Energy and Cost Efficient Pre-trained Convolutional Neural Network (ResNet-50) Model for Tomato Disease Recognition Using Cloud Platform

[1]Sachin Kumar*
PhD, Research Scholar, Department of Computer Application, V.B.S.P.U, Jaunpur, U.P,INDIA
Email: jaiswalsachin009@gmail.com

[2]Saurabh Pal
Department of Computer Application V.B.S.P.U, Jaunpur, U.P,INDIA
Email: drsaurabhpal@yahoo.co.in

[3]Satya Singh
Department of Computer Science & Application M.G.K.V.P, Varanasi, U.P, INDIA
Email: satyasinghmgkvp@gmail.com

[4]Arun Kumar Shahi
Department of Computer Science & Enggi. K.I.T, Varanasi, U.P, INDIA
Email: shahiarun48@gmail.com

[5]Kavita Patel
Department of Computer Science & Enggi. A.I.T.M, Varanasi, U.P, INDIA
Email: kaviata.works22@gmail.com

[6]Priya Jaiswal
V.B.S.P.U, Jaunpur, U.P,INDIA
Email: jaiswalsachin001@gmail.com

Abstract:

Tomato (Solanum Lycopersicum) is a popular crop worldwide. The harvesting of tomatoes is highly profitable for farmers but due to some diseases in a tomato plant, farmers suffer from several problems. Due to these diseases, the overall production of tomato fruits is reduced. The main goal of this research is to raise the overall production of tomato fruits. To accomplish this task, we have to identify the infection in tomato leaves at its initial stage and work to make the tomato plant infection-free. To recognize the disease at its initial stage, we have used an image classification model to resolve such types of problems. We have used a CNN model with residual network-50 as a learning algorithm for tomato disease classification. With the help of this algorithm, we have classified the infected tomato leaves and healthy tomato leaves separately by using MATLAB 2019(a). We have collected 6594 images of tomato leaves for training and evaluation. We have taken fifty percent of the entire data set for training & fifty percent of the dataset for testing purposes for ResNet-50. These leaves are contaminated by six different tomato diseases. After applying the above model, we obtained an effective outcome with 99.83% accuracy in a time span of 47.68 min only. The novelty of this research paper is to calculate time with accuracy. This model can be useful for farmers to defend tomato plants in opposition to disease. Our experiments exhibit the use of a pre-trained model for disease classification with better accuracy.

Keywords: Tomato disease classification, deep convolution neural network, ResNet-50.

DOI Number: 10.14704/nq.2022.20.9.NQ44232 **Neuro Quantology 2022; 20(9):2009-2023**

1. Introduction:

Tomatoes are among the most popular crops around the world. The biological name of the tomato is Solanum Lycopersicum. The harvesting of tomatoes is highly gainful for farmers but due to some diseases in tomato plants, farmers do not get the desired profit. To overcome such types of disease problems there are several image classification algorithms available in deep learning, which show the potential results for disease classification in tomato leaves.

1.1: Steps for the tomato disease identification: There are several steps for the identification of infected tomato leaves; we may start to analyze the part which has an infection in tomato leaves after that to focus on the difference such as black patches and brown spots and then holes on the tomato plant and then to stare for the insect also [1]. Figures 1 given below are the steps for tomato disease identification.

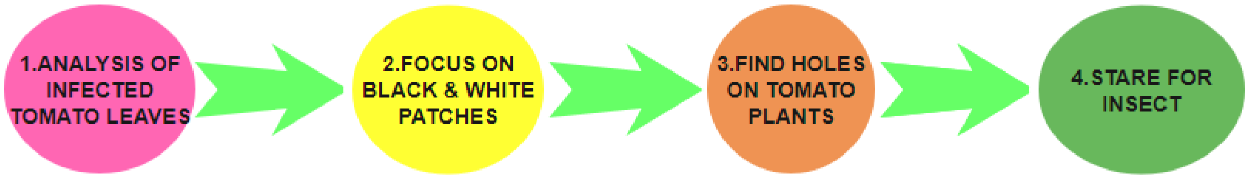

Fig 1: Steps for the tomato disease identification

1.2: Infection in tomato leaves: Infection in tomato leaves: Generally there are several infections in tomato leaves, and due to these infections, the overall productivity of tomato fruits is decreasing. To increase the production of tomato fruits, we should identify the infected leaves at their early stage, and separate the infected leaves from other parts of the tomato plant. After that, we will work to remove the infection from tomato leaves. Figure 2 given below; demonstrate the infection in tomato leaves from several diseases with healthy leaves of tomato.

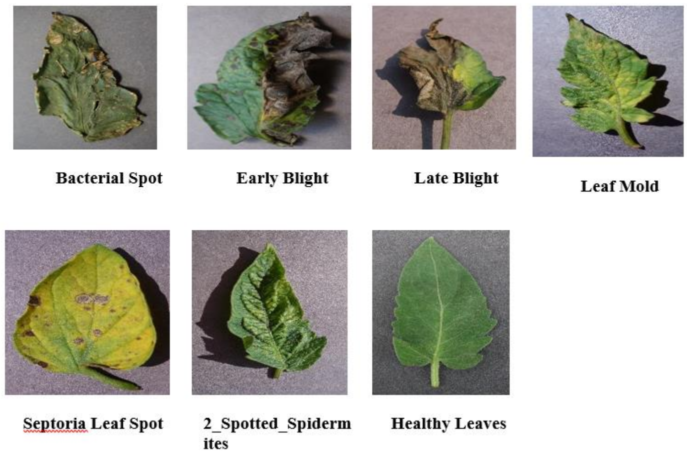

Fig 2: Infected tomato leaves with healthy leaf

- **Early Blight:** It is a frequent tomato infection due to the Alternaria solani fungus. Generally, early blight attacks the tomato plant in continuous rain or in deep dews. Early Blight can infect almost every part of the tomato plants, including the fruits, stems, and leaves. The tomato plants may not die, but they will be damaged and will set fewer tomatoes than usual. Early blight normally attacks older tomato plants. Stressed tomato plants or tomato plants in poor health are particularly susceptible [2].

- **Bacterial Infection:** The bacterial infection in tomato leaves is due to the bacteria named bacterium Xanthomonas vesicatoria. It does not infect red tomatoes. The bacterial infection is more common during wet seasons [3].

- **Late Blight:** Late blight is a severe infection of tomato leaves. It is due to the fungus namely Phytophthora infestans. It is most destructive during wet and cool weather. This fungus can harm every part of the tomato plant. Lesions on tomato leaves are small and appear as water-soaked and dark spots. Infection management strategies mainly depend on hygienic practices and well-timed applications [4].

- **Septoria leaf spot:** Septoria leaf spot is a critical infection of tomato stems, petioles, and foliage (tomato fruit is not infected). It is caused by the fungus named Septoria lycopersici. This infection in tomato leaves close to the earth after tomato plants begin to set tomato fruits [5].

- **Spider mites:** Spider mites are another infection, which is tiny but we can see them with healthy eyes or lenses. This type of infection generally infects the bottom of tomato leaves. There are three familiar symbols of spider mites: white, yellow, and tan spots on the leaf of the plant. If we find the sign that our tomato leaves are infected by spider mites, without any delay separate the tomato plant and take the necessary steps to destroy the spider mites [6].

- **Curl leaf:** Curl leaf in tomatoes is carried by insects named whiteflies. Curling leaves of tomatoes may be due to viral infection. Generally, such a type of disease is transmitted by infected transplants or by whiteflies. There are some common signs of curling tomato leaves: the curling of the tomato leaves in an upward direction and the yellowness of tomato leaves. If we will not cure the tomato leaves of this disease then the production of tomato fruits will be affected [7].

- **Healthy tomato leaves:** We have also taken a dataset of healthy tomato leaves for a better comparison with infected tomato leaf.

The main aim of this paper is to present a deep CNN as a method for identifying tomato infections, focusing on pictures of tomato leaf. This research paper proposed 2 major contributions to tomato plant infection classifications:

i. **Development in the classification of tomato leaves using deep convolutional neural network model:** Deep learning models show better results in image classification. Deep CNN used raw data without feature engineering.

ii. **Recognition of Tomato disease using ResNet-50:** The identification of contaminated tomato leaves assists the users of the proposed model by providing them with information concerning the infection. Also, this organic information is taken out without the interference of farming specialists. Our model performs better with 96.33 % accuracy in 46.853 min only.

In this research article, we project a new technique for identifying infestation in tomato crops after analyzing tomato leaf images. These efforts will solve farmers' problems related to identifying tomato plant infection without going after tomato botanists. Thus, it will assist them to heal the tomato plant infection in a timely manner and thus increase the number & quality of tomato crop production and thus help increase farmers' income. For the purpose of the experiment, we collected images of tomato leaves from a plant village [8]. After collecting the leaf data set, we used a deep CNN model to identify tomato leaf images. The accuracy of the presented model was analyzed based on different parameters for example training accuracy and testing of precision parameters with respect to a pre-trained deep convolutional neural network model.

The remaining part of the research article is ordered as given below:

ii. The literature survey has discussed the technique presented in this section.

iii. Proposed Methodology is explained in this section.

iv. Discussion about the image dataset collection and image dataset preparation is presented in this section.

v. Proposed model and training the model discussed in this part.

vi. Performance Evaluation of projected work discussed in this section.

vii. The Result analysis and result discussion are explained in this section.

viii. Conclusion & Future work about the presented work.

2. Literature Review:

In our proposed manuscript, our focus is to classify tomato disease by using a convolution neural network. This article recommended a convolutional neural network model which can give a satisfactory outcome. There are various algorithms to train a CNN. In this article, 10905 pictures of infected and healthy tomato leaves are used. First of all, we have processed the image data and after that, we have sent the processed data to CNN for training purposes. We have compared some different deep learning architectures. [9-20] after comparison we have found CNN with ResNet-50 performs better in comparison to other models for image classification.

In table 1, we have done a comparison between different transfer learning techniques for image classification of different plant diseases. It will help to make decisions for better transfer learning techniques. After a comparison between different transfer learning techniques, we have found the performance of ResNet-50 is better in comparison to others

TABLE 1: COMPARISON BETWEEN TECHNIQUES FOR CLASSIFICATION OF PLANT DISEASE

Authors	Year	No. of classes	No. of images	CNN Architecture	Measures	Accuracy
Y.Kawasaki,et al [21]	2K15	Three	Eight Hundred	Customized	Sensitivity, Specificity, Accuracy	94.89%
S.Sladojevic, et al.[22]	2K16	Fifteen	4K483	CaffeNet	Accuracy	96.29%
Mohanty et al.[23]	2K16	Thirty Eight	54K306	GoogleNet ,Alexnet	Mean Position, F1 Score, Mean Recall	99.33%
L.G ,Nachtigall, et al.[24]	2K16	Five	1K450	AlexNet	Precision, Accuracy ,Recall	97.29%
E.Fujita,et al.[25]	2K16	Seven	14K840	Customized	Specificity, Accuracy Sensitivity	82.29%
M.Brahmini, et al.[26]	2K17	Nine	14K828	GoogleNet ,AlexNet	Macro Precision, Accuracy, Macro Recall	99.17%
C. De Chant .et al. [27]	2K17	Two	1K796	Customized	Precision, Accuracy, Recall	96.69%
Y. Lu, et al.[28]	2K17	Ten	Five Hundred	AlexNet	Correctness	95.47%
G. Wang, et al.[29]	2K17	Four	20K86	VGG19 ,VGG16, ResNet-50,InceptionV3,	Correctness	90.41%
M. Brahimi, et al. [30]	2K18	Thirty Eight	54K323	DenseNet 169,AlexNet, InceptionV3,Squeeze Net 1.1, ReNet-34,VGG 13	Correctness	99.75%
J. Wang, et al. [31]	2K18	Five	2K430	Customized	Correctness	90.84%
A.K Rangarajan, et al.[32]	2K18	Seven	13K262	VGG 16,AlexNet	Correctness	97.49%
I. Khandelwal, et al.[33]	2K18	Fifty Seven	86K198	ResNet 50,Inception V3	Correctness	99.374%
This Work	2K20	Seven	6K594	ResNet-18,ResNet-50, ResNet-101	Precision, Accuracy, F1 Score, Recall,	99.83%

- In 2019, Ahmed Foysal, Md. Ferdouse et.al printed their conference paper on A Novel Approach for Tomato Diseases Classification Based on Deep Convolutional Neural Networks. In This research paper, the researchers have described various effects and causes of infection in tomatoes [34].

- In 2018, Aravind Krishnaswamy Rangarajan, Raja Purushothaman*, and Anirudh Ramesh published their paper on Tomato crop disease classification using a pre-trained deep learning algorithm [35]. In this paper, they explain the job of images and the implication of weight, bias, and minibatch size, and the rate of learning in the classification implementation time and categorization precision have been analyzed.

- Mohit Agarwala,⁎, Abhishek Singh, Siddhartha Arjariac, Amit Sinha, Suneet Gupta. They proposed Tomato Leaf Disease Detection using the Convolution Neural Network to classify images [36].

- Qimei Wang, Feng Qi, Minghe Sun, Jianhua Qu, Identification of Tomato Disease Types and Detection of Infected Areas Based on Deep Convolutional Neural Networks and Object Detection Technique [8].In this research, object detection models and different deep convolutional neural networks are combined. This model can help to find disease spots and is also be used to recognize the types of tomato disease [37].

- In December – 2019, Mahmoud A. Alajrami, and Samy S. Abu-Naser proposed a paper on the Type of Tomato Classification Using Deep Learning [38].

3. Proposed Methodology:

In this part, we have discussed the procedure to classify tomato disease. With the help of our technique, we have created an algorithm for CNN. The structure of CNN has a small number of sequences like a compilation of RGB images of tomato leaves, segmentation of tomato leaves, feature extraction of tomato leaves, classification of tomato leaves, and the final segment explains the operation and outcome of the job. To compute all these methods we have used MATLAB-2019(a). Figure 3 represents the procedure to classify tomato disease.

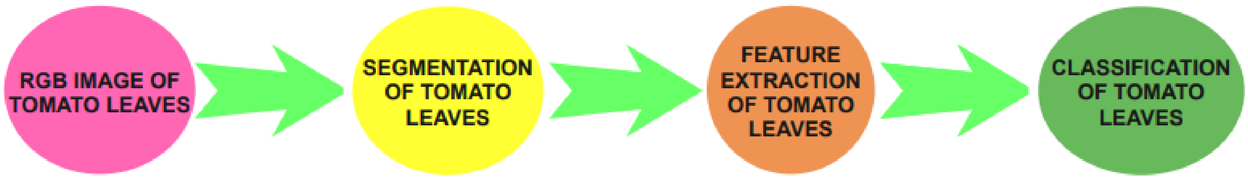

Figure 3: Procedure to classify tomato disease

3. 1: Convolutional Neural Network (CNN): It is an influential machine learning technique; a huge collection of images is required to train CNN. Training our CNN from a big collection of divergent images is not an easy task. However there is an easy way, we can use a pre-trained CNN. Figure 4 given below represents the structure of CNN.

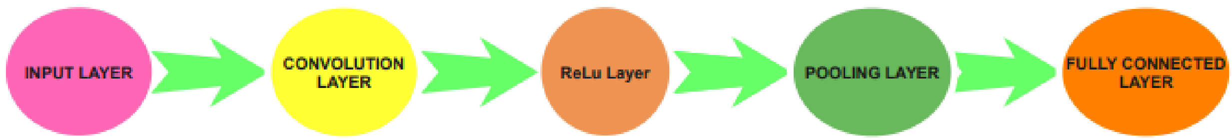

Fig 4: Convolutional Neural Network

Layers of CNN: There are several layers of CNN mentioned below.

A. Input Layer: CNN takes images as input; here the measurement of images is 224 x 224 x 3.

B. Convolution Layer: It describes a collection of filters that accomplish the convolution operations over the whole image. Involve a sequence of convolution operations among an input volume 'I' and a collection of 'n' convolutional filters 'F^E' followed by a non-linear activation. This ultimately yields an output volume 'O' as obtainable in Eq.(2)

$$Om(i,j) = a\left(\left(\sum_{d=1}^{D}\sum_{u=-2k-1}^{2k+1}\sum_{v=-2k-1}^{2k+1} {}_{\mathrm{md}}^{\mathrm{E}}\mathrm{F}(\mathrm{u},\mathrm{v})\ I_d\ (\mathrm{i}-\mathrm{u},\mathrm{j}-\mathrm{v})\right) + b_m\right) \tag{2}$$

After 'Om' is computed, it undergoes a max-pooling operation. Instinctively, each convolutional layer in this architecture learns diverse traits that capture discriminatory patterns to distinguish the type of disease in the tomato leaf.

C. ReLU Layer: The full form of ReLU is the Rectified Linear Unit. ReLu transform function just makes a node active if the input is above a definite extent, when the input is less than zero, the yield is zero but when input is growing up above a definite threshold, it has a linear relationship with the reliant variable. We have reasoned a straightforward function with the values as observed in table 2 given below, so the function just executes a process if that value is obtained by a variable.

Table 3: Rectified Linear Unit

S.NO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
X	-5	-4	-3	-2	-1	0	1	2	3	4	5
F(X)	0	0	0	0	0	0	1	2	3	4	5

For table 2 given above, the subsequent graph is obtained by using MATLAB 2019 (a) mentioned below as figure 5.

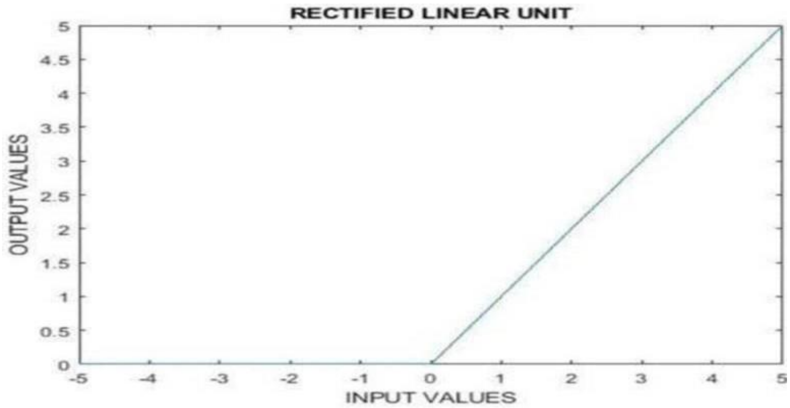

Fig 5: Rectified Linear Unit

3.2: **Pooling Layer:** It usually comes after the activation layer. We have used the max pool layer for our model. If the pool size is 2 x 2 and stride 2 and it is used with the max-pooling layer. We are restricting the x and y-axis by 150 and 170 regions. We have converted a 4 x 4 matrix into four 2 x 2 matrices. Each 2 x 2 matrix has its color code as shown in figure 6. When we pass it through the max pool with 2 x 2 filters with stride 2 then we will get the maximum numerical value of each slice. Finally, we will get a 2 x 2 matrix as output.

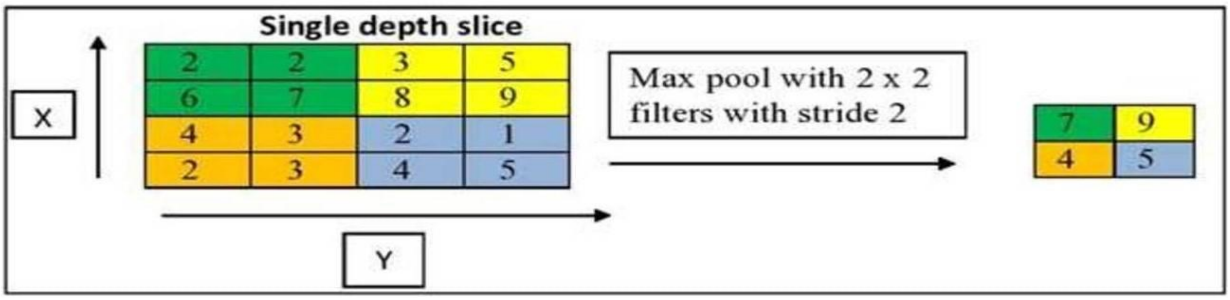

Figure 6: Max Pooling Layer

3.3: Fully connected layer: It perverts the given array into a one-dimensional array. This one-dimensional array is identical to the dimension of the no. of classes.

4. Image Dataset Collection & Image Dataset Preparation:

In this section of our presented paper, we have explained image dataset collection and image dataset preparation (resizing) for processing purposes.

4.1: Image Dataset Collection:

We prepared a dataset from the composed pictures, which contains 6594 pictures. These pictures are infected by six different diseases? We have taken 1978 images of tomato leaves for training the model and 4615 images of tomato leaves for validation purposes. We have classified our image dataset in the order given below in table 3.

TABLE 3: DIFFERENT CLASS OF TOMATO DISEASE

SR.NO	CLASS	NAME OF TOMATO DISEASE
1	A	Tomato Bacterial spot
2	B	Late blight
3	C	Early blight
4	D	Tomato leaf mold
5	E	Tomato septoria leaf spot
6	F	Two spotted spider mite
7	G	Healthy tomato leaves

4.2 : Image Dataset Preparation:

By using a machine-learning algorithm we can train our model on the present image dataset. We required a suitable number of pictures to train our projected model. It is always difficult to have accurate data. If we do not process the data properly, then we will not get the desired output. So we need to arrange it and then launch it to train the model. For our planned model, we have redrawn pictures into 224 x 224 x 3 dimensions by using MATLAB 2019(a) as shown in figure 7.

Fig 7: Resized (224 x 224 x 3) image of infected and healthy Tomato Leaves

5. Proposed Model & Training the Model:

In this part, we have projected a novel model based on a convolutional neural network and trained the model by using ResNet-50.

5.1 Proposed Model:

We proposed a model that is a pre-trained CNN with ResNet-50.It is developed to classify the diseases of tomatoes. In our model, we have used ResNet-50.In our projected model; we designed a deep CNN by merging the transfer learning method with Residual Network-50.In our projected model. We have designed the architecture of deep CNNs, which are trained for object recognition jobs on the ImageNet dataset. There are a total of 50 layers in ResNet-50 which have a total number of 23.6 million trainable parameters.

We would like to emphasize the transfer learning model of residual network-50 with a minor size image dataset that contains patterns of our handled multi-class detection problem.

In Table 4 and fig 8 we have done a comparison of time consumption between ResNet-18, ResNet-50, and ResNet-101 and we found ResNet-18 takes a minimum time i.e 12.461min on 70% training and 30% testing.

TABLE 4: COMPARISON OF ResNet-18, ResNet-50,ResNet-101 FOR TIME CONSUMPTION

SR.NO	TRAINING%	TESTING%	TIME(IN MIN) BY ResNet-18	TIME(IN MIN) BY ResNet-50	TIME(IN MIN) BY ResNet-101
1	50	50	17.396	46.853	61.421
2	60	40	17.349	35.571	68.726
3	70	30	12.461	25.67	48.893
4	80	20	13.302	25.518	43.825

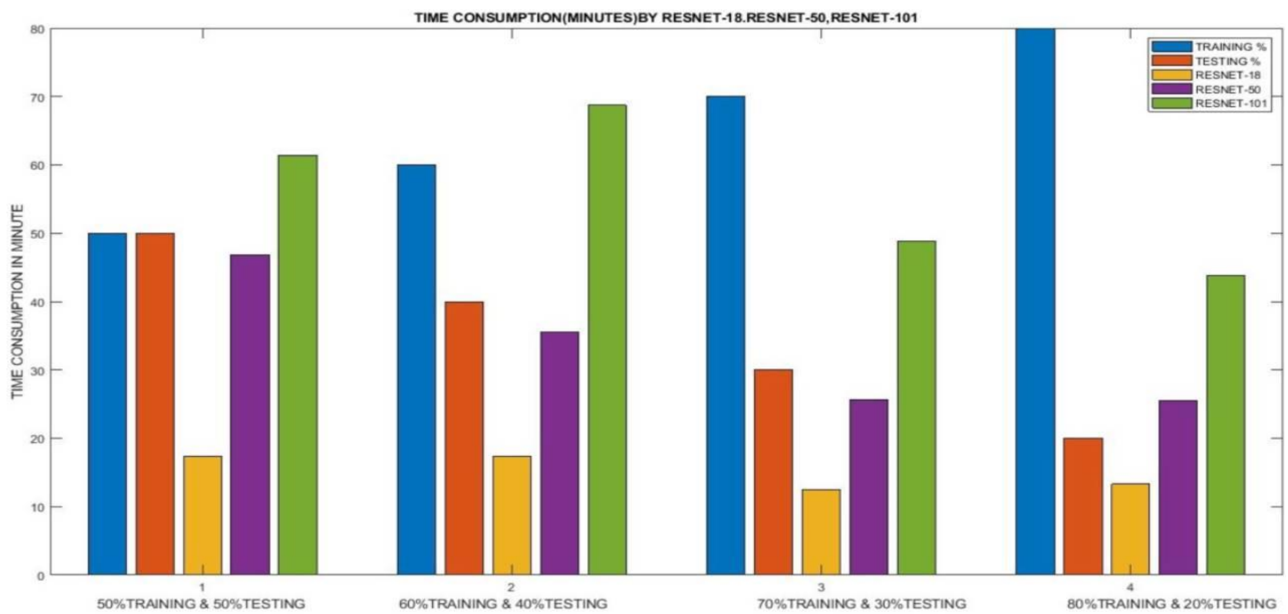

Fig 8: Time Consumed by ResNet-18, ResNet-50, ResNet-101

In Table 5 and Fig 9, we have done a comparison of ResNet-18, ResNet-50, and ResNet-101 for accuracy and we found better accuracy by ResNet-50 i.e 96.33% accuracy on 50% training and 50% testing.

TABLE 5: COMPARISON OF ResNet-18,ResNet-50,ResNet-101 For ACCURACY

SR.NO	TRAINING%	TESTING%	ACCURACY % BY ResNet-18	ACCURACY % BY ResNet-50	ACCURACY % BY ResNet-101
1	50	50	92.05	96.33	94.66
2	60	40	91.44	96.17	93.14
3	70	30	90.31	92.22	93.89
4	80	20	89.59	95.74	95.36

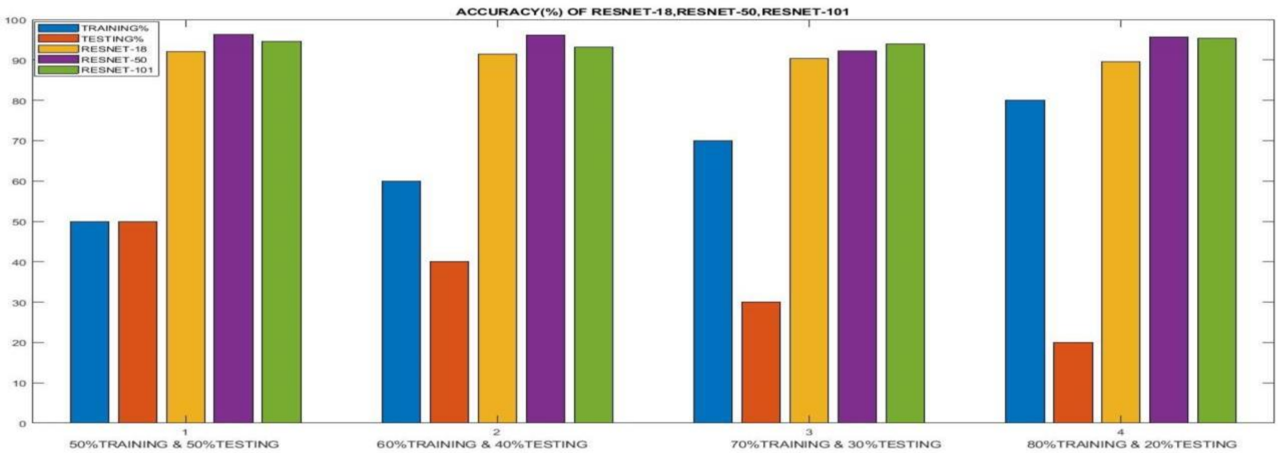

Fig 9: Comparison of ResNet-18, ResNet-50, ResNet-10

5.2 Training the Model:

To train our model we have to load a pre-trained convolutional neural network with ResNet 50. It has been trained on the Imagenet dataset. The imagenet dataset has 1000 object categories and 1.2 million training images. Let's plot the network to visualize its architecture as shown in figure (8). To plot the network, we have used the plot (net) function using MATLAB 2019(a). After using this function we have got network architecture, which is too large in size. Due to this, we have inserted the first section of the network.

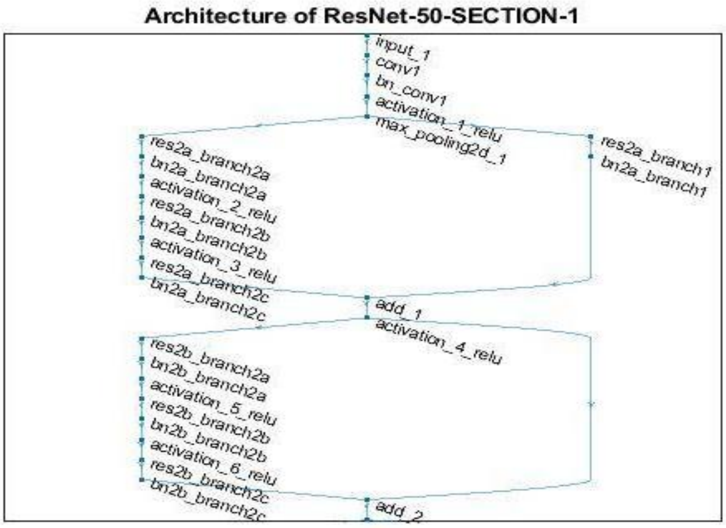

Fig 10: Architecture of ResNet-50

There is a loss-type whose name is "cross-entropyex". Cross entropyex is primarily required for multiclass classification.splitEachLabel ( ) function was used to split training and testing data in our model. We want to use 50% of the image for training and the rest of the images for testing and we will select the image randomly that's why we will use function randomize ( ). It seems complex architecture and difficult to understand. We need to resize the figure. To resize the network we have used the set (gca) function using MATLAB 2019(a).

6. Performance Evaluation:

After validation and training of tomato leaves, we get the desired outcome. When our model is functional on data for training then the precision is said to be training precision. When our model is functional on data for testing from dissimilar classes then the precision is called validation precision. The figure given below demonstrates the first convolutional layer weight. We can observe the weight of the first convolutional layer as shown in figure (9).

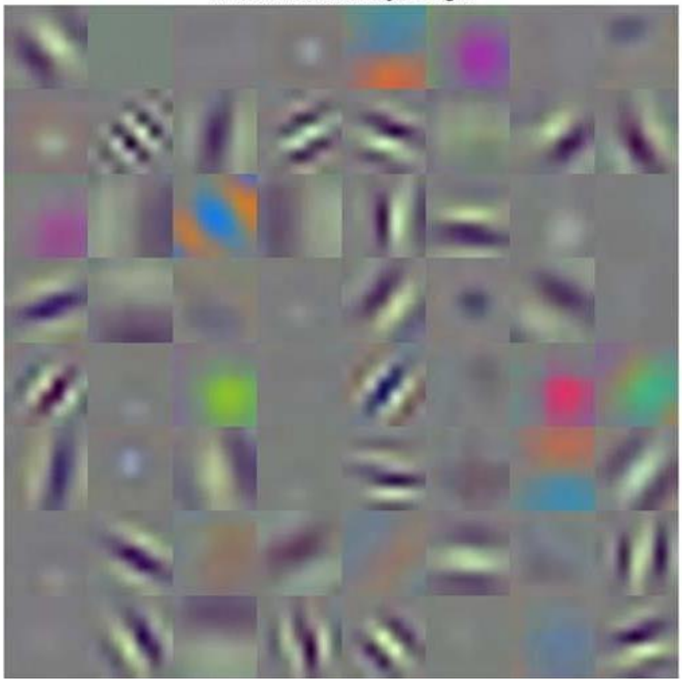

Fig 11: First Convolutional Layer Weight of ResNet-50

7. Result Analysis & Discussion:

We have used 50% of the data for training and 50 % of the data for testing the model using ResNet-50. We have found a LossFunction named 'crossentropyex' as output is shown below.

Output for lossfunction:

ClassificationOutputLayer with properties:

Name:'ClassificationLayer_fc1000'

Classes:[1000x1 categorical]

OutputSize:1000

LossFunction:'crossentropyex'

By using the function confusionmat ( ), we have plotted a confusion matrix. With the help of the confusion matrix, we can show the performance of our algorithm as shown in table 6.

Table 6: Confusion Matrix

CLASS	A	B	C	D	E	F	G
A	186	0	2	0	0	0	0
B	15	139	14	7	8	3	2
C	1	1	181	3	1	1	0
D	2	1	0	182	1	2	0
E	2	0	1	2	183	0	0
F	0	0	0	1	1	186	0
G	0	0	0	0	0	2	186

The execution of the projected technique is assessed by comparing the previously trained deep CNN model with dissimilar metrics. The value of any learning algorithm is calculated by checking how better they execute on test data. Here TP-True Positive; FP: False Positive; TN: True Negative; FN: False Negative; AUC: Area under Curve

• **ROC (Receiver Operating Characteristic Curve):** The first one is the ROC, which is known as AUC, it is broadly used for routine measurement in supervised classification and it totally depends upon the relationship between specificity & sensitivity [14].To perform this work, we have used an Area under the curve for various classes. This function is the resultant of the mean of a number of AUC values [15].

• **Recall (True Positive Rate/Sensitivity):** The recall or sensitivity corresponds to the accurateness of positive examples, and it represents how many numbers of examples of the positive classes were correctly labeled; it can be estimated by using equation number 1. True positives are the number of occurrences, which are correctly identified and positive. FN is the number of positive cases that are wrongly classified as negative.

$$\boldsymbol{Recall\ (Senstivity) = \frac{TP}{TP + FN}} \qquad (1)$$

• **Specificity (True Negative Rate):** Specificity stands for the provisional possibility of true negatives for specified a derived class; the meaning of this is to approximate the negative label being true; it is expressed by equation number 2.

$$\mathbf{Specificity(True\ Negative\ Rate) = \frac{TN}{TN + FP}} \qquad (2)$$

• **Accuracy:** Commonly, Specificity and sensitivity calculate the efficiency of an algorithm on a particular class, negative and positive correspondingly. Generally, for evaluation of classification performance, we measure the accuracy of classification. In the valuation phase, the correctness of classification was measured every twenty iterations. This measurement counts the percentage of the sample that is properly classified and it is clearly expressed in equation number 3.

$$\mathbf{Accuracy} = (\boldsymbol{TP + TN}) \div (\boldsymbol{TP + TN + FP + FN}) \quad (3)$$

• **Precision:** For the evaluation of the predictive power of an algorithm, we have measured the precision value. Precision is defined by the formula given below. This is represented by equation number 4.

$$\mathbf{Precision} = \boldsymbol{\frac{TP}{TP + FP}} \qquad (4)$$

• **F_β Measure:** Fβ is a weighted harmonic mean of precision and sensitivity. We have used the harmonic mean at the place of average because it punishes acute values. Here, we have used the parameter β, which determines the significance of precision over sensitivity. If the value of β is higher it means that more the weight specified to precision. We will set the value of β according to the problems. For some problems, a higher precision value is required. When we will take the value of β equal to .5 and put this value in the equation number (5) then this measurement is said to be f measure or f1 score. It means the value of precision and recall adds equally to resolve the performance of the model.

Table 6 given below represents the value of Recall, False positive rate, Accuracy, Precision, and F1 score, which helps to find out the overall accuracy and area under the curve value.

Table 7: Statistic calculation for the confusion matrix (Table 6)

SR. NO	Class	Recall (Sensitivity)	1-Specificity (FPR)	Accuracy	Precision	F1 Score
1	**A**	0.9	0.0177	0.9833	0.99	0.94
2	**B**	0.99	0.0018	0.9612	0.74	0.84
3	**C**	0.91	0.0151	0.9818	0.96	0.94
4	**D**	0.93	0.0115	0.9856	0.97	0.95
5	**E**	0.94	0.0098	0.9878	0.97	0.96
6	**F**	0.96	0.0071	0.9924	0.99	0.97
7	**G**	0.99	0.0018	0.9970	0.99	0.97

We have also represented table 6 as graph. This is shown below in figure 10.

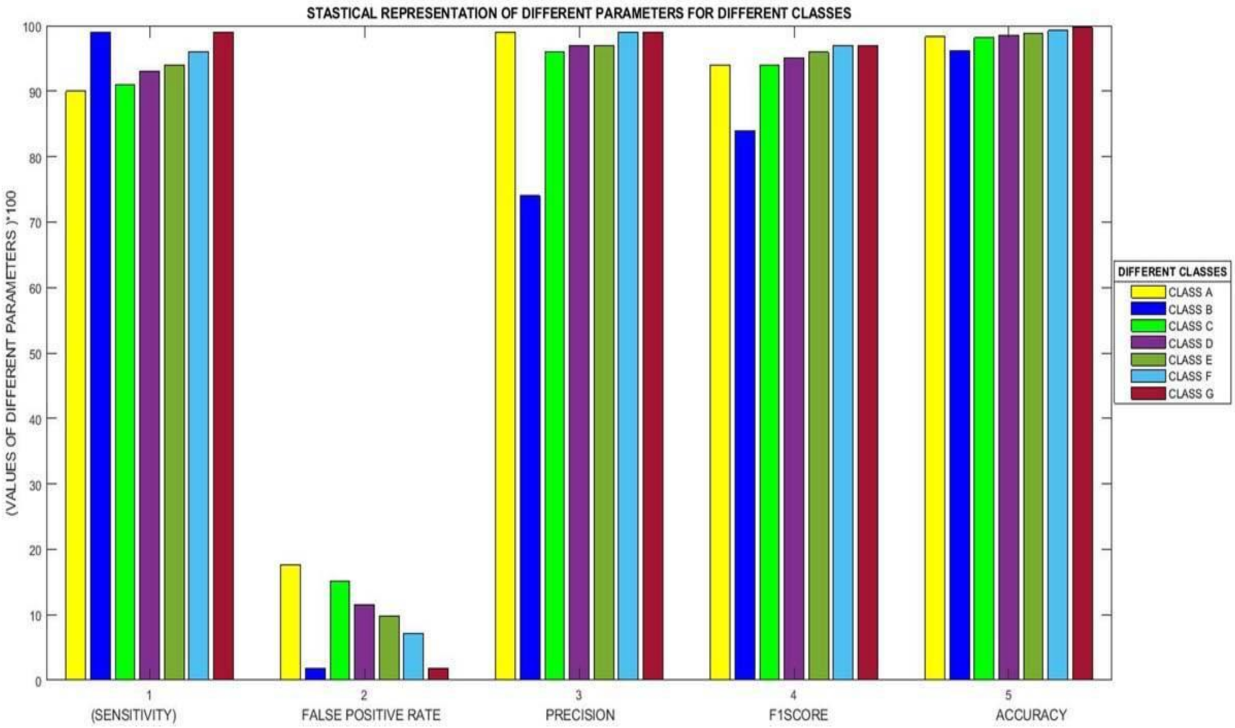

Fig 12: Statistic representation for the Table 5

We have drawn the ROC curve by using false positive rate and sensitivity, which is given below.ROC curve is a curve drawn between false positive rate (1-Specificity) and True positive rate (Sensitivity) as shown in figure 11 given below.

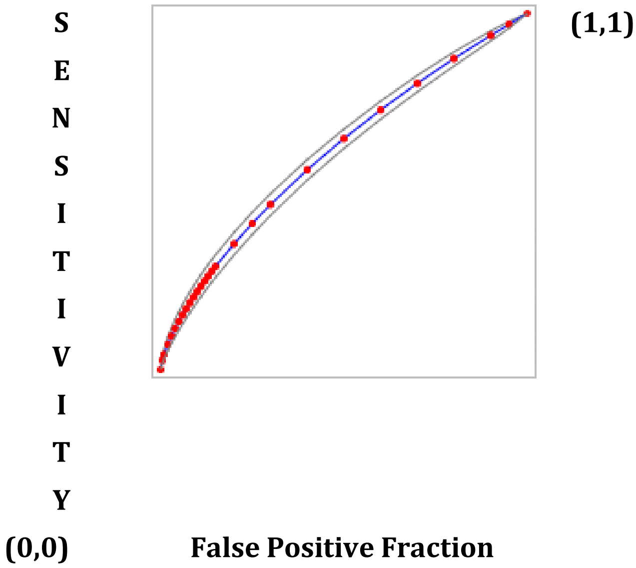

Fig 13: ROC curve for Table 7

With the help of the above ROC curve, we have calculated Fitted ROC Area (0.611) and Empiric ROC Area (0.599).Which shows a good result for calculation of overall accuracy. With the help of above table 7 and ROC Area, We have achieved 94.45% of overall accuracy in 26.75 minutes only, which is a good accuracy.

8. Conclusion & Future Work:

In this research paper, we have projected an algorithm for a CNN to construct a classifier for disease classification. Our outcome shows that CNN significantly overcame the state of the art of this field. In addition, our research shows the use of a pre-trained model. As we know, we cannot survive without food. So we need good production of healthy food to overcome this problem, we have to provide a better solution by using novel methods like disease prevention and disease detection. These days, many researchers attain a better precision by Convolutional Neural Network Model. They have faced a lot of difficulties to resolve the tomato disease problem. To solve such a type of problem, there are several classification techniques but the convolutional neural networks are functional in the application area as well as theoretical. A novel method is used for the classification and detection method to increase the capacity of deep convolutional neural networks. Our central objective is to identify the six most frequent diseases of tomato by its leaves. To work with pictures of tomato leaves of dissimilar pose, orientation, resolution, and size is difficult and challenging. We have developed a technique for better outcomes after completing the training and testing of tomato leaf images. We can state from every part of the above experiments and experiences that with the help of our technique anyone can able to distinguish exaggerated leaves of tomato correctly.

We have proposed a technique in which we have classified six types of tomato disease. For tomato disease classification; we have used an algorithm namely convolutional neural network (CNN) to construct a model for tomato leaves dataset. Our future goal will be to increase accuracy on a variety of tomato diseases. More than half of the population of several countries is dependent on agriculture. We would like to provide a good answer for tomato disease recognition by using an image dataset.

Declaration

- **Availability of Data** - Data available on request from the authors.
- **Competing interests** - No conflict of interest.
- **Research funding** - No funding was received to assist with the preparation of this manuscript.
- **Acknowledgments** - All authors are acknowledged for this research paper.
- **Author contributions** - All authors have accepted responsibility for the entire

content of this manuscript and approved its submission.

References:

1. Hanssen, I. M., & Lapidot, M. (2012). Major tomato viruses in the Mediterranean basin. In *Advances in virus research* (Vol. 84, pp. 31-66). Academic Press.
2. Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. *Applied Artificial Intelligence*, *31*(4), 299-315.
3. http://www.omafra.gov.on.ca/english/crops/facts/05-069.htm#3
4. Fry, W. E., & Goodwin, S. B. (1997). Re-emergence of potato and tomato late blight in the United States. Plant disease, 81(12), 1349-1357.
5. https://www.gardeningknowhow.com/edible/vegetables/tomato/septoria-leaf-spot.htm
6. https://www.gardeningknowhow.com/plant-problems/pests/insects/spider-mite-control.htm
7. https://www.gardeningknowhow.com/edible/vegetables/tomato/tomato-leaves-curling.htm
8. https://www.kaggle.com/kaustubhb999/tomatoleaf
9. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504-507.
10. Ranzato, M. A., Poultney, C., Chopra, S., & Cun, Y. L. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems* (pp. 1137-1144).
11. Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011, January). Contractive auto-encoders: Explicit invariance during feature extraction. In *Icml*.
12. Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011, June). Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks* (pp. 52-59). Springer, Berlin, Heidelberg.
13. Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.
14. Salakhutdinov, R., & Hinton, G. (2009, April). Deep boltzmann machines. In *Artificial intelligence and statistics* (pp. 448-455).
15. Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, *1*(2), 270-280.
16. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.
17. Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, *160*(1), 106.
18. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103).
19. Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011, January). Contractive auto-encoders: Explicit invariance during feature extraction. In *Icml*.
20. Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011, June). Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks* (pp. 52-59). Springer, Berlin, Heidelberg.
21. Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2015, December). Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In *International Symposium on Visual Computing* (pp. 638-645). Springer, Cham.
22. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*, *2016*. [CrossRef]
23. Prasanna Mohanty, S., Hughes, D., & Salathe, M. (2016). Using Deep Learning for Image-

Based Plant Disease Detection. arXiv, arXiv-1604.. [CrossRef]

24. Nachtigall, L. G., Araujo, R. M., & Nachtigall, G. R. (2016, November). Classification of apple tree disorders using convolutional neural networks. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 472-476). IEEE.
25. Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2016, December). Basic investigation on a robust and practical plant diagnostic system. In 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 989-992). IEEE.
26. Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: classification and symptoms visualization. Applied Artificial Intelligence, 31(4), 299-315.
27. DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E. L., Yosinski, J., Gore, M. A., ... & Lipson, H. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology, 107(11), 1426-1432.
28. Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). Identification of rice diseases using deep convolutional neural networks. Neurocomputing, 267, 378-384.
29. Wang, G., Sun, Y., & Wang, J. (2017). Automatic image-based plant disease severity estimation using deep learning. Computational intelligence and neuroscience, 2017.
30. Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., & Moussaoui, A. (2018). Deep learning for plant diseases: detection and saliency map visualisation. In Human and machine learning (pp. 93-117). Springer, Cham.
31. Wang, J., Chen, L., Zhang, J., Yuan, Y., Li, M., & Zeng, W. (2018, April). Cnn transfer learning for automatic image-based classification of crop disease. In Chinese Conference on Image and Graphics Technologies (pp. 319-329). Springer, Singapore.
32. Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science, 133, 1040-1047.
33. Khandelwal, I., & Raman, S. (2019). Analysis of transfer and residual learning for detecting plant diseases using images of leaves. In Computational Intelligence: Theories, Applications and Future Directions-Volume II (pp. 295-306). Springer, Singapore.
34. Foysal, M. F. A., Islam, M. S., Abujar, S., & Hossain, S. A. (2020). A novel approach for tomato diseases classification based on deep convolutional neural networks. In *Proceedings of International Joint Conference on Computational Intelligence* (pp. 583-591). Springer, Singapore.
35. Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science, 133, 1040-1047.
36. Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, *167*, 293-301.
37. Wang, Q., Qi, F., Sun, M., Qu, J., & Xue, J. (2019). Identification of Tomato Disease Types and Detection of Infected Areas Based on Deep Convolutional Neural Networks and Object Detection Techniques. *Computational Intelligence and Neuroscience*, *2019*.
38. Alajrami, M. A., & Abu-Naser, S. S. (2020). Type of Tomato Classification Using Deep Learning.