Plant Leaf Disease Detection Using Data Augmentation and CNN

Akhyar Ahmed Kathawala
Student, I^2IT , Pune
akhyarahmed369@gmail.com

Ankit Chauhan
Student, I^2IT, Pune
ankitchauhan2104@gmail.com

Ashutosh Chalmal
Student, I^2IT, Pune
ashutoshchalmal@gmail.com

Onkar Ransubhe
Student, I^2IT, Pune
onkarransubhe@gmail.com

Anand A. Bhosale
Assistant Professor, I^2IT, Pune
anandb@isquareit.edu.in

***Abstract*—Today agriculture is a key source of livelihood for around 50% of people in India. To bring positive growth in India's GDP modifications in the agriculture industry are required. Every other field possesses some new technologies as compared to the agricultural field. Around 42% of the farm produce goes to waste due to the inability of farmers to detect and take precautions against plant leaf diseases. In this paper we have devised a technique to overcome this issue, this plant leaf disease detection technique can be used to detect the disease from input images. This will be helpful to maintain the better quality of the farm produce and decrease the amount of produce that goes to waste due to undetected diseases.**

***Index Terms*—CNN, Plant leaf disease, ResNet, Data Augmentation.**

I. Introduction

The primary source of income for the majority of people in India is agriculture. Due to the diverse climate all over India, a wide variety of crops are produced in India. The production of crops is affected by various factors such as climatic conditions, soil conditions, various diseases, etc. Currently, farmers rely on their eyes for detecting any plant disease which results in improper judgment and hampers the quality of the farm produce and, increases the amount of produce thrown in waste. Inefficient disease detection leads to improper pesticide usage that can cause the development of long-term resistance of the pathogens and resultantly reduces the ability of the crop to fight back. Plant leaf disease detection can be achieved by identifying various spots on the leaves of the affected plant. We will use pretrained convolution neural network (CNN) to detect plant leaf diseases.

With the advancement of artificial intelligence technology, it is now possible to identify plant leaf disease by detecting spots on raw leaf images. Deep learning is an AI function that imitates the functioning of the human brain in processing data for use in identifying objects, detecting speech, translating languages, and making decisions and determining outcomes. In this paper, we will use convolution neural networks (CNN). CNNs are a class of neural networks that are proven very efficient in fields such as image detection and classification. A Convolution Neural Network (CNN) consists of either one or multiple convolution layers (often with a sub-sampling step) and then followed by one or more than one fully connected layer just like a standard multi layer neural network. The architecture of a CNN is created to exploit the 2D structure of an input image (or other 2D input). This is accomplished with local connections and tied weights and subsequently by some form of pooling which results in translation-invariant features. One advantage of CNN is that it is easier to train and has many fewer parameters than fully connected networks with an equal number of hidden units.

In recent years, CNN models are widely utilized in image classification problems. In this work [1] authors have introduced a hybrid model for extracting contextual information of leaves features by using CNN and Deconvolutional Networks (DN). In this work [2] authors have used Squeeze and AlexNet pre-trained CNN models on tomato leaves using an open data set to detect diseases. In this work [3] authors have performed multiple pre-trained CNN models on a large open data set of leaves. Their research shows that CNN is highly recommended for automatic plant leaf disease detection. In this work [4] authors suggested a three-channel CNN model depending on RGB colors to identify vegetable leaf diseases. In this work [5] authors, fine-tuned a pre-trained model and constructed a new CNN model to execute tomato leaf disease detection. Their study indicates that a custom CNN model gives better results than a pre-trained model. In this work [6] authors designed a system to detect and classify diseases in tomato leaf, by implementation of a slight difference of the CNN design namely ResNet. In the following paper we designed a CNN model based on RGB components of the tomato leaf imageson an open data set acquired from Kaggle website. We are classifying the leaves between 9 classes (diseases) namely late blight, early blight, bacterial spot, leaf mold, septoria leaf spot, target spot, mosaic virus, yellow leaf curl virus and healthy leaf.

The paper is organized as follows: Section II provides details of CNN. Section III describes ResNet algorithm. Section IV provides the proposed method for plant leaf disease detection and classification. Section V evaluates experimental results. Finally, section VI concludes the paper.

II. CNN

Deep learning is a class of machine learning algorithms that has sequential layers. Every layer utilizes the output of the previous layer as input. This learning process can be supervised, unsupervised or semi-supervised. Deep learning does not have to divide the feature extraction and the classification separately since the model automatically extracts the features while training the model. It is used in many research areas such as image restoration, image processing, natural language processing, speech recognition and, bioinformatics. In this study, we prefer CNN as a Deep Learning method. Convolutional Neural Network (CNN) is a Deep Learning algorithm that takes in an input image, assigns importance (learnable weights and biases) to various aspects/objects in the image and, can differentiate one aspect from another. The pre-processing required in a CNN is much lower in comparison to other classification algorithms. The architecture of a CNN is similar to the neuron connections in a human brain.

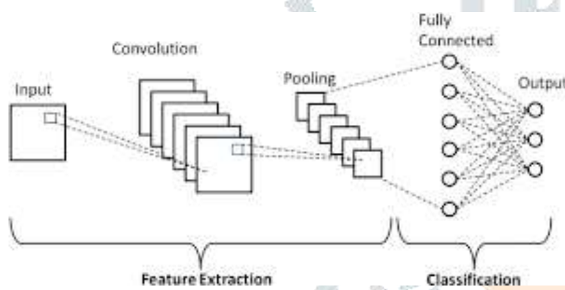

Fig. 1. CNN MODEL

It has 4 important operations involved: 1. Convolution, 2. Pooling, 3. Activation functions and 4. Fully connected layer. Fig. 1 shows a general CNN architecture.

- Convolution - The name CNN is derived from this layer. The aim of this operation is to extract features from the input image. CNNs are not limited to only one Convolutional Layer. Conventionally, the first Convolutional Layer is responsible for capturing the low-Level features such as edges, color, gradient orientation, etc. With the use of added layers, the architecture adapts to the high-level features as well, giving us a network that has a good understanding of images in the data set, similar to how a human brain would comprehend. The feature map of the input image is extracted by performing a series of mathematical operations on the input image. The input image is reduced to a smaller size by using a filter. The filter is shifted step by step commencing from the upper left corner of the image. At each step, the values in the image are multiplied by the values of the filter and the result is then summed. Resultantly, a new matrix with a smaller size is created from the input image [1]. Fig. 2 shows the convolution operation on a 5x5 input image using a 3x3 filter.

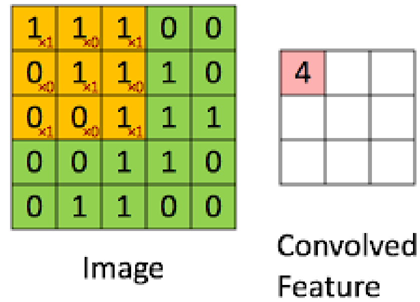

Fig. 2. CONVOLUTION OPERATION

- Pooling - Similar to the convolution operation, the operation of pooling is to reduce the spatial size of the convolved Feature. The size of the output matrix obtained from the convolution layer is reduced in this layer. Moreover, it is used for extracting the dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model. Although filters of different sizes can be used for pooling, generally 2x2 size filter is used [1]. Pooling operations can be of different types such as average pooling, max pooling and, minimum pooling can be used in this layer. In this research, max pooling and average pooling has been applied. Max pooling is achieved by selecting the maximum element from the region of the feature map covered by the filter. Hence, the output after max-pooling operation will be a feature map having the most prominent features of the previous feature map. Fig. 3 shows a pooling operation.

Fig. 3. AVERAGE POOLING OPERATION

- Activation Functions- The activation function is a node that is placed at the end or in the middle of a Neural Network. They play a role in determining whether or not a neuron can fire. The activation function in artificial neural networks creates a curvilinear relationship between the input and output layers. It has an effect on network efficiency. The activation function is used to achieve non-linear network learning. There are a variety of activation functions available, including linear, sigmoid, and hyperbolic tangent, but the nonlinear ReLU (Rectified Linear Unit) activation function is most commonly used in CNN. Values less than zero are set to zero in ReLU, while values greater than zero remain unchanged. Fig. 4 shows various activation functions.

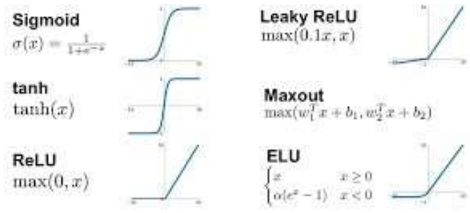

Fig. 4. ACTIVATION FUNCTIONS

- Fully Connected Layers-The final obtained matrix is fed into the fully connected layer (also known as the dense layer) as input after the convolution, pooling, and activation operations are completed. This layer is in charge of learning the image's parameters. We apply an output layer at the end of the Dense Layer that classifies the images into their respective classes. We're using a SoftMax function in the Output layer. Fig. 5 shows the formula of the SoftMax function.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Fig. 5. SoftMax FUNCTION

Z-Input Vector
K-Number of classes in the multi-class classifier
e^{z_i} -Standard exponential function for input vector
e^{z_j} -Standard exponential function for output vector

III. RESNET ALGORITHM

In this study, the ResNet algorithm also known as Residual Neural Network is used to train the data.
ResNet is a network structure proposed in 2015 by He Kaiming, Sun Jian, and others of Microsoft Research Asia in 2015, it won the first place in the ILSVRC-2015 classification task. It also took first place in the ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation tasks at the same time.
ResNet, also known as Residual Neural Network, is a concept that combines residual learning with the traditional convolutional neural network to solve the problem of vanishing gradient and accuracy degradation (training set) in deep networks, allowing both accuracy and speed to be controlled as the network gets deeper and deeper.

The idea of residual learning is the Fig.6, which can be understood as a block, defined as follows:

$$\mathbf{y} = x_i + \mathbf{F}(x_i, (W_i))$$

The residual learning block contains two branches or two mappings:

1. Identity mapping refers to the curve on the right side of Figure 6. As its name implies, identity mapping refers to its mapping, which is x itself;
2. F(x) Residual mapping refers to another branch, that is, part. This part is called residual mapping (y-x).

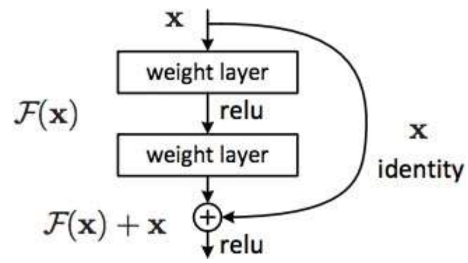

Fig. 6. WORKING OF ResNet

The idea behind Residual Networks is illustrated in Fig 7.

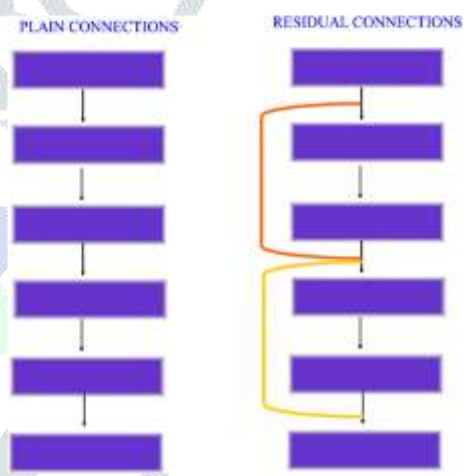

Fig. 7. IDEA BEHIND RESIDUAL NETWORKS

In Figure 7, The plain network only sends information from one layer to the next; information about the image's previous state is highly limited, and all activations must be based on new features; on the other hand, the residual connections take the future map from layer T and adds it to the output of layer T+2. This is equal to learning the residual function y = f(x)+x. A layer T in a direct feed-forward network without residual connections relies solely on data from layer T-1, with layer T-1 encoding the consequences of all previous layers. Residual connections, on the contrary, look further back in time, taking into account information from layer T-2.
This very simple but powerful idea enables to train over a 100 layers network with increasing accuracy. Different variants of ResNet are shown in Figure 8.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2.x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3.x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4.x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5.x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 8. VARIANTS OF ResNet

IV. Proposed Methodology

In this study, 15745 training and 4000 test tomato leaf images have been used from the Tomato Leaf dataset(Kaggle). The images in the selected data set have been cropped to the size of 224x224. The intended leaf diseases to classify in this study are late blight, early blight, bacterial spot, leaf mold, septoria leaf spot, target spot, mosaic virus, yellow leaf curl. Nine different classes have been used, eight of them are for leaf diseases and one of them is for healthy leaves.

- Early Blight - Initially, leaves have small dark spots form on older foliage near the ground. Leaf spots are round, brown, and can grow up to half an Early Blight - Initially, leaves have small dark spots form on older foliage near the ground. Leaf spots are round, brown and can grow up to half-inch in diameter. Larger spots have target-like concentric rings. The tissue around spots often turns yellow
- Late Blight - Leaves have large, dark brown blotches with a green-gray edge; not confined by major leaf veins. Stem infections are firm and dark brown with a rounded edge.
- Septoria Leaf Spot - Septoria leaf spots start somewhat circular and first appear on the undersides of older leaves, at the bottom of the plant. They are small, 1/16 to 1/8 inches (1.6 to 3.2 millimeters) in diameter, with a dark brown margin and lighter gray or tan centers. A yellow halo may surround the spot. As the disease develops, the spots will get larger and may merge.
- Bacterial Spot - Leaf lesions are initially circular and water-soaked and may be surrounded by a faint yellow halo. In general, spots are dark brown to black and circular on leaves and stems. Spots rarely develop to more than 3 mm in diameter. Lesions can coalesce causing a blighted appearance of leaves and a general yellowing may occur on leaves with multiple lesions. Tomato Yellow Leaf Curl - Leaves of infected plants are small and curl upward, and show strong crumpling and interveinal and marginal yellowing. The internodes of infected plants become shortened and, together with the stunted growth, plants often take on a bushy appearance, which is sometimes referred to as 'bonsai' or broccoli'-like growth.
- Leaf Mold - Symptoms of the disease include yellow spots on the upper leaf surface. Discrete masses of olive-green spores can be seen on the underside of the affected leaves. The older leaves become infected first and die prematurely. The pathogen may spread rapidly during periods of prolonged relative humidity.
- Target Spot - The disease starts on the older leaves and spreads upwards. The first signs are irregular-shaped spots (less than 1 mm) with a yellow margin. Some of the spots enlarge up to 10 mm and show characteristics rings, hence the name of "target spot". Spread to all leaflets and other leaves is rapid, causing the leaves to turn yellow, collapse, and die.
- Tomato Mosaic Virus - Mottled light and dark green on leaves. If plants are infected early, they may appear yellow and stunted overall. Leaves may be curled, malformed, or reduced in size. Spots of dead leaf tissue may become apparent with certain cultivars at warm temperatures. Fruits may ripen unevenly.

Fig.9 shows sample images of a. Healthy Leaf, b. Early Blight, c. Late Blight, d. Septoria Leaf Spot, e. Bacterial Spot, f. Yellow Leaf Curls, g. Leaf Mold, h. Target Spot, i. Mosaic virus.

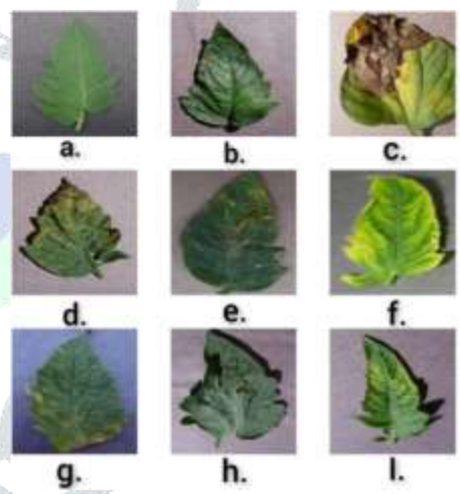

Fig. 9. SAMPLE PHOTOS OF AFFECTED LEAVES

In this paper, we use transfer learning to implement a novel deep neural network architecture for tomato plant leaf disease detection. In the feature extraction layers of the proposed system, state-of-the-art CNN is used and its features are extracted. The extracted features are then given to fully connected layers to generate classification outputs.

A. Pretrained CNN for Feature Extraction

In this section, we adopt one deep CNN architecture, namely ResNet152 as the feature extractor of the proposed method for tomato plant leaf disease detection tasks. The CNN is pre-trained on a nature image data set (ImageNet) for distinct generic image descriptors and it can be applied to extract discriminative features from biomedical images based on transfer learning theory. We select ResNet152 as it achieves the best accuracy among ResNet family members. Fig. 10 illustrates the basic architecture of ResNet152.

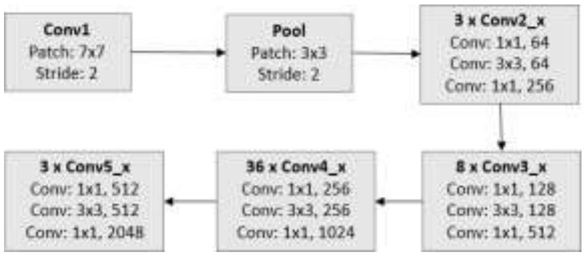

Fig. 10. ResNet152 ARCHITECTURE

B. The Proposed Network Structure

To begin with, the CNN model is trained over more than a million natural images from 1000 categories in ImageNet [9]. As discussed in [8], after being trained on a very large labelled dataset (e.g., ImageNet), transfer learning technique can be used, i.e., Deep CNNs can learn generic image features that can be applied to other image datasets without needing to be trained from scratch. Fig. 11 illustrates the transfer learning structure for a single CNN. In this figure, The pretrained network serves as a feature extractor for generic image features, while the two final layers are fully-connected layers for classification. This structure is known as a single transfer learning network. It is shown in Fig. 11.

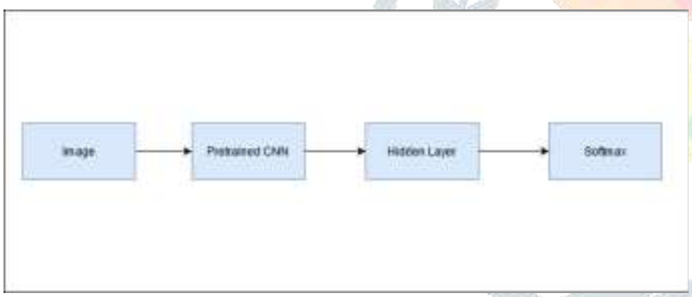

Fig. 11. SINGLE TRANSFER LEARNING NETWORK

Following pretraining, we create a 2048-dimensional feature vector.

Lastly, the feature vector is feeded into two fully-connected layers for classification. We adjust the classification architecture by adding a output layer. This modification improves our network's learning capabilities and helps us to adopt the generic features extracted by the pertained CNN to the image data.

The output layer has 9 neurons to represent one neuron for each class. To summarize, we propose a transfer learning network structure based on feature extraction and design a two layer fully-connected structure for generic feature adoption to tomato plant leaf image data. The maximum number of epochs has been selected as 50 in all experiments. The learning rate has been selected as 0.01. Fig. 12 shows the entire architecture of the proposed network.

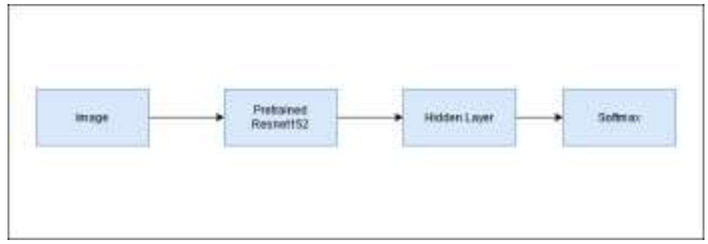

Fig. 12. ARCHITECTURE OF PROPOSED NETWORK

V. Experiment and Results

In this research, to get the most accurate result, we trained and tested our data set with 4 different CNN architectures namely LeNet, AlexNet, VGG16 and ResNet152.

- LeNet- LeCun et al. first proposed the LeNet architecture in their paper Gradient-Based Learning Applied to Document Recognition, published in 1998. LeNet CNN architecture consists of 7 layers. The layer composition is made up of 3 convolutional layers, 2 subsampling layers and 2 fully connected layers.
- AlexNet- AlexNet architecture is made up of of 5 convolutional layers, 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer.
- VGG16- VGG16 architecture was used to win ILSVR(Imagenet) competition in 2014. it has 16 layers that have weights. VGG16 focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 fully connected layers followed by a softmax function for output.
- ResNet152- ResNet, also known as Residual Neural Network, is a concept that combines residual learning with the traditional convolutional neural network to solve the problem of vanishing gradient and accuracy degradation (training set) in deep networks, allowing both accuracy and speed to be controlled as the network gets deeper and deeper.

Table 1 shows the results for different CNN architecture used.

TABLE I
EXPERIMENT AND RESULTS

Architecture	Training Accuracy	Testing Accuracy
LeNet	95.87%	77.24%
AlexNet	97.59%	89.24%
VGG16	97.59%	93.49%
ResNet	98.55%	94.01%

VI. CONCLUSION

In this paper, we have proposed a plant leaf disease detection technique using ResNet152 CNN architecture. We also compared our proposed system with several traditional classification methods.The proposed method achieves better accuracy of classification (94.01%) as compared to the best competitive method. Since transfer learning is used, the proposed approach produces good classification results without the need to train deep neural networks from the ground up.

REFERENCES

[1] Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. Pattern Recognition, 71, 1-13.

[2] Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In 2017 6th International Conference on Agro-Geoinformatics (pp. 1-5). IEEE.

[3] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311-318.

[4] Zhang, S., Huang, W., & Zhang, C. (2019). Three-channel convolutional neural networks for vegetable leaf disease recognition. Cognitive Systems Research, 53, 31-41.

[5] Atabay, H. A. (2017). DEEP RESIDUAL LEARNING FOR TOMATO PLANT LEAF DISEASE IDENTIFICATION. Journal of Theoretical & Applied Information Technology, 95(24).

[6] Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In 2018 Eleventh International Conference on Contemporary Computing (IC3) (pp. 1-5). IEEE.

[7] Sardogan, M., Tuncer, A., & Ozen, Y. (2018, September). Plant leaf disease detection and classification based on CNN with LVQ algorithm. In 2018 3rd International Conference on Computer Science and Engineering (UBMK) (pp. 382-385). IEEE.

[8] Kumar, A., Kim, J., Lyndon, D., Fulham, M., & Feng, D. (2016). An ensemble of fine-tuned convolutional neural networks for medical image classification. IEEE journal of biomedical and health informatics, 21(1), 31-40.

[9] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[10] Narvekar, P., & Patil, S. N. (2015). Novel algorithm for grape leaf diseases detection. International Journal of Engineering Research and General Science, 3(1), 1240-1244.

[11] Nguyen, L. D., Lin, D., Lin, Z., & Cao, J. (2018, May). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In 2018 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IEEE.

[12] Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., & Menaka, R. (2020). Attention embedded residual CNN for disease detection in tomato leaves. Applied Soft Computing, 86, 105933.

[13] Wu, Z., Shen, C., & Van Den Hengel, A. (2019). Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition, 90, 119-133.