**INTRODUCTION**

India is a cultivated country and about 70% of the population depends on agriculture. Farmers have large range of diversity for selecting various suitable crops and finding the suitable pesticides for plant. Disease on plant leads to the significant reduction in both the quality and quantity of agricultural products. The studies of plant disease refer to the studies of visually observable patterns on the plants. Monitoring of health and disease on plant plays an important role in successful cultivation of crops in the farm. In early days, the monitoring and analysis of plant diseases were done manually by the expertise person in that field. This requires tremendous amount of work and also requires excessive processing time. The image processing techniques can be used in the plant disease detection. In most of the cases disease symptoms are seen on the leaves, stem and fruit. The plant leaf for the detection of disease is considered which shows the disease symptoms.

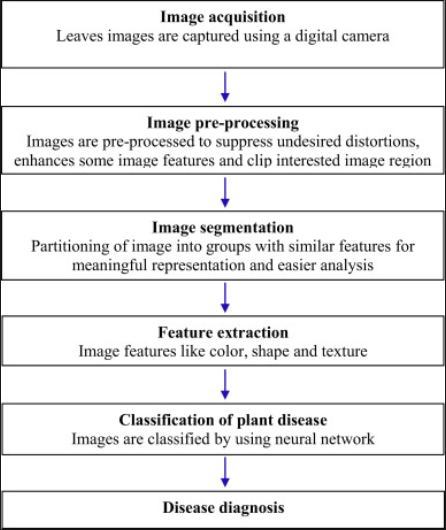
Agriculture has played a key role in the development of human civilization. If there is decrease in agro products, total economy will get affected. Therefore judicious management of all input resources such as soil, seed, water, fertilizers etc. is essential for sustainability. As diseases are inevitable, detecting them plays major role. One can refer incident that occurred in 2007, Georgia (USA), it is estimated that approximately 539 USD was the loss incurred due to plant diseases as well as controlling them. The naked eye observation of farmers followed by chemical test is the main way of detection and classification of agricultural plant diseases. In developing countries, farming land can be much larger and farmers cannot observe each and every plant, every day. Farmers are unaware of non-native diseases. Consultation of experts for this might be time consuming & costly. Also unnecessary use of pesticides might be dangerous for natural resources such as water, soil, air, food chain etc. as well as it is expected that there need to be less contamination of food products with pesticides. There are two main characteristics of plant disease detection machine-learning methods that must be achieved, they are: speed and accuracy. There is need for developing technique such as automatic plant disease detection and classification using leaf image processing techniques. This will prove useful technique for farmers and will alert them at the right time before spreading of the disease over large area. Solution is composed of four main phases; in the first phase we create a color transformation structure for the RGB leaf image and then, we apply color space transformation for the color transformation structure. Then image is segmented using the K-means clustering technique. In the second phase, unnecessary part (green area) within leaf area is removed. In third phase we calculate the texture features for the segmented infected object. Finally, in the fourth phase the extracted features are passed through a pre-trained neural network

**STEPS FOR DISEASE DETECTION**

First the images of various leaves are acquired using high resolution camera so as to get the better results & efficiency. Then image processing techniques are applied to these images to extract useful features which will be required for further analysis.

When a leaf area is to be measure in field condition, it is difficult to keep cameras optical axis vertical with leaf plane. As this can affect the measurements so the leaf is separated and placed horizontally on white background to take photograph.

The basic steps for plant disease detection and classification using image processing are shown (Fig. 1).



**A] Image Acquisition**

The images of the plant leaf are captured through the camera. This image is in RGB (Red, Green And Blue) form. color transformation structure for the RGB leaf image is created, and then, a device-independent color space transformation for the color transformation structure is applied.

**B] Image Pre-processing**

Noise gets added during acquisition of leaf images. So we use different types of filtering techniques to remove noise. We create device independent color space transformation structure. Thus we create the color transformation structure that defines the color space conversion. The next step is that we apply device-independent color space transformation, which converts the color values in the image to color space specified in the color transformation structure. The color transformation structure specifies various parameters of transformation. A device independent color space is the one where the resultant color depends on the equipment used to produce it. For example the color produced using pixel with a given RGB values will be altered as brightness and contrast on display device used. Thus the RGB system is a color space that is dependent. To improve the precision of the disease detection and classification process, a device independent color space is required. In device independent color space, the coordinates used to specify the color will produce the same color regardless of the device used to take the pictures.

To remove noise in image or other object removal, different pre-processing techniques is considered. Image clipping i.e. cropping of the leaf image to get the interested image region. Image smoothing is done using the smoothing filter. Image enhancement is carried out for increasing the contrast.

The RGB images into the grey images using colour conversion using equation (1). f(x)=0.2989\*R + 0.5870\*G + 0.114.\*B - - - - - - - - - - - - - (1)

Then the histogram equalization which distributes the intensities of the images is applied on the image to enhance the plant disease images. The cumulative distribution function is used to distribute intensity values.

**C] Image Segmentation**

Segmentation means partitioning of image into various part of same features or having some similarity. The segmentation can be done using various methods like otsu’ method, k-means clustering, converting RGB image into HIS model etc.

**1] Segmentation using Boundary and spot detection algorithm:**

The RGB image is converted into the HIS model for segmenting. Boundary detection and spot detection helps to find the infected part of the leaf as discussed in . For boundary detection the 8 connectivity of pixels is consider and boundary detection algorithm is applied.

**2] K-means clustering:**

The K-means clustering is used for classification of object based on a set of features into K number of classes. The classification of object is done by minimizing the sum of the squares of the distance between the object and the corresponding cluster.

The algorithm for K –means Clustering:

1. Pick center of K cluster, either randomly or based on some heuristic. 2. Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center. 3. Again compute the cluster centers by averaging all of the pixels in the cluster. Repeat steps 2 and 3 until convergence is attained.

**3] Otsu Threshold Algorithm:**

Thresholding creates binary images from grey-level images by setting all pixels below some threshold to zero and all pixels above that threshold to one. The Otsu algorithm defined in is as follows:

i) According to the threshold, Separate pixels into two clusters

ii) Then find the mean of each cluster.

iii) Square the difference between the means.

iv) Multiply the number of pixels in one cluster times the number in the other

The infected leaf shows the symptoms of the disease by changing the color of the leaf. Hence the greenness of the leaves can be used for the detection of the infected portion of the leaf. The R, G and B component are extracted from the image. The threshold is calculated using the Otsu’s method. Then the green pixels is masked and removed if the green pixel intensities are less than the computed threshold.

**D] Feature Extraction**

Feature extraction plays an important role for identification of an object. In many application of image processing feature extraction is used. Color, texture, morphology, edges etc. are the features which can be used in plant disease detection.

We can consider color, texture and morphology as a feature for disease detection. Morphological result gives better result than the other features. Texture means how the colour is distributed in the image, the roughness, hardness of the image. It can also be used for the detection of infected plant areas.

**E] Classification**

The extracted features are given as inputs to pre-trained neural network for automatic classification of diseases. BPNN, SVM, Radial basis functions, K-nearest neighbors are some well-known neural networks. Neural network is chosen as a classification tool due to its well-known technique as a successful classifier for many real applications. The training and validation processes are among the important steps in developing an accurate process model using NNs. The dataset for training and validation processes consists of two parts

1. The training feature set which are used to train the NN model.

2. The testing features sets are used to verify the accuracy of the trained NN model.

1) **Using SVM CLASSIFER:**

Support Vector machine (SVM) is a non-linear Classifier. This is a new trend in machine learning algorithm which is used in many pattern recognition problems, including texture classification. In SVM, the input data is non-linearly mapped to linearly separated data in some high dimensional space providing good classification performance. SVM maximizes the marginal distance between different classes. The division of classes is carried out with different kernels.SVM is designed to work with only two classes by determining the hyper plane to divide two classes.

This is done by maximizing the margin from the hyper plane to the two classes. The samples closest to the margin that were selected to determine the hyper plane is known as support vectors. Fig below shows the support vector machines concept. Multiclass classification is also applicable and is basically built up by various two class SVMs to solve the problem, either by using one-versus-all or one versus-one. The winning class is then determined by the highest output function or the maximum votes respectively.

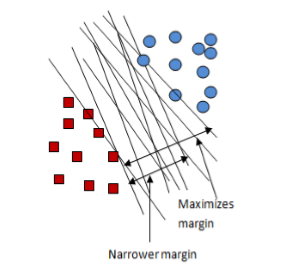


Fig.2 Support Vector Machine

**Main advantages of SVM are:**

1. Its prediction accuracy is high.

2. Its working is robust when training examples contain errors.

3. Its simple geometric interpretation and a sparse solution.

4. Like neural networks the computational complexity of SVMs does not depend on the dimensionality of the input space.

**Drawbacks of SVM are:**

1. This classifier involves long training time.

2. In SVM it is difficult to understand the learned function (weights).

3. The large number of support vectors used from the training set to perform classification task.

**Programs:**

Detect.m

% Project Title: Pomegranate Leaf Disease Detection

clc

close all

clear all

[filename, pathname] = uigetfile({'\*.\*';'\*.bmp';'\*.jpg';'\*.gif'}, 'Pick a Leaf Image File');

I = imread([pathname,filename]);

I = imresize(I,[256,256]);

%figure, imshow(I); title('Query Leaf Image');

% Enhance Contrast

I = imadjust(I,stretchlim(I));

figure, imshow(I);title('Contrast Enhanced');

% Otsu Segmentation

I\_Otsu = im2bw(I,graythresh(I));

% Conversion to HIS

I\_HIS = rgb2hsi(I);

%% Extract Features

% Function call to evaluate features

%[feat\_disease seg\_img] = EvaluateFeatures(I)

% Color Image Segmentation

% Use of K Means clustering for segmentation

% Convert Image from RGB Color Space to L\*a\*b\* Color Space

% The L\*a\*b\* space consists of a luminosity layer 'L\*', chromaticity-layer 'a\*' and 'b\*'.

% All of the color information is in the 'a\*' and 'b\*' layers.

cform = makecform('srgb2lab');

% Apply the colorform

lab\_he = applycform(I,cform);

% Classify the colors in a\*b\* colorspace using K means clustering.

% Since the image has 3 colors create 3 clusters.

% Measure the distance using Euclidean Distance Metric.

ab = double(lab\_he(:,:,2:3));

nrows = size(ab,1);

ncols = size(ab,2);

ab = reshape(ab,nrows\*ncols,2);

nColors = 3;

[cluster\_idx cluster\_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...

'Replicates',3);

%[cluster\_idx cluster\_center] = kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);

% Label every pixel in tha image using results from K means

pixel\_labels = reshape(cluster\_idx,nrows,ncols);

%figure,imshow(pixel\_labels,[]), title('Image Labeled by Cluster Index');

% Create a blank cell array to store the results of clustering

segmented\_images = cell(1,3);

% Create RGB label using pixel\_labels

rgb\_label = repmat(pixel\_labels,[1,1,3]);

for k = 1:nColors

colors = I;

colors(rgb\_label ~= k) = 0;

segmented\_images{k} = colors;

end

figure, subplot(3,1,1);imshow(segmented\_images{1});title('Cluster 1'); subplot(3,1,2);imshow(segmented\_images{2});title('Cluster 2');

subplot(3,1,3);imshow(segmented\_images{3});title('Cluster 3');

set(gcf, 'Position', get(0,'Screensize'));

% Feature Extraction

x = inputdlg('Enter the cluster no. containing the ROI only:');

i = str2double(x);

% Extract the features from the segmented image

seg\_img = segmented\_images{i};

% Convert to grayscale if image is RGB

if ndims(seg\_img) == 3

img = rgb2gray(seg\_img);

end

%figure, imshow(img); title('Gray Scale Image');

% Evaluate the disease affected area

black = im2bw(seg\_img,graythresh(seg\_img));

%figure, imshow(black);title('Black & White Image');

m = size(seg\_img,1);

n = size(seg\_img,2);

zero\_image = zeros(m,n);

%G = imoverlay(zero\_image,seg\_img,[1 0 0]);

cc = bwconncomp(seg\_img,6);

diseasedata = regionprops(cc,'basic');

A1 = diseasedata.Area;

sprintf('Area of the disease affected region is : %g%',A1);

I\_black = im2bw(I,graythresh(I));

kk = bwconncomp(I,6);

leafdata = regionprops(kk,'basic');

A2 = leafdata.Area;

sprintf(' Total leaf area is : %g%',A2);

%Affected\_Area = 1-(A1/A2);

Affected\_Area = (A1/A2);

if Affected\_Area < 0.1

Affected\_Area = Affected\_Area+0.15;

end

sprintf('Affected Area is: %g%%',(Affected\_Area\*100))

% Create the Gray Level Cooccurance Matrices (GLCMs)

glcms = graycomatrix(img);

% Derive Statistics from GLCM

stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');

Contrast = stats.Contrast;

Correlation = stats.Correlation;

Energy = stats.Energy;

Homogeneity = stats.Homogeneity;

Mean = mean2(seg\_img);

Standard\_Deviation = std2(seg\_img);

Entropy = entropy(seg\_img);

RMS = mean2(rms(seg\_img));

%Skewness = skewness(img)

Variance = mean2(var(double(seg\_img)));

a = sum(double(seg\_img(:)));

Smoothness = 1-(1/(1+a));

Kurtosis = kurtosis(double(seg\_img(:)));

Skewness = skewness(double(seg\_img(:)));

% Inverse Difference Movement

m = size(seg\_img,1);

n = size(seg\_img,2);

in\_diff = 0;

for i = 1:m

for j = 1:n

temp = seg\_img(i,j)./(1+(i-j).^2);

in\_diff = in\_diff+temp;

end

end

IDM = double(in\_diff);

feat\_disease = [Contrast,Correlation,Energy,Homogeneity, Mean, Standard\_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness, IDM];

%%

% Load All The Features

load('Training\_Data.mat')

% Put the test features into variable 'test'

test = feat\_disease;

result = multisvm(Train\_Feat,Train\_Label,test);

%disp(result);

% Visualize Results

if result == 0

helpdlg(' Alternaria Alternata ');

disp(' Alternaria Alternata ');

elseif result == 1

helpdlg(' Anthracnose ');

disp('Anthracnose');

elseif result == 2

helpdlg(' Bacterial Blight ');

disp(' Bacterial Blight ');

elseif result == 3

helpdlg(' Cercospora Leaf Spot ');

disp('Cercospora Leaf Spot');

elseif result == 4

helpdlg(' Healthy Leaf ');

disp('Healthy Leaf ');

end

%% Evaluate Accuracy

load('Accuracy\_Data.mat')

Accuracy\_Percent= zeros(200,1);

for i = 1:500

data = Train\_Feat;

%groups = ismember(Train\_Label,1);

groups = ismember(Train\_Label,0);

[train,test] = crossvalind('HoldOut',groups);

cp = classperf(groups);

svmStruct = svmtrain(data(train,:),groups(train),'showplot',false,'kernel\_function','linear');

classes = svmclassify(svmStruct,data(test,:),'showplot',false);

classperf(cp,classes,test);

Accuracy = cp.CorrectRate;

Accuracy\_Percent(i) = Accuracy.\*100;

end

Max\_Accuracy = max(Accuracy\_Percent);

sprintf('Accuracy of Linear Kernel with 500 iterations is: %g%%',Max\_Accuracy)

Evalualte.m

% Function to call and evaluate features

function [feat\_disease seg\_img] = EvaluateFeatures(I)

% Color Image Segmentation

% Use of K Means clustering for segmentation

% Convert Image from RGB Color Space to L\*a\*b\* Color Space

% The L\*a\*b\* space consists of a luminosity layer 'L\*', chromaticity-layer 'a\*' and 'b\*'.

% All of the color information is in the 'a\*' and 'b\*' layers.

cform = makecform('srgb2lab');

% Apply the colorform

lab\_he = applycform(I,cform);

% Classify the colors in a\*b\* colorspace using K means clustering.

% Since the image has 3 colors create 3 clusters.

% Measure the distance using Euclidean Distance Metric.

ab = double(lab\_he(:,:,2:3));

nrows = size(ab,1);

ncols = size(ab,2);

ab = reshape(ab,nrows\*ncols,2);

nColors = 3;

[cluster\_idx cluster\_center] = kmeans(ab,nColors,'distance','sqEuclidean', ...

'Replicates',3);

%[cluster\_idx cluster\_center] = kmeans(ab,nColors,'distance','sqEuclidean','Replicates',3);

% Label every pixel in tha image using results from K means

pixel\_labels = reshape(cluster\_idx,nrows,ncols);

%figure,imshow(pixel\_labels,[]), title('Image Labeled by Cluster Index');

% Create a blank cell array to store the results of clustering

segmented\_images = cell(1,3);

% Create RGB label using pixel\_labels

rgb\_label = repmat(pixel\_labels,[1,1,3]);

for k = 1:nColors

colors = I;

colors(rgb\_label ~= k) = 0;

segmented\_images{k} = colors;

end

figure, subplot(3,1,1);imshow(segmented\_images{1});title('Cluster 1'); subplot(3,1,2);imshow(segmented\_images{2});title('Cluster 2');

subplot(3,1,3);imshow(segmented\_images{3});title('Cluster 3');

% Feature Extraction

x = inputdlg('Enter the cluster no. containing the disease affected leaf part only:');

i = str2double(x);

% Extract the features from the segmented image

seg\_img = segmented\_images{i};

% Convert to grayscale if image is RGB

if ndims(seg\_img) == 3

img = rgb2gray(seg\_img);

end

%figure, imshow(img); title('Gray Scale Image');

% Evaluate the disease affected area

black = im2bw(seg\_img,graythresh(seg\_img));

%figure, imshow(black);title('Black & White Image');

m = size(seg\_img,1);

n = size(seg\_img,2);

zero\_image = zeros(m,n);

%G = imoverlay(zero\_image,seg\_img,[1 0 0]);

cc = bwconncomp(seg\_img,6);

diseasedata = regionprops(cc,'basic');

A1 = diseasedata.Area;

sprintf('Area of the disease affected region is : %g%',A1);

I\_black = im2bw(I,graythresh(I));

kk = bwconncomp(I,6);

leafdata = regionprops(kk,'basic');

A2 = leafdata.Area;

sprintf(' Total leaf area is : %g%',A2);

%Affected\_Area = 1-(A1/A2);

Affected\_Area = (A1/A2);

if Affected\_Area < 1

Affected\_Area = Affected\_Area+0.15;

end

sprintf('Affected Area is: %g%%',(Affected\_Area\*100))

% Create the Gray Level Cooccurance Matrices (GLCMs)

glcms = graycomatrix(img);

% Derive Statistics from GLCM

stats = graycoprops(glcms,'Contrast Correlation Energy Homogeneity');

Contrast = stats.Contrast;

Correlation = stats.Correlation;

Energy = stats.Energy;

Homogeneity = stats.Homogeneity;

Mean = mean2(seg\_img);

Standard\_Deviation = std2(seg\_img);

Entropy = entropy(seg\_img);

RMS = mean2(rms(seg\_img));

%Skewness = skewness(img)

Variance = mean2(var(double(seg\_img)));

a = sum(double(seg\_img(:)));

Smoothness = 1-(1/(1+a));

Kurtosis = kurtosis(double(seg\_img(:)));

Skewness = skewness(double(seg\_img(:)));

% Inverse Difference Movement

m = size(seg\_img,1);

n = size(seg\_img,2);

in\_diff = 0;

for i = 1:m

for j = 1:n

temp = seg\_img(i,j)./(1+(i-j).^2);

in\_diff = in\_diff+temp;

end

end

IDM = double(in\_diff);

feat\_disease = [Contrast,Correlation,Energy,Homogeneity, Mean, Standard\_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis, Skewness, IDM];

Rgb2hsi.m

function hsi = rgb2hsi(rgb)

%RGB2HSI Converts an RGB image to HSI.

% HSI = RGB2HSI(RGB) converts an RGB image to HSI. The input image

% is assumed to be of size M-by-N-by-3, where the third dimension

% accounts for three image planes: red, green, and blue, in that

% order. If all RGB component images are equal, the HSI conversion

% is undefined. The input image can be of class double (with values

% in the range [0, 1]), uint8, or uint16.

%

% The output image, HSI, is of class double, where:

% hsi(:, :, 1) = hue image normalized to the range [0, 1] by

% dividing all angle values by 2\*pi.

% hsi(:, :, 2) = saturation image, in the range [0, 1].

% hsi(:, :, 3) = intensity image, in the range [0, 1].

% Copyright 2002-2004 R. C. Gonzalez, R. E. Woods, & S. L. Eddins

% Digital Image Processing Using MATLAB, Prentice-Hall, 2004

% $Revision: 1.5 $ $Date: 2005/01/18 13:44:59 $

% Extract the individual component images.

rgb = im2double(rgb);

r = rgb(:, :, 1);

g = rgb(:, :, 2);

b = rgb(:, :, 3);

% Implement the conversion equations.

num = 0.5\*((r - g) + (r - b));

den = sqrt((r - g).^2 + (r - b).\*(g - b));

theta = acos(num./(den + eps));

H = theta;

H(b > g) = 2\*pi - H(b > g);

H = H/(2\*pi);

num = min(min(r, g), b);

den = r + g + b;

den(den == 0) = eps;

S = 1 - 3.\* num./den;

H(S == 0) = 0;

I = (r + g + b)/3;

% Combine all three results into an hsi image.

hsi = cat(3, H, S, I);

multisvm.m

function [itrfin] = multisvm( T,C,test )

%Inputs: T=Training Matrix, C=Group, test=Testing matrix

%Outputs: itrfin=Resultant class

itrind=size(test,1);

itrfin=[];

Cb=C;

Tb=T;

for tempind=1:itrind

tst=test(tempind,:);

C=Cb;

T=Tb;

u=unique(C);

N=length(u);

c4=[];

c3=[];

j=1;

k=1;

if(N>2)

itr=1;

classes=0;

cond=max(C)-min(C);

while((classes~=1)&&(itr<=length(u))&& size(C,2)>1 && cond>0)

%This while loop is the multiclass SVM Trick

c1=(C==u(itr));

newClass=c1;

%svmStruct = svmtrain(T,newClass,'kernel\_function','rbf'); % I am using rbf kernel function, you must change it also

svmStruct = svmtrain(T,newClass);

classes = svmclassify(svmStruct,tst);

% This is the loop for Reduction of Training Set

for i=1:size(newClass,2)

if newClass(1,i)==0;

c3(k,:)=T(i,:);

k=k+1;

end

end

T=c3;

c3=[];

k=1;

% This is the loop for reduction of group

for i=1:size(newClass,2)

if newClass(1,i)==0;

c4(1,j)=C(1,i);

j=j+1;

end

end

C=c4;

c4=[];

j=1;

cond=max(C)-min(C); % Condition for avoiding group

%to contain similar type of values

%and the reduce them to process

% This condition can select the particular value of iteration

% base on classes

if classes~=1

itr=itr+1;

end

end

end

valt=Cb==u(itr); % This logic is used to allow classification

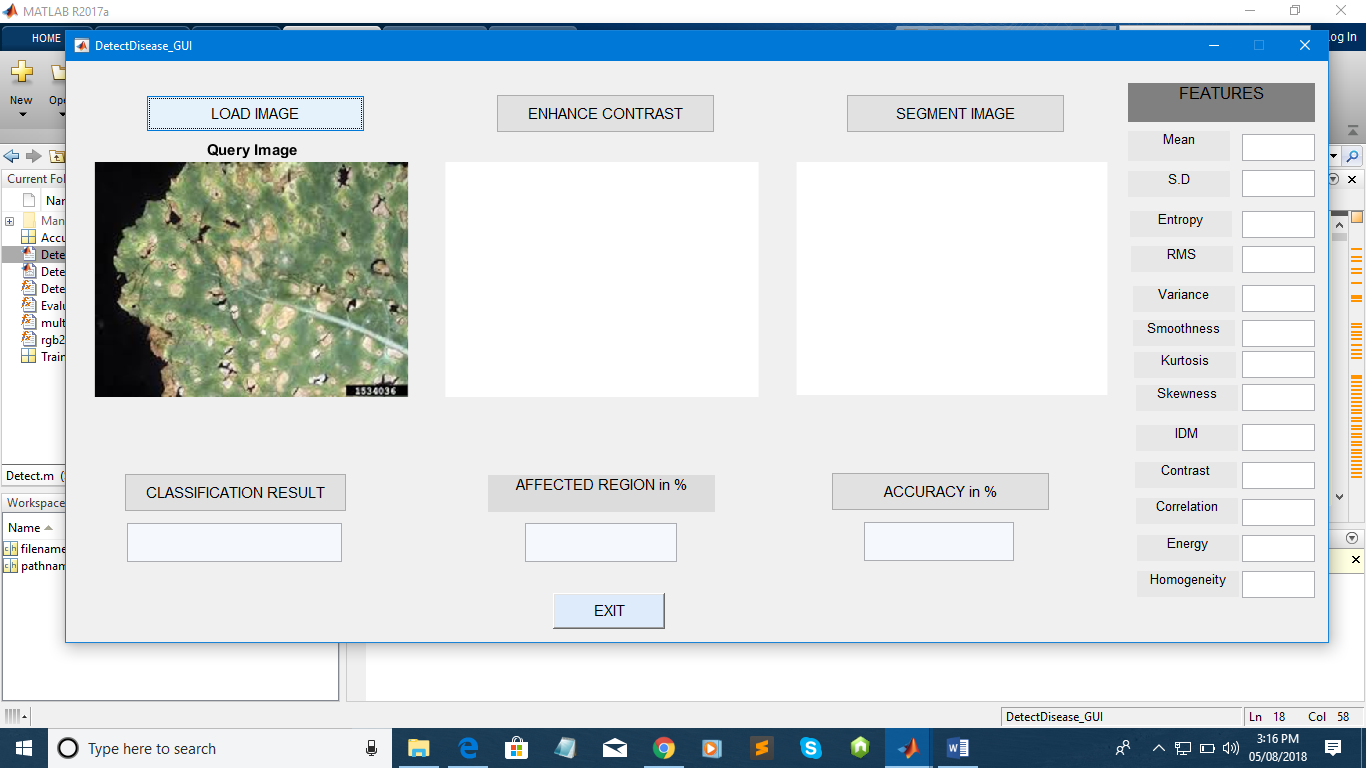
val=Cb(valt==1); % of multiple rows testing matrix

val=unique(val);

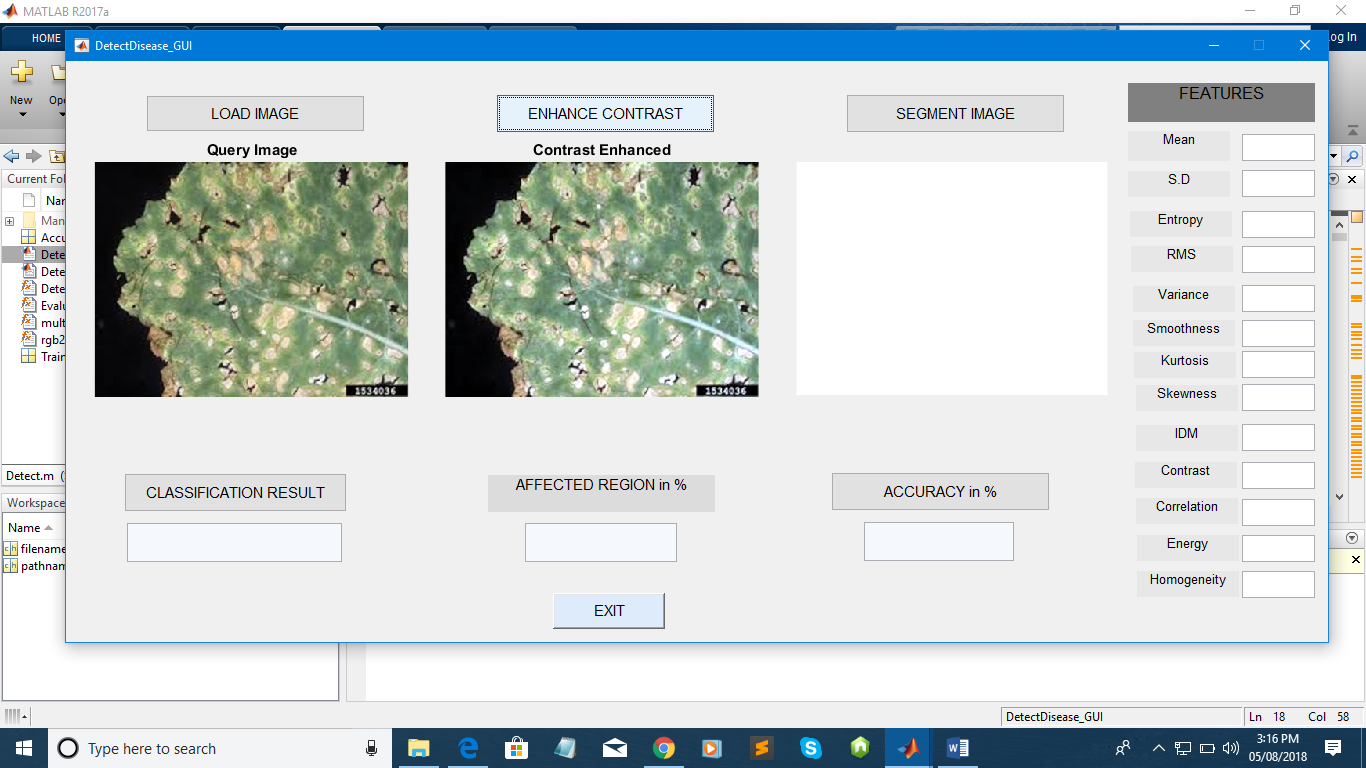
itrfin(tempind,:)=val;

end

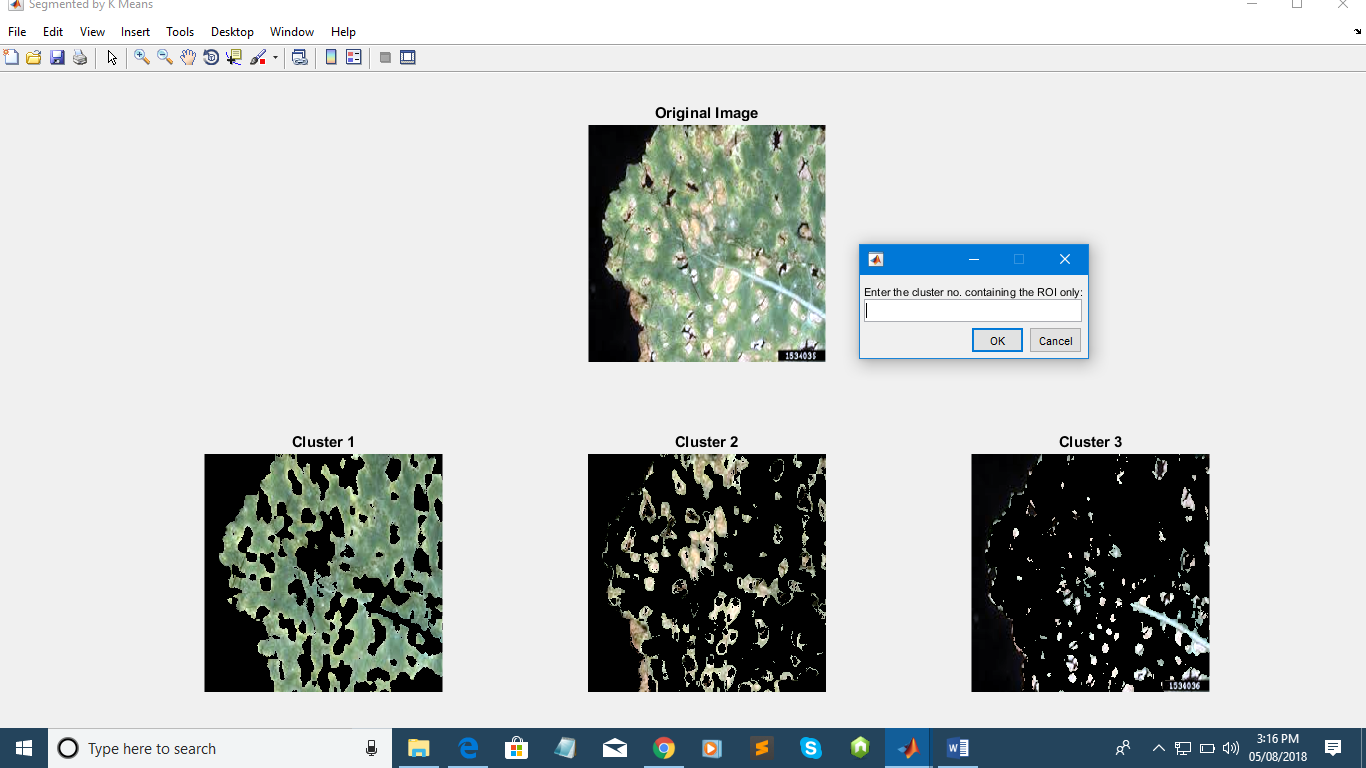
end



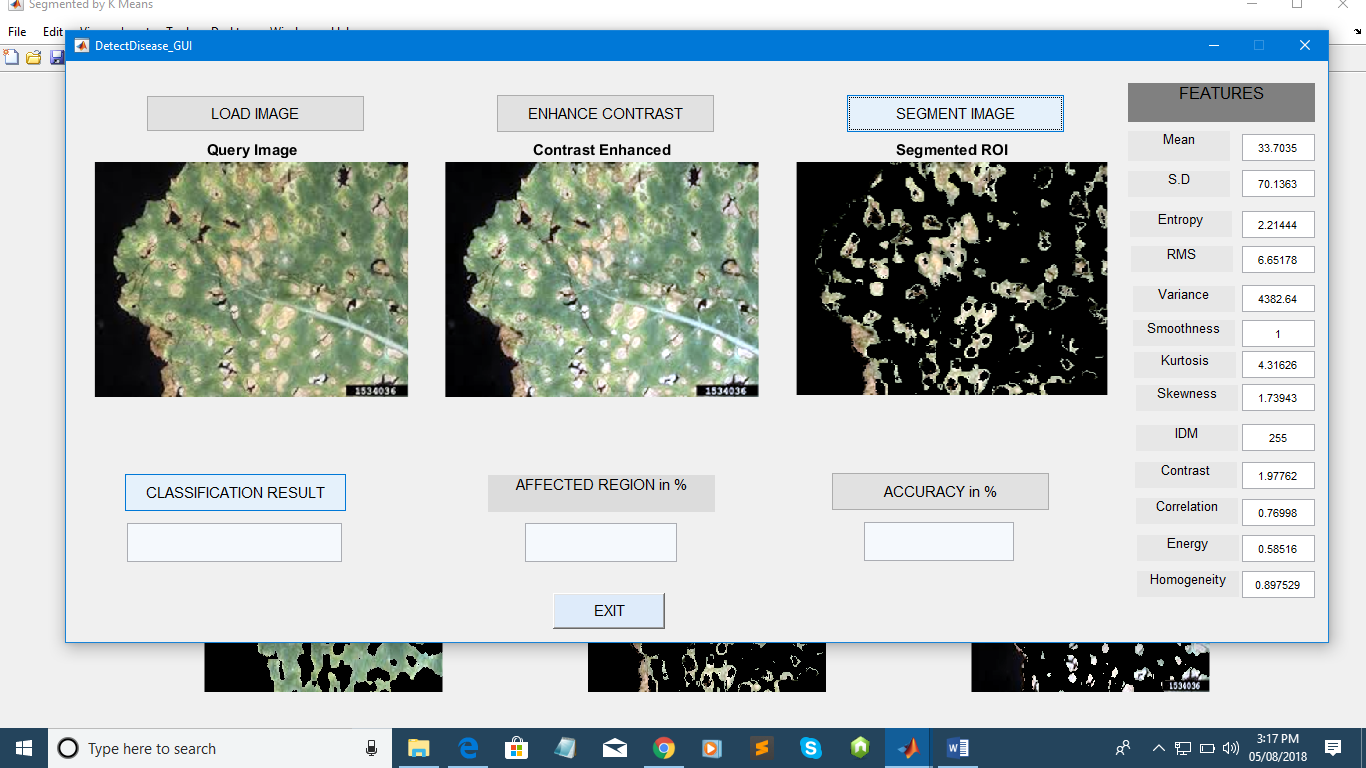
Load image

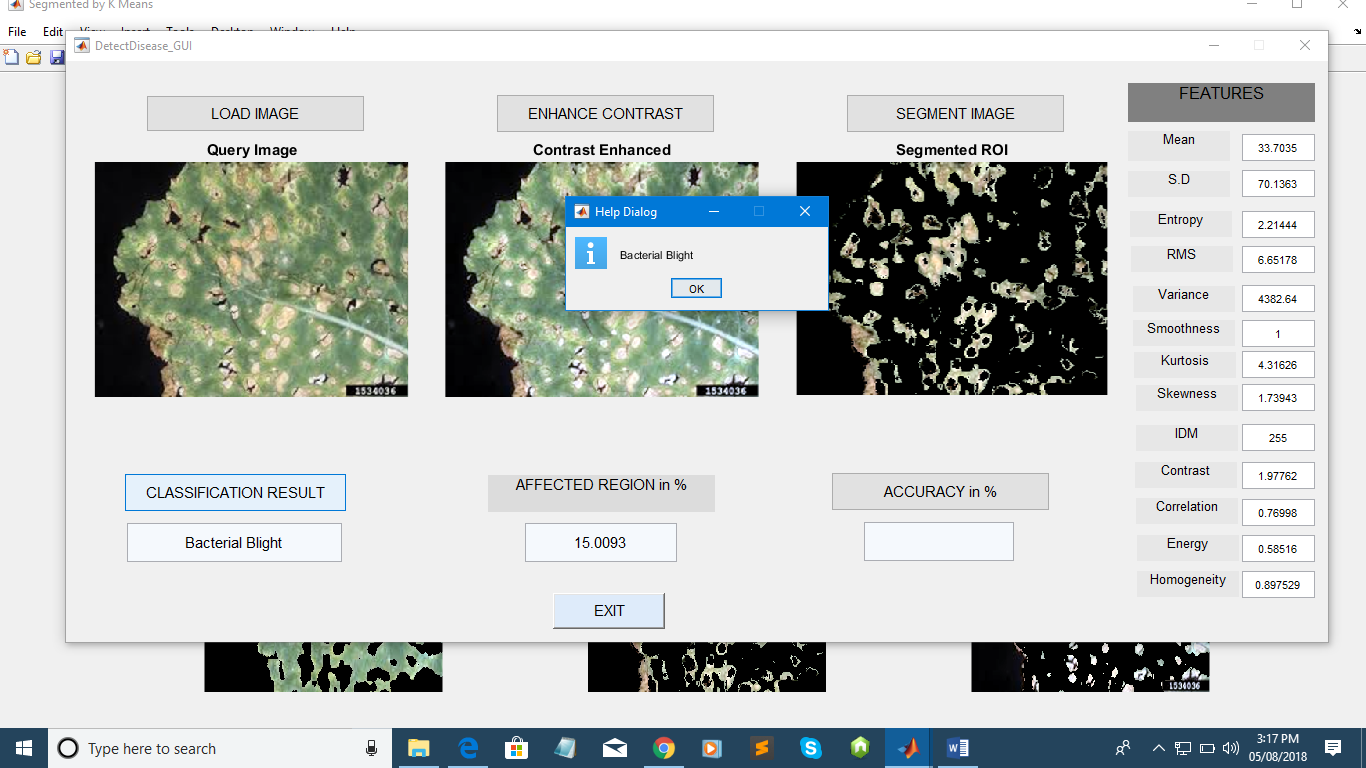


Enhance contrast

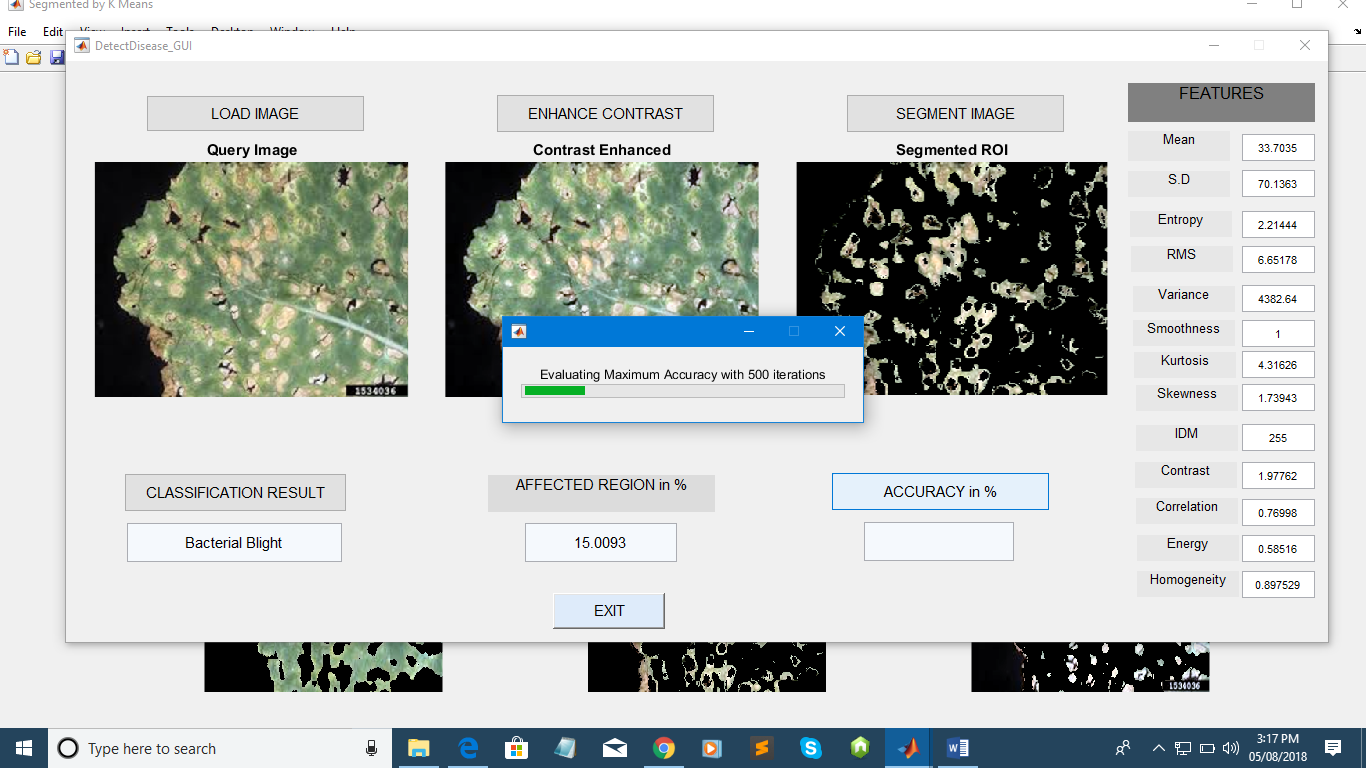


Segment Image & Select region of Interest

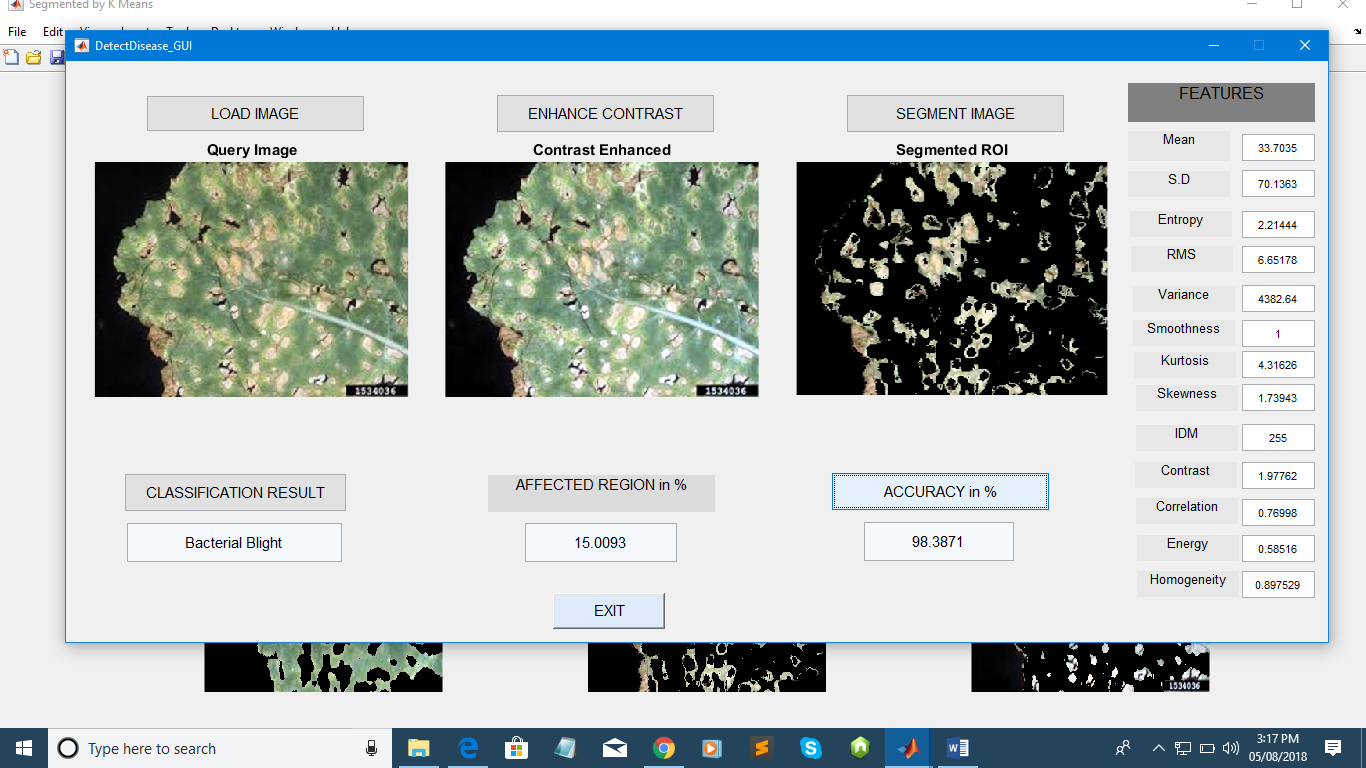




Result



Calculate Accuracy



**CONCLUSION**

The accurately detection and classification of the plant disease is very important for the successful cultivation of crop and this can be done using image processing. In this report we discussed various techniques to segment the disease part of the plant. Here we also discussed some Feature extraction and classification techniques to extract the features of infected leaf and the classification of plant diseases. The use of SVM methods for classification of disease in plants such as self-organizing feature map, back propagation algorithm can be efficiently used. From these methods, we can accurately identify and classify various plant diseases using image processing techniques.

REFERENCES

[1] Smita Naikwadi, NiketAmoda, “Advances in image processing for detection of plant diseases”, International journal of application or innovation in engineering and managrment(IJAIEM) Volume 2,Issue11,November 2013   
[2] www.mathworks.in   
[3] S. Ananthi, S. Vishnu Varthini, Detection and classification of plant leaf diseases‖, IJREAS, 2012, 2(2), 763-773.   
[4] Arti N. Rathod, Bhavesh A. Tanawala, Vatsal H. Shah, International Journal of Advance Engineer ing and Research Development (IJAERD) Volume 1,Issue 6,June 2014, e-ISSN: 2348 - 4470   
[5] Simona E. Grigorescu, Nicolai Petkov, and Peter Kruizinga, Comparison of Texture Features Based on Gabor Filters ,IEEE transaction on Image processing vol 11,No.10,pp1160-1167, oct 2002   
[6] Book: Gonzalez and Woods,”Digital image processing using MATLAB”,Pearson Education   
[7] S. Arivazhagan, R. Newlin Shebiah\* , S. Ananthi, S. Vishnu Varthin, Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features   
[8] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. ALRahamneh Fast and Accurate Detection and Classification of Plant Diseases, International Journal of Computer Applications.   
[9] Arti N. Rathod, Bhavesh A. Tanawala,Vatsal H. ShahLeaf disease detection using image processing and neural network, International Journal of Advance Engineer ing and Research Development (IJAERD),Volume 1,Issue 6,June 2014. [10] Mr. Hrishikesh P. Kanjalkar, Prof. S.S.Lokhande, Detection and Classification of Plant Leaf Diseases using ANN, International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August-2013   
[11] Book: Principles of Soft Computing, By S.N Sivanandam and S.N.Deepa   
[12] Book: Neural Networks, A classroom Approach by Satish Kumar, Tata McGraw Hill, publications.   
[13] www.pdkv.ac.in   
[14] www.ikisan.com