

# DETECTION OF PLANT DISEASES IN TOMATO LEAVES

## ABSTRACT

The transmission of diseases from unhealthy to healthy plants is one of the most disastrous threats to the agriculture industry. Diseases transferred spread like wildfire and have the potential to infest the whole farm if not detected early. Plant disease detection methods aid in identifying infected plants in their very early stages and also help the user in scaling the identification of plant diseases to a variety of plants in a cost-effective manner. Rapid improvements in deep learning (Deep Learning) techniques have made it possible to detect and recognize objects from images. Deep Learning approaches have recently entered various agricultural and farming applications after being successfully employed in various fields. Automatic identification of plant diseases can help farmers manage their crops more effectively, resulting in higher yields. Detecting plant disease in crops using images is an intrinsically difficult task. In addition to their detection, individual species identification is necessary for applying tailored control methods. In addition, we identified and summarized several problems and solutions corresponding to the CNN used in plant leaf disease detection.

## 1. Introduction

Plant diseases are one of the most critical elements impacting food production. They are responsible for a significant drop in the economic productivity of crops, as well as being an obstruction to this activity in some cases. According to, disease management and control procedures must be carried out effectively to reduce output losses and ensure agricultural sustainability, underlining the importance of continual crop monitoring paired with prompt and accurate disease detection. In addition, as the world's population continues to rise, a significant increase in food production is required (FAO). This must be combined with the preservation of natural ecosystems through the use of environmentally-friendly farming methods. Food must keep a high nutritious value while still being secure worldwide. This can be accomplished by using new scientific methodologies for leaf disease diagnosis and crop management, as well as applying these new technologies to large-scale ecosystem monitoring.

Tomatoes (biological name: *Solanum lycopersicum*) grows on mostly any well drained soil and Nine out of 10 farmers grow tomatoes in their field. Many gardeners also grow tomatoes in their garden to use fresh grown tomato in their kitchens and get a good taste of food. However, farmers and gardeners are sometimes unable to get proper progress of the plant growth. The tomatoes may not sometime appear on plant or sometimes the tomatoes may get bad looking and disease-ful black spots at bottom part.

The identification of tomato plant disease may start by, diagnosing the portion having infection in the plant then noting the differences such as brown or black patches and holes

on the plant, and then looking for the insects also. Tomatoes and similar vegetables like potatoes or brinjal must not be planted on the same farm for more than one time in three years. To maintain soil fertility, we should ideally precede tomato planting by any member of the grass family e.g. wheat, corn, rice, sugarcane, etc.

In the research article, we have proposed a novel method to identify the disease in tomato crops after analyzing the images of leaves. The work will solve farmers' problems of plant disease identification without running after plant scientists. It will thus help them cure the plant's disease in a timely fashion and will thus increase both the quality and quantity of food crops produced and therefore help in increasing farmers' profit. For the experiment purpose, we have downloaded the tomato leaves dataset from plant village. After downloading the dataset, we developed a Convolution Neural Network model to classify the images. The performance of the model has been analyzed based on various parameters such as training accuracy, validation accuracy, and testing accuracy and a number of trainable and number trainable parameters with respect to the pre-trained model. The rest of the paper is organized as follows: The survey of related studies is discussed in section 2. Discussion of the dataset is presented in section 3. Experimental setup and discussion of the results are discussed, presented in section 4, followed by the conclusion.

## 2. Related Work

Many studies have been conducted to find an ideal solution to the problem of crop disease detection by creating techniques that can assist in identifying crops in an agricultural environment. This section will provide the most recently reviewed studies on CNN's applicability in the broad field of agriculture; this section includes papers from peer-reviewed articles that use CNN methods and plant datasets.

Abade reviewed CNN algorithms for the detection of plant diseases. The authors studied 121 papers that were published between 2010 and 2019. PlantVillage was selected as the most widely used dataset, while TensorFlow was identified as the most frequently used framework in this review. Dhaka outlined the basic methods of CNN models used to identify plant diseases using leaf images. They also compared CNN models, pre-processing approaches, and frameworks. The study also looks at the datasets and performance measures used to assess model efficiency. Moreover, Nagaraju et al also provided a review to find the best datasets, pre-processing approaches, and DL techniques for various plants. They reviewed and analyzed 84 papers on DL's applicability in plant disease diagnosis. They observed that so many DL methods are limited in their ability to analyze original images and that effective model performance necessitates using a suitable pre-processing technique.

Kamilaris found that DL approaches were used to solve various agricultural challenges. According to the study, DL methods performed better than standard image processing techniques. Fernandez-Quintanilla evaluated weed-monitoring technologies in crops. They focused on weed monitoring devices in agricultural fields that were both remote-sensed and ground-based. Weed monitoring is critical for weed control, according to them. They predicted that data acquired by various sensors would be saved in a public cloud and used in appropriate contexts at the optimal time. Lu et al. introduced a review for plant disease classification using a CNN. They evaluated the significant problems and solutions of CNN used for plant disease classification and DL criteria in plant disease classification. They discovered that additional research with more complex datasets was required to obtain a more satisfactory result.

Golhani presented a review paper on hyperspectral data for plant leaf disease identification, highlighting existing problems and potential prospects. They also presented NN approaches for SDI development in a short time. They discovered that, as long as SDIs remain relevant for proper crop protection, they must be tested on various hyperspectral sensors at the plant leaf scale. Bangari presented a review on disease detection using CNN, focusing on potato leaf disease. They reviewed several papers and concluded that convolutional neural networks work better at detecting disease. They also identified that CNN contributed significantly to the maximum possible accuracy for disease identification.

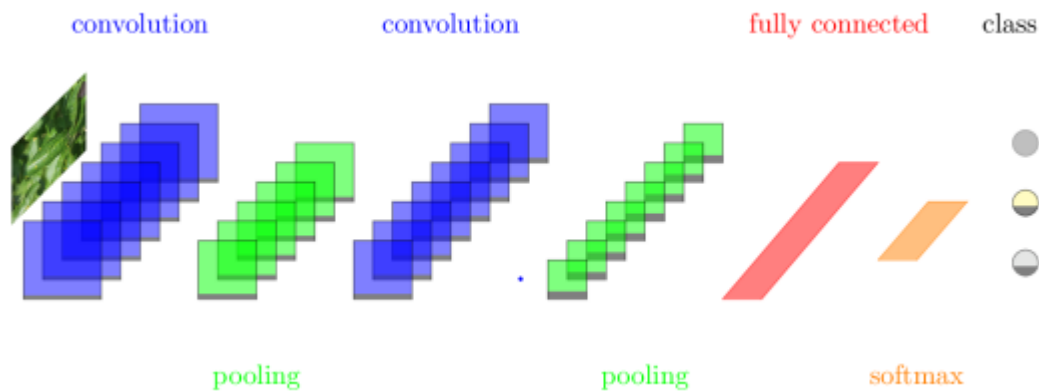
### 3.Convolutional Neural Networks (CNN)

Regular Neural Networks such as the Multi-layer Perceptron (MLP), in the past were used for image classification purposes however as the resolution of the images being used to classify became higher and higher the networks became computationally hard to deal with and the number of total parameters used for classification would be far too many.

Convolutional Neural Networks are very similar in working to regular neural networks such as the MLP, however, what changes in Convolutional Neural Networks is that the layers of a CNN have three dimensional arrangement (width, height and depth) of neurons instead of the standard two-dimensional array and for this simple reason CNNs are widely used on image data for the purpose of classification as the architecture of a CNN is designed to take advantage of the 3d form of an image.

A simple Convolutional Neural Network's architecture consists of three main layers, namely, Convolutional layer, Pooling layer and the Fully connected layer. The Convolutional layer is regarded as the main building block of a CNN, it consists of learnable parameters known as filters/kernels. The filter is responsible for finding patterns

(textures, edges, shapes, objects, etc) in the input image. Each filter slides/convolves over the height and width of the input image, computing the dot product between the filter and the pixels present in the input image. The resultant of a Convolutional layer is a feature map that summarizes all the features found in the input image.



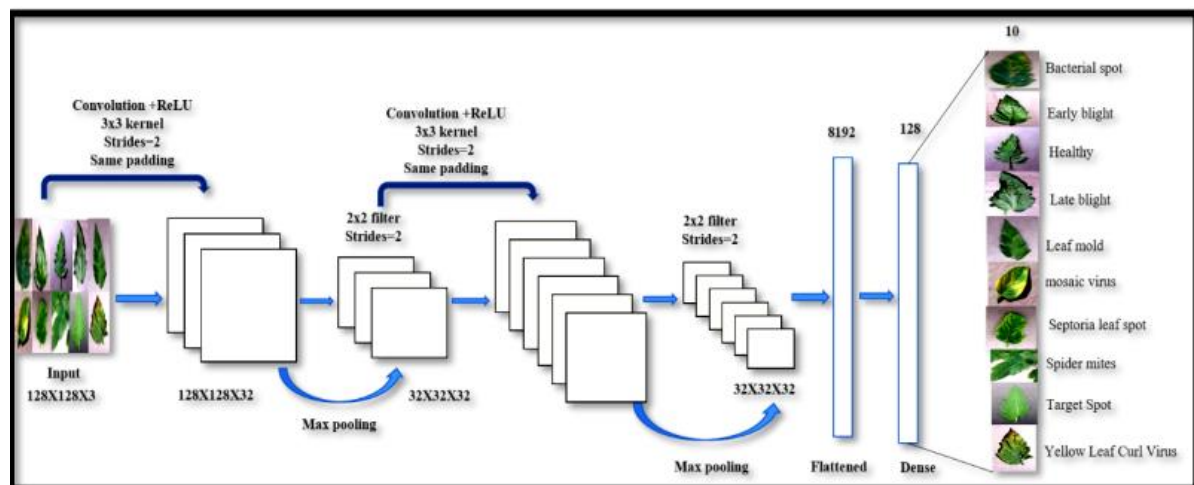
**Figure 1. A common CNN architecture**

The Pooling layer is another building block of a CNN that is used to perform downsampling in order to reduce the spatial size of the feature map. This is done to reduce the number of parameters, computations in the network and also to make sure overfitting is controlled. There are two different types of pooling in CNN namely, Max Pooling and Average Pooling. Max Pooling returns the maximum value that is present in the portion of the image convolved by the kernel and Average Pooling returns the average of all the values present in the portion of the image convolved by the kernel.

This layer is often followed only after non-linearity is introduced into the network using a Rectified Linear Unit (ReLU) activation function, this layer is used to turn all the negative values present in the feature map into zeros. The addition of ReLU is very important in a CNN as it helps increase the non-linearity in the images. Images as such are known to contain non-linear features such as different objects or the boundaries between those objects, however, when imposing an image through a Convolutional layer to create a feature map, there might be some linearity introduced in the image and in order to bring back the non-linearity in the image the ReLU activation function is used.

After adding multiple Convolutional layers and Pooling layers, the fully connected layers are introduced in the architecture. The input coming to the fully connected layer is the flattened output from the final convolutional and pooling layer. The flattening is done to convert the 3d matrix data from the last pooling layer into a 1d array of vectors so that it can be used by the fully connected layers which perform the same operations to an ANN

to carry out the final classification and compute the class scores. And in this way, a simple Convolutional Neural network is structured to read and classify images.



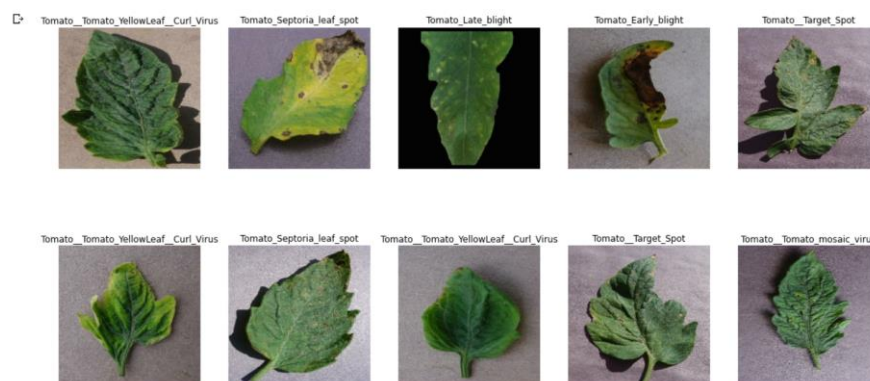
**Figure 2. The Architecture of Convolutional Neural Network**

As a result, the convolutional layers extract features from the input images, which are then reduced in dimensionality by the pooling layers. Typically, the fully connected layers use the high-level features learned to classify input images into predefined classes at the final layer. Moreover, the classification layer can extract features for classification and detection tasks. Figure 1 provides an overview of the architecture of a typical CNN.

#### 4. Dataset

Images of Tomato disease have been taken from Plant Village dataset. The dataset includes over 50,000 images of 14 crops, such as tomatoes, potatoes, grapes, apples, corn, blueberry, raspberry, soybeans, squash and strawberry. We selected tomato as our target crop.

Pictures of diseases on tomato leaves are as follows (refer to Figure 2).



**Fig. 3. Class wise sample image of the dataset.**

There are mainly nine types of diseases in tomato:

1. Tomato\_Bacterial\_spot
2. Tomato\_Early\_blight
3. Tomato\_Late\_blight
4. Tomato\_Leaf\_Mold
5. Tomato\_Septoria\_leaf\_spot
6. Tomato\_Spider\_mites\_Two\_spotted\_spider\_mite
7. Tomato\_\_Target\_Spot
8. Tomato\_\_Tomato\_YellowLeaf\_\_Curl\_Virus
9. Tomato\_\_Tomato\_mosaic\_virus
10. Tomato\_healthy.

In proposed work, there are 10000 images in training dataset, 7000 images in validation dataset and 500 images in testing dataset. Out of 10000 training images, 1000 images belong to healthy category and 1000 images belong to each tomato disease category described above. In validation set each class has 700 images and test set has 50 images in each class.

For testing, we randomly picked 50 images from each class from training set and removed them from those folders. From remaining training dataset, we built our project training dataset by putting same number of images (1000) in each class. When the images in any class were less than 1000, we used data augmentation technique to generate some new images. Augmentation was done using Augmentor package of python and it helps to build similar new images by rotating, flipping, cropping and resizing the existing images. When images in any class in training dataset were more than 1000, we picked first 1000 images. We followed same process for validation dataset and made all classes have 700 images each. This process is necessary to prevent bias for any particular class during training of CNN. Size of all the images is  $256 \times 256$  and format is jpeg.

### **Pre-processing of the Dataset:**

The pre-processing step for any machine learning model is of great importance and ideally shapes the performance and results of the models chosen. In this report, the following were the steps that were carried out in order to make sure that the models produced optimal results.

- Read images – cv2.imread() was used to read the images from the root directory that was specified.

- Resize images – cv2.resize() was used in order to change the dimensions of the images. The images were 256 x 256 in size initially and were resized to 150 x 150 due to the inability of the system's (Random Access Memory) RAM to carry out operations on all 10,000 images of size 256 x 256 using both CNN and KNN model.
- After reading and resizing the images, we then convert the images into an array form using np.array().
- The labels of each plant images are then also mapped to a unique value using LabelBinarizer().
- Finally, the plant village dataset is split into two different sets, namely, train and test set with a 75:25 ratio respectively.

## 5.Evaluation Metrics

The accuracy, precision, recall, and f1-score of the proposed method were all evaluated. The proposed plant recognition system's accuracy has been calculated using the following expression, which incorporates numerical details such as true positive (TP) (the number of correctly identified leaf images), false positive (FP) (the number of incorrectly detected leaves), true negative (TN) (the number of correctly detected leaf images), and false negative (FN) (it is a parameter for representation of the number of leaf images that are correctly recognized).

**Accuracy:** Accuracy is the number of right predictions that are made by the model with respect to the total number of predictions that were made. It is mathematically represented by Equation.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

**Precision:** Precision is defined as the number of true positive results (TP) divided by the number of positive results (TP + FP) that are predicted by the model. The range of the precision is between 0 and 1 and is calculated using Equation (2). It is used to find the proportion of positive identifications that is true.

**Recall:** The recall is the number of true positives (TP) divided by the number of all relevant sample data (TP + FN). Equation (3) represents the mode of calculation of the recall. It is used to determine the proportion of actual positives that were correctly identified. These concepts are represented mathematically by Equations (2) and (3), respectively:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$



$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

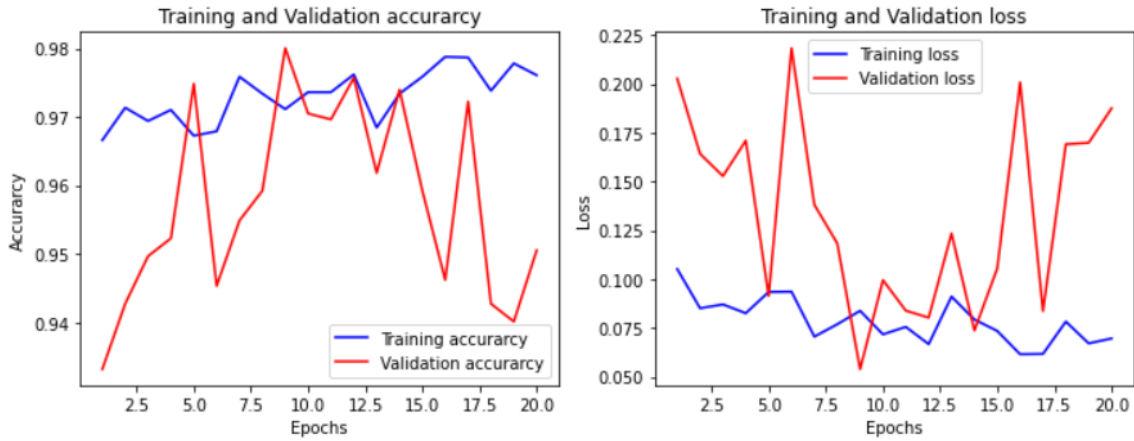
F1 Score: Being one of the widely used metrics for the performance evaluation of machine learning algorithms, the F1 score is the harmonic mean of precision and recall. The range of the F1 score is between 0 and 1, and it is calculated as shown by Equation (4). It reflects the number of instances that are correctly classified by the learning model.

$$F1 = \frac{2 \times P \times R}{P+R} \quad (4)$$

## 6. Experimental result and Discussion

The complete experiment was performed on Google Colab. The result of the proposed method is described with different test epochs and learning rates and explained in the next subsection.

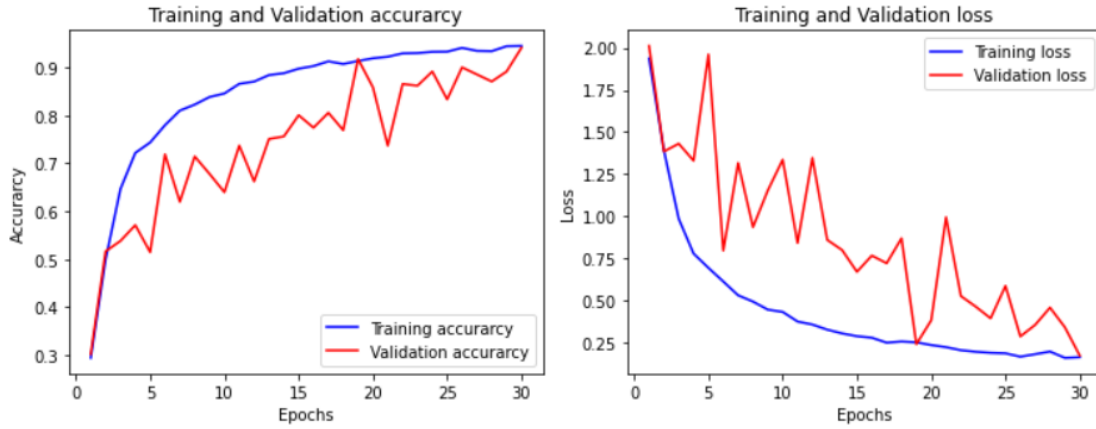
This research used epoch 20 and epoch 30 for comparison, though learning rates were 0.0001. Figure 3a shows the comparison between training and validation loss, and Figure 3b shows the comparison between training accuracy and validation accuracy.



**Figure 4. (a) Training accuracy vs validation accuracy (rate of learning 0.0001 and epoch 20).  
(b) Training loss vs validation loss (rate of learning 0.0001 and epoch 20).**

Figure 4a shows the comparison between training loss and validation loss, and Figure 4b shows training accuracy and validation accuracy. Here, Figure 4b shows that the accuracy rate of 94.00% is achieved with a training step at 30 epochs, and the rate of learning is 0.0001. Therefore, it is reasonable to infer that more iterations will result in higher data accuracy based on the research technique. However, the number of epochs increases as the training phase lengthens.





**Figure 5. (a) Training accuracy vs validation accuracy (rate of learning 0.0001 and epoch 30)**  
**(b) Training loss vs validation loss (rate of learning 0.0001 and epoch 30).**

tomato bacterial spot	99	0	0	0	0	0	0	1	0	0
tomato early blight	0	100	0	0	0	0	2	0	0	0
tomato late blight	1	1	98	0	0	0	0	1	0	0
tomato leaf mold	0	2	0	100	1	0	0	2	0	0
tomato Septoria leaf spot	0	0	0	0	98	0	0	0	0	0
two-spotted spider mite	0	0	0	0	0	100	0	0	0	0
tomato target spot	0	0	0	0	0	2	98	0	0	0
tomato yellow leaf curl virus	1	0	0	0	0	0	0	99	0	0
tomato mosaic virus	0	0	0	0	0	0	0	0	100	0
tomato healthy	1	0	0	0	0	0	0	0	0	100
	tomato bacterial spot	tomato early blight	tomato late blight	tomato leaf mold	tomato Septoria leaf spot	two-spotted spider mite	tomato target spot	tomato yellow leaf curl virus	tomato mosaic virus	tomato healthy

**Figure 6. Confusion matrix of the proposed model.**

According to the results, which are reflected in the confusion matrix, we can see that the proposed model was able to predict half of the classes that were evaluated using the

test dataset with a 100% accuracy. For the rest of the classes, the model reached an accuracy level of at least 98%, thus obtaining better values than those of several of the works proposed in the literature.

Table 1 presents the results obtained in the classification performance of the proposed model on each of the classes defined within the experimental dataset. According to the data reflected in the table, the value obtained for the recall metric is high for each category defined in the dataset; this allows inferring the performance of the proposed model, which is able to correctly classify the corresponding disease with accuracy higher than 98%.

Class	Precision	Recall	F1 Score	Support
Tomato bacterial spot	0.99	0.990	0.99	100
Tomato early blight	1	1	0.98	100
Tomato late blight	0.97	0.98	0.97	100
Tomato leaf mold	1	1	0.99	100
Tomato Septoria leaf spot	0.99	0.98	0.97	100
Tomato Two-spotted spider mite	1	1	0.98	100
Tomato target spot	0.99	0.98	0.99	100
Tomato yellow leaf curl virus	0.98	0.98	0.98	100
Tomato mosaic virus	1	1	0.99	100
Tomato healthy	1	1	0.99	100

**Table 1. Class-wise Performance of the Proposed Model.**



**Figure 7. The results of the predictive model**

The architecture and weights obtained from the proposed model were saved as a hierarchical data file to be used during the prediction process. The prediction process uses a dataset with a total of 1350 images. The matplotlib library was used to visualize the prediction result. For each prediction, the image, the true result, and the result of the prediction made with the proposed model were displayed, together with the percentage of accuracy. Figure 7 shows some results of the predictions made by the model.

## 7. CONCLUSION

This article has proposed the most efficient and accurate model for the identification of tomato crop diseases. Because of tomato leaf diseases, there is a huge loss in the production of tomato. The proposed method classifies 10 types of tomato crop diseases and a healthier one to gain the productivity and quality of tomato crops. Recently, the deep learning-based approach has become popular for plant disease classification as it automatically extracts features from images. Hence in this study, the deep convolutional network model has been proposed for the identification of tomato crop diseases. To evaluate the performance of the proposed model, the diseased and healthy leaf images of tomato crops were collected from the plantVillage repository. The dataset contains a total of 11,603 images in jpeg format. The dataset is preprocessed and split into a train-test set of 75-25 for this experiment. Finally, the models are trained on training samples and tested the performance on the test samples. Two parameters namely accuracy and loss have been used to compare the performance of models. The experiment results show the proposed deep convolutional neural network outperformed all pre-trained model and the state of art identification model with identification accuracy and loss of 96.35% and 3.65% respectively. Along with, loss and accuracy plot is presented, which shows the loss and accuracy of each model per epochs. This study also presents a confusion matrix, which can be used to visualize the performance of the model for each class of diseases. Finally, the proposed model performance is tested by splitting the dataset into train, validation, and test set. The model achieves validation and test accuracy of 95.23% and 94.12% respectively.

## 8. References

- [1] [https://www.researchgate.net/publication/355030791\\_Tomato\\_Plant\\_Disease\\_Detection\\_and\\_Classification\\_Using\\_Convolutional\\_Neural\\_Network\\_Architectures\\_Technologies](https://www.researchgate.net/publication/355030791_Tomato_Plant_Disease_Detection_and_Classification_Using_Convolutional_Neural_Network_Architectures_Technologies)
- [2] <https://www.hindawi.com/journals/cin/2019/9142753/>
- [3] <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9832884>