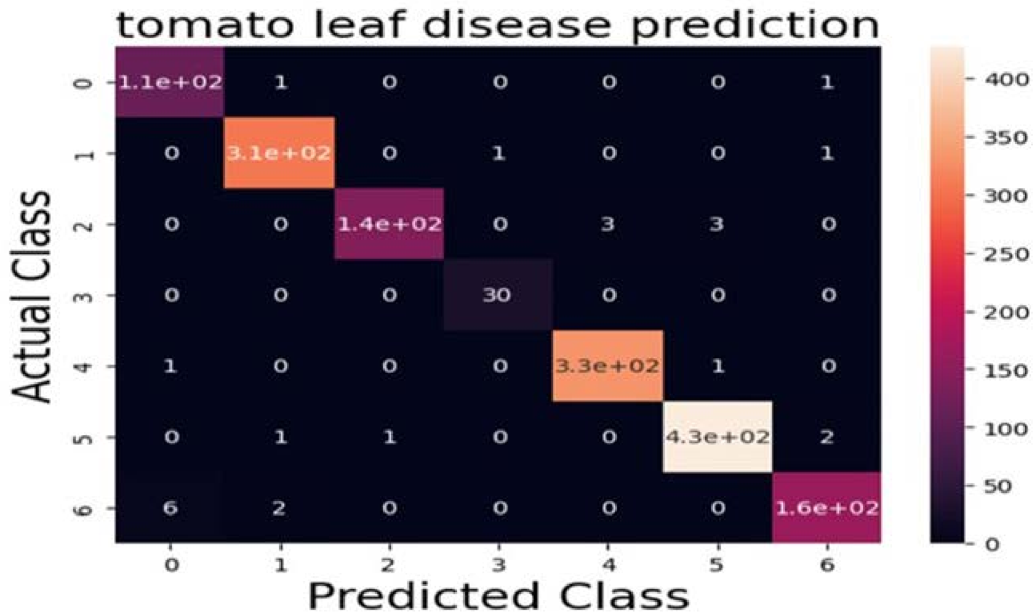


Figure 6: Matrix of Confusion for the improved CNN using images of diseased tomato leaf collected from the field.

The total number of correctly predicted images is called ***"true positive" (Q),*** whereas the total number of incorrectly predicted images is called ***"false positive" (R).*** Parallel to this***, "true negative" (S)*** denotes the total number of correctly predicted images in the negative class, while ***"false negative"(T)*** denotes the total number of incorrectly predicted images in the same class.

We take into account various performance metrics, including F1 score based on the confusion matrix, accuracy, precision, recall, and recall.

The following formula calculates accuracy as the Q+S divided by the sum of Q+R+S+T:

$$\textbf{Accuracy=Q+S/Q+R+S+T} \quad (1)$$

Q divided by (Q+R) as shown below, can be used to define precision.

$$\textbf{Precision=Q/Q+R} \quad (2)$$

Q divided by Q + T can be used as Recall given below:

$$\textbf{Recall=Q/Q+T} \quad (3)$$

F1-score can be calculated as

$$\textbf{F1-score=2*Q/Q+R*Q/Q+T/Q/Q+R+Q/Q+T} \quad (4)$$

4.1 Performance Analysis

In this section depicted the performance matrix such as precision, recall, F1-score, and accuracy of diseased tomato leaf. The following Table 7 shows a performance analysis of each class of diseased tomato leaf based on every leaf for precision, recall, and f1-score.

Table 7: The performance analysis of each class of tomato leaf diseases collected from the field using an Improved CNN

Class	*"precision"*	*"recall"*	*"F1-score"*	*"support"*
Alternaria	0.94	0.98	0.96	116
Blight	0.99	0.99	0.99	308
Cutwork infected	0.99	0.96	0.98	147
Healthy	0.97	1.00	0.98	30
Curl	0.99	0.99	0.99	335
Miner	0.99	0.99	0.99	431
Spot	0.98	0.95	0.96	172

The following Figure 7 shows the accuracy of both training and validation set of our Improved CNN model. Improved Convolutional Neural Network gives 99.98% training accuracy and 98.4% validation accuracy with fewer number of parameters.

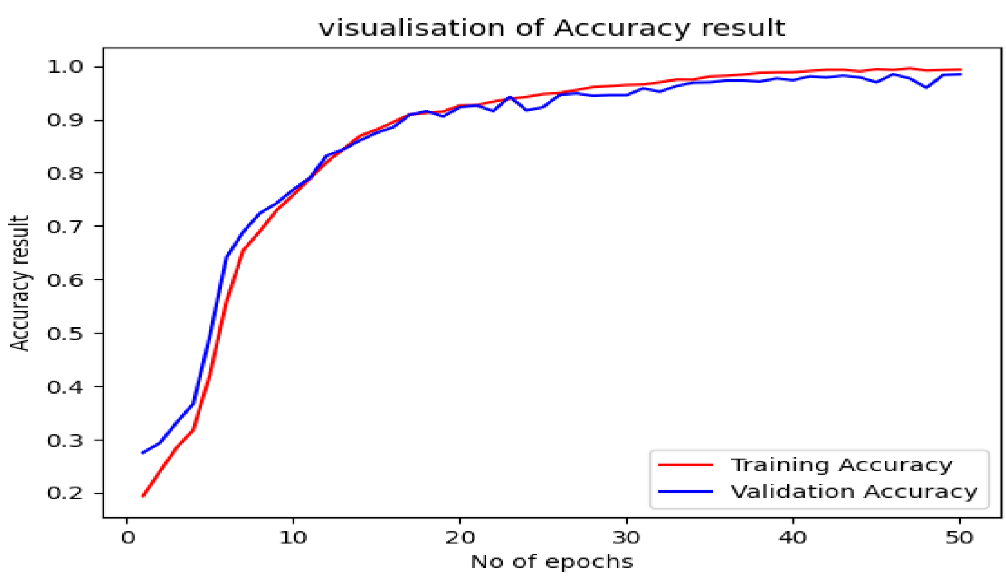


Figure 7: Using augmentation of images of diseased tomato leaf collected from the field, the accuracy of the training set and validation sets of Improved CNN is visualized.

The following Figure 8 shows that our Improved CNN Model can identify the name of the diseased tomato leaf or not according to the dataset shown in Figure 2. Our Improved CNN can identify the correct name of the diseased tomato leaf.

Figure 8: Tested images of blight and cutwork infected leaf.

Similarly, We utilize the confusion matrix to show the Improved CNN using images of tomato leaf collected from internet sources (plant village dataset) class-wise performance to assess its performance. The following Figure 9 shows the heat map or confusion matrix of each class of diseased tomato leaf starting from class 0 to class 9

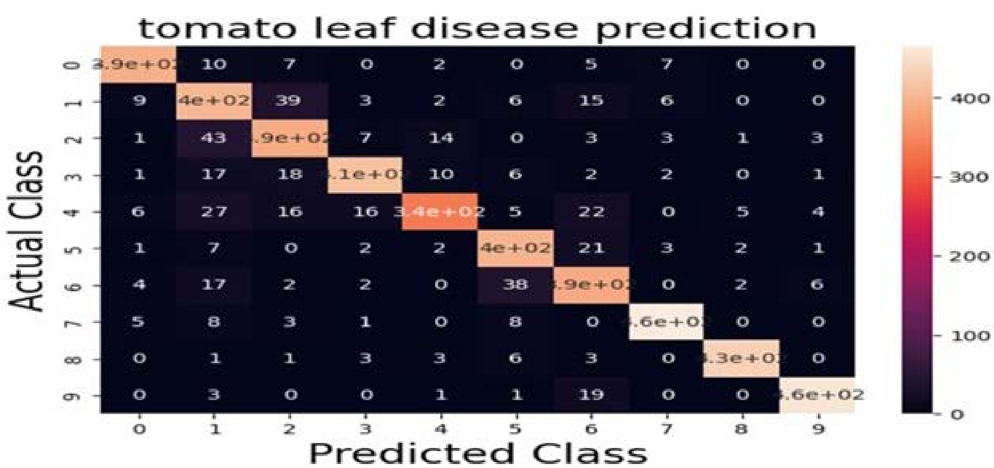


Figure 9: Matrix of confusion of Improved CNN using images of tomato leaf collected from the internet sources(plant village dataset).

The following Table 8 shows a performance analysis of each class of diseased tomato leaf based on every leaf for precision, recall, and f1-score

Table 8: The performance analysis of each class of tomato leaf diseases of plant village dataset (internet sources) using an Improved CNN

Class	*"precision"*	*"recall"*	*"F1-score"*	*"support"*
Tomato bacterial spot	.94	.93	.93	425
Tomato early blight	.75	.83	.79	480
Tomato late blight	.82	.84	.83	463
Tomato leaf mold	.92	.88	.90	470
Septoria leaf spot	.91	.77	.83	436
Spider mites	.85	.91	.88	435
Target Spot	.81	.84	.83	457
Yellow leaf curl virus	.96	.95	.95	490
Tomato mosaic virus	.98	.96	.97	448
Tomato healthy	.97	.95	.96	481

The following Figure 10 shows the accuracy of both training and validation set of our Improved CNN model. Improved Convolutional Neural Network gives 99.68% training accuracy and 89% validation accuracy with fewer number of parameters

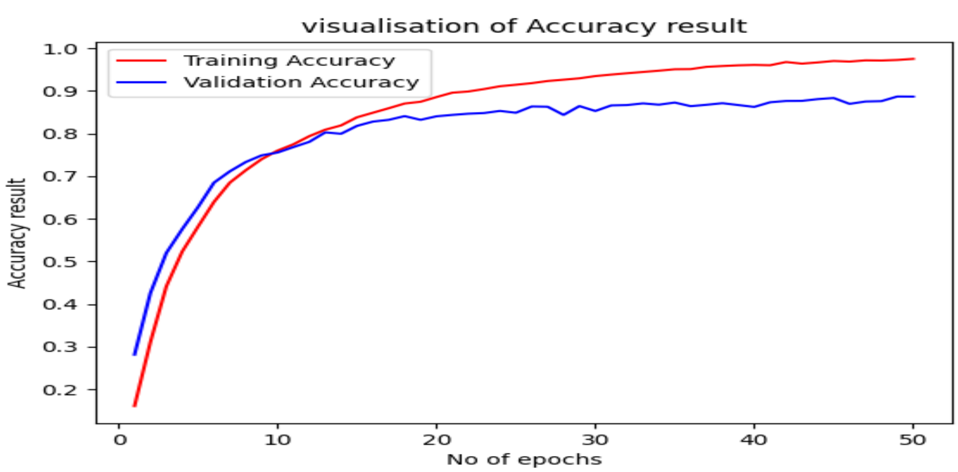


Figure 10: The accuracy of the training set and validation sets of disease tomato leaf collected from internet sources by using Improved CNN is visualized.

The following Figure 11 shows that our Improved CNN Model can identify the name of the diseased tomato leaf or not according to the dataset shown in Figure 5. Our Improved CNN is able to identify the correct name of the diseased tomato leaf.

Figure 11: Tested image of septoria leaf spot.

5 Conclusion

In this paper, we improved CNN to increase the interpretation of the model with 27807 trainable parameters and reduced the computational time. To minimize the loss of significant features we lowered the number of pooling layers in the third block of Improved CNN. We checked the interpretation of the Improved CNN model using photos of diseased tomato leaf collected from the field. The Improved CNN outperforms with an overall training accuracy of ninety nine point nine eight percent (99.98%) and validating accuracy of ninety eight point four percent (98.4%). Also, our improved CNN outperform with an overall training accuracy of 99.68% and validation accuracy of 89% with 152850 trainable parameters using images of diseased tomato leaf collected from the internet sources (plant village dataset). Therefore, we can conclude that our model performs better accuracy in terms of training and validation accuracy and can identify the name of the disease corrected when tested with images of diseased tomato leaf collected from the field as well as the images of diseased tomato leaf collected from internet sources.

Authors' Contribution: All the authors contributed equally

Funding: This research received no grant from any funding agency

Conflicts of Interest: The authors report no conflict of interest

Acknowledgments: Special thanks were extended to the School of Technology, Department Of Computer Science, University of Science and Technology, Meghalaya.

References

[1] Deepalakshmi P,etal(2021).”Plant leaf Disease Detection Using CNN”. *International Journal of Information System Modeling and Design (IJISMD)* *12(1)*. DOI: 10.4018/IJISMD.2021010101

[2]Al-Adhaileh,etal(2023)."Potato Blight Detection Using Fine-Tuned CNN Architecture" *Mathematics* 11, no. 6: 1516. https://doi.org/10.3390/math11061516

[3] Fatihah Sahidan, Nurul, et al(2019). "Flower and leaf recognition for plant identification using convolutional neural network." *Indonesian Journal of Electrical Engineering and Computer Science* Vol.16,No.2 ,pp:737-743. ISSN: 2502-4752, DOI: 10.11591/ijeecs.v16.i2.pp737-743

[4] Geetharamani, G., & J.Arun Pandian. (2019). Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Computers & Electrical Engineering*.vol.76 pp 323-338

[5] Paryadi, C., Diqi, M., & Mulyani, S. H. (2021). Implementation of CNN for plant leaf classification. *International Journal of Informatics and Computation (IJICOM) Vol*, *2*.

[6] R. Akter & M. I. Hosen, "CNN-based Leaf Image Classification for Bangladeshi Medicinal Plant Recognition," *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, Bangladesh, 2020, pp. 1-6, doi: 10.1109/ETCCE51779.2020.9350900.

[7] Selvam.L.,& Kavitha.P.(2020). Classification of ladies finger plant leaf using deep learning. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-02671-y

[8] Nandhini, S.et al (2021). Automatic Detection of Leaf Disease Using CNN Algorithm. *Machine Learning for Predictive Analysis*. https://doi.org/10.1007/978-981-15-7106-0_24

[9] Anandhakrishnan, T., & Jaisakthi, S. M.(2022) . Deep Convolutional Neural Networks for image based tomato leaf disease detection. *Sustainable Chemistry and Pharmacy*, *30*, 100793.

[10] Bisen, D (2021). Deep convolutional neural network based plant species recognition through features of leaf. *Multimedia Tool and Application*, pp:6443–6456. https://doi.org/10.1007/s11042-020-10038-w

[11] Arivazhagan, S., & Ligi, S. V. (2018). Mango leaf diseases identification using convolutional neural network. *International Journal of Pure and Applied Mathematics*, *120*(6), 11067-11079.

[12] V. K. Vishnoi,et al(2022), Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network ,in *IEEEAccess*,vol.11, pp. 6594-6609.

[13]Trivedi, N. K,etal(2021). Early detection and classification of tomato leaf disease using high-performance deep neural network. IEEE *Sensors*, *21*(23)

[14] Singh, R.K., Tiwari, A. & Gupta, R.K.(2022). Deep transfer modelling for the classification of maize diseases. *Multimedia tools and applications*. https://doi.org/10.1007/s11042-021-117636

[15] Upadhyay, S.K.& Kumar, A(2022).A novel approach for rice plant diseases classification with deep convolutional neural network. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-021-00817-5

[16] Baser.P,et al(2023). TomConv: An Improved CNN Model for Diagnosis of Diseases in Tomato Plant Leaves,Procedia Computer Science,Volume 218,pp.1825-1833, ISSN 1877-0509,

https://doi.org/10.1016/j.procs.2023.01.160.

[17] Juncheng Ma,et al(2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network, *Computers and Electronics in Agriculture*, Vol.154, pp18-24,ISSN 0168-1699.

[18] Praveena, S., *et al(2024)*. CNN-based Indian medicinal leaf type identification and medical use recommendation. *Neural Computing and Application*. https://doi.org/10.1007/s00521-023-093529

[19] Rai, C.K., Pahuja, R.(2023) Classification of Diseased Cotton Leaves and Plants Using Improved Deep Convolutional Neural Network. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-023-14933-w

[20]Khan, S., & Narvekar, M..(2020). Disorder detection in tomato plant using deep learning. In *Advanced computing technologies and applications* (pp. 187-197).

[21] Math, R. M., & Dharwadkar, N. V. (2022). Early detection and identification of grape diseases using convolutional neural networks. *Journal of Plant Diseases and Protection*, *129*(3), 521-532.

[22] Fajri, D. M. N., Mahmudy, W. F., & Yulianti, T. (2021). Detection of Disease and Pest of Kenaf Plant using Convolutional Neural Network. *Journal of Information Technology and Computer Science*, *6*(1), 18-24.
[23] Pravin, A., & Deepa, C. (2022). Piper plant classification using deep cnn feature extraction and hyperparameter tuned random forest classification. *Transdisciplinary Journal of Engineering & Science*, *13,233-258.* doi: 10.22545/2022/00202

About the Authors

Mr.Debabrat Bharali,Research Scholar,Department of Computer Science,University of Science and Technology,Megalaya,India completed B.Tech (IT) from North Eastern Hill University and Mtech(IT) from Gauhati University.He is doing Phd in University of Science and Technology,Megalaya under the supervision of Dr kanak Chandra Bora and co supervisor of Dr.Rashel Sarkar.Doing his research He published one paper in scopus journal and currently he has 3 national journals and one conference paper.

Dr.Kanak Chandra Bora,Associate Professor, Department of Computer Science,University of Science and Technology,Megalaya,India.He completed his Phd in the field of Graph Theory,from Gauhati University.He has an experience of about 20 years in the field of academics and teaching.He published many paper in scopus,springer as well as Elsevier journal and many more.

Dr.Rashel Sarkar,Associate Professor, Department of Computer Science and Engineering,Royal Global University ,Assam ,India.He completed his Phd in the field of Neural Network,from Gharwal University.He has an experience of about 15 years in the field of academics and teaching.He published many paper in scopus,springer etc.