A Novel Framework For Automatic Detection Of Plant Leaf Disease Using 2d-Deep Convolutional Neural Network Architecture

D. Poornima[1], G. Arul Selvi[2], S. Praylashyry[3], B. Kirubagari[4]

[1]Research scholar, Department of Computer Science and Engineering, Annamalai University, Chidambaram,
[2]Associate Professor, Department of Computer Science and Engineering, Annamalai University, Chidambaram.
[3]Associate Professor, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology
[4]Associate Professor, Department of Computer Science and Engineering, Annamalai University
[1]poorniramesh2011@gmail.com, [2]arulselvidhanasekaran@gmail.com,[3] praylashyry@gmail.com,
[4]kirubacdm@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: The economy of a country largely depends on its agricultural productivity. Hence, identification of plant leaf diseases plays a vital role in the field of agriculture. In this research, we propose a novel framework for the automatic detection of plant leaf disease based on Deep Convolutional Neural Network (DCNN) architecture. The proposed framework involves steps like image restoration, enhancement, clustering, thresholding and classification. The image restoration is performed using a novel filter called 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter (2D–AHBAD). This filter is used for the elimination of various noise such as salt and pepper noise, Gaussian noise, random noise, thermal noise, speckle noise etc. Image enhancement is done using Edge Preservation–Modified Histogram Contrast Brightness Equalization (EP-MHCBE) algorithm. The enhanced images are then segmented using clustering and thresholding algorithms. A new technique called Hybrid Fast Fuzzy C Means Improved Expectation Maximization (HFFCM IEM) Clustering technique was used for the computation of clusters. The generated clusters are then segmented based on the Iterative Mean Shift Thresholding (IMST) algorithm. The segmented images are classified using DCNN architecture. A total of 2000 images are used in this framework out of which 1600 images were utilized for training the DCNN architecture. The remaining 400 images were used for testing. The leaf images are categorized into four categories namely, normal, mild, moderate and severe. It was inferred that the proposed AHBAD image restoration algorithm achieved a high PSNR of 54 and very low MSE of 0.0039. Similarly, the proposed DCNN classification system attained a high classification accuracy of 92.25%.

Keywords: Plant leaf disease, DCNN, classification, restoration, enhancement, segmentation.

1. Introduction

Plant leaf disease detectionplays an important role in the enhancement of agricultural sector. Detection of plant disease at earlier stages helps to avoid the spreading of the disease. Automatic detection schemes are widely being employed for the detection of such leaf diseases. Computer vision techniques uses various algorithms for the identification of leaves having diseases [1].Plants are susceptible to a variety of diseases at their young stages. These diseases spread all over the plants as they grow and make the plants unsuitable for human consumption. This leads to tremendous loss in the crop production. Other parameters like climatic conditions also influence the spreading of plant diseases [2].

The plant diseases are categorized into two broad categories namely, the infectious diseases and the non-infectious diseases. The infectious diseases are a major thread for crop production as they spread rapidly. Further, this type of diseases are spread using microorganisms like bacteria, fungus and virus[3]. Various algorithms are employed for the identification of plant leaf diseases. These include machine learning classification algorithms like support vector machine, deep learning, sparse classification etc. These automatic disease detection techniques helps in decreasing the number of diseased leaves, there by increases the profit of the farmers [4].

The leaves identified as diseased leaves are removed either using manual hand-picking methods or using automatic robotic systems.Various automatic systems are now available based on embedded systems for the automatic removal and disposal of diseased leaves [5]. Machine learning techniques are fast and accurate. These techniques require extensive training samples from both the diseased and healthy leaf categories. These samples are trained to form training models. These models are employed for the classification and identification of diseased leaves. Further, the Neural Network based classifiers are found to produce good results around 95% of accuracy [6].

Recently, deep learning techniques are popularly employed in this field. These deep learning architectures comprises of models like AlexNet, ResNetetc that are used for classification. The parameters for these models are selected based on the input training data. These models are designed such that the complete cycle of the diseases are detected [7]. Popular type of plant diseases includes anthracnose, bacterial blight, leaf spot, etc. These diseasesare commonly classified based on the texture features. The texture details of the leaf images are collected and the features are extracted. These features include statistical and other higher order features [8].

Feature extraction based on k-means clustering algorithms also provide good level of accuracy. The clusters that contain the diseased regions are separated. From these regions, features are extracted and used in the classification stage[9]. Artificial neural network and fuzzy-based systems are other type of classifiers employed in plant leaf disease identification. Fuzzy-based classifiers makes use of fuzzy predictions for predicting the leaves that contain diseases regions[10].

The overall contributions of this paper are twofold:

a) A novel deep convolutional neural network architecture for the detection of plant leaf diseases is proposed.

b) Image filtering using a new 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter is proposed.

c) Image enhancement using Edge Preservation–Modified Histogram Contrast Brightness Equalization is presented.

d) Image classification using Hybrid Fast Fuzzy C Means Improved Expectation Maximization Clustering technique with Iterative Mean Shift Thresholding is proposed.

e) Classification of diseases using DCNN.

2. Related Works

Pantazi et al. [11]presented a method for the automatic disease detection of plant leaves based on the One class classifier technique. Here, vine leaves were employed for the disease detection. Feature extraction was done using Local Binary Pattern classifier. This scheme achieved high accuracy of around 95%. This scheme shows very good generalization behavior.Sandhu et al. [12] did a review on various plant disease detection methodologies. Different phases involved in the disease detection were evaluated and analyzed. Accuracy was compared for various machine learning algorithms. Further, the computation time analysis was also performed for different algorithms. Different features that can be used for representing the leaf images was also discussed.

Annabel et al. [13] performed a survey on various machine learning techniques that can be used for the detection of plant diseases. Three main properties of plant leaves were considered for feature generation. The first property was the color of the leaves. The second and third were the intensity and the dimension of the leaves. Various machine learning classifiers were compared and discussed.Kumar et al. [14] did a review on various image processing techniques employed for disease detection. Different kinds of plant leaf diseases, their origin, spreading rate and color of the disease were discussed. Further, the importance of automation in the field of disease detection was presented along with the challenges and drawbacks of these automation techniques.

Pandian et al. [15]presented a study on data augmentation for plant disease classification. This article developed datasets for the plant leaf diseases based on data augmentation techniques. Techniques such as flipping, cropping, transformation, etc., were discussed and compared. Further, the usage of deep learning techniques for the identification of diseases were also discussed.Tulshan et al. [16] proposed a scheme based on machine learning for the identification of leaf diseases. The classifier employed in this framework was k-nearest neighbor algorithm. It was found that the average area of affected portion in a leaf was around 18.5%. This scheme achieved accuracy of 98.56% that was higher compared to that of support vector machine classifier.

Geetharamani et al. [17]proposed a 9-layer neural network architecture for the classification of plant diseases. 39 different varieties of plant diseases were considered for classification. Data augmentation was done using 6 different methodologies. The proposed framework was evaluated in terms of reliability and consistency. This scheme has an accuracy of 96.46%. Militante et al. [18]discussed the usage of deep learning in leaf disease identification. A dataset comprising of 35,000 leaf images were employed for classification. Various plant varieties like apple, potato, tomato, etc., were considered for the study. The proposed scheme was designed to detect various types of plant diseases. The system had a training accuracy of 96.5%.

Kaur et al. [19]did a survey on the leaf disease identification techniques. The various advantages and disadvantages of techniques employed for disease classification was discussed. Various types of bacterial and viral diseases that affects the plants were analyzed. Various segmentation, feature extraction and classification schemes were also discussed.Ahmed et al. [20] proposed a scheme for the detection of diseases in rice leaves. Three types of diseases were considered in this study for classification. The use of machine learning for the disease detection was analyzed. The input images considered in this study used white background. The proposed technique achieved accuracy of 97%.

3. Proposed Methodology

The proposed methodology comprises of 5 main steps. This is shown in Figure 1. The first step is the input leaf image acquisition. The next step is filtering using 2D-ADBAD filter. The filtered image is enhanced using EP-MHCBE algorithm. Then, the image is segmented using HFFCM IEM-IMST segmentation technique. Finally, the image is classified into 4 categories namely, mild, moderate, severe and normal using DCNN architecture.

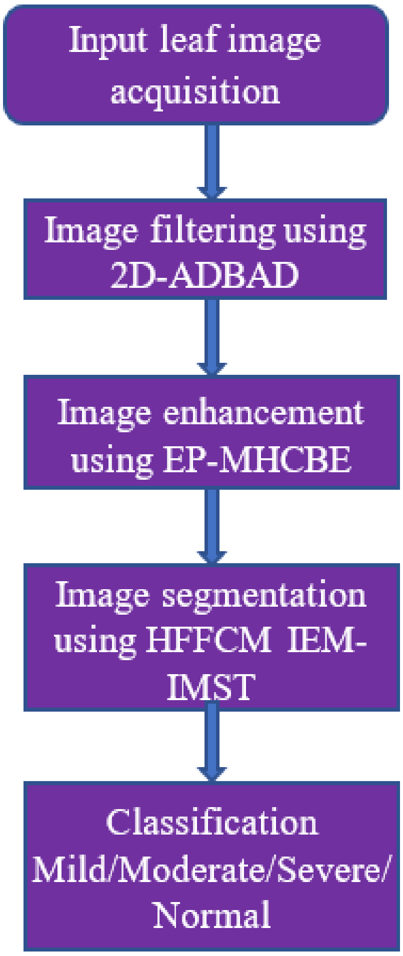


Figure 1. Flow chart of the proposed leaf disease detection methodology

3.1. Image Preprocessing

Image preprocessing is performed using two steps namely, filtering and enhancement. The image filtering is used to filter various types of noises such as salt and pepper noise, speckle noise and Gaussian noise. The filtered output image is further undergone to image enhancement. The image enhancement technique is used to improve the quality of image in terms of contrast and brightness.

3.1.1. Image Filtering using 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter (2D – A
3.1.2. HBAD)

The block diagram of the proposed 2D-AHBAD filter is shown in Figure 2. The first step is the acquisition of the input image. The acquired image is then assigned spacing between the X and the Y coordinates. The image is then convoluted using 2D convolutional masking. The bilateral filtering is then applied using convolution. Using the convolved matrix, the reciprocity theorem is then applied. Then, the partial differential equation is applied using anisotropic diffusion. Finally, unsigned 8 image format conversion is applied to obtain the filtered image.

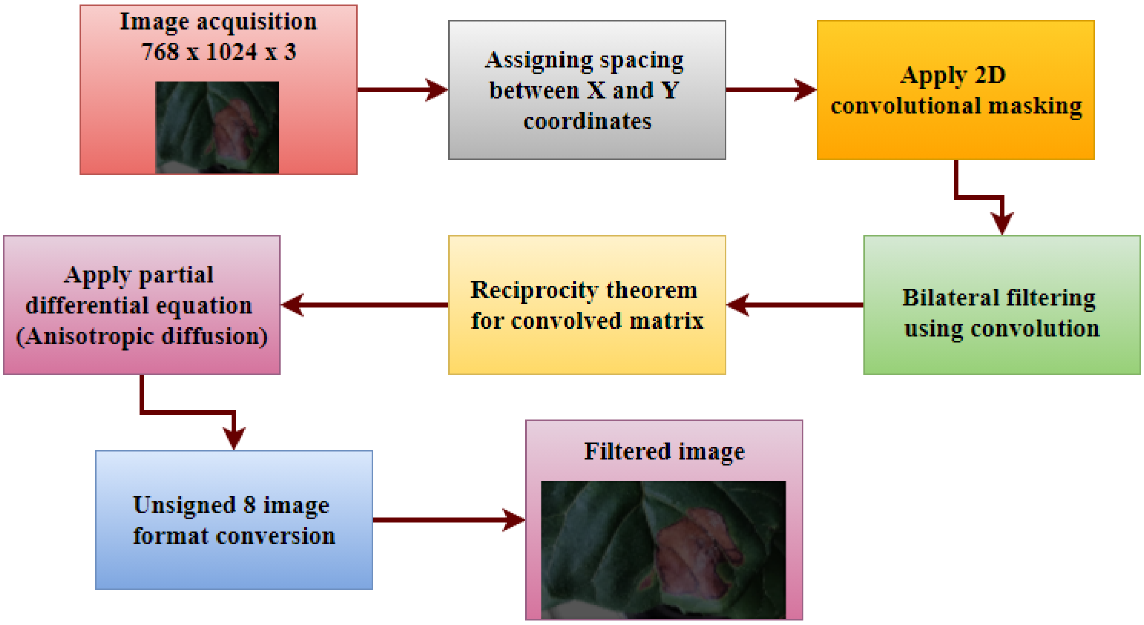


Figure 2. Block diagram of proposed 2D-AHBAD image filtering technique

Algorithm1: Proposed 2D – AHBAD algorithm.
Input:
Input image $II \in R^{M \times N}$.
Output:
Filtered image $FI \in R^{M \times N}$.
Algorithmic steps:
Step 1: Acquire input image of size ($M \times N \times 3$).
Step 2: Initialize the number of Bilateral iterations and diffusion parameters.
Step 3: Convert Uint8 image format into double precision format.
Step 4: Adjust the spacing between X and Y directional coordinates.
Step 5: Generate 8, 3x3 Matrix 2D convolution masks using 2D Impulse response coefficients.
Step 6: Perform random noise filtering using anisotropic diffusion principle. The weights are computed as

$$w_n = \exp(-K / II(x, y) - II(x+i), y+j) \quad (1)$$

Step 7: Apply reciprocity theorem for performing elimination of salt and pepper noise using

$$Diffusion_n = 1/(1+(Filter_n / w_n).\wedge 2) \quad (2)$$

Step 8: Restore the image using the following Partial Differential Equation (PDE)

$$I(x, y) = 0.1429 * \sum_{n=1}^{8}(((1/(Y\_dist^2)) * Diffusion_n .* Filter_n)^{1/2} \quad (3)$$

Step 8: Calculate PSNR & MSE for input and output image
Step 9: Display the calculated values and image.

3.1.3. Image Enhancement using Proposed Edge Preservation–Modified Histogram Contrast Brightness Equalization (EP-MHCBE) algorithm

The filtered images are enhanced using proposed EP-MHCBE algorithm as shown in Figure 3. In this technique, the filtered images are initially subjected to intensity thresholding. Then the image matrix in RBB format is converted to NTSC format. The bandwidth threshold is initiated to the upper and lower limits. The mean value is then estimated based on the bandwidth and intensity. The mean adjust is applied followed by normalization to improve the contrast and brightness of the image.

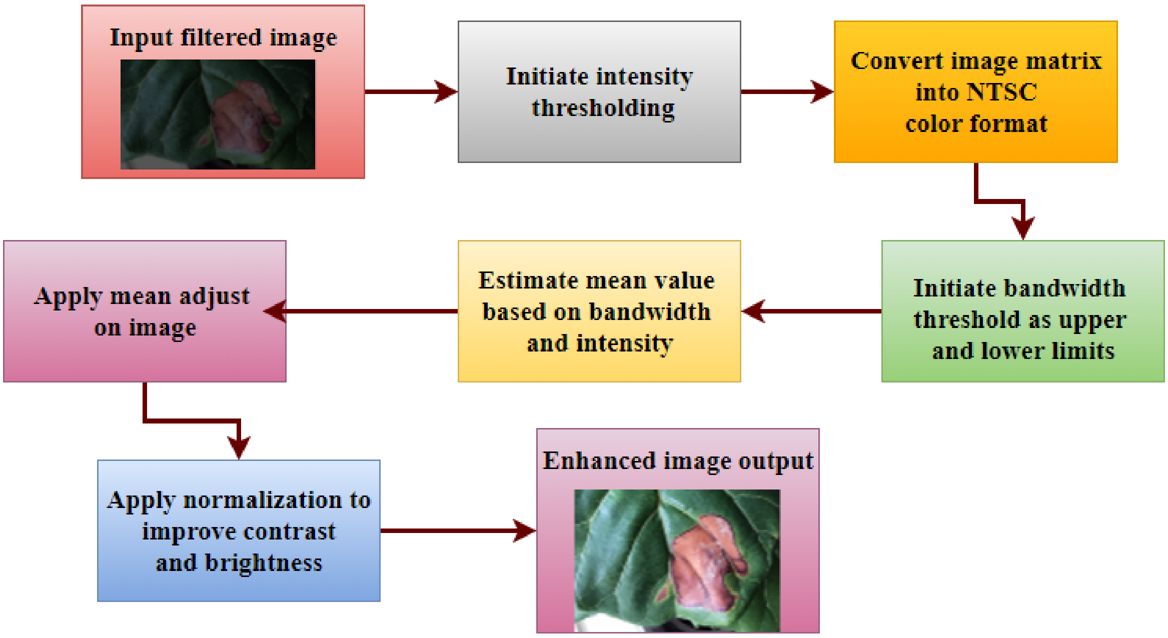


Figure 3. Block diagram of proposed EP-MHCBE image enhancement technique

Algorithm 2: Proposed EP-MHCBE algorithm.
Input:
Filtered image $FI \in R^{M \times N}$.
Output:
Enhanced image $EI \in R^{M \times N}$.
Algorithmic steps:
Step 1: Acquire the filtered Image
Step 2: Set the lower and upper threshold values to calculate the Minima and Maxima Intensity Thresholding.
Medium thresholding threshold τ_m =0.5;
Lower thresholding threshold τ_l =0.008;
Upper thresholding threshold τ_u =0.992;
Step 3: Set the colour Image Threshold (Image Bandwidth) as
Color image upper threshold C_u =0.04;
Color image lower threshold C_l =-0.04;
Step 4: Convert from RGB to NTSC Colour format to obtain $NTSC_g$, $NTSC_b$ and $NTSC_r$ components.
Step 5: Calculate the Mean adjust value for Green layer using color image upper threshold C_u as

$$M_g = C_u - avg(0.596 - NTSC_g) \tag{4}$$

Step 6: Calculate the Mean adjust value for the Blue layer color image lower threshold C_l.

$$M_b = C_l - avg(0.523 - NTSC_b) \tag{5}$$

Step 7: Mean Adjustment for Red Layer using both the thresholds.

$$M_r = (C_u - C_l) - avg(0.59 - NTSC_r) \tag{6}$$

Step 8: Calculate Minima and Maxima

$$Mn(x, y) = FI(x, y) < \tau_u \tag{7}$$

$$Mx(x, y) = FI(x, y) > \tau_l \tag{8}$$

Step 9: Compute the enhanced image using

$$EI(x,y)=\frac{FI(x,y)-Mn(x,y)}{Mx(x,y)-Mn(x,y)} \tag{9}$$

3.2. Image Segmentation using proposed HFFCM IEM Clustering-IMST Thresholdimage segmentation technique

The enhanced images are then segmented using clustering and thresholding algorithms. A new technique called Hybrid Fast Fuzzy C Means Improved Expectation Maximization (HFFCM IEM) Clustering technique was used for the computation of clusters. The clustered images are subject to non-homogeneity pixels grouping. The generated clusters are then segmented based on the Iterative Mean Shift Thresholding (IMST) algorithm as shown in Figure 4. Finally, classification is done using CNN architecture in which the images are classified into 4 categories.

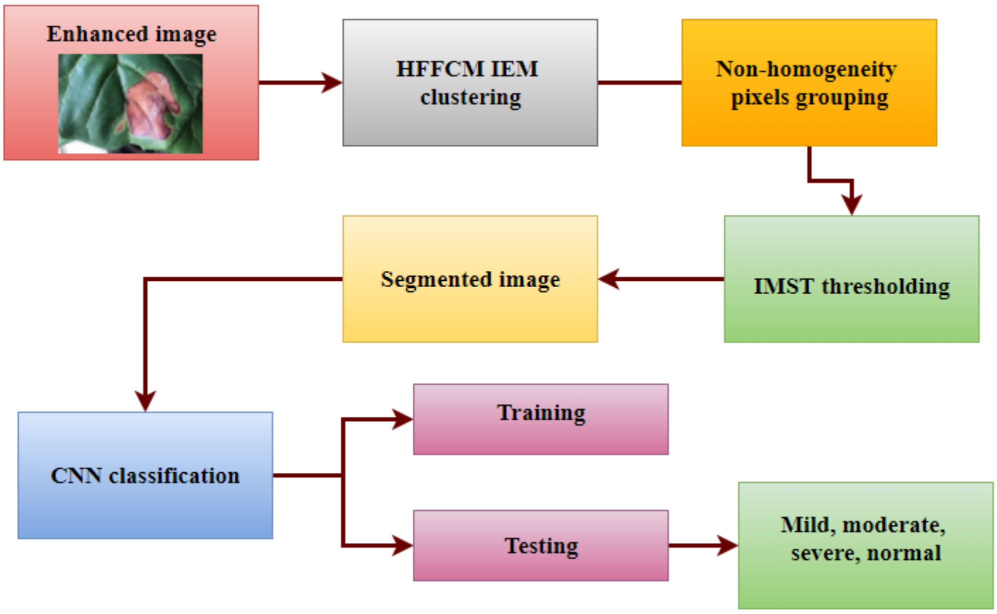


Figure 4. Block diagram of proposed HFFCM IEM Clustering-IMST Thresholdimage segmentation technique

Algorithm 3: Proposed HFFCM IEM Clustering-IMST Threshold
Input:
Enhanced image $EI \in R^{M\times N}$.
Output:
Segmented image SI.
Algorithmic steps:
Step 1: Initialize the number of Clusters (3 Clusters-0,1,2). Convert $M\times N$ matrix into $MN\times 1$ matrix to simplify the complexity.
Step 2: Estimate various intensity pixels. Find the location of the intensity pixels. Further, the length of the Intensity Pixels is estimated.
Step 3: Calculate the center point of intensity pixels using

$$C_{ent}=(1/C)*Hist/(C+1) \tag{10}$$

Where $Hist$ represents the histogram of the image.
Step 4: Calculate the absolute centroid position between central point and the location of the intensity pixels
Step 5: Find the location of classified pixels. Calculate the mean of the classified pixels. Apply the mean value as threshold ∂ for IMST.
Step 6: Based on the thresholding value ∂, Region of Interest (ROI) is computed as the affected pixels. The ROI pixels are restored and other pixels are Non-ROI. The Non-ROI Pixels are then suppressed.

3.3. Classification

The segmented images are classified using the proposed deep convolutional neural network architecture (DCNN) as shown in Figure 5. Three convolutional layers, three pooling layers and two fully-connected layers are employed in this architecture. The input image is converted to feature maps using convolutional layers (1, 2, 3). Pooling layer (1, 2, 3) are using for the generation of pooled feature maps. Finally, the classification is done using the fully-connected layers. The Output has classified into four classes.

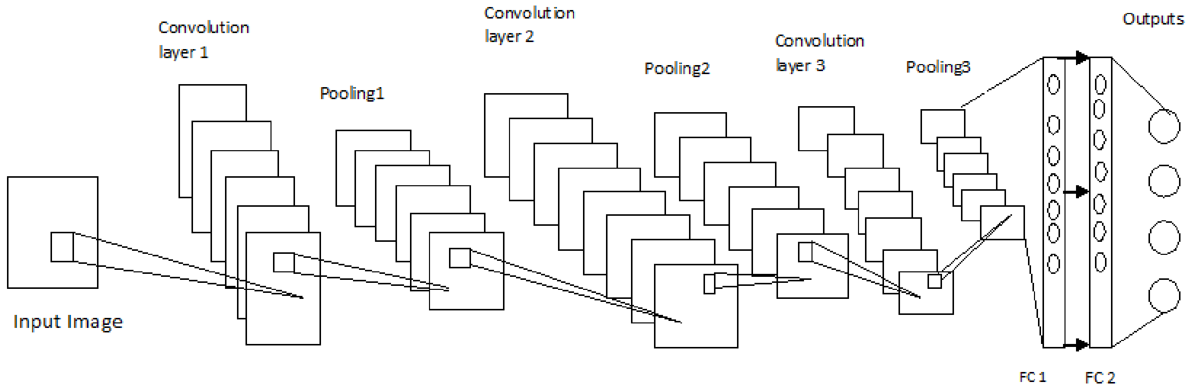


Figure 5. Architecture of proposed Deep Convolutional Neural Network

Figure 6 shows the classification scheme. The segmented images are used for training the neural network. A total of 2000 images are used in this framework out of which 1600 images were utilized for training the DCNN architecture. The remaining 400 images were used for testing. The leaf images are categorized into four categories namely, normal, mild, moderate and severe.

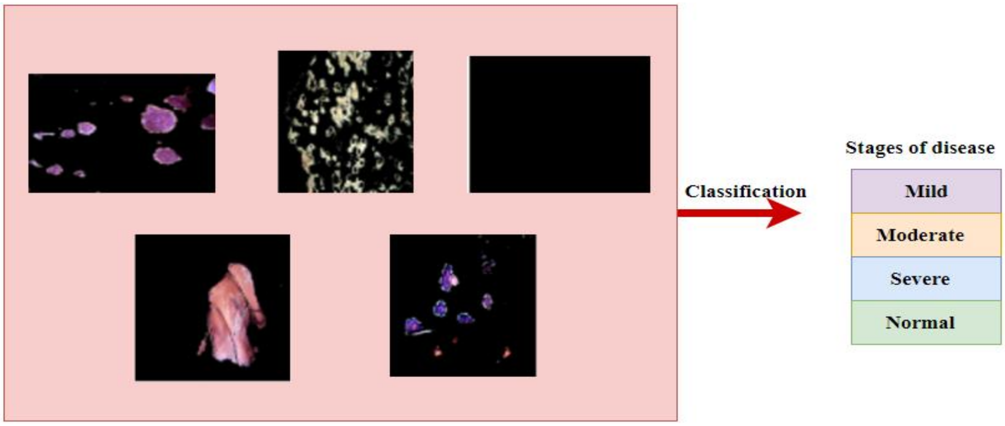


Figure 6. Proposed classification scheme

4. Results and Discussion

4.1. Parameter Settings

The input parameters using the proposed framework is shown in Table 1. The size of the input images considered are 768×1024. They were in RGB format. The total file size was 2359296 bytes. The total memory size was around 9Kb.

Table 1. Input parameters

S. No	Parameters	Description
1	Image width (rows)	768
2	Image height (columns)	1024
3	No. of layers	3 (r g b)
4	File size	2359296 bytes
5	No. of pixels	768×1024=786432

6	Class	Uint8
7	Minima, maxima	[0, 0, 0], [255, 255, 255]
8	Memory size	9Kb
9	Total number of images	2000

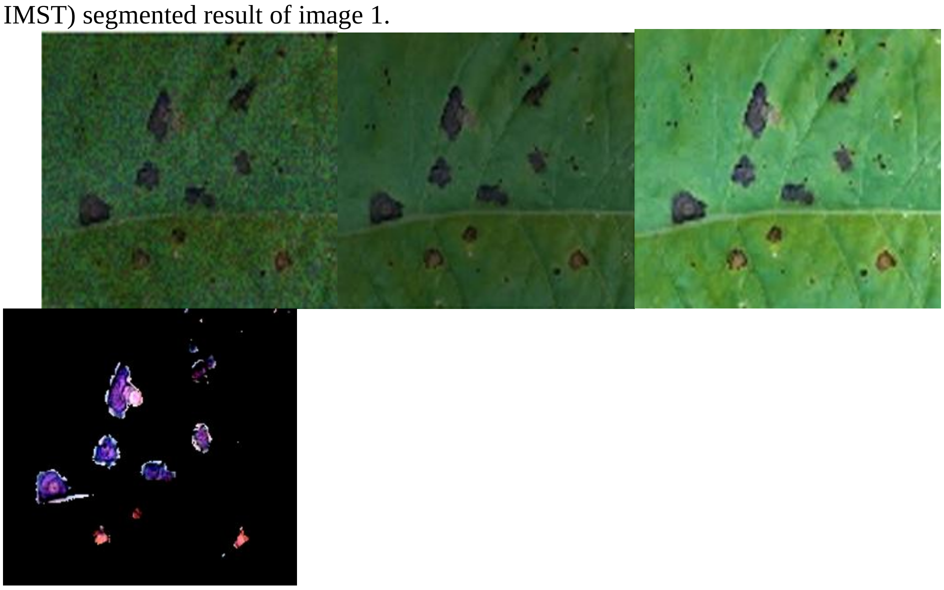
Table 2 shows the parameters of the proposed DCNN architecture. The number of filters used in the first convolutional layer was 24, second convolutional layer was 48 and third convolutional layer was again 48. The activation was done using ReLU activation layer. Classification was done using Softmax layer.

Table 2. Parameters of DCNN architecture

Layer	Number of filters	Activation	Padding
Input data	-		
Conv2D (k=5x5, s=1)	24	ReLU	-
Maxpool2D (4×2)	-		
Conv2D (k=5x5, s=1)	48	ReLU	Same
Maxpool2D (4×2)	-		
Conv2D(k=5x5, s=1)	48	ReLU	same
Maxpool2D (4×2)	-		
Flatten(2)	-		
Softmax (4)	-		

4.2. Simulation results

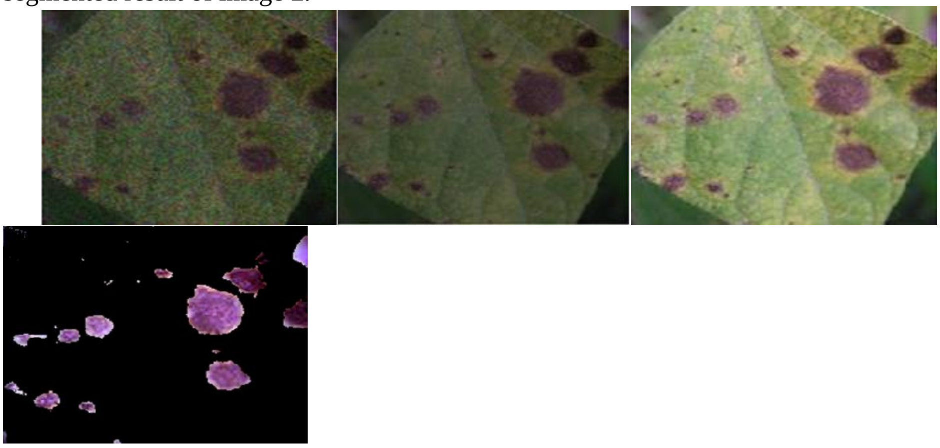
The simulation results obtained using the proposed algorithm is shown in Figure 7 to Figure 11 for 5 different input images. From Figure 7 we see that, each image comprises of 4 sub-images. The first sub-image is the input image 1. The second sub-image is the 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter (2D-AHBAD) filtered output of image 1. The third sub-image is the Edge Preservation–Modified Histogram Contrast Brightness Equalization (EP-MHCBE) enhanced output of image 1 and final sub-image is the Hybrid Fast Fuzzy C Means Improved Expectation Maximization Clustering technique with Iterative Mean Shift Thresholding (HFFCM IEM-IMST) segmented result of image 1.



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

Figure 7. (a) Input image 1 (b) 2D-AHBAD filtered output(c) EP-MHCBE enhanced output (d) HFFCM IEM-IMST segmented result

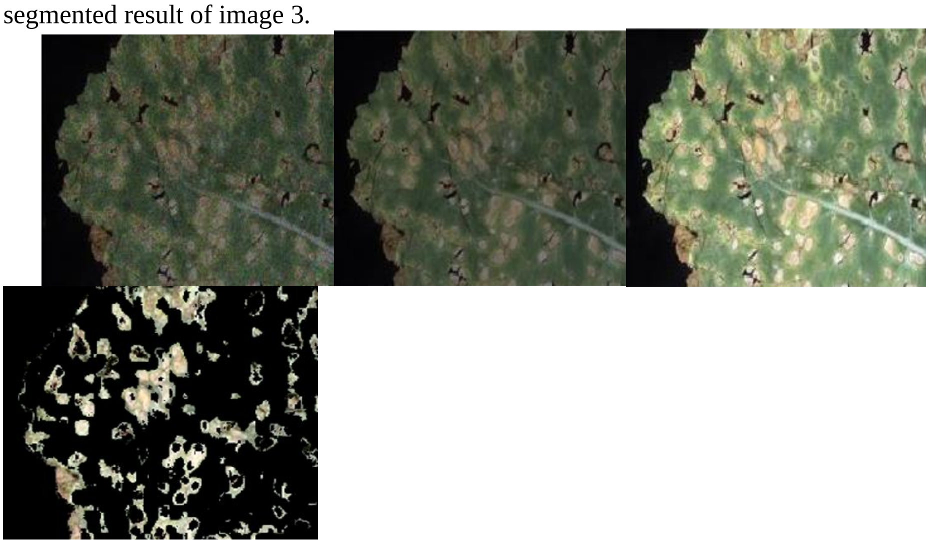
From Figure 8 we see that, there are 4 sub images. The first sub-image is the input image 2. The second sub-image is the 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter (2D-AHBAD) filtered output of image 2. The third sub-image is the Edge Preservation–Modified Histogram Contrast Brightness Equalization (EP-MHCBE) enhanced output of image 2 and final sub-image is the Hybrid Fast Fuzzy C Means Improved Expectation Maximization Clustering technique with Iterative Mean Shift Thresholding (HFFCM IEM-IMST) segmented result of image 2.



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

Figure 8. (a) Input image 2 (b) 2D-AHBAD filtered output (c) EP-MHCBE enhanced output (d) HFFCM IEM-IMST segmented result

From Figure 9 we see that, there are 4 sub images again. The first sub-image is the input image 3. The second sub-image is the 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter (2D-AHBAD) filtered output of image 3. The third sub-image is the Edge Preservation–Modified Histogram Contrast Brightness Equalization (EP-MHCBE) enhanced output of image 3 and final sub-image is the Hybrid Fast Fuzzy C Means Improved Expectation Maximization Clustering technique with Iterative Mean Shift Thresholding (HFFCM IEM-IMST) segmented result of image 3.



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

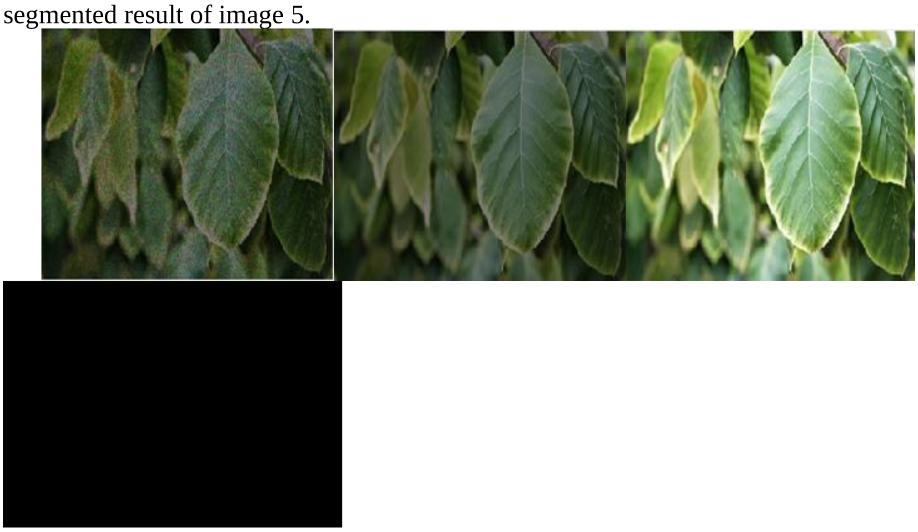
Figure 9. (a) Input image 3 (b) 2D-AHBAD filtered output (c) EP-MHCBE enhanced output (d) HFFCM IEM-IMST segmented result

From Figure 10 we see that, there are 4 sub images. The first sub-image is the input image 4. The second sub-image is the 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter (2D-AHBAD) filtered output of image 4. The third sub-image is the Edge Preservation–Modified Histogram Contrast Brightness Equalization (EP-MHCBE) enhanced output of image 4 and final sub-image is the Hybrid Fast Fuzzy C Means Improved Expectation Maximization Clustering technique with Iterative Mean Shift Thresholding (HFFCM IEM-IMST) segmented result of image 4.

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

Figure 10. (a) Input image 4 (b) 2D-AHBAD filtered output (c) EP-MHCBE enhanced output (d) HFFCM IEM-IMST segmented result

From Figure 11 we see that, there are 4 sub images. The first sub-image is the input image 5. The second sub-image is the 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter (2D-AHBAD) filtered output of image 5. The third sub-image is the Edge Preservation–Modified Histogram Contrast Brightness Equalization (EP-MHCBE) enhanced output of image 5 and final sub-image is the Hybrid Fast Fuzzy C Means Improved Expectation Maximization Clustering technique with Iterative Mean Shift Thresholding (HFFCM IEM-IMST) segmented result of image 5.



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

Figure 11. (a) Input image 5 (b) 2D-AHBAD filtered output (c) EP-MHCBE enhanced output (d) HFFCM IEM-IMST segmented result

4.3. Evaluation of the proposed framework

To evaluate the proposed framework, peak signal to noise ratio (PSNR) and mean square error (MSE) measures were employed. The PSNR achieved by the proposed 2D AHBAD filter was the highest compared to all that filters popularly employed in the literature. The proposed scheme achieves a high PSNR of 54. However, other schemes like 2D Median filter, 2D Double Density Wavelet Transform (2D DDWT) filter, 2D Bilateral filter and 2D Anisotropic Diffusion filter attained PSNR of 27, 34, 36 and 42 respectively. Similarly, the MSE of 2D Median filter, 2D Double Density Wavelet Transform (2D DDWT) filter, 2D Bilateral filter, 2D Anisotropic Diffusion filter and the proposed 2D AHBAD filter were 78.3898, 56.9738, 52.3878, 11.9379 and 0.0039 respectively. Thus, the proposed filter has the minimum MSE.

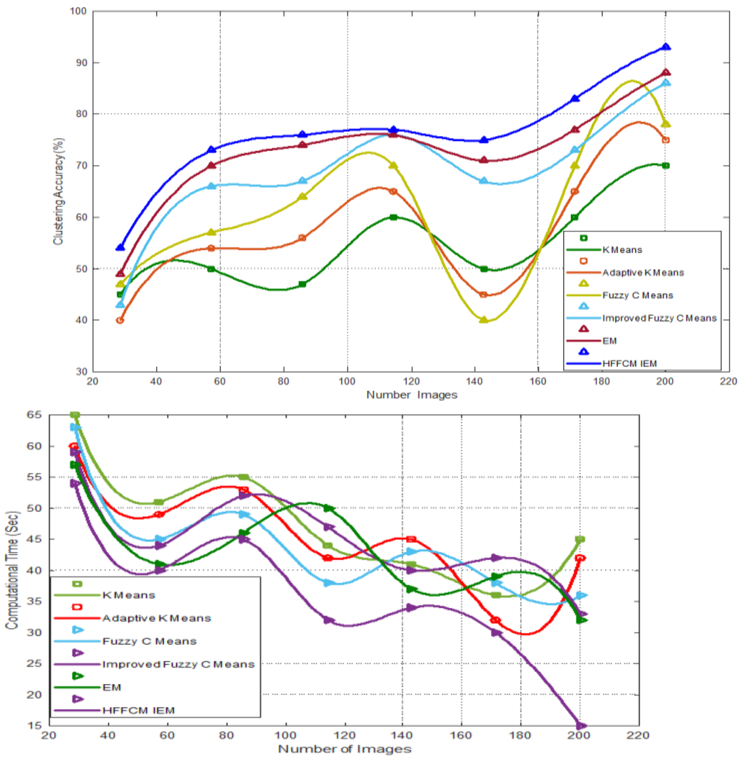
Table 3. PSNR and MSE of different filters

S. No	Type of filters	PSNR in dB	MSE
Image 1	2D Median filter	27	78.3898
Image 1	2D Double Density Wavelet Transform (2D DDWT) filter	34	56.9738
Image 1	2D Bilateral filter	36	52.3878
Image 1	2D Anisotropic Diffusion filter	42	11.9379
Image 1	2D AHBAD (Proposed)	54	0.0039

The classification performance of the proposed scheme was compared with other algorithms like k-nearest neighbour (k-NN) and support vector machine (SVM). The proposed DCNN achieved highest accuracy of 92.25% compared to that of k-NN and SVM that attained 85.75% and 89.30% respectively.

Table 4. Comparison of Overall Accuracy

Classification algorithms	Overall accuracy (%)
k-NN	85.75
SVM	89.30
DCNN	**92.25**



(a) (b)

Figure 12. (a) Comparison of clustering accuracy (b) Comparison of computational time for clustering algorithms

Figure 12 (a) shows the comparison of clustering accuracy for various clustering algorithms like k-means clustering, adaptive k-means clustering, fuzzy-C means, improved fuzzy-C means, Expectation Maximization and the proposed HFFCM IEM algorithm. The system was trained with varying number of images and the accuracy was identified. It was observed that, the proposed clustering algorithm achieved the highest clustering accuracy of 93.1% for 200 training images. Figure 12 (b) shows the comparison of computation time for clustering using various algorithms. Among all the algorithms it was observed that the computational time is minimal for the proposed HFFCM IEM algorithm. In particular, for 180 images, the computational time was as low as 26 s.

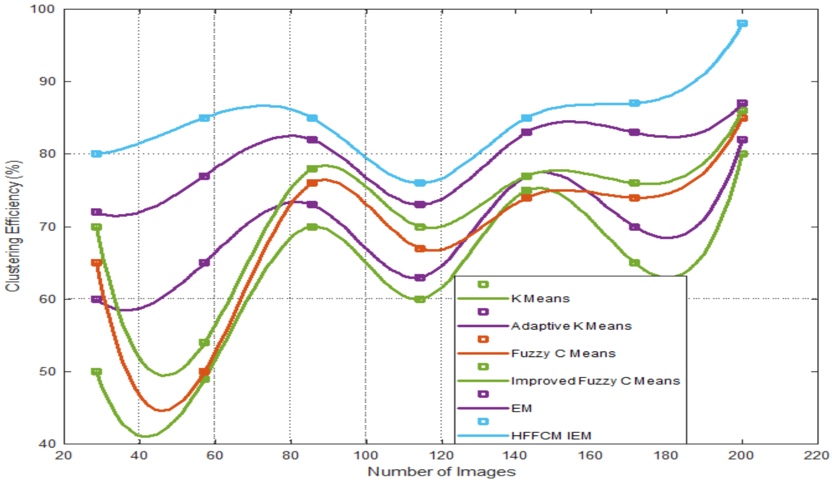
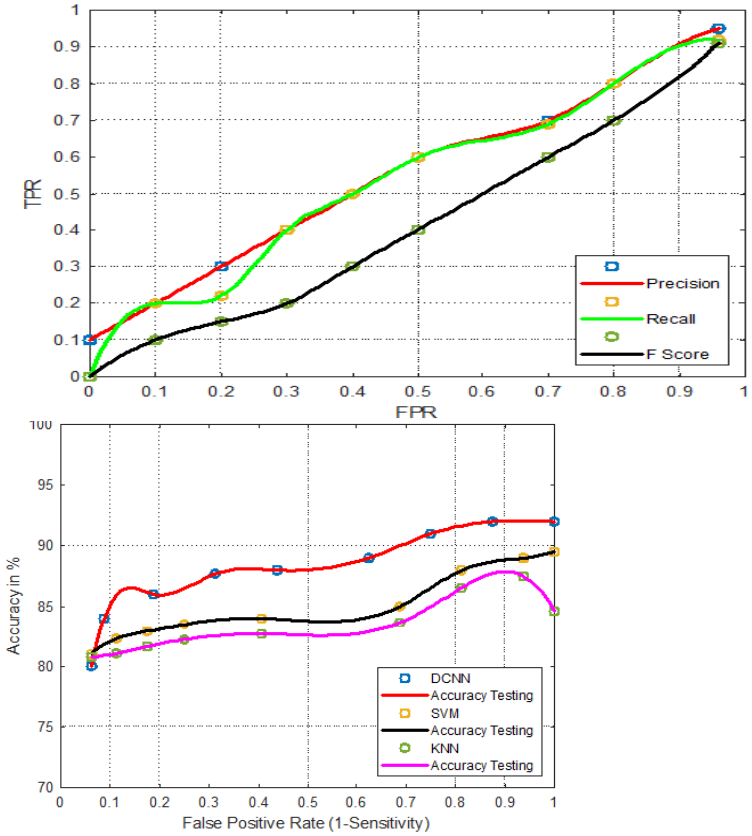


Figure 13. Comparison of clustering efficiency

Figure 13 shows the comparison of clustering efficiency. It is evident for the Figure 13 that the efficiency remains higher than other algorithms. For 200 images, the efficiency using k-means clustering, adaptive k-means clustering, fuzzy-C means, improved fuzzy-C means, Expectation Maximization and the proposed HFFCM IEM algorithm are 80.1%, 82.3%, 84.1%, 84.5%, 84.7% and 98.1% respectively.

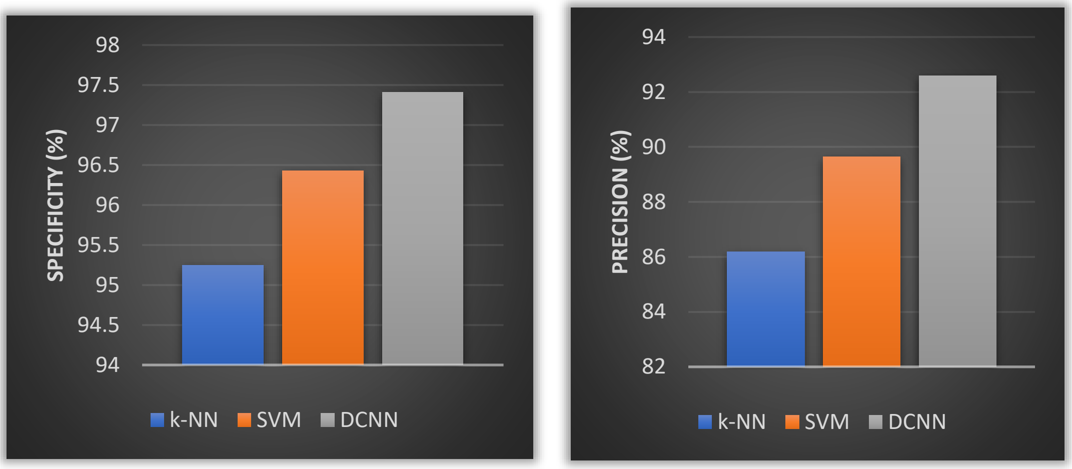


(a) (b)

Figure 14. (a) Variation of AUC curve (b) Variation of classification accuracy

Figure 14 (a) shows the variation of true positive rate vs true negative rate for the proposed classifier. It is clear that the proposed classifier produces the highest AUC in terms of precision, recall and F-score. Figure 14 (b) shows the variation of classification accuracy. The proposed DCNN scheme achieves the highest accuracy of 92.25% which is higher than other classifiers like SVM and k-NN.

Figure 15 (a) represents the comparison of specificity. The k-NN scheme attains a specificity rate of 95.25% and SVM achieves a rate of 96.43%. However, the proposed DCNN achieves a rate of 97.41% that is exceedingly high compared to that of k-NN and SVM. This high rate is due to the usage of efficient training with 1600 images.Figure 15 (b) represents the comparison of precision. The k-NN scheme attains a precision rate of 86.18% and SVM achieves a rate of 89.65%. However, the proposed DCNN achieves a rate of 92.59% that is exceedingly high compared to that of k-NN and SVM. This high rate of precision is due to the implementation of efficient convolutional neural network architectures.



(b) (b)

Figure 15. (a) Comparison of specificity (b) Comparison of precision

Figure 16 (a) represents the comparison of recall. The k-NN scheme attains a recall rate of 85.75% and SVM achieves a rate of 89.30%. However, the proposed DCNN achieves a rate of 92.25% that is exceedingly high compared to that of k-NN and SVM. This high rate of recall is due to the implementation of Softmax layer for classification.Figure 16 (b) represents the comparison of F-score. The k-NN scheme attains a F-score rate of 85.81% and SVM achieves a rate of 89.36%. However, the proposed DCNN achieves a rate of 92.32% that is exceedingly high compared to that of k-NN and SVM. This high rate of F-score is due to incorporation of effectively segmented images during training and classification.

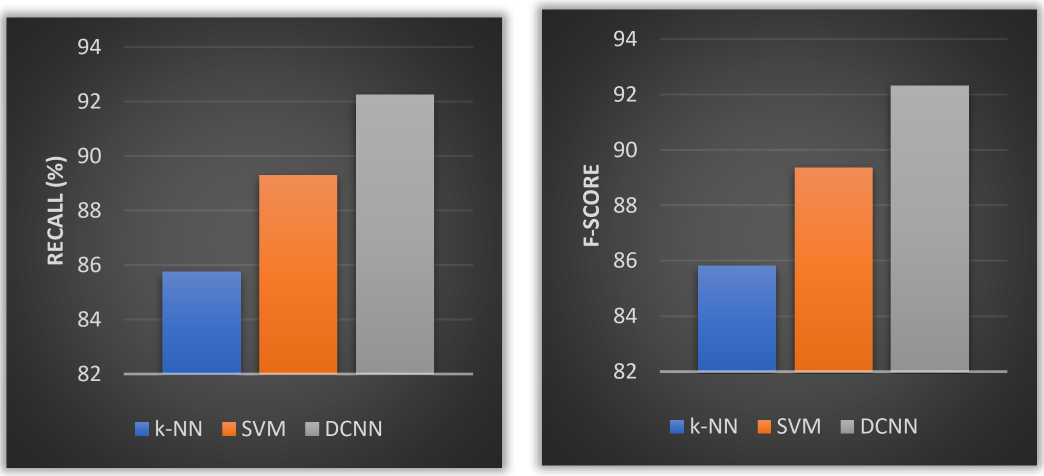


Figure 16 (a). Comparison of recall (b) Comparison of F-score

Figure 17 represents the comparison of classification time. The k-NN scheme uses a time of 19.6ms and SVM uses a time of 16.3ms for classification. However, the proposed DCNN utilizes a minimum time of 11.2ms for classification that is exceedingly low compared to that of k-NN and SVM. This low time consumption is due to the usage of simple convolutional neural network structure for the data classification by the proposed scheme.

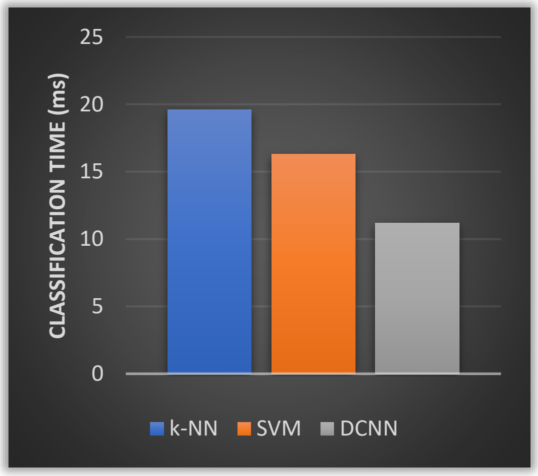


Figure 17. Comparison of classification time

5. Conclusion

In this research we have introduced a new framework for the detection of leaf diseases. In this framework, the image filtration was performed using a novel filter called 2D Adaptive Hybrid Bilateral Anisotropic Diffusion Filter. The filtered images were enhanced using Edge Preservation–Modified Histogram Contrast Brightness Equalization algorithm. A new algorithm called Hybrid Fast Fuzzy C Means Improved Expectation Maximization Clustering technique was used for the computation of clusters and the generated clusters were segmented using Iterative Mean Shift Thresholding algorithm. Finally, classification was done using DCNN architecture. The images were classified into four categories namely, mild, moderate, severe and normal.

The proposed filter achieved a high PSNR of 54 with minimal MSE of 0.0039. Similarly, the proposed classification algorithm achieved highest accuracy of 92.25%. Further, this scheme attained highest values of 97.41%, 92.59%, 92.25% and 92.32% in terms of specificity, precision, recall and F-score respectively.

References

1. R. Nalawade, A. Nagap, L. Jindam, and M. Ugale, "Agriculture Field Monitoring and Plant Leaf Disease Detection," *2020 3rd Int. Conf. Commun. Syst. Comput. IT Appl. CSCITA 2020 - Proc.*, pp. 226–231, 2020, doi: 10.1109/CSCITA47329.2020.9137805.
2. R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention embedded residual CNN for disease detection in tomato leaves," *Appl. Soft Comput. J.*, vol. 86, p. 105933, 2020, doi: 10.1016/j.asoc.2019.105933.
3. N. Ganatra and A. Patel, "A multiclass plant leaf disease detection using image processing and machine learning techniques," *Int. J. Emerg. Technol.*, vol. 11, no. 2, pp. 1082–1086, 2020.
4. M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 293–301, 2020, doi: 10.1016/j.procs.2020.03.225.
5. R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocess. Microsyst.*, vol. 80, no. December 2020, p. 103615, 2021, doi: 10.1016/j.micpro.2020.103615.
6. A. K. Rath and J. K. Meher, "Disease detection in infected plant leaf by computational method," *Arch. Phytopathol. Plant Prot.*, vol. 52, no. 19–20, pp. 1348–1358, 2019, doi: 10.1080/03235408.2019.1708546.
7. M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11. 2019, doi: 10.3390/plants8110468.
8. E. Hossain, M. F. Hossain, and M. A. Rahaman, "A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier," *2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019*, pp. 1–6, 2019, doi: 10.1109/ECACE.2019.8679247.
9. C. Usha Kumari, S. Jeevan Prasad, and G. Mounika, "Leaf disease detection: Feature extraction with k-means clustering and classification with ANN," *Proc. 3rd Int. Conf. Comput. Methodol. Commun.*

ICCMC 2019, no. Iccmc, pp. 1095–1098, 2019, doi: 10.1109/ICCMC.2019.8819750.
10. U. Shruthi, V. Nagaveni, and B. K. Raghavendra, "A Review on Machine Learning Classification Techniques for Plant Disease Detection," *2019 5th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2019*, pp. 281–284, 2019, doi: 10.1109/ICACCS.2019.8728415.
11. X. E. Pantazi, D. Moshou, and A. A. Tamouridou, "Automated leaf disease detection in different crop species through image features analysis and One Class Classifiers," *Comput. Electron. Agric.*, vol. 156, no. July 2018, pp. 96–104, 2019, doi: 10.1016/j.compag.2018.11.005.
12. G. K. Sandhu and R. Kaur, "Plant Disease Detection Techniques: A Review," *2019 Int. Conf. Autom. Comput. Technol. Manag. ICACTM 2019*, pp. 34–38, 2019, doi: 10.1109/ICACTM.2019.8776827.
13. L. Sherly Puspha Annabel, T. Annapoorani, and P. Deepalakshmi, "Machine learning for plant leaf disease detection and classification - A review," *Proc. 2019 IEEE Int. Conf. Commun. Signal Process. ICCSP 2019*, pp. 538–542, 2019, doi: 10.1109/ICCSP.2019.8698004.
14. S. N. Ghaiwat and P. Arora, "Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review," *Int. J. Recent Adv. Eng. Technol. ISSN (Online*, no. 2, pp. 2347–2812, 2014.
15. J. Arun Pandian, G. Geetharamani, and B. Annette, "Data Augmentation on Plant Leaf Disease Image Dataset Using Image Manipulation and Deep Learning Techniques," *Proc. 2019 IEEE 9th Int. Conf. Adv. Comput. IACC 2019*, pp. 199–204, 2019, doi: 10.1109/IACC48062.2019.8971580.
16. A. S. Tulshan and N. Raul, "Plant Leaf Disease Detection using Machine Learning," *2019 10th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2019*, pp. 1–6, 2019, doi: 10.1109/ICCCNT45670.2019.8944556.
17. G. Geetharamani and A. P. J., "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Comput. Electr. Eng.*, 2019, doi: 10.1016/j.compeleceng.2019.04.011.
18. J. G. Arnal Barbedo, "Plant disease identification from individual lesions and spots using deep learning," *Biosyst. Eng.*, 2019, doi: 10.1016/j.biosystemseng.2019.02.002.
19. S. Kaur, S. Pandey, and S. Goel, "Plants Disease Identification and Classification Through Leaf Images: A Survey," *Arch. Comput. Methods Eng.*, vol. 26, no. 2, pp. 507–530, 2019, doi: 10.1007/s11831-018-9255-6.
20. K. Ahmed, T. R. Shahidi, S. M. Irfanul Alam, and S. Momen, "Rice leaf disease detection using machine learning techniques," *2019 Int. Conf. Sustain. Technol. Ind. 4.0, STI 2019*, vol. 0, pp. 1–5, 2019, doi: 10.1109/STI47673.2019.9068096.