Grenze International Journal of Engineering and Technology, Jan Issue

Plant Disease Detection and Classification using Convolutional Neural Networks (CNN)

Mahalakshmi Sreenivasan[1] and P.Selvaraj[2]
[1-2]Department of Computing Technologies, Faculty of Engineering and Technology, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai, 603203, Tamil Nadu, India
Email: ms4934@srmist.edu.in, selvarap@srmist.edu.in

***Abstract*— Farmers and producers may suffer significant financial losses due to plant diseases. Effective control of plant diseases requires early identification and categorization of the illnesses. However, traditional plant disease detection techniques are frequently labor- and time-intensive. Plant disease detection has found a possible collaborator in deep learning and machine learning. Consequently, a novel deep-learning technique is proposed for classifying and diagnosing plant illnesses. It uses a transfer learning methodology built on a convolutional neural network architecture consisting of 5 convolutional, 2 fully- connected, and 3 max pooling layers. The algorithm has a 98% accuracy rate when tested on a publicly accessible dataset of images of plant diseases. The CNN model was trained with the PlantVillage dataset, which is available on Kaggle. From the results and a comparison with previous works, it is evident that the chosen approach performed well in classifying plant diseases with expected accuracy.**

***Index Terms*— convolutional neural network, detection system, plant disease**

I. INTRODUCTION

Plant diseases represent an enduring and profound challenge to global agriculture, with ramifications that extend beyond the boundaries of fields and orchards. The demand for sustainable food production has never been more pressing in a world marked by population growth and a rapidly changing climate. However, the menace of plant diseases casts a shadow over this imperative, jeopardizing crop yields, economic stability, and food security on a global scale. Though valuable, traditional approaches to plant disease diagnosis, rooted in visual inspection by experts, are fraught with limitations [1]. Subjectivity, labor intensity, and a lack of scalability are intrinsic to these methods. The human element introduces variability, rendering diagnosis susceptible to individual expertise. Moreover, the labor-intensive nature of these approaches hinders their application across vast expanses of agricultural landscapes. Amidst these obstacles, a technological revolution is taking place that offers a more efficient, precise, and scalable method of classifying and detecting plant diseases. Recently, three revolutionary technologies have come together: computer vision, machine learning, and digital imagery. Affordable, high-quality digital imaging equipment, from smartphones to drones, has empowered stakeholders to capture detailed images of plants and their associated diseases. These images are at the heart of advanced disease detection systems. Machine learning and Deep learning are emerging as potent tools for automated image analysis. Through computer vision, machines can decipher and interpret visual data, effectively identifying even subtle disease symptoms that elude human perception. Remarkable accuracy has been demonstrated in identifying healthy from diseased plants

Grenze ID: 01.GIJET.10.1.95

using machine learning and deep learning models, especially convolutional neural networks.

II. State-Of-The-Art

A. Plant Disease Detection: Datasets

Expert Systems are widely used for identifying and managing agricultural plants due to advancements in agricultural technology. This has increased considerably increased plant production. Different disease detection models are developed because of multiple datasets about plant diseases. The most well-known ones are Plant Doc and Plant Village. The Plant Village is the most extensive collection presently available, with over 163,000 pictures representing 14 crop species: Raspberry, Apple, Cherry, Orange, Blueberry, Pepper Bell, Squash, Tomato, Grape, Peach, Soybean, Strawberry, Corn, and Potato. These images of crop leaves—both healthy and diseased—can be accessed on the Kaggle platform [2]. The Plant Doc dataset consists of 2598 images gathered from 13 distinct crop species: Potato, Apple, Strawberry, Cherry, Squash, Peach, Pepper Bell, Raspberry, Corn, Blueberry, Tomato, Soybean, and Grape [3]. This model has been developed using approximately 20,600 images from the Plant Village Dataset, with plants ranging from pepper bells to potatoes and tomatoes. The leaf images consist of healthy and diseased ones, including Bacterial Spots, Blights, Viruses, Mold, and Spots.

B. Models for Plant Disease Detection

Plant diseases have been identified and diagnosed using a variety of artificial intelligence (AI) approaches, the most often used being Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and K-Nearest Neighbours (KNN). Using the GoogleNet and AlexNet models that had already been trained (pre-trained models), Mohanti et al. [4] categorized plant diseases. They used images of 54,306 healthy and diseased plant leaves from a public dataset to get a 99.35 percent accuracy rate. Ahmad et al. [5] used existing, pre-trained neural networks to build their disease detection model. Using the pre-trained weights from MobileNetV2 on the PlantVillage and Pepper datasets, they obtained 99% and 99.69% accuracy, respectively.

Amin et al. [8] utilized the pre-trained CNNs, EfficientNetB0 and DenseNet121, to obtain detailed information from images of maize plants. Their proposed model surpassed the performance of current pre-trained CNN models, such as ResNet152 and InceptionV3, achieving an impressive classification accuracy of 98.56%. Zhou et al. [9] introduced a deep learning model with a hybrid approach that conjugates the strengths of deep residual and dense networks for the detection of tomato leaf diseases. This innovative model effectively reduced the number of parameters for training processes, enhancing computational efficiency, gradient optimization, and information flow. As a result, their proposed model obtained an accuracy of 95% on the Tomato dataset. Wang et al. [10] introduced a model involving dual-stream hierarchical bilinear pooling. It was designed to perform multi-task classification of plants and diseases in field environments. It improved the model's representational capabilities by extracting discriminative fine-grained features through the use of a fine-grained image recognition technique. The results showed that the plant disease accuracy was 84.71% and 75.06%, respectively. Although basic CNN architectures such as VGGNet, ResNet, and AlexNet have been widely used, they have disadvantages, including slow processing and calculation and multiple parameters. Consequently, a deep learning approach that focuses on deep feature extraction, a large dataset, and no pre-trained model has been proposed.

Table II(1) Comparison Of Proposed And Pre-Trained Model Accuracies

Method	**Plant**	**No. of Parameters**	**Accuracy**
ResNet-50 [11]	Apple	25,636,712	78%
AlexNet [12]	Apple	62,378,344	87%
ResNet-50 [13]	Tomato	25,636,712	97%
InceptionV3 [14]	Apple	23,851,784	97%
CNN [15]	Tomato	208,802	91%
Proposed CNN	Assorted cultures	58,102,671	98%

The above table (Table 2.1) compares the accuracies of the proposed CNN model and the works of various authors, as discussed above. The suggested model consists of 5 convolutional, 2 fully- connected, and 3 max pooling layers. To introduce nonlinearity and resolve the vanishing gradients issue, an activation function called ReLU is applied at each activation layer. At the output layer, the softmax function is used. This classifies diseases by normalizing the network's output into a distribution of probabilities across the expected output classes. Four dropout layers are used to cut off a few randomly selected neurons to prevent overfitting. The proposed model has been compared to various authors' works and achieved a high accuracy of 98%. Also, it does not involve the use of pre-trained CNN models. Creating a CNN model with multiple layers, training and testing it against a large dataset of around 20,600 images and achieving a high accuracy makes the model unique and novel.

III. Implementation Of The Plant Disease Detection System Using Cnn

A. Dataset and Preprocessing

The PlantVillage dataset is chosen for the project. It contains 20,639 RGB images of leaves belonging to 3 different plants: Bell Pepper, Potato, and Tomato. They are divided into 15 classes, each depicting a disease or healthy plant. A few of the images are displayed in Fig 3.1. The PlantVillage dataset has served as a foundational resource in numerous research endeavors. The inclusivity of a diverse range of leaf images in the dataset is pivotal, as it enables the model to assimilate crucial variations throughout the training process. This inclusivity significantly enhances the capacity of the deep CNN model to generalize effectively.

Fig 3.1 Random leaf images from PlantVillage dataset

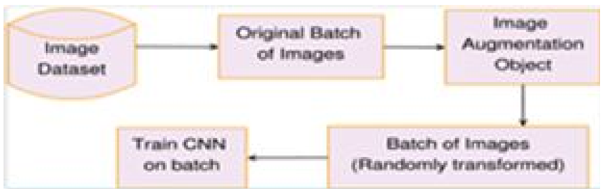

Fig 3.2 Data Augmentation Process

The process in which image variations are created artificially is called image augmentation. Transformations like shifting, shearing, scaling, zooming, and flipping are applied to alter the images. Fig 3.2 represents an overall Augmentation process as used in the proposed model. By introducing small variations in the images, these transformations encourage diversity in the training set.

An ANN with deep learning and feed-forward at its core is known as a Deep Convolutional Neural Network. It includes components such as the fully connected, pooling, and convolutional layers with non-linear activation units. Deep CNN does not require further feature engineering, giving it various advantages over typical machine learning-based methods. It has proven effective in several applications, including precision agriculture, natural language processing (NLP), and text and picture categorization. Extrapolating several lower-level characteristics is the task of a deep CNN's first layer. Consequently, this lessens overfitting and enhances the model's ability to generalize. A few examples of the augmented images from the PlantVillage dataset are present in Fig 3.3.

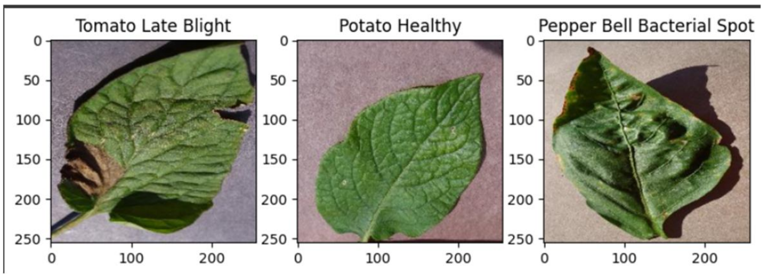

Fig 3.3 Examples of enhanced PlantVillage dataset leaves

B. CNN Model Architecture

1) Convolutional Layer

The convolutional layer is required when incorporating a filter (kernel) into a source image for processing. Locating local patterns in the layers that originated before generates feature maps. The two primary components of the convolutional layer are a linear convolutional operation and a non-linear activation unit. Convolution is used across multiple images, such as RGB images, when working with multi-channel ones. In mathematical terms, this is stated as:

$$Conv(I, K)_{x,y} = \sum_{i=1}^{nH}\sum_{j=1}^{nW}\sum_{k=1}^{nC} K_{i,j,k}\, I_{x+i-1,\, y+j-1,\, k} \quad (3.1)$$

Here, K(fh, fw, nC) represents the kernel, which undergoes convolution with an image represented as I(nH, nW, nC). The image and kernel can have different dimensions but share the same number of channels, denoted as nC. The width and height of the input image are nW and nH, respectively, while the kernel's dimensions are fh and fw. As a standard practice, the kernel is treated as an odd-dimensional window with fh and fw both equal to a standard value of f. This typical value determines the size of the resulting feature map:

$$Feature\_map(oH, oW, z) = (\lfloor nH + 2P - f/s + 1 \rfloor, \lfloor nW + 2P - f/s + 1 \rfloor, z) \quad (3.2)$$

In the equation above, "z" refers to the number of kernels convolved with the input image, "s" is the stride, and "P" represents the padding value. Rectified Linear Units (ReLU) is a frequently used activation function that activates neurons when the output of a convolution unit or other linear transformation is greater than or equal to zero. This can be expressed as follows:

$$f(z) = MAX(0, z) \quad (3.3)$$

2) Pooling Layer

To minimize the large number of parameters in activation maps, the pooling layer is utilized to down-sample the convolutional layers generated feature maps. This technique serves to decrease computational workload, accelerate training, and assist in controlling overfitting. Among various pooling methods, max pooling is the most popular, and the following mathematical expression defines it:

$$Max\_Pooling: y_j = \max_{i \in R_j}(P_i) \quad (3.4)$$

In max pooling, the highest value from each input patch is chosen. The following equation can be used to determine the dimensions of the resulting feature map, where "R" represents a receptive field consisting of "P" pixels:

$$Feature\_map(oH, oW, z) = (\lfloor nH + 2P - f/s + 1 \rfloor, \lfloor nW + 2P - f/s + 1 \rfloor, nC) \quad (3.5)$$

It's important to note that the pooling process only alters the dimensions nH and nW, while nC remains unchanged.

3) Fully Connected Layer (FC)

Like traditional neural networks, convolutional neural networks also incorporate FC or dense layers, typically positioned in the later process of the CNN. Output layers with a predefined number of outputs are constructed. A flattened layer is employed to adapt these layers towards 1-dimensional data, which converts the 2-dimensional output from the preceding layers into a 1-dimensional representation. The fully connected layers use both linear and non-linear modifications. The definition of these transformations is as follows:

$$z = W^T \cdot X + b \quad (3.6)$$

$$O = f(z) \quad (3.7)$$

The input feature map is represented by "X," the weight by "W," the bias terms by "b," and the fully connected layer's output by "O."

C. Model Training

During the model training phase, the CNN model's parameters must be optimized to minimize the error between the predicted and actual disease labels. Crucial components of this phase include:

Loss Function: The model's training procedure must be guided by the choice of loss function. Especially for problems requiring numerous classes, categorical cross-entropy is a widespread loss function in plant disease diagnosis.

Optimizers: Stochastic gradient descent (SGD), Adam, and RMSprop are optimizers that control how the model's parameters change during training. The optimizer selection can significantly impact model convergence and training speed.

Hyperparameter Tuning: A few examples of hyperparameters that significantly affect a training process' efficacy include batch size, learning rate, and number of epochs to be trained. Finding the optimal collection of hyperparameters to provide the highest possible model performance is known as hyperparameter tuning.

Regularization: Regularization techniques, such as dropout and L2 regularization, are employed in deep neural network training to mitigate overfitting.
Model Checkpoints: If training is interrupted, the optimal model parameters are preserved, enabling the recovery of the top-performing model.

D. Model Evaluation

The model needs to be thoroughly evaluated to gauge its performance. Evaluation involves several components:

- *Validation Set Evaluation*: We assess the model's performance using a validation set. These results are crucial for calculating the model's predictive accuracy for novel and unknown data as well as for modifying hyperparameters.
- *Testing Set Evaluation*: It is ultimately used to evaluate the performance and efficiency of the model. This set objectively assesses the model's accuracy in identifying crop diseases.

E. Real-World Deployment

To make plant disease detection accessible and practical for farmers and agricultural stakeholders, the model needs to be deployed in the real world. It includes:

- *Real-time processing*: In scenarios where real-time disease detection is required, the system can be integrated into IoT devices or edge computing solutions for on-site analysis. This is particularly beneficial for early disease detection in the field.
- *Batch Processing*: In cases where real-time processing is not essential, batch processing can be employed. Collected plant images are periodically analyzed in bulk, and disease classification results are reported.

IV. PROPOSED CNN MODEL

The incorporation of deep learning techniques has recently revolutionized plant disease detection, specifically Convolutional Neural Networks. However, it remains challenging to close the gap between advanced machine learning methods and farmers' ability to use them.
To bridge this gap, we propose an integrated system where farmers can directly input leaf images for disease diagnosis. The system subsequently extracts features, and classifies the images, enabling timely intervention and reduced crop losses. The disease detection process begins with the user uploading a leaf image into the model. It then undergoes numerous Image Processing techniques and CNN layers to generate the result in a short period. The output helps the user decide whether the plant is healthy or not. The system architecture of the proposed CNN model is shown in Fig 4.1.
The integrated system for disease detection and classification in crops begins with the farmer's input of leaf images. The system is divided into three primary phases:

- Farmer's Input
- Image Processing
- Training

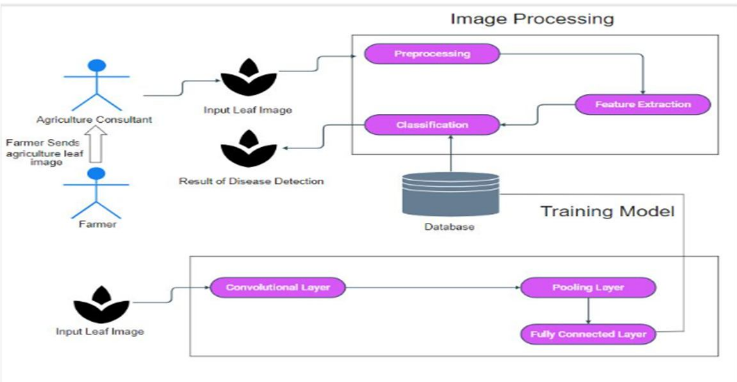

Fig 4.1 System Architecture

A. Farmer Input: Capturing Leaf Images

The system begins with a farmer or user encountering a plant with suspected disease symptoms. To begin the disease detection process, the farmer captures a digital image of the affected plant's leaf using a smartphone or another digital device equipped with a camera. Acquired images are submitted to the plant disease detection system via an application or platform. This user-friendly step makes it accessible to farmers without specialized technical knowledge.

B. Preprocessing: Enhancing Image Quality

Upon receiving the farmer's input, the image processing phase commences. This phase involves three critical steps: Preprocessing, Feature Extraction, and Classification, which are essential for accurate disease detection.

- *Image Denoising*: Any noise or artifacts present in the image, such as sensor noise or lighting variations, are removed using denoising filters or techniques. This step ensures a cleaner input image.
- *Image Enhancement*: Image resizing, contrast, brightness, and color balance are adjusted to enhance the visibility of disease symptoms. Histogram equalization and contrast stretching are commonly used techniques.

C. Feature Extraction: Identifying Disease-Related Patterns

After preprocessing, the system moves to feature extraction, a critical phase for understanding the image's content. Features are distinctive patterns or characteristics within the image that are relevant to identifying plant diseases. Feature extraction can encompass various techniques:

- *Color-Based Features*: Color features are extracted to determine plant-leaf color variations. It includes methods such as color moments and color histograms.
- *Texture Analysis*: Texture features describe an image's spatial arrangement of color or intensity. Standard texture analysis techniques include Gabor filters and Haralick features.
- *Shape Descriptors*: Shape-based features capture the geometry of disease symptoms, such as lesions, spots, or deformities on the leaves. Methods like contour analysis and Hu moments may be employed.

This developed model works against 58,102,671 parameters and is considered better than the pre-trained models (e.g., VGG16, AlexNet,, GoogleNet). This is because they generate high accuracy at the cost of processing speed and CPU resources. Significant works have been performed on one crop with fewer parameters (such as Apple, Tomato, and so on). However, the proposed model lies in the novelty of utilizing multiple crops, that is, Assorted Cultures (Potato, Tomato, Pepper Bell) against multiple layers and a more significant number of parameters. A satisfactory 98% accuracy is achieved, which is better than most of the other works.

D. Classification: Disease Identification

The extracted features are then sourced as the input images for a classification model. This model is responsible for recognizing the type of disease based on the identified features. CNNs play a pivotal role in this stage:

- *CNN Architecture*: The CNN architecture comprises numerous convolutional and pooling layers, allowing it to acquire hierarchical features. The CNN architecture can be a custom design or a pre-trained model like VGG, Inception, or ResNet, fine-tuned for the specific plant disease classification task.
- *Training*: The CNN model undergoes training on a dataset consisting of labeled images containing healthy and ill plants. During the training process, model parameters are fine-tuned utilizing techniques such as backpropagation and gradient descent to minimize classification errors.
- *Classification Output*: The CNN model's last layer produces a probability distribution across potential disease classes. The predicted disease type belongs to the highest probability class of disease. The program may display the results as "healthy" or identify the exact type of disease, such as "virus," "blight," or "spot”.

V. Implementation

The following steps were followed during the implementation process.

A. Inclusion of Required Library Modules

The importation of the required modules and packages has been done during the implementation. The imported packages are Numpy, Pickle, Sklearn, Keras, Tensorflow, Matplotlib.

B. Data Loading and Processing

a) CNN Parameters are identified

b) Images are converted into Array
c) Image pixels are loaded into separate lists
d) Loading of Images into the program
e) Images are converted into Binary form

C. CNN Building

The following steps were followed during the CNN building stage.

a) Splitting the dataset into Training and Testing (70, 30 ratio)
b) CNN Layers are identified
c) Model parameters are fixed
d) Layer filters are tuned to maximize the required level of accuracy without any biasing or over fitting problem
e) Adam Optimizer was used for the overall optimization.

D. Running the CNN Model

The suggested model consists of 5 convolutional, 2 fully- connected and 3 max- pooling layers and it runs for 25 epochs in approximately 6 hours.

E. Outcome

Training and Validation Accuracy helps assess how well a model has learned from the training data and how well it generalizes to new, unseen data, respectively. Fig 5.2 shows that the Training Accuracy increases and iterations of epochs are implemented; the Validation Accuracy rises and falls due to the data being unseen. Training and Validation Loss calculates the difference in error between the model's predictions and actual target values in the dataset. From Fig 5.1, it can be observed that the Validation Loss gradually decreases as the newer data keeps being seen, and the Training Loss is consistently low since data is being read from the training dataset (already in process). The trained model was evaluated for accuracy and generated the best test results. The accuracy was achieved as 98 %.

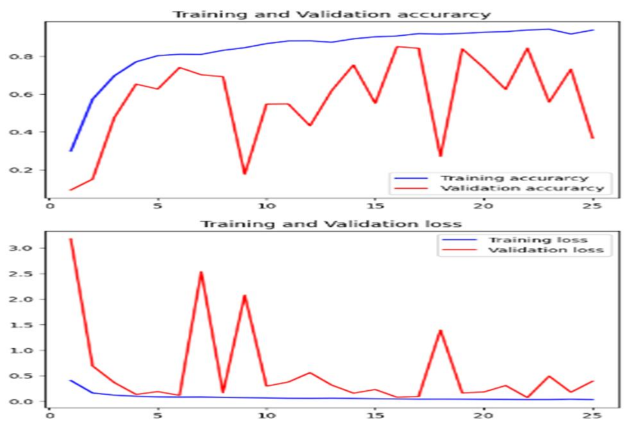

Fig 5.1 Accuracy and Loss Graphs for the CNN model

VI. Results And Discussion

The model has been successfully trained and executed. Using the PlantVillage dataset consisting of around 20600 images, a CNN model with 5 convolutional, 2 fully- connected, and 3 max-pooling layers has been built. It has achieved a high accuracy of 98%. In comparison to the works of several authors, the proposed model is made up of multiple layers and does not involve the use of pre-trained models. Pre-trained models such as ResNet-50 and AlexNet would make accurate predictions but at the cost of performance. The processing speed would be slow with multiple calculations involved. The proposed model works against 58,102,671 parameters and is considered better than pre-trained models, making it effective in detecting and classifying plant diseases.

VII. Conclusion And Future Enhancement

In summary, using Convolutional Neural Networks (CNN) for disease detection in plants is a novel strategy for tackling the agriculture sector's significant challenges. The inadequacy and labor-intensiveness of traditional manual inspection methods impact the economy's sustainability and global food security. Using CNN technology presents a viable way to deal with this problem. These models have demonstrated remarkable performance in image recognition, making them ideal for identifying and classifying plant diseases. By training CNNs on extensive plant image datasets, the system can accurately distinguish between healthy and ill plants and generate results with high accuracy and efficiency (For example, the proposed model generates an accuracy of 98%). The future of disease detection in plants using CNN technology is full of exciting possibilities. By advancing the accuracy of models, expanding crop coverage, incorporating real-time monitoring, forecasting, and risk assessment, ensuring accessibility, addressing ethical concerns, and promoting collaboration, we can create a sustainable, efficient, and equitable approach to managing plant diseases. This technology is about improving crop yields and transforming agriculture into a more sustainable and resilient sector that can feed a growing global population while minimizing its environmental footprint. The journey ahead is challenging, but the potential rewards are immense.

References

[1] B. Aglave, Handbook of Plant Disease Identification and Management, Boca Raton, FL, USA:CRC Press, Apr. 2018

[2] PlantVillage Dataset: www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset/data

[3] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat and N. Batra, "PlantDoc: A dataset for visual plant disease detection", Proc. 7th ACM IKDD CoDS 25th COMAD, pp. 249-253, Jan. 2020.

[4] S. P. Mohanty, D. P. Hughes, M. Salath'e, Using deep learning for image-based plant disease detection, Frontiers in plant science 7 (2016) 1419.

[5] M. Ahmad, M. Abdullah, H. Moon and D. Han, "Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning", IEEE Access, vol. 9, pp. 140565-140580, 2021.

[6] M. A. Khan, M. I. U. Lali, M. Sharif, K. Javed, K. Aurangzeb, S. I. Haider, et al., "An optimized method for segmentation and classification of apple diseases based on strong correlation and genetic algorithm-based feature selection", IEEE Access, vol. 7, pp. 46261-46277, 2019.

[7] S. Chakraborty, S. Paul and M. Rahat-uz-Zaman, "Prediction of apple leaf diseases using multiclass support vector machine", Proc. 2nd Int. Conf. Robot. Electr. Signal Process. Techn. (ICREST), pp. 147-151, Jan. 2021.

[8] H. Amin, A. Darwish, A. E. Hassanien and M. Soliman, "End-to-end deep learning model for corn leaf disease classification", IEEE Access, vol. 10, pp. 31103-31115, 2022.

[9] C. Zhou, S. Zhou, J. Xing and J. Song, "Tomato leaf disease identification by restructured deep residual dense network", IEEE Access, vol. 9, pp. 28822-28831, 2021.

[10] D. Wang, J. Wang, Z. Ren and W. Li, "DHBP: A dual-stream hierarchical bilinear pooling model for plant disease multi-task classification", Comput. Electron. Agricult., vol. 195, Apr. 2022.

[11] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang and Y. Shi, "MobileNet based apple leaf diseases identification", Mobile Netw. Appl., vol. 27, no. 1, pp. 172-180, Aug. 2020.

[12] S. Kodors, G. Lacis, O. Sokolova, V. Zhukovs and I. Apeinans, "Apple scab detection using CNN and transfer learning", Agronomy Res., vol. 19, no. 2, pp. 507-519, 2021.

[13] P. Wspanialy and M. Moussa, "A detection and severity estimation system for generic diseases of tomato greenhouse plants", Comput. Electron. Agricult., vol. 178, Nov. 2020.

[14] M. B. Tahir, M. A. Khan, K. Javed, S. Kadry, Y.-D. Zhang, T. Akram, et al., "Recognition of apple leaf diseases using deep learning and variances-controlled features reduction", Microprocess. Microsyst., 2021

[15] M. Agarwal, S. Gupta and K. K. Biswas, "A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant", Sustain. Comput. Informat. Syst., vol. 30, Jun. 2021.