e-ISSN : 0976-5166
p-ISSN : 2231-3850

LEAF DISEASE IDENTIFICATION BY EXTRACTING GRADIENT LOCAL TERNARY PATTERN & ZERNIKE MOMENT

Nagamani H S
Research Scholar, University of Mysore, India
Asst. Prof. Dept of Computer Science,
Smt V H D Central Institute of Home Science, (MCU),
Bangalore, India
nagamanihs.pubications@gmail.com

Dr. Sarojadevi H
Prof. Nitte Meenakshi Institute of Technology (NMIT),
Bangalore, India

Abstract

The agricultural field is one of the major occupations and plays a significant role in India. The rural people depend on agriculture as their primary income. Plant diseases play a vital issue seen by the farmers. They spend much money—the disease management without sufficient technical help results in poor disease control. Plant disease can also ruin the ecosystem and create new environmental problems due to poor land management. Monitoring the diseases in plants with the support of professionals is expensive. Therefore, a requirement arises for an automatic system to identify the plant diseases as it helps to keep track of large field crops and take precautions at early stages. The presented approach aims to develop an automatic system to detect plant disease using leaf images. This paper introduces automatic identification and classifying leaf diseases by adopting color thresholding and flood filling techniques for segmentation and extracting relevant information gradient local ternary patterns and Zernike moments are used. The CNN classifier is trained to detect leaf diseases. Our experimental results proved that the presented method had generated an accuracy of 95.49%.

Keywords: **Disease Management, Automatic System, Color Thresholding, Flood Filling, Gradient Local Ternary Pattern, Zernike Moments, CNN Classifier.**

1. Introduction

Plant disease management is always challenging to grow healthy and produce a good yield. The diseases affect the plants by interrupting many processes like absorbance, transferring water and required nutrients, developing flowers and fruits, photosynthesis, development of plants, enlargement, and cell division. The diseases on plants are due to many types of fungus, bacterial, phytoplasma, viruses, viroids, nematodes, and many others through other resources. The symptoms and effects of pathogens show from lower to death of the affected plants based on the type of pathogens, environmental conditions, the resistance of a type of host, and other factors. The infection on plants varies from the symptoms and on different plants like leaves, roots, stems, fruit rots, and fruit spots.

The plant leaves, roots, and production determine the health of the plants. Leaves are the surfaces from which plants collect sunlight and conduct photosynthesis. The leaf diseases majorly affect the number of crops [Vivek and Deepak (2016)]. Agriculturists are affected through economic drawback as the leaf diseases increases [Ms. Kiran (2014)]. The leaves of the plants are most easily seen and provide more information on the plant's health. For example, dark green leaves may have a surplus amount of nitrogen. Light green or yellow color leaves have a shortage of nitrogen storage. This results in a lag of growth and reduces chloroplast production in plants. Changes in leaf color indicate the deficiency. A discolor in a single leaf is no reason for worry. When it spreads on a large scale, severe observations and precautions as to be started; these problems are caused by two factors: Biotic parameters and Abiotic elements [Priyanka et al. (2017)], the diseases generated by organisms added to biotic elements like fungi, bacteria, and viruses. The diseases originated from a nutrition deficiency, lower value of soil pH, insufficient light energy, and extreme condition of weather from abiotic elements.

Types of Plant Diseases are partitioned into three groups: Fungal, Bacterial and viral diseases.

1.1. *Fungal Diseases*

Fungal Disease instances of fungal diseases are Wilt, Powdery mildew, Downy mildew, Anthracnose, Alternia, Leaf spot, Grey mildew, Rots, Cankers, Molds. The fungi cause these diseases. The fungal disease symptoms are seen as marks on leaves of plants; leaves turn yellow, and birdseye points on berries. Fig. 1 shows a few of the diseases caused by fungus on leaves.

(a) Late blight (b) Early bight (c) Downy mildew

Figure 1: Plant leaves due to fungus.

1.2. *Bacterial Diseases*

Bacterial diseases are complicated to find out. Common diseases by bacteria are Bacterial blight, Crown gall, Wilts, Soft spots. Symptoms of these diseases are appeared to be green pale in color marks on plant leaves. The marks seem to be dead spots. Fig. 2 shows some of the diseases caused by bacteria on leaves.

(a) Crown gall (b) Bacterial blight (c) Soft spots

Figure 2: Leaf affected by bacteria

1.3. *Viral diseases*

Virus-caused diseases are Leaf curl, Leaf crumple, Leafroll. The virus affects these diseases, and their identification is the most difficult. The leaves which are infected seem to be frizzed and furrowed. Fig. 3 shows some of the diseases caused by viral on leaves.

(a) Leaf Cankers (b) Leaf Rots (c) Mosaic Virus

Figure 3: Leaf affected by the virus

Detecting the leaf disease manually by observing the naked eye requires a team of experts and continuous monitoring. It is expensive when the farm is enormous. So, image processing techniques can be used to automatically identify diseases in the leaf that need less time, money, and hard work compared with manual ways. Early identification of diseases in leaves increases the production of crops. Adopting many image processing approaches like segmentation, identification, and classification, the diseases affected leaves can be discovered at the initial level, and the quantity and quality of crops are upgraded. Many agriculturists lack facilities or concepts in contacting the experts, which is more expensive, time-consuming, and less accurate. In this scenario, the proposed system resulted being more beneficial in observing the crops. Detection of plant disease using the symptoms of leaves makes the system more accessible and less expensive. Using an automatic detection process consumes less time, less effort, and more accuracy. In this paper, features are analyzed of various plant leaves had affected by various plant diseases. The dataset is created by leaves collecting images infected by diseases such as black rout, fungal, powdery mildew, scorch, and regular leaves. We categorize these plant leaves depending on features trained by the CNN classifier.

2. Literature Survey

[Akila and Deepan (2018)] has proposed an approach of deep learning to identify diseases in the leaf of various plants with the help of plant leaves. They have used different detectors: Convolutional Neural Network Region-based Fully Convolutional Network and Single Shot Multibox Detector, to detect and classify the leaf diseases in plants. Their main contribution of the system is that it provides the status of infection of the disease in leaves and suggests the solution by naming the fertilizers for the particular disease.

[Sanjay *et al.* (2018)] have built a database with different diseases like black spot disease, yellow Sigatoka disease, frog eye disease, powdery mildew disease, tobacco ringspot disease, tomato plant disease, and valencia leafspot disease. Their proposed system has three levels: 1) identifying the infected part of leaves: the process is carried out by converting RGB into Gray, resizing the image, and applying median filtering. Using a k-means algorithm, the infected part is segmented from the other region. 2) Extracting the features from the segmented disease region, the various features like Energy, Entropy, Co-variance, Standard deviation, and Background intensity average are collected. 3) Using the ANN classifier, images identify and classify the type of plant disease.

[Mrs. Gaganpreet *et al.* (2018)] in this paper have reviewed and provided any information that the researchers can utilize to identify diseases in plants in their initial stages. They have presented methods of image processing applied in the agricultural field, basic steps to be adopted, various filters that can be used to de-noise the images, which have been discussed more clearly.

[Ravindra *et al.* (2016)] presented an algorithm for automatically segmenting the region of interest and automatic detection and classification of plant diseases. Initially, the image undergoes filtering as a pre-processing step, and efficient segmentation is formed by using a genetic algorithm. Energy, Entropy, Contrast Correlation Coefficient, and Homogeneity are extracted features. Classification is carried out for automatic detection and classification by combining SVM, genetic algorithm, and neural network.

[Shantanu *et al.* (2019)] developed a user-friendly web-based application for farmers. Their study mainly focuses on cotton leaf diseases like Alternaria Macrospora and Bacterial Blight. The farmers upload a leaf image to the model; image processing begins with digitizing diseased colored images passing through three Convolutional Layers for extracting features, pooling, and flattening. Finally, the CNN classifier detects the plant disease. The developed application also provides preventive information for the diseases. The database used for training consists of 513 leaf images, and to perform validating database includes 207 leaf images. The training performance is 80%, and the testing accuracy is 89%.

[Kanaka and Anuradha (2019)] build a model to recognize the disease of tomato and maize crops. To obtain maximum accuracy, they have tried to identify the disease based on machine learning and artificial neural networks. The method called label edge detection is used for segmentation. It is very efficient as it calculates the gradient of photograph intensities at each pixel within the image. HOG features are extracted from the segmented image. Finally, SVM and ANN classifiers classify the healthy and unhealthy disease leave images.

[Peng *et al.* (2019)] from the initial stage, the input images of all the leaf samples are taken and converted into HIS format, for segmentation masking of the green pixel is performed to obtain the region of interest marked green components are removed. The segmented region texture feature called Color-Co- Occurrence vector is extracted and trained using the ANN classifier. The proposed algorithm produced an Efficient and successful detection and classification of the diseases, providing an accuracy rate of 94%. Experiments and analyses on a database were carried for 500 leaf plants to confirm the adopted approach's efficiency.

3. Methodology

The Overall Architecture of the proposed method is shown in Fig. 4 shows. The architecture includes input image, pre-processing block, segmentation block, Feature extraction block, and classification block. Here the workings of the system are partitioned into two phases: training and testing process. Training processes are performed by the features of diseased leaf images and saved in the knowledge base. The input image is resized by the pre-processing. The pre-processed image is segmented by color thresholding and flood filling. The segmented image consists of either a healthy region or a disease-affected region of the leaf. Features of the segmented image are extracted using descriptors such as Zernike Moments and Gradient Local Ternary Pattern (GLTP) methods.

The extracted feature vectors are classified using CNN classifier to identify the leaf disease.

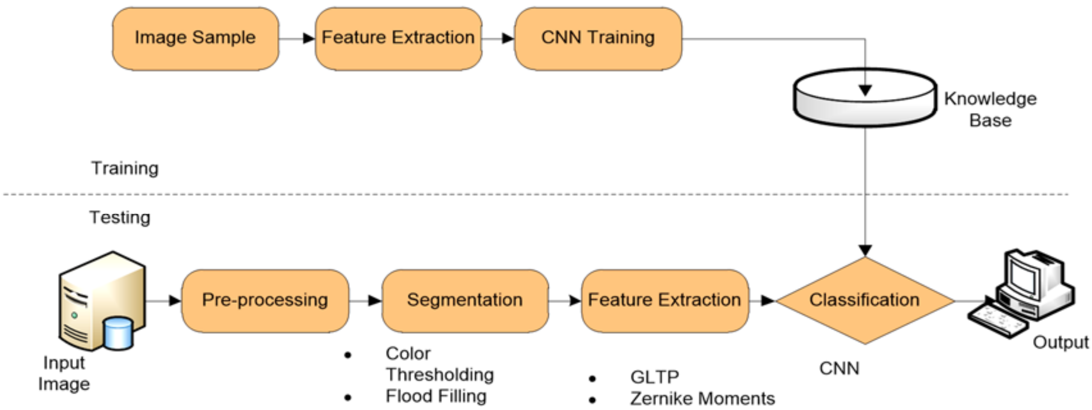


Figure 4: Overall system architecture

3.1. *Image Acquisition*

The camera is employed in recording the leaf image, and a database of different classes is created. In the proposed system, seven different classes are considered and classified as healthy and unhealthy leaf images. The database contains seven different tomato leaf diseases classes: healthy leaf images, Bacterial Spot, Mosaic Virus, Yellow leaf curl virus, early blight, late blight, and leaf mold. Fig. 5 shows the image of the leaf affected by the yellow Spot Curly Leaf.

Figure 5: Input sample images

3.1.1. Pre-processing

The natural plant leaf image includes shadow-based or illumination, which is based on the brightness parameter of the image. The existence of shadow or illumination causes the degradation of leaf region detection. Hence the images captured have to be cleaned before processing. Noises from the images must be removed, and unclear information must be enhanced. The well-known techniques of image processing known as pre-processing are utilized for enhancing the quality of input images. Usually, images captured through the camera vary in size. Most deep learning algorithms require the entire input image to be similar in size.

3.1.2. Image RESIZE

The image resizes method is applied to resize all the images in the database to an exact size. The equation below is used to resize the input leaf image

$$ResizedImg = imresize\,(Img, [512\ 512]) \tag{1}$$

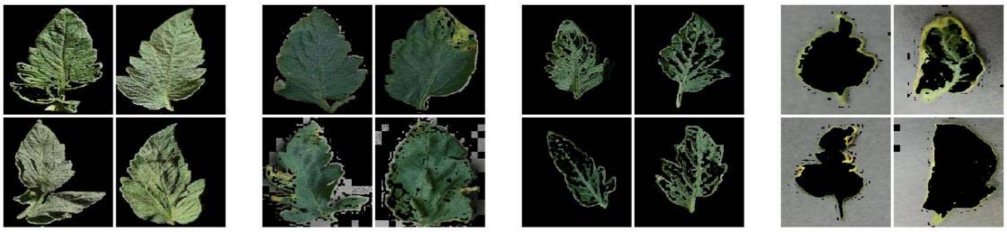
Image enhancement is the method for improving the quality and information of the input image before processing. In this paper, we have used contrast enhancement to enhance the input image. The contrast of an image can be defined as a spectrum of the presence of dark and lighter pixels [Tanmoy (2019)]. A contrast lower image shows the minor changes within its lighter and darker pixel values, and the histogram generated is narrow. The human eye is very effective on contrast other than the intensity of absolute pixel; a much good image is needed by extending the image histogram; therefore, the range of the fully dynamic image is filled up. Fig. 6 shows an enhanced image for the input leaf image.

3.2. *Segmentation*

Image segmentation [Nameirakpam et al. (2015)] is utilized to detect the objects and boundaries inside the pictures. The pre-processed picture is used to segment the Region of Interest (ROI), with the outcome of the segmentation being highlighted as an item of interest. Color thresholding and flood filling are two segmentation methods used in the proposed system. For greater accuracy, the output of both algorithms is integrated using ROI (region of interest).

3.2.1. Color Thresholding

[Alaa and Hisham (2010)] Color-based segmentation includes dividing an input color picture into multiple color models, such as RGB, HSV, YCbCr, and the CIE Lab color model. Then perform segmentation. Fig. 6 shows the output from the color thresholding segmentation method for the respective input images.



(a) (b) (c) (d)

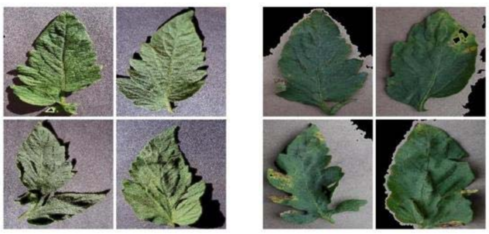
Figure 6: (a), (b), (c) and (d) are the Color Thresholding Segmentation Output

3.2.2. Flood Fill

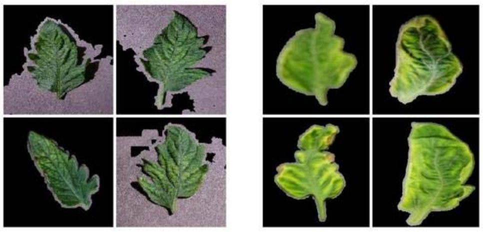
Flood filling is a method that locates the region in a multi-dimensional array associated with a particular node. In most cases, the approach is used to paint a linked pixel's whole region with the same color. The first pixel is known as a seed, and if it matches the proper color, more and more seed pixels are picked. Testing for additional pixels must be altered each time the added seed pixel is colored with the initial seed. There are a variety of flood filling algorithms available: Scan line based, 4-way, 8-way. This study employs a four-way flood filling method. Three changeable beginning nodes, a destination color, and a substituted color are included in the method. The following steps make up the recursive algorithm:

(1) At this point, a starting position and a color are allocated.
(2) Its four neighbors recursively contribute the following fresh seed.
(3) The entire operation is performed for each fresh seed that meets the following criteria:
    (i) The pixel is located within the screen.
    (ii) The pixel's color has faded.
    (iii) The new color is not applied to the pixel.

A grayscale image is created from the RGB input image. On a grayscale picture, the flood fill is employed. With a tolerance of 0.4, the seed point is chosen in R=112 and C=92. The outcomes of using the flood filling algorithm are shown in Fig. 7.



(a) (b)



(d) (e)

Figure 7: Flood Filling Segmentation Output.

3.3. *Feature extraction*

The segmented region is used to extract features of the diseased region of the leaf. Zernike Moments and Gradient Local Ternary Pattern (GLTP) methods are used.

3.3.1. Zernike Moments

Zernike Moments belongs to the type of orthogonal moments, which are efficient for image representation depending on orthogonal Zernike radial polynomials. The algorithm of Zernike Moments for an image function $I(i,j)$ with spatial dimension$M \times N$, their Zernike Moments of order n with repeating l are written by:

$$A_{nl} = \frac{n+1}{\pi}, \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i,j), R_{nl}(r_{ij}). e^{-il\theta_{ij}} \quad (2)$$

where the discrete polar coordinates r_{ij} and θ_{ij} are respectively defined as follows:

$$r_{ij} = \sqrt{x_j^2 + y_i^2} \quad (3)$$

$$\theta_{ij} = arctan\left(\frac{y_i}{x_j}\right) \quad (4)$$

The Cartesian coordinates x_j and y_i are given by:

$$x_j = c + \frac{j.(d-c)}{N-1} \; 0 \leq j \leq N-1 \quad (5)$$

$$y_i = d - \frac{i.(d-c)}{M-1} \; 0 \leq i \leq M-1 \quad (6)$$

where c & d are real numbers selected based on whether the image function is mapped outside or inside a unit circle. Circle outside c = −1 and d = 1. Circle inside, c = −1/ √ 2 and d = 1/ √ 2. The real values radial polynomials $R_{nl}(r)$, are given by:

$$R_{nl}(r) = \sum_{s=0}^{\frac{n-|l|}{2}} (-1)^2 \frac{(n-1)!}{S!\left(\frac{n+|l|}{2} - S\right)!\left(\frac{n-|l|}{2} - S\right)!} \quad (7)$$

where $|l| \leq n$ and $n - |l|$ is always even.

3.3.2. Gradient Local Ternary Pattern (GLTP)

The gradient local ternary pattern is a texture-dependent feature extraction technique [Qiang *et al.* (, 2010)] by computing gradient magnitudes of neighborhoods within the segmented region and then quantizing the values based on three different levels of discriminations. The patterns generated as a result are utilized as feature descriptors. GLTP is more robust against variations in illuminations and noise as GLTP selects values that are more accurate w.r.t to gradient magnitude and performs three types of encoding schemes to differentiate between smooth and textured regions. The steps are explained in the algorithm.

Algorithm of Gradient Local Ternary Pattern

Inputs: Pre-Processed input leaf image.
Output: Feature Vector of GLTP histograms

Step.1 *Initial by applying Sobel operator derivatives are calculated along the horizontal* (Gx) *and vertical* (Gy) *approximation of the input image.*

Step.2 *Each pixel gradient magnitude is calculated using*

$$G_{x,y} = \sqrt{G_x^2 + G_y^2}$$

Step.3 *To separate smooth and highly textured areas, a threshold* $\pm t$ *is adopted around the center of computed gradient values* (Gc) *which of size 3x3 pixels throughout the gradient magnitude image to find out*

$$S_{GLTP}(G_c, G_i) = \begin{cases} -1 & G_i < G_c - t \\ 0 & G_c - t \leq G_i \leq G_c + t \\ +1 & G_i > G_c + t \end{cases}$$

Step.4 *Computing both positive* P_{GLTP} *and negative* N_{GLTP} *GLTP coded image representations from SGLTP values*

$$P_{GLTP} = \sum_{i=0}^{7} Sp(S_{GLTP}(i)) x\, 2^i$$

$$Sp(v) = \begin{cases} One\ if\ v > 0 \\ 0 \quad else \end{cases}$$

$$N_{GLTP} = \sum_{i=0}^{7} Sn(S_{GLTP}(i)) x\, 2^i$$

$$Sn(v) = \begin{cases} One\ if\ v < 0 \\ 0 \quad else \end{cases}$$

Step.5 *Divide coded images as m × n areas.*

Step.6 Calculate *positive* $H_{P_{GLTP}}$ *and negative* $H_{N_{GLTP}}$ *GLTP histogram for each region.*

$$H_{P_{GLTP}}(\tau) = \sum_{r=1}^{\frac{M}{n}} \sum_{c=1}^{\frac{N}{n}} f(P_{GLTP}(r,c), \tau)$$

$$H_{N_{GLTP}}(\tau) = \sum_{r=1}^{\frac{M}{n}} \sum_{c=1}^{\frac{N}{n}} f(N_{GLTP}(r,c), \tau)$$

$$f(\alpha, \tau) = \begin{cases} One\ if\ \alpha = \tau \\ 0 \quad else \end{cases}$$

Step.7 *Adding positive* $(H_{P_{GLTP}})$*and negative* $(H_{N_{GLTP}})$*GLTP histograms from each region to obtain feature vector.*

3.4. *CNN Classifier*

CNN is a Deep Neural Networks version. NN uses convolutions to extract essential data or patterns from the input characteristics, subsequently used to build the sub-layers of neural network computations. An input layer, four convolutional layers, five rectified linear units (ReLu), two stochastic pooling layers, a single dense layer, and a SoftMax output layer make up the system.

The CNN network has four convolutional layers in its model. The first two layers simultaneously eliminate the lower-level elements such as lines, corners, and edges. The final two levels look into the more complex characteristics. The following equation is commonly used to express the outcome of a convolutional layer:

$$y^n{}_j = f(\sum_{i \in c_j} y_i{}^{n-1} * k^n{}_{ij} + \zeta_j{}^n \tag{8}$$

Where n denotes the n^{th} layer, kij is the convolutional kernel, ζj denotes bias, and the input maps are denoted by cj. The CNN utilizes a tanh activation function by an additive bias formulated as

$$h_{ni}{}^{xy} = tanh(\zeta_{mi} + \sum_{w=0}^{w_i-1} \sum_{h=0}^{h_j-1} w_{ij}{}^{wh}\, h^{(x+w)(y+h)}{}_{i-1}) \tag{9}$$

ζni Describes the feature map bias that is non-supervisory tunned and wi, hj are the kernel width and height, respectively. $W\ wh$ denotes the kernel's weight at point (w, h). with an area, the maximum value of a feature is taken by employing the pooling method, which decreases the information variance. We coded our model with a stochastic pooling approach by computing the probabilities of each area. For every feature map c, the probability is expressed as:

$$\chi_{w,h}^{n,k} = Stochastic_{(w,h,i,j) \epsilon p} \left(\chi_{w,h}^{n-1,k} u(i,j) \right) \tag{10}$$

where $\chi n, k_{,h}$ is the neuron activation function at a point (w, h) in spatial coordinates, and $u(i, j)$ is the weighing functionof the window. While comparing with other pooling approaches, stochastic pooling builds CNN converge at a higher rate andincreases the ability of generalization in processing invariantfeatures.

4. Experimental Results

This section discusses detailed results of the proposed method. The implementation of the proposed methodology has been carried out as an extension of our previous paper on Plant leaf disease detection.

4.1. *Database Description*

We have created a Plant leaf dataset for training and testing purposes, which consists of 735 numbers of healthy- and infected-plant leaves. The database contains seven different tomato leaf diseases classes: healthy leaf images, Bacterial Spot, Mosaic Virus, Yellow leaf curl virus, early blight, late blight, and leaf mold. All images were captured in laboratory conditions. All leaf images were divided into a training set and a testing set. We split-leaf images into two sets to evaluate performance, namely 80–20 (80% training images and 20% testing images).

4.2. *Feature Extraction*

The segmented region is used to extract features of the diseased region of the leaf. This paper uses Eleven different Zernike Moments and Gradient Local Ternary Pattern (GLTP) methods. Fig. 8(a) & (b) shows the standard deviation derived from the features from all the seven classes of the database

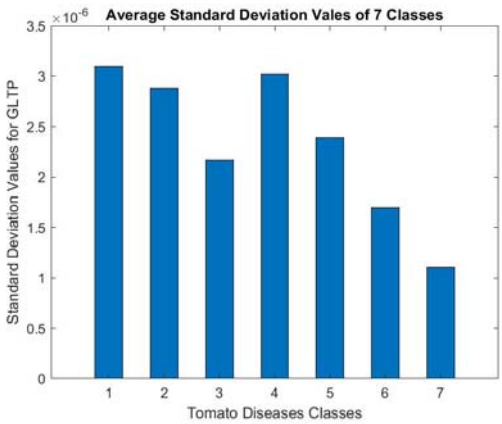


Figure 8(a): Standard Deviation Values for GLTP Feature.

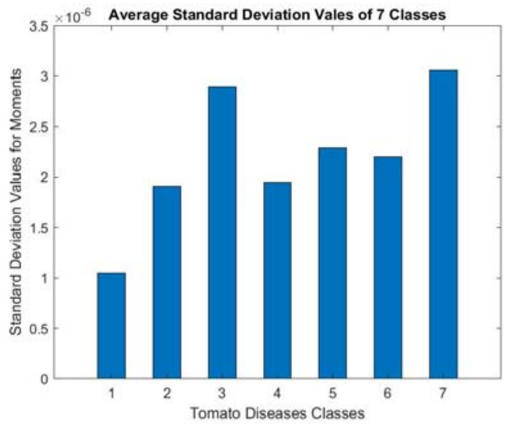


Figure 8(b): Standard Deviation Values for Zernike Moments Feature.

4.3. *Classifier - CNN*

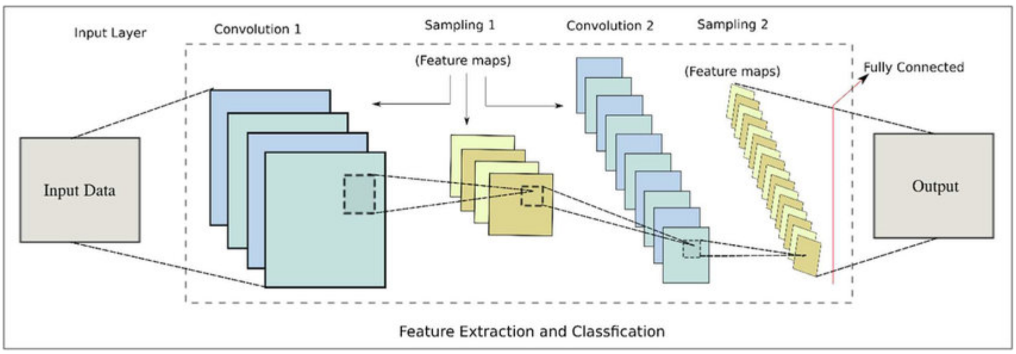


Figure 9: Architecture of CNN

The CNN model, which comprises convolutional activation, pooling, and fully connected layers, was utilized to classify the tomato leaf diseases. Figure 9 depicts the model utilized in this work. Each block comprises three layers: convolutional, activation, and max-pooling. This design employs three blocks, followed by fully linked layers and softmax activation. For feature extraction, convolutional and pooling layers are utilized, whereas fully connected layers are used for classification. Non-linearity is introduced into the network via activation layers. The architecture is presented in depth in Table 1:

Layer	**Type**	**Output Shape**	**Parameter**
Imageinput	Imageinput	256x256x3	
Conv_1	Convolution	256x256x8	Weights 3x3x8 Bias 1x1x8
Batchnorm_1	Batch Normalization	256x256x8	Offset 1x1x8 Scale 1x1x8
Relu_1	ReLu	256x256x8	-
Maxpool_1	Max pooling	128x128x8	-
Conv_2	Convolution	128x128x16	Weights 3x3x8x16 Bias 1x1x16
Batchnorm_2	Batch Normalization	128x128x16	Offset 1x1x16 Scale 1x1x16
Relu_2	ReLu	128x128x16	-
Maxpool_2	Max pooling	64x64x16	-
Conv_3	Convolution	64x64x32	Weights 3x3x16x32 Bias 1x1x32
Batchnorm_3	Batch Normalization	64x64x32	Offset 1x1x32 Scale 1x1x32
Relu_3	ReLu	64x64x32	-
dropout	Dropout	64x64x32	-
FC	Fully Connected	1x1x7	Weight 7x131072 Bias 7x1
softmax	softmax	1x1x7	-
class output	Classification output	-	-

Table 1: Architecture Details of CNN

The training was carried out across twenty epochs, with 7x131072 parameters. The first layer filters detect colors, edges, and primary forms (Layer Conv 3 and batch norm 3). The filters grow on top of one other as we progress further into the network, learning to encode more complicated patterns. The visualization of characteristics is shown in Figure 10.

e-ISSN : 0976-5166
p-ISSN : 2231-3850

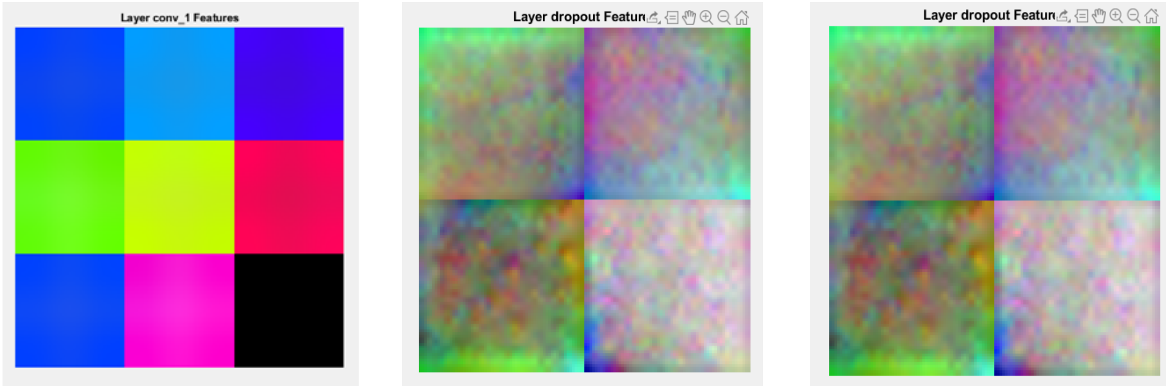


Figure 10: Feature map of inner layers of CNN

A set of quantitative criteria, including accuracy, precision, sensitivity, and specificity, were employed to assess the suggested model's performance. Table 2 summarises the findings. They display the most significant quantitative indicators' values until the epoch number is reached.

Accuracy	95.494558
Sensitivity	95.494558
Specificity	95.5000
Precision	95.5000

Table 2. Results and Analysis

In Fig. 11, the plots of train and test accuracy and loss versus the epochs give a visual representation and indicator of model convergence pace. With a training duration of 12 minutes 51 seconds, a validation accuracy of 100 percent is achieved.

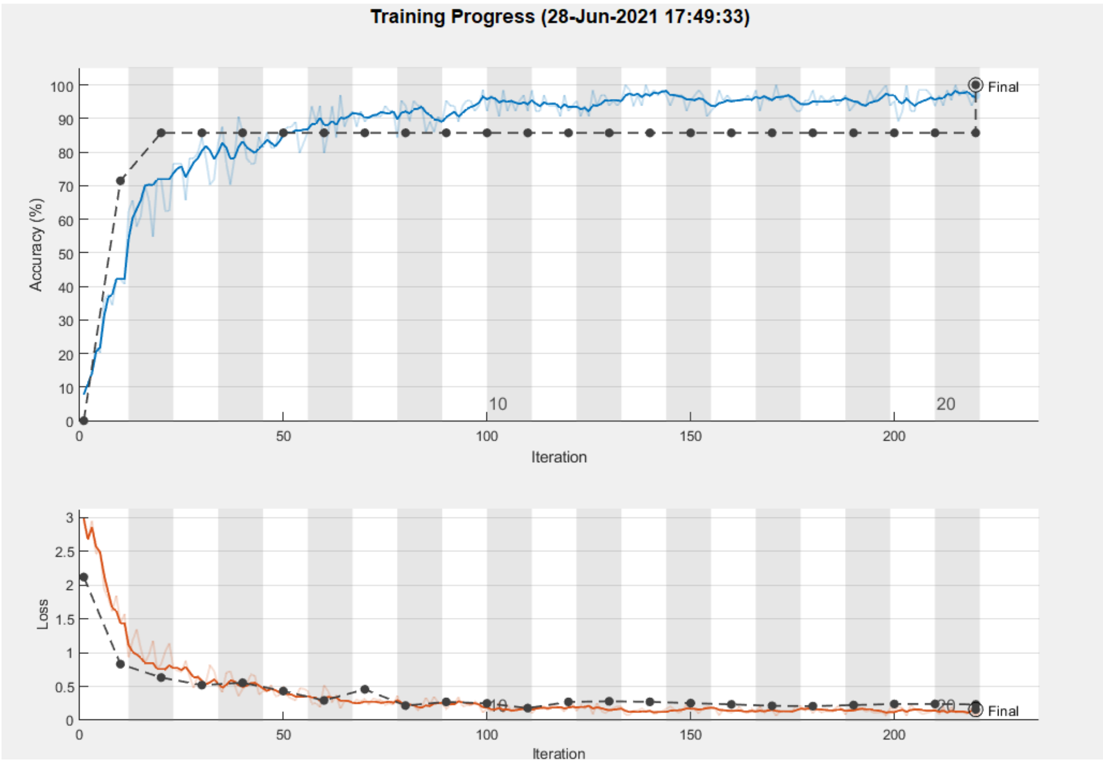


Figure 11: CNN training run (with a Matlab) with altered learning rates and layer filter sizes.

The confusion matrix, also known as the error matrix, is a visual depiction of a machine learning or statistical classification algorithm's efficacy. The rows reflect expected ground facts, whereas the columns represent actual ground truths. The identification rate of the CNN Classifier is graphed in Fig. 12, and the confusion matrix for my model for the 7classes is shown in Fig. 13.

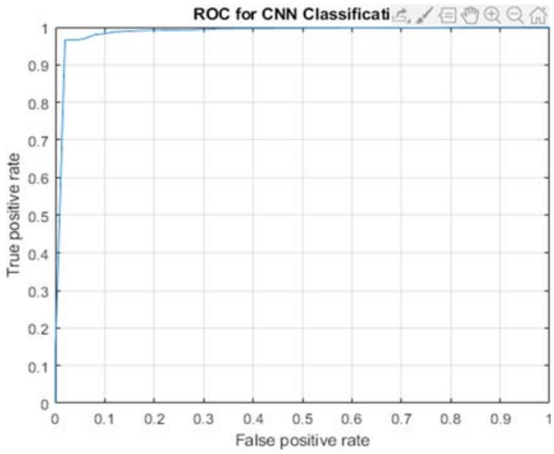


Figure 12: ROC for CNN Classification

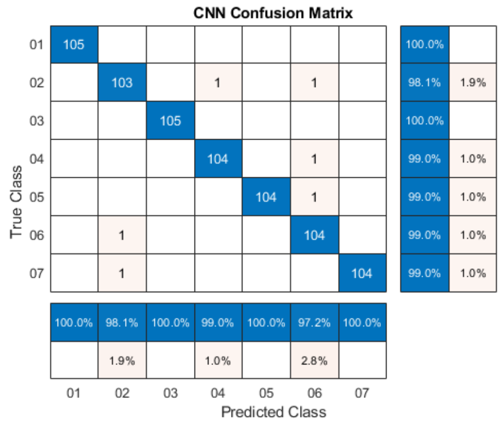


Figure 13: Confusion Matrix

5. Conclusion

Detection of disease is the main aim of the presented method to identify leaf diseases with fewer calculation efforts. This method can be utilized as an application in agriculture for detecting & classifying diseases present in the leaf using CNN classifier. This research work addresses the way of analyzing the diseases utilizing various plant leaf diseases can be performed for identification in their initial stages before it spreads and damages the entire plant. The efficiency of the proposed work is about 95.4929%.

References

[1] Vivek Ugale and Deepak Gupta, “A comprehensive survey on agricultural image processing,” International Journal of Science and Research (IJSR), Vol.5, Issue.1, pp-133-135, 2016.
[2] Ms. Kiran R. Gavhale and Prof. Ujwalla Gawande, “An overview of the plant research leave disease detection using image processing techniques,” IOSR Journal of Computer Engineering (IOSR-JCE), Vol.16, Issue.1, pp-10-16, 2014.
[3] Priyanka Thakur, Preeti Aggarwal, and Mamta Juneja, “Plant disease detection and classification using image processing: A review,” International Journal of Recent Research Aspects, Vol.4, Issue.3, pp-22-27, 2017.
[4] M. Akila, P. Deepan, “Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm,” International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, 2018.
[5] Mr. Sanjay Mirchandani, Mihir Pendse, Prathamesh Rane, Ashwini Vedula, “Plant Disease Detection And Classification Using Image Processing And Artificial Neural Networks,” International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 06, June 2018.
[6] Mrs. Gaganpreet Kaur, Sarvjeet Kaur, Amandeep Kaur, “Plant Disease Detection: a Review of Current Trends,” International Journal of Engineering & Technology, 7 (3.34) (2018) 874-881
[7] M. Ravindra Naik1, Chandra Mohan Reddy Sivappagari, “Plant Leaf and Disease Detection by Using HSV Features and SVM Classifier,” International Journal of Engineering Science and Computing, Volume 6 Issue No. 12 December 2016
[8] Shantanu Kumbhar, Amita Nilawar, Shruti Patil, Bodireddy Mahalakshmi, “Farmer Buddy-Web Based Cotton Leaf Disease Detection Using CNN,” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 11 (2019) pp. 2662-2666
[9] N. Kanaka Durga, G. Anuradha, “Plant Disease Identification Using SVM and ANN Algorithms” International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-5S4, February 2019.
[10] Peng Jiang, Yuehan Chen, Bin Liu, Dongjian He, and Chunquan Liang, “Real-Time Detection of Tomato Leaf Diseases using Deep Learning Approach Based On Improved Convolutional Neural Networks,” IEEE Access, Vol. 7, 2019, pp. 59069-59080.
[11] Tanmoy Bera, Ankur Das, Jaya Sil, and Asit K. Das, “A Survey on Rice Plant Disease Identification using Image Processing and Data Mining Techniques,” In Emerging Technologies in Data Mining and Information Security, Springer, Singapore, 2019, pp. 365-376.
[12] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu, “Image Segmentation using K -means Clustering Algorithm and Subtractive Clustering Algorithm,” Procedia Computer Science, Elsevier, Vol. 54, 2015.
[13] Alaa Al-Hamami and Hisham Al-Rashdan, “Improving the Effectiveness of the Color Coherence Vector,” The International Arab Journal of Information Technology, Vol. 7, No. 3, July 2010, pp. 324-332.
[14] Qiang Deng, YupinLuo, JunfengGe. "Dual Threshold-Based Unsupervised Face Image Clustering," International Conference on Industrial Mechatronics and Automation, 2010.

e-ISSN : 0976-5166
p-ISSN : 2231-3850

Authors Profile

Nagamani H. S Received an MS in the Computer System from the University of Mysore. M.Phil in Computer Science and Engineering from Annamalai University, TN. Presently She is working as an Assistant Professor in the Department of Computer Science, Smt VHD Central Institute of Home Science (Maharani Cluster University), Bangalore. Her Academic experience is 18 years. She has published her research work in IEEE conferences and International Journals. She has attended many seminars and workshops.

Dr. Sarojadevi H. is a professor in the Computer Science & Engineering department at Nitte Meenakshi Institute of Technology (NMIT), Bangalore. She is a Ph.D. graduate from the Indian Institute of Science, Bangalore. She has received B.E. and M.E. qualifications from the Visvesvaraya College of Engineering, Bangalore. She has published more than 100 papers, mainly in SCOPUS indexed journals & conferences. She has more than 23 years of research, industry, and academics. She has guided six research scholars for Ph.D. in Visvesvaraya Technological University, Belagavi. She has won several national and international recognitions, specifically Marquis Who in the World 2016 & CSI TechNext India best faculty award 2018.