Middle East Research Journal of Engineering and Technology
ISSN: 2789-7737 (Print) ISSN: 2958-2059 (Online)
Frequency: Bi-Monthly
DOI: https://doi.org/10.36348/merjet.2024.v04i04.007

KUWAIT SCHOLARS PUBLISHER
Website: http://www.kspublisher.com/
Email: office@kspublisher.com

Plant Disease Detection Using CNN

Mrs. Reshma[1*], Mrs. Varalakshmi B D[2], Prajwal B S[3], Rithvik K T[4], Sai Nikhil S[5], Shashank M[6]
[1,2]Asst Prof., Department of CSE Acharya Institute of Technology, Bengaluru, Karnataka
[3-6] B.E in Department of CSE Acharya Institute of Technology, Bengaluru, Karnataka

Abstract: Agriculture forms a cornerstone of the Indian economy, with food and cash crops playing a critical role in sustaining both the environment and human livelihoods. However, crop yields are significantly impacted each year by various plant diseases. The lack of efficient diagnostic methods, combined with limited awareness of disease symptoms and treatment options, often leads to widespread crop losses. This study explores the application of machine learning for plant disease detection, focusing on Convolutional Neural Networks (CNNs) to identify and classify diseases. The proposed approach employs advanced image processing techniques to analyze infected leaf regions, examining metrics such as time complexity and lesion area. The model was trained and tested on a curated dataset comprising 15 cases, including 12 disease categories such as Bell Pepper Bacterial Spot, Potato Early Blight, and Tomato Leaf Mold, alongside 3 categories of healthy leaves. The system achieved a test accuracy of 88.8%, demonstrating its potential for accurate plant disease detection. Performance evaluation was conducted using standard metrics to validate the model's reliability.

Keywords: CNN, image processing, dataset, test set.

Research Paper

***Corresponding Author:**
Mrs. Reshma
Asst Prof., Department of CSE Acharya Institute of Technology, Bengaluru, Karnataka

How to cite this paper:
Mrs. Reshma *et al* (2024). Plant Disease Detection Using CNN. *Middle East Res J. Eng. Technol, 4*(4): 138-145.

Article History:
| Submit: 21.11.2024 |
| Accepted: 25.12.2024 |
| Published: 29.12.2024 |

I. INTRODUCTION

India, with a population of approximately 1.38 billion as of April 2020, relies heavily on agriculture as a cornerstone of its economy. The agricultural sector contributes around 18% to the country's GDP and supports nearly 95.8 million farmers. Enhancing agricultural practices can lead to significant economic benefits, improved livelihoods for farmers, and the creation of new employment opportunities. Despite advancements in research on pesticides, fungicides, and herbicides, millions of tons of crops are lost annually due to diseases caused by natural factors. Early and efficient detection of plant diseases is essential to mitigate these losses and alleviate the economic challenges faced by Indian farmers.

Modern technology offers promising solutions to address these challenges. With access to the internet and mobile devices, farmers can now leverage tools to capture images of affected plants and use automated systems for disease diagnosis. Such systems not only identify the specific disease but also recommend appropriate treatments and pesticides. The majority of plant diseases are caused by fungal and bacterial infections, and factors like climate change and increasing population further exacerbate the issue. Regular monitoring of plant leaves is crucial to identify diseases early and implement timely interventions to prevent severe crop damage.

Several researchers have proposed various techniques for plant disease detection and monitoring. Usama Mokhtar utilized Gabor wavelet transform techniques to extract features from tomato leaves. Support Vector Machines (SVM) were employed to detect diseases like early blight and powdery mildew. The experiments used real sample images of tomato leaves, resized to 512×512 resolution during preprocessing to reduce computational time. Background subtraction was applied to eliminate unnecessary elements, and the SVM classifier used kernel functions for training and testing [3].

Ganesan introduced a fuzzy-based segmentation method paired with computer vision for early identification of plant diseases. The diseased portion of a plant leaf was isolated using image segmentation, and color space analysis was performed to determine the characteristics of the affected areas [4].

Arthit Srikaew, Kitti Attakitmongcol, and Prayoth Kumsawat developed a neural network-based diagnosis method using unsupervised learning with color imagery. Their system consisted of two main components: disease feature extraction, leveraging co-occurrence matrices and texture equations, and classification using a fuzzy ARTMAP neural network. This method classified grape leaf images into five categories, including healthy, rust, scab, downy mildew, and powdery mildew, achieving 90% accuracy [5].

H. Sabrol and K. Satish studied five types of tomato diseases, including late blight, bacterial spot, Septoria spot, bacterial canker, and leaf curl. They used features like color, texture, and shape extracted from images of healthy and diseased tomato plants. These features were input into a classification tree, achieving a classification accuracy of 97.3% [6].

N. Petrillis reported a smartphone-based application for plant disease diagnosis, utilizing color normalization for better accuracy [7,8]. M. Islam implemented a multiclass plant disease detection approach using SVM [9].

Haiguang Wang *et al.*, employed four types of neural networks—backpropagation (BP), radial basis function (RBF), generalized regression (GRNN), and probabilistic (PNN)—to distinguish between diseases like wheat stripe rust, wheat leaf rust, grape downy mildew, and grape powdery mildew. Their model used color, texture, and shape features for disease recognition [10].

K.K. Singh proposed a cloud-based collaborative platform for plant disease identification and forecasting to assist farmers in preventive measures [11]. Various segmentation methods, such as K-means clustering and thresholding, were also explored for lesion detection. These methods provided moderately effective outcomes for diverse datasets, though K-means clustering required prior information about cluster centers [12].

Anton Louise P. de Ocampo and Elmer P. Dadios presented a lightweight neural network model for plant disease detection on a mobile platform. Their approach involved a two-step training process: pre-training on the ImageNet dataset and fine-tuning with specific plant disease datasets, achieving a test accuracy of 89.0% [13].

S. Dubey and M. Dixit explored the use of deep convolutional neural networks for image-based classification. They demonstrated its effectiveness in detecting and recognizing visual patterns with high accuracy, laying the groundwork for similar approaches in plant disease detection [14].

This paper evaluates the role of image processing in plant disease detection using a CNN-based approach and outlines potential areas for future improvement. The structure of this work is as follows: Section II discusses the methodology, Section III elaborates on the CNN method and its implementation, Section IV presents simulation results, and Section V concludes the study.

II. METHODOLOGY

The proposed plant disease prediction method uses images of plant leaves as input. The workflow of the proposed approach is illustrated in Fig. 1. Initially, the input images undergo preprocessing, which includes resizing the images and converting them into NumPy arrays. Subsequently, the dataset is segregated into image data and their corresponding labels.

The model training process utilizes a curated dataset containing images of both healthy and diseased plant leaves relevant to this study. The labeled data is stored in pickle files, which are later extracted during the training phase.

The architecture of the model includes convolutional layers followed by max-pooling layers. To prevent overfitting, 25% of the data is dropped out during training. The output from the convolutional layers is flattened and fed into a dense network. The final layer employs a softmax activation function, which outputs probability values corresponding to the various disease labels.

Adam optimizer is utilized. The framework consequently distinguishes the picture of leaf given and pre-processes the picture further for prediction. The model will produce 15 distinctive probability values for 15 labels respectively among which the probability value with highest score to the relating name will be the anticipated disease or result for that particular image.

The predicted disease label can then be used to provide actionable insights and recommendations for farmers or agricultural practitioners. These insights may include information about the disease symptoms, its potential impact on crop health, and suggested treatments such as appropriate pesticides, fungicides, or other remedial measures. To ensure the system's practical applicability, it can be integrated with a user-friendly interface, such as a mobile application or a web- based dashboard, allowing users to upload images and receive instant diagnosis and guidance. Furthermore, the system can store historical data to track disease patterns over time, which could aid in proactive monitoring and decision- making in agricultural practices. This historical data can be analyzed to identify trends and predict potential outbreaks, allowing for timely interventions. Additionally, it could be shared with agricultural experts or extension services to enhance knowledge sharing and provide localized recommendations.

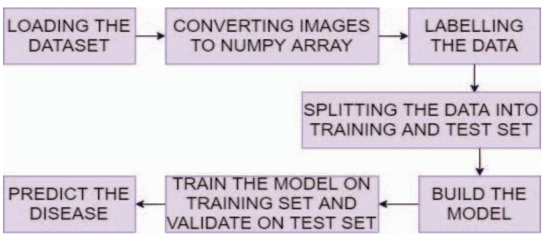

Fig. 1: Block diagram of the proposed scheme

I. CNN

In machine learning, Convolutional Neural Networks (CNNs) employ a unique approach to regularization, which is simpler and more effective compared to traditional regularization techniques. The following outlines the layers of the CNN architecture.

A. *Input Layer*

The input layer serves as the entry point for the data into the model. In this stage, the number of neurons corresponds to the number of features, with each feature typically representing a pixel in an image. For example, in an image, the total number of pixels becomes the total number of features used in the model. The input data is split into two subsets: one for training and the other for testing. The majority of the data is used for training the model, while a smaller portion is reserved for testing its performance. Fig. 2 illustrates two different images of tomato leaves, which are used for training and testing in this process.

Fig. 2. Sample Images from the database (from left to right) Potato Early Blight, Tomato Healthy

B. *Hidden Layer*

The hidden layer takes the output from the input layer and processes it further. The number of neurons in the hidden layer(s) depends on the specific model architecture and the complexity of the data.

Inputs: An image matrix (volume)of dimension(h*w*d). A filter (fh*fw*d).

Output: Output volume dimension (h-fh+1)*(w-fw+1)*1

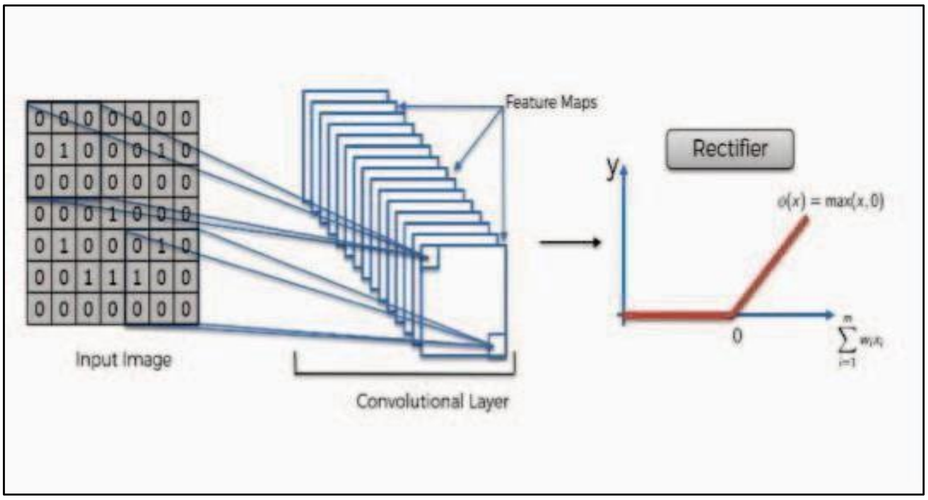

Fig. 3: Rectified Linear Unit [14]

1. Pooling Layer - The pooling layer functions such a way that a 2D filter slides over every channel of the feature map and conveys the features lying within the area enveloped by the filter. Given a specified dimension of any feature map the pooling layer output dimension is expressed as follows:

$$\frac{(n_h - f + 1)}{s} * \frac{(n_w - f + 1)}{s} * n_c$$

nh – Feature map height
nw – Feature map width.
nc- Number of channels included in each feature map.
f- Filter size
s- Length of stride

2. Max Pooling Layer – It is the region of the feature map where the maximum values are selected and retained by the filter. Therefore, the max-pooling layer's output is a feature map that highlights the most important features from the previous layer. Fig. 4 illustrates the max-pooling layer.

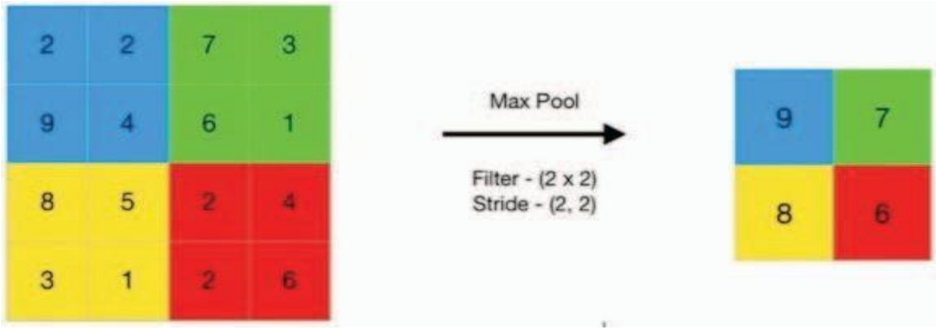

Fig. 4: Max Pooling Layer [15]

3. **Fully Connected Layer-** The fully connected (FC) layer in a CNN represents the aggregated feature vector for the input data. This layer contains essential information derived from the previous layers, encapsulating key patterns and features necessary for classification or regression tasks. During training, the feature vector is used to make predictions and calculate the loss function, which aids in optimizing the network. The convolutional layers preceding the FC layer extract important low-level and high-level features from the input, which are then compressed into this feature vector. These extracted features are crucial for determining the accuracy of the model's predictions and contribute significantly to the network's learning process. The FC layer plays a critical role in transitioning from raw feature maps to actionable insights or final output predictions.

Table I: Model Summary 1

LAYER	OUTPUT	PARAMETER
CONV2D_1	(256,256,32)	896
MAX_POOLING2D_1	(85,85,32)	0
CONV2D_2	(85,85,64)	18496
CONV2D_3	(85,85,64)	36928
MAX_POOLING2D_2	(42,42,64)	0
FLATTEN_1	(56448)	0
DENSE_1	(1024)	57803776
DENSE_2	(15)	15375

Filters in CNNs are not pre-programmed; instead, they are learned through training. These filters progressively learn to detect abstract features, such as edges, textures, or facial features, depending on the complexity of the model. Multiple convolution layers work in tandem to extract increasingly sophisticated and deeper information from an image. Think of filters as selective barriers that allow specific patterns or characteristics to pass through, while blocking others. Feature maps are the results generated by these filters in the convolution layer, capturing the most relevant information about the input. Figures 5 through 8 illustrate the filters and corresponding feature maps from the first and second convolution layers.

In addition to their role in feature extraction, filters and feature maps allow CNNs to develop a hierarchical understanding of the image. Initially, in the first convolution layers, filters detect basic features like edges and simple textures. As the network progresses through deeper layers, these basic features combine and form more complex patterns, such as shapes, objects, or even entire scenes. This multi-layered processing helps the network understand an image at various levels of abstraction, enabling it to recognize intricate structures or specific objects with high accuracy.

The process of learning the optimal filters is what enables CNNs to perform well on a wide range of tasks, from object recognition to plant disease detection. Filters evolve during training through backpropagation, where the network adjusts their weights based on the error in its predictions. The feature maps, which are generated from the output of the filters, play a critical role in making these predictions. They provide a spatial representation of the image that helps the network capture both the location and the presence of important features.

Moreover, the depth of the network allows it to progressively capture more refined details. With each additional layer, the network can analyze the input image from a broader perspective. As a result, the CNN becomes highly effective in detecting patterns that may not be immediately apparent in the raw input data, thus improving the model's accuracy in tasks like plant disease classification. This hierarchical learning enables CNNs to generalize well across various types of images, making them highly adaptable to different datasets and applications. For example, in plant disease detection, the network can learn to identify subtle variations in texture, color, and shape that are indicative of specific diseases.

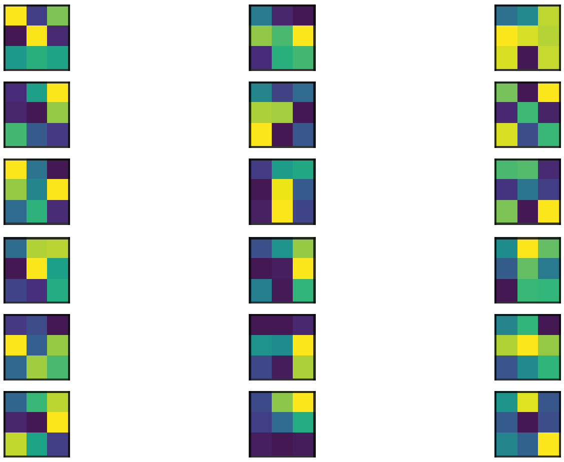

Fig. 5: Visualization of filter for 1st convolution layer

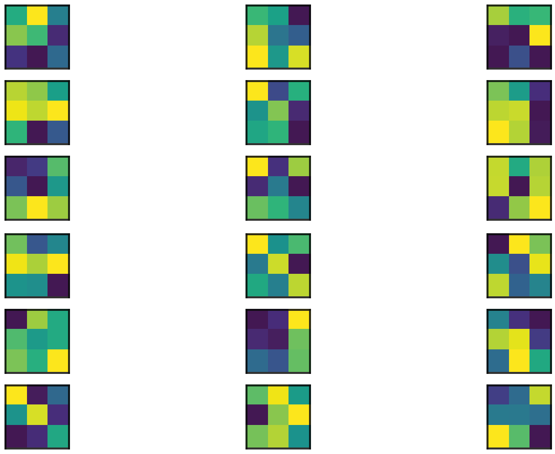

Fig. 6: Visualization of filter for 2nd convolution layer

III. SIMULATION RESULT

The simulation study was conducted using Python programming along with Deep Learning frameworks such as TensorFlow and Keras. A dataset comprising 200 images for each of the 15 classes, totaling 3000 images, was used for the study. The dataset was split in an 80:20 ratio, allocating 2400 images for training and 600 images for testing. Training was performed on a system equipped with an Intel Core i7-9700k processor running at a clock speed of 3.60 GHz with 12 MB cache, coupled with 16 GB RAM and an NVIDIA GeForce GTX 1660 Super GPU for enhanced computational efficiency.

The model was trained over 50 epochs with a batch size of 32, and categorical cross-entropy was used as the loss function. The Adam optimizer was employed to accelerate convergence and optimize the learning process. Early stopping was integrated to prevent overfitting by monitoring validation loss and halting the training if no improvement was observed over 10 consecutive epochs.The performance of the model was evaluated using metrics such as accuracy, precision, recall, and F1-score. During testing, the model achieved an overall accuracy of 88.8%, with the diseased leaf classes showing better precision compared to healthy leaf classes. The results highlight the model's ability to distinguish between various plant diseases effectively, even in cases of overlapping symptoms or visually similar diseases. The incorporation of dropout layers and optimized training parameters played a significant role in enhancing the model's robustness and reducing overfitting.

The simulation was conducted on a system equipped with 16 GB DDR4 RAM, running Windows 10 (64-bit). The total computational time for the process was approximately 30 minutes, which included about 3 minutes dedicated to data preprocessing. While the input feed from the webcam currently lacks the capability to effectively eliminate the background from captured frames, the prediction performance on provided input images remains satisfactory. To mitigate overfitting—a scenario where there is a significant discrepancy between training and testing accuracy—a dropout layer was strategically implemented after the 1st (p=0.25), 3rd (p=0.25), and 5th (p=0.5) layers of the model. This adjustment resulted in a training accuracy of 97.42% and a testing accuracy of 88.80%. The relatively small gap between these accuracies confirms that the model is well-generalized and not overfitted. A comparative analysis of the proposed method against existing approaches is presented in Table II, showcasing its effectiveness. Key performance metrics, outlined in Table III, demonstrate that the proposed approach achieves superior accuracy and overall performance, indicating its reliability and improvement over previous methodologies.

Table II: Comparative study of two model

Image Type	Number of Images	Architecture			Test Accuracy
Grayscale	20639	L1	Conv	3x3	74.19
			Pool	2x2	
		L2	Conv	3x3	
			Pool	2x2	
Rgb	3000	L1	Conv	3x3	88.80%.
			Pool	3x3	
		L2	Conv	3x3	
		L3	Conv	3x3	
			Pool	3x3	
		L4	Conv	3x3	
		L5	Conv	3x3	
			Pool	2x2	

Table III: Performance metrices of the model

Class	Precision	Recall	F1 Score	Support
0	1.00	0.02	0.04	45
1	1.00	0.03	0.05	40
2	0.36	0.17	0.23	54
3	0.00	0.00	0.00	43
4	1.00	0.10	0.18	20
5	0.00	0.00	0.00	37
6	0.00	0.00	0.00	41
7	1.00	0.14	0.25	35
8	0.56	0.24	0.33	38
9	0.21	0.39	0.27	33
10	0.00	0.00	0.00	36
11	0.00	0.00	0.00	51
12	1.00	0.19	0.32	32
13	0.00	0.00	0.00	39
14	0.10	1.00	0.18	47
Micro Avg	0.16	0.16	0.16	591
Macro Avg	0.42	0.15	0.12	591
Weighted Avg	0.38	0.16	0.12	591

Fig. 7 and Fig. 8 is showing training and validation loss and accuracy graph respectively.

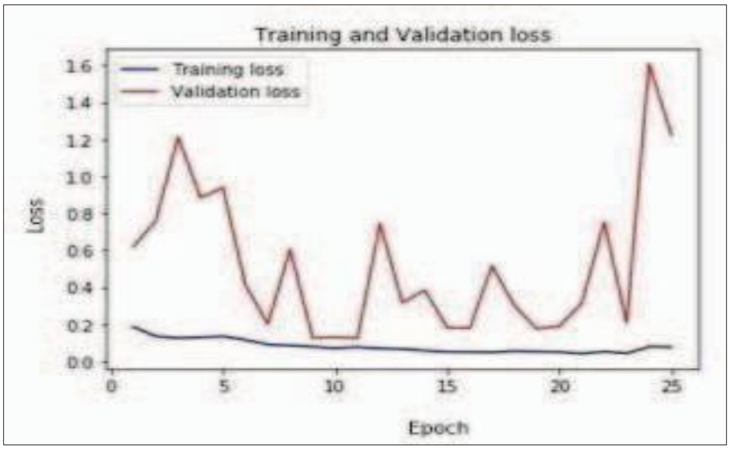

Fig. 7: Training loss and Validation loss graph

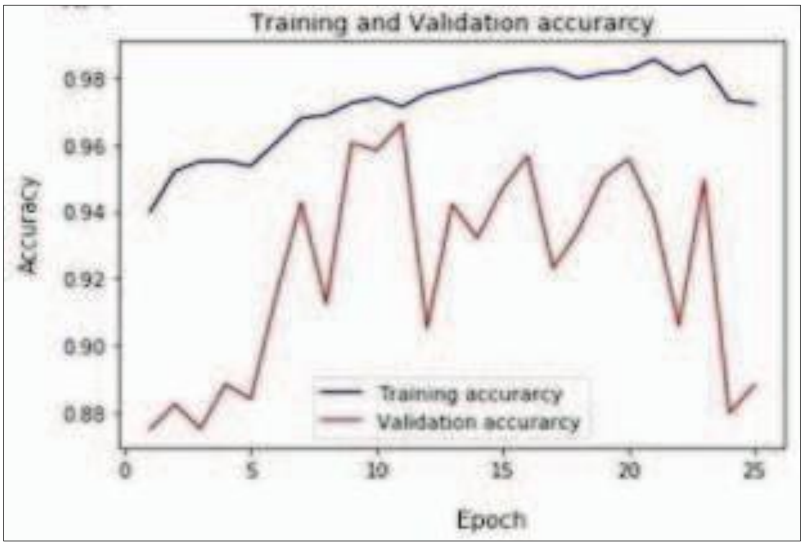

Fig. 8. Training accuracy and Validation accuracy graph

IV. CONCLUSION

The proposed algorithm has been effectively implemented, achieving a test accuracy of 88.80% without signs of overfitting. However, there remains a 12.20% gap that presents an opportunity for future enhancement. This work holds significant promise for the agricultural domain, offering farmers a reliable tool for monitoring crops and tracking plant health. It can also aid individuals in maintaining the health of their houseplants. Expanding this system into a mobile application could provide users with disease detection capabilities along with recommended remedies, making it a comprehensive solution for plant care and management.

REFERENCES

1. U. Mokhtar, M.A.S.Ali, A.E.Hassenian, H.Hefny, “Tomato leaves disease detection approach based on support vector machines”, 11th international computer engineering conference (ICENCO),Cairo, Egypt, 29-30 Dec.2015, pp. 246-250.
2. Ganesan, S.Sajiv, L.Megalanleo Leo, “CIELuv Color Space for Identification and Segmentation of Disease Affected Plant Leaves Using Fuzzy based Approach”, 3rd International Conference on Science Technology Engineering & Management (ICONSTEM), Chennai India, 23-24 March 2017, pp. 889-894.
3. M.Sardogan, A.Tuncer, Y.Ozen, “Plant Leaf Disease Detection and Classification based on CNN with LVQ Algorithm”, 3rd International Conference on Computer Science and Engineering (UBMK), Sarajevo, 20-23 Sept. 2018, pp. 382-385.
4. A.Sehgal, S.Mathur, “Plant Disease Classification Using SOFT COMPUTING Supervised Machine Learning”, 3rd International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 12-14June 2019, pp.75-80.
5. C. Khitthuk, A.Srikaew, K.Attakitmongcol, P.Kumsawat “Plant Leaf Disease Diagnosis from Color Imagery Using Co-Occurrence Matrix and Artificial Intelligence System”, International Electrical Engineering Congress(iEECON), Krabi, Thailand, 7-9 March 2018.
6. H.Sabrol, K.Satish, “Tomato Plant Disease Classification in Digital Images using Classification Tree”, International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 6-8 April 2016, pp. 1242-1246.
7. N.Petrillis, “Plant Disease Diagnosis With color Normalization”, 8th International Conference on Modern Circuits and Systems Technologies(MOCAST), Thessaloniki, Greece,13-15 May 2019.
8. N.Petrellis, “A Smart Phone Image Processing Application for Plant Disease Diagnosis”, 6th International Conference on Modern Circuits and Systems Technologies(MOCAST), Thessaloniki, Greece, 4-6May2016.
9. M.Islam, AnhDinh, K.Wahid, P.Bhowmik, “Detection of Potato Diseases Using Image Segmentation and Multiclass Support Vector Machine”, IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, Canada, 30 April- 3 May 2017 .
10. H.Wang, G.Li, Z.Ma, X.Li, “Application of Neural Networks to Image Recognition of Plant Diseases”, International Conference on Systems and Informatics(ICSAI2012), Yantai, China, 19-20 May 2012, pp. 2159-2164.
11. K.K.Singh, “An Artificial Intelligence and Cloud-Based Collaborative Platform for Plant Disease Identification, Tracking and Forecasting for Farmers”, IEEE International Conference on Cloud Computing in Emerging Markets(CCEM), Bangalore, India, 23-24 Nov 2018, pp. 49-56.
12. E.Agustina, I.Pratomo, A.D.Wibawa, S.Rahayu, “Expert System for Diagnosis Pests and Diseases of The Rice Plant using Forward Chaining and

Certainty Factor Method", International Seminar on Intelligent Technology and Its Applications(ISITIA), Surabaya, Indonesia, 28-29 Aug 2017, pp. 266-270.
13. A.Louise P.de Ocampo, E.P.Dadios, "Mobile Platform Implementation of Lightweight Neural Network Model for Plant Disease Detection and Recognition", IEEE 10th International conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Enviroment and Management(HNICEM), Baguio, Philippines, 29 Nov-2 Dec2018.
14. S.Dubey, M.Dixit, "Facial Expression Recognition using Deep Convolutional Neural Network", International Symposium on Advanced Intelligent Informatics(SAIN) , Vol.8, No.1 pp. 96- 101,Aug 2018.
15. GeeksforGeeks- https://www.geeksforgeeks.org/cnn-introduction-to- pooling-layer/