IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Classification of Healthy and Diseased Cactus plants using SVM

Hailay Beyene[1*], Narayan A.Joshi[2]

[1]Department of Computer Science, Parul University, Gujarat, India
[2]Department of MCA, Dharmsinh Desai University, Gujarat, India

Corresponding Author: berhekihshen@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i5.14121426 | Available online at: www.ijcseonline.org

Accepted: 13/May/2019, Published: 31/May/2019

Abstract-Machine learning is very important technology that can support people in different disciplines (Agriculture, health centers, household, transportation, etc) and different levels of life. Machine learning increases accuracy. It uses various types of data (image, video, audio and text) for different purposes and applications. Our work mainly focuses on cactus diseases detection to early prevent the reduction of productivity (quantitatively and qualitatively) of the cereal. To do this, the researchers have used 500 unhealthy and 72 healthy cactus images. The images were enhanced, noises were removed and images were segmented to create good model using imadjust, guided filter and K-means clustering techniques respectively. These image preprocessing techniques were selected from many techniques after implementing each technique and measuring their performances. As part of creating the model, feature extraction techniques (Color histogram, Bag of features and GLCM) were applied to extract color, bag of features and texture and respectively. After testing the model applying these features, bag of features were found to be best for creating better model and they were selected as features of our model. We created our machine learning model using bag of features applying linear SVM. Other machine learning algorithms were used to train and test the model for detecting the diseases, but linear SVM was found with best performance (97.2%). In this task, 75% of each class were used for training and 25% were used for testing the model. Finally, the similarity for classification was checked using linear kernel, RBF kernel and Polynomial kernel and an average accuracy of 94% was achieved though linear kernel is the best classifying method with an accuracy of 98.951%.

Keywords: Machine learning, supervised learning, unsupervised learning, training, classification, feature, bag of features, algorithm, k-means, MSE, PNSR, and linear SVM.

1. Introduction

Machine learning is the semi-automated extraction of knowledge from data. It applies algorithms to the data using machines (computer) to provide the required knowledge and makes many more smart decisions in order to make the process successful [1]. Machine learning, as part of artificial intelligence, can also be defined as a method that computers can learn to make predictions based on data (statistical or image data). It gives computers the ability to learn without being explicitly programmed. Machine learning can be illustrated as a learning system that can distinguish spam or non-spam email messages, i.e. categorizing spam messages into spam folder and non-spam messages into non-spam container. Machine learning is frequently used in various fields, such as medical diagnosis, self driving cars, computational biology, astrophysics, public policy, stack market analysis etc. Machine learning is very important when there is shortage of skilled professionals to detect the phenomenon and assures the accuracy of the result (predicted outcome) because people may not be accurate, for example, in reading images of diseases. Generally, machine learning is preferable because [2] (1) it is much more accurate than human-crafted rules (since it is data driven) (2) humans are often incapable of expressing what they know (3) it does not need a human expert or programmer (4) it is automatic method to search for hypotheses explaining data (5) it is cheap and flexible (can apply to any learning task).

The aim of this work is to classify Cactus images as healthy or unhealthy to early detect the plant's diseases to assure the quality and quantity of its products. In this article, architectural model of the system (classifier) is proposed showing the necessary steps to achieve the goal. Therefore, in the following sections, image enhancement, noise removal, segmentation, features extraction and classification of the plants are done using different techniques for each task.

2. Architectural Design of the Proposed System

System architecture is the high level description of components of a system and their communication or it is building and/or designing the system structures [3]. As it can be seen in Fig 1, the architecture of the proposed system has two phases, namely, model creation (training) and testing phases. The model creation phase encompasses 'Beles' image acquisition, enhancement (contrast adjustment), noise removal, segmentation, feature extraction and model creation. The testing phase also includes new image acquisition, enhancement, noise removal, segmentation, feature extraction and system evaluation. This means that the same procedures are done in both phases except in creating and testing the model. After the model is created, the features are also created from the testing phase and these features are compared with the features of the model to classify the image into its respective class. This system classifies the image data into two classes (Diseased or healthy) as it can be seen in the implementation part.

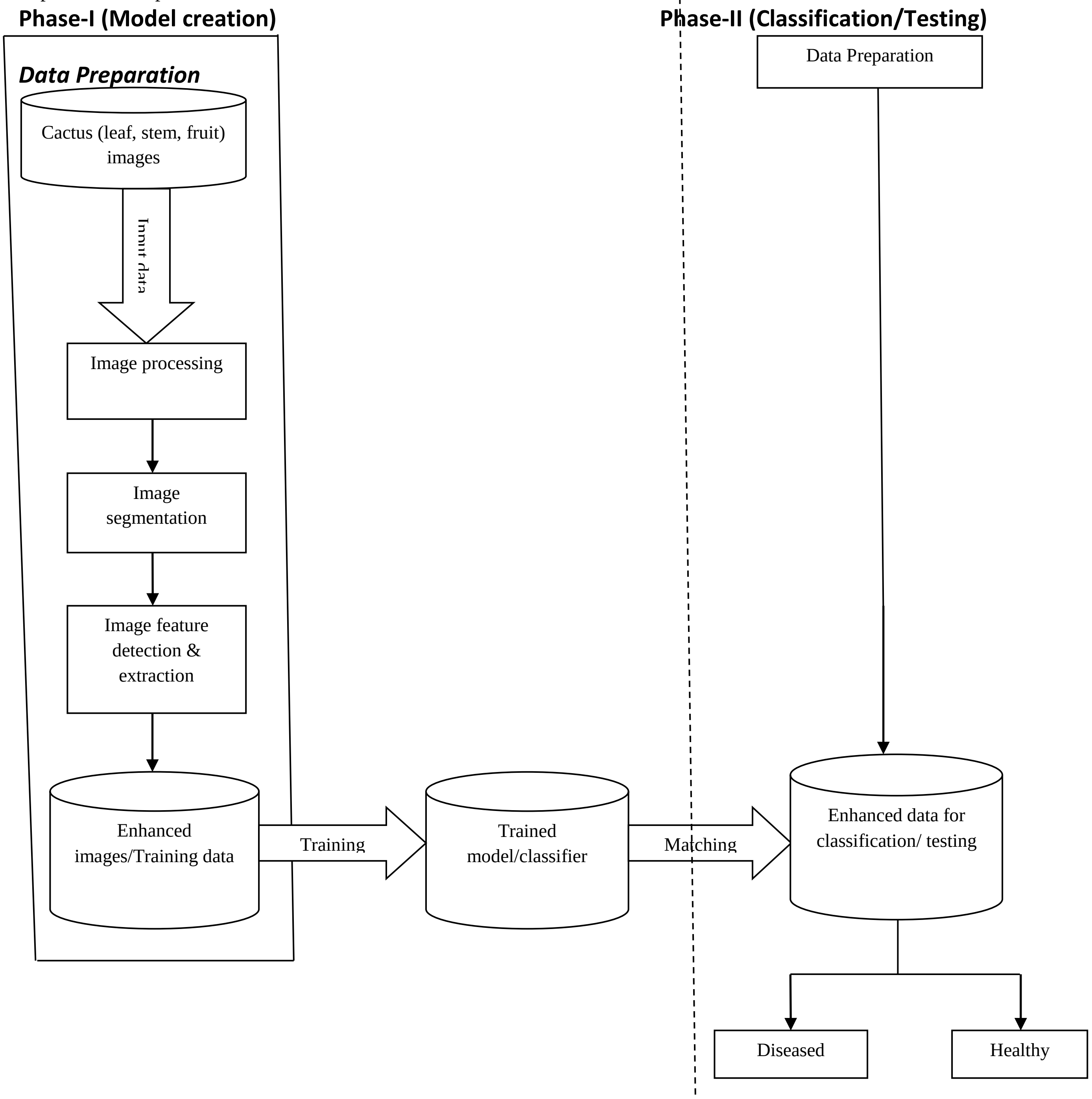

Phase-I (Model creation)
Data Preparation
Cactus (leaf, stem, fruit) images
Input data
Image processing
Image segmentation
Image feature detection & extraction
Enhanced images/Training data
Training
Trained model/classifier
Matching
Phase-II (Classification/Testing)
Data Preparation
Enhanced data for classification/ testing
Diseased
Healthy

Fig 1: Architectural Model of the proposed system

3. Prototype and Experimental Results

This section discuses about the materials and tools used and the experimental results of the prototype.

3.1 Materials and tools used

To develop the prototype of the system, we have used a 2 cores Dell computer with 4 GB RAM, 2.50 GHZ processing speed and 1000 GB hard disk. We have also used standard mobile camera to capture the images taken from the field. In this prototype development, 572 images, of which 500 images were unhealthy and 72 images were healthy, were used. The 408 unhealthy and 72 healthy images were captured by the mobile camera from the field and the remaining ones were taken from the web. To process (capture, enhance, de-noise, segment and extract features) the images used in this prototype development, windows platform MATLAB R2015a was used because MATLAB is the state of the art for image processing and machine learning [4].

3.2 Prototype Development and Results

The prototype of the machine learning technique was done by employing image enhancement, image de-noising, segmentation and the images' features extraction. To do this, image enhancement was employed over the above mentioned images to adjust the brightness of the images using imadjust() matlab function from their directory as it can be seen in the sample below.

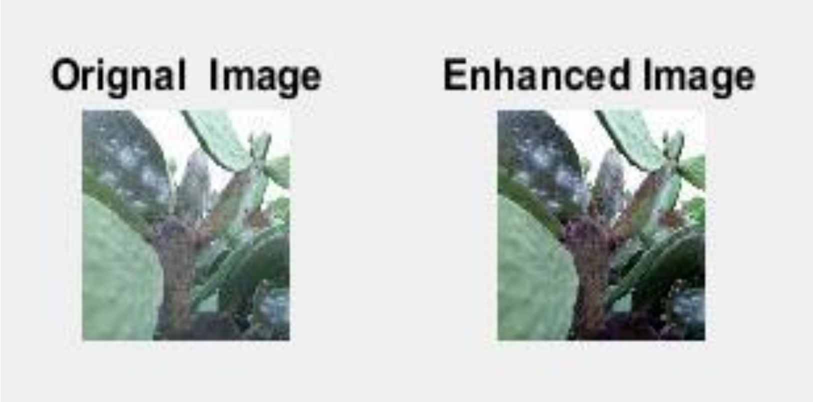


Fig 2: Cactus image enhancement [16].

After enhancing all the images, it was necessary to remove noises from the images not to affect the classification result of the model because noise is unnecessary data that can be found in images. Although there are many types of image noises, the most common are Gaussian, salt & pepper, speckle and Poisson noises. There are also different techniques to remove image noises, but we have used (implemented) mean, median, guided, Gaussian, adaptive and linear filters to remove the above noted noises from our images. The performance of the noise removal techniques varies with data. For our purpose, we implemented all the above techniques and applied on each noise. Since their performance varies with data, we used MSE (mean squared error) and PSNR (Peak-signal to noise ratio) to select the best image noise remover from the noise removing techniques.

The greater the PSNR value the better is the filter [5] and the smaller the MSE value the better is the filtering technique [6]. Therefore, as it can be seen from the charts below, guided filter has smallest MSE and largest PSNR values in removing each type of noise. So, this filtering technique is selected to be the best filter for cactus noise image.

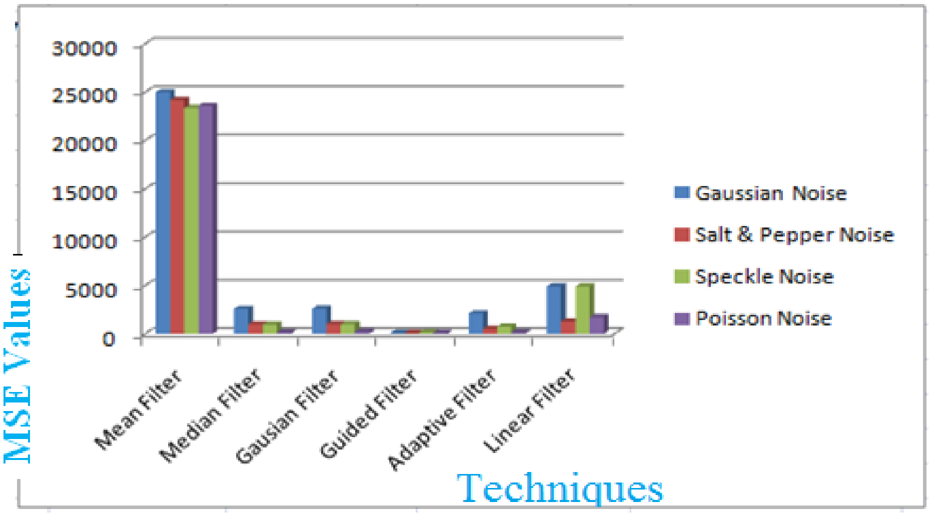


Fig 3: MSE values distribution in chart [16].

The above figure (fig 3) shows that MSE (Mean squared error) is used to select the best noise removing technique. It is shown that the Guided filter has the smallest MSE value and mean filter has the largest MSE value. Since it is true that the smaller the MSE value the best noise remover it is, Guided filter has performed good to remove the noise in our data.

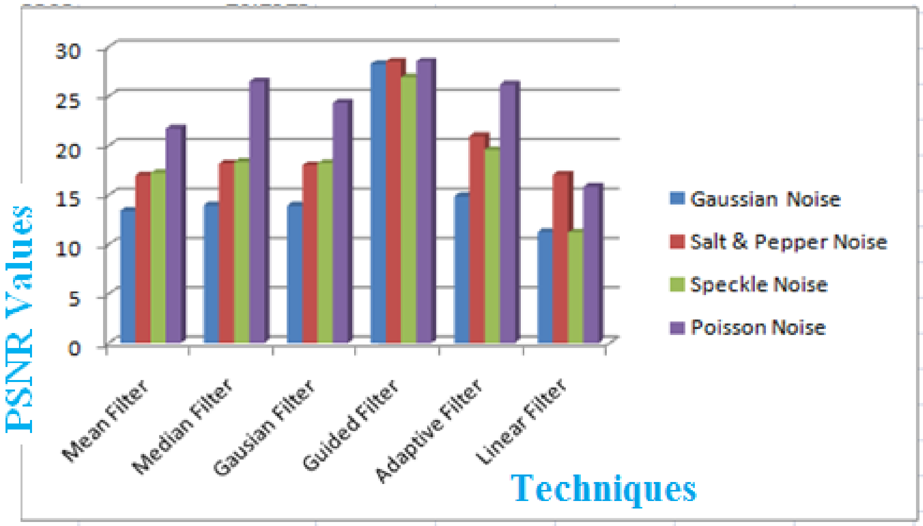


Fig 4: PSNR values distribution in chart [16].

Figure 4 illustrates the values of each noise filtering technique using PSNR (Peak-signal to noise ratio) measure. In this figure, it is shown that linear filter has the lowest PSNR values for removing each noise and Guided filter has highest PSNR values to remove each noise. Therefore, because it is true that the higher the value of the MSN value the better is the filter, Guided filter is again a best de-noising technique for our data.

After noises are removed from images, the next step is to segment an image into groups of pixels. Image segmentation is the process of partitioning an image into important constituents (pixels or regions) to change the representation of the image into its meaningful and easier regions that can collectively cover the entire image [7].

To do image segmentation, some features, such as edge, color, shape, texture, etc are considered. Therefore, we have analyzed and identified *Region Based*, *Edge Based*, *Feature Based and Model Based Image Segmentation Techniques [8, 9, 10]*. There are also different edge based segmentation techniques (operations), namely, *Roberts Detection*, *Prewitt Detection, Sobel Detector, Canny Detection and Laplacian Detection.*

There must be a quantitative measurement so that better segmentation technique will be selected. Therefore, for this purpose, correlation and structural similarity methods are implemented and color based K-means clustering (Feature Based Image segmentation) is selected as a better segmentation technique for our data as it can be seen from the figure below.

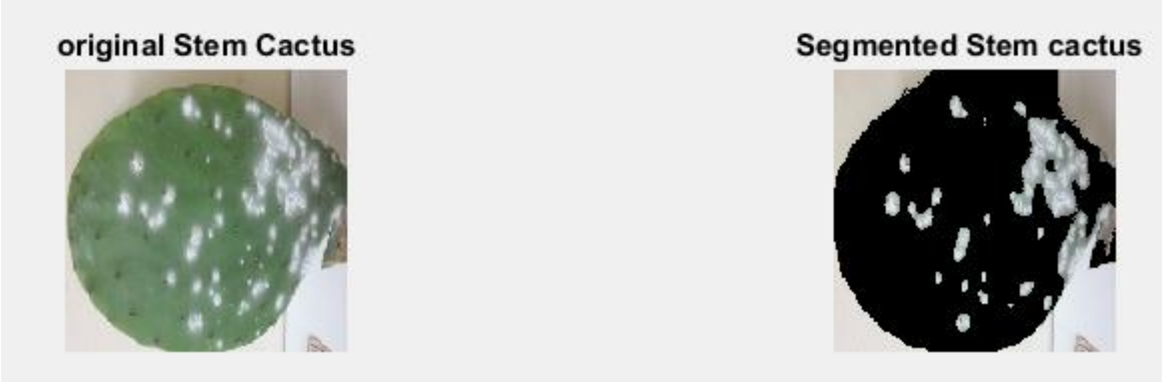


Fig 5: Feature Based (k-means clustering) Image segmentation [16]

3.2.1 Features and Feature Extraction Techniques

Feature is a characteristics or piece of information that describes an image or part of the image [11]. Image features include shape, color, point, line, edge, corner, circle, ellipse, blobs, histogram, texture and spectral features. In order to use the image for further processing, suitable features must be detected from where they are in the image. The selected features are used for matching purpose. Therefore, features are detected in two steps, namely, detection and extraction. Image features are very important in both training and testing models. After the above (image enhancement, noise removal and segmentation) mentioned image processing tasks are done, features are detected and extracted to create the model using image extraction techniques and features. The same process is also conducted in testing the model. During testing, the extracted features are compared with the features where the model is created. Based on the similarity of the features, the new image is classified as '*Diseased*' or '*Healthy*'. To create (train) and test the model, we have extracted three features (color, bag of features and texture) of cactus image using color histogram, bag of feature and GLCM (Gray Level Coocurrence Matrices) feature extraction techniques.

Bag of features method is used to extract image features (characteristics) or represents images as orderless collections of local features. It mainly depends on the orderless collection of quantized local image descriptors (extracted feature vectors) to classify images, detect objects, retrieve images and visualize robots [12]. This is to mean that this method extracts image descriptors, observes the descriptors' frequencies, clusters them into k-clusters using k-means clustering technique and uses these features for creating and testing the model. Image descriptors can be dense, color, texture and shape descriptors [13].

Color features are the most import general (as image features are classified as general and domain specific features) features that are commonly used in image retrieval and classification. It is independent of image size and orientation. Image color features are extracted using color histogram to crate and test a machine learning model [14]. Color histogram tells us the probability of each pixel of the image to be of one of the extracted colors. Computationally, color histogram for a given image is defined as a vector:

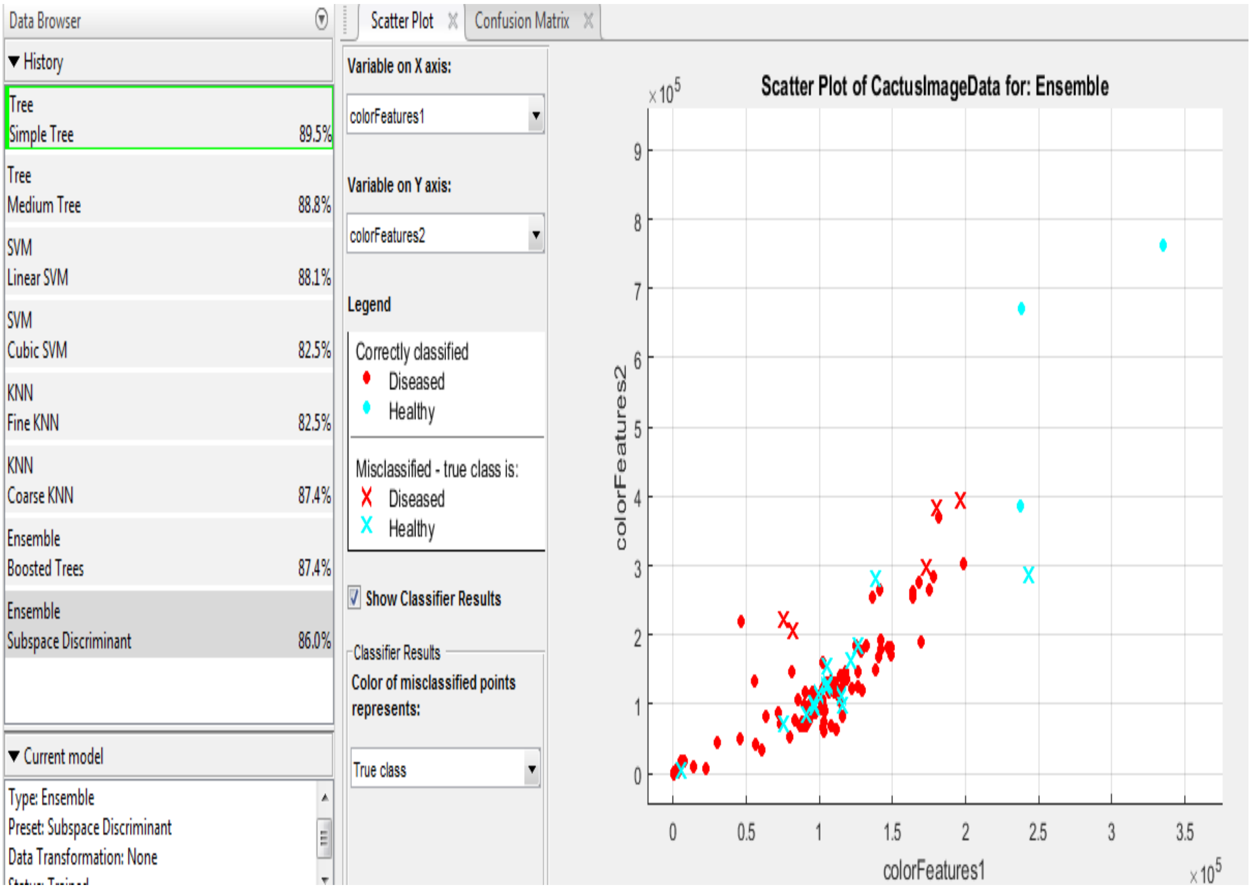
$$H = \{H[0], H[1], H[2], \dots, H[i], \dots, H[N]\}.$$

where i is a color in the color histogram and corresponds to a sub-cube in the RGB color space, $H[i]$ is the number of pixels in color i in that image, and N is the number of colors in the adopted color model.

For our work, we considered RGB color space (because the images are RGB images) and took 30 color features for each color in each image and a total of 90 color features.

The other important image features extraction technique is GLCM. This method is used to extract texture features from an image. It is the way of extracting second order statistical texture features [15]. Since the input images in this method are grayscale images, the number of rows and columns is equal to the number of gray levels, G, in the image. Although there are many GLCM measures, we have implemented autocorrelation, contrast, correlation, Cluster Prominence, Cluster Shade, Dissimilarity, Energy, Entropy, Homogeneity, Maximum probability, Variance, Sum average, Sum variance, Sum entropy, Difference variance, Difference entropy, Information measure of correlation, maximal correlation coefficient, Inverse difference (INV) , Inverse difference normalized (INN) and Inverse difference moment normalized

In the following sections, we have used color, texture and bag of features to create our model and classify the plants (images) into two classes, namely, 'Diseased' and 'Healthy'. For doing this, we have used 500 unhealthy images (75% for training and 25% for testing) and 72 healthy images (75% for training and 25% for testing). We extracted color, texture and bag of features of each image applying the above mentioned feature extraction techniques as it can be seen from the following screenshots.



Data Browser

▼ History

Tree	Simple Tree	89.5%
Tree	Medium Tree	88.8%
SVM	Linear SVM	88.1%
SVM	Cubic SVM	82.5%
KNN	Fine KNN	82.5%
KNN	Coarse KNN	87.4%
Ensemble	Boosted Trees	87.4%
Ensemble	Subspace Discriminant	86.0%

▼ Current model

Type: Ensemble
Preset: Subspace Discriminant
Data Transformation: None
Status: Trained

Variable on X axis: colorFeatures1

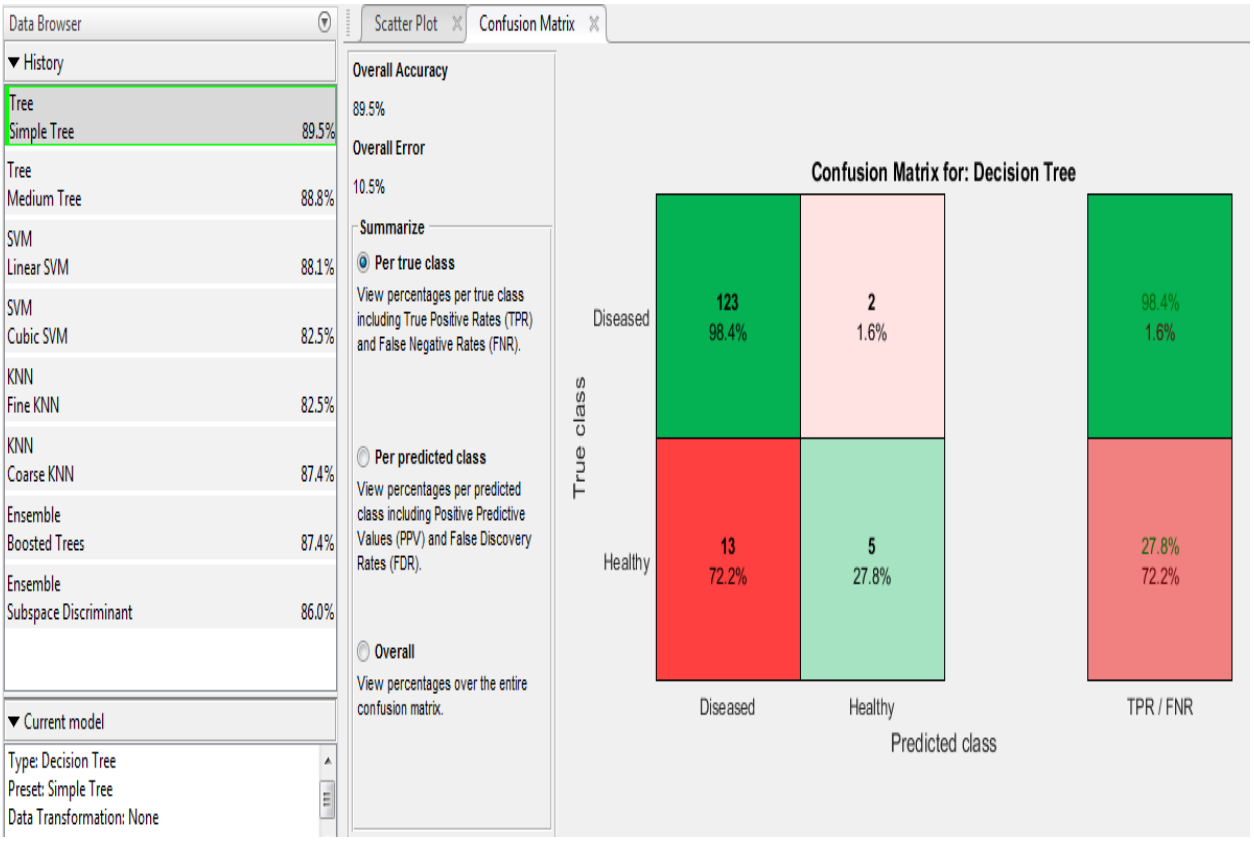
Variable on Y axis: colorFeatures2

☑ Show Classifier Results

Classifier Results
Color of misclassified points represents: True class

Fig 6: Scatter plot for color features based Cactus classification

The above scatter plot (Fig 6) demonstrates that the classifier is created using color features of 75% of the 500 diseased images and 75% of the 72 healthy images. It also shows that the model is tested by 25% of each of the image categories using simple tree, medium tree, linear SVM, cubic SVM, fine KNN, coarse KNN, bagged trees and Subspace Discriminant techniques as it can be seen from the scatter plot. Of these techniques, in this case, a simple tree is found to have with good accuracy (89.5%). The correctly classified and incorrectly classified images are shown by dot (.) and cross(x) respectively.



Model	Accuracy
Tree Simple Tree	89.5%
Tree Medium Tree	88.8%
SVM Linear SVM	88.1%
SVM Cubic SVM	82.5%
KNN Fine KNN	82.5%
KNN Coarse KNN	87.4%
Ensemble Boosted Trees	87.4%
Ensemble Subspace Discriminant	86.0%

Type: Decision Tree
Preset: Simple Tree
Data Transformation: None

Overall Accuracy
89.5%
Overall Error
10.5%

Summarize

● Per true class
View percentages per true class including True Positive Rates (TPR) and False Negative Rates (FNR).

○ Per predicted class
View percentages per predicted class including Positive Predictive Values (PPV) and False Discovery Rates (FDR).

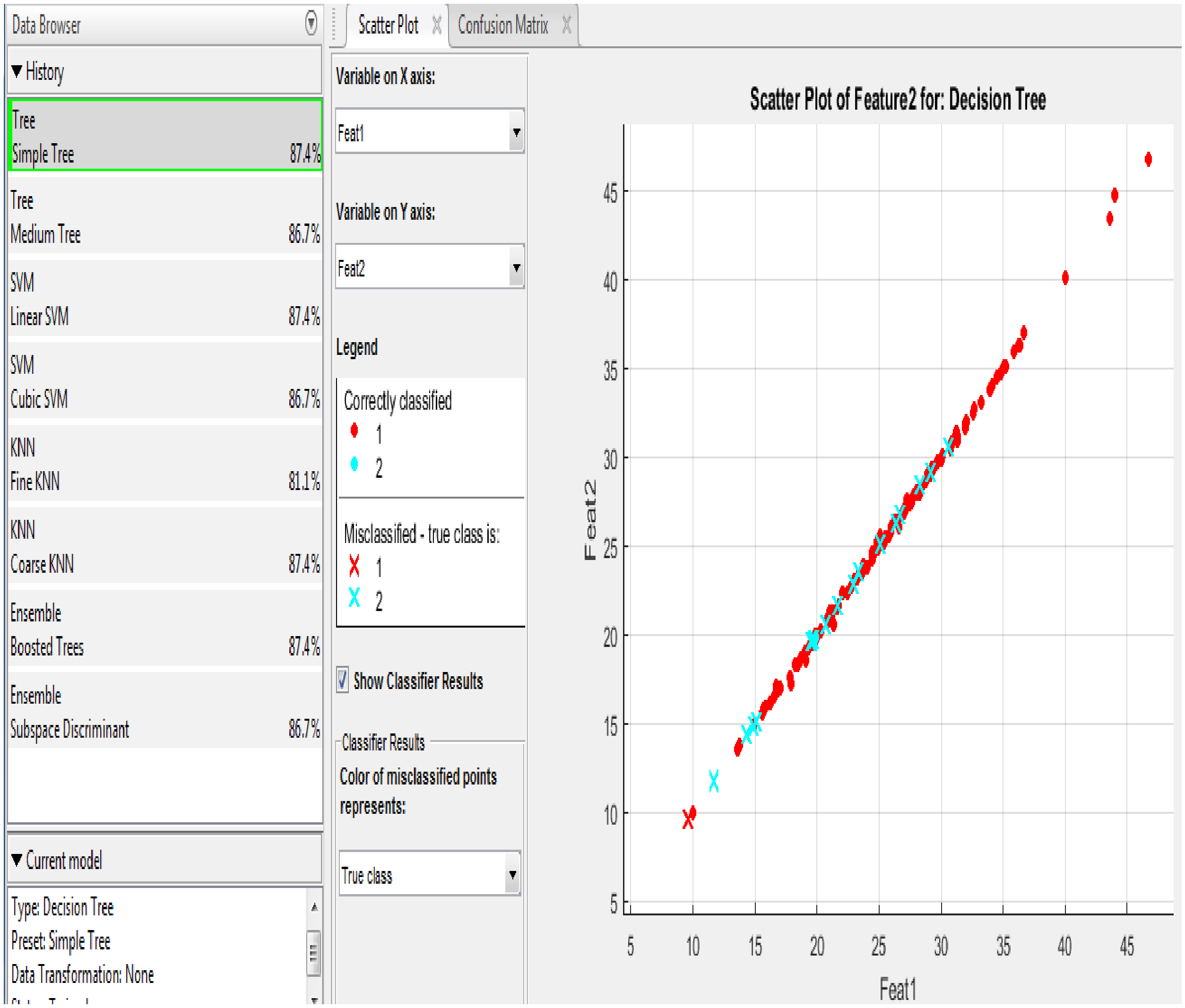
○ Overall
View percentages over the entire confusion matrix.

Confusion Matrix for: Decision Tree

True class \ Predicted class	Diseased	Healthy	TPR / FNR
Diseased	123 98.4%	2 1.6%	98.4% 1.6%
Healthy	13 72.2%	5 27.8%	27.8% 72.2%

Fig 7: Confusion matrix for color features based Cactus classification

The above confusion matrix (Fig 7) depicts that of the 125 (25%) diseased images 123 are correctly classified and 2 images are misclassified while using Simple tree classifier. In the other class, 13 healthy images are misclassified and 5 images are correctly classified using the same technique. Therefore, the accuracy of this model to correctly classify the images into predicted ('Diseased' and 'Healthy') classes is 98.4% and 27.8% respectively. To train and test the model simple tree, medium tree, linear SVM, cubic SVM, fine KNN, coarse KNN, boosted trees and Subspace Discriminant are used and have different accuracies as it can be seen from the figure. Of these techniques, in this case, a simple tree is found to have with good accuracy (89.5%).



Tree	Simple Tree	87.4%
Tree	Medium Tree	86.7%
SVM	Linear SVM	87.4%
SVM	Cubic SVM	86.7%
KNN	Fine KNN	81.1%
KNN	Coarse KNN	87.4%
Ensemble	Boosted Trees	87.4%
Ensemble	Subspace Discriminant	86.7%

Variable on X axis: Feat1

Variable on Y axis: Feat2

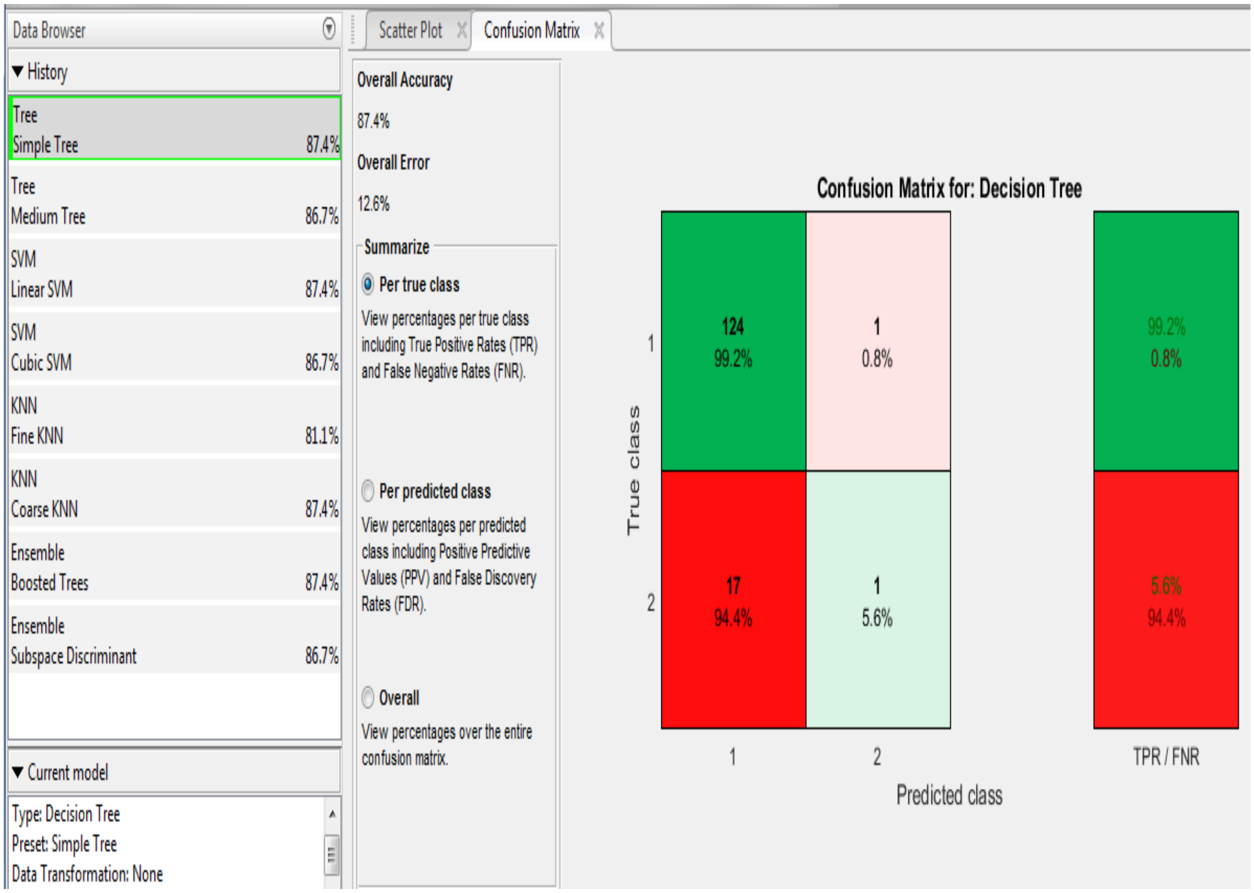
☑ Show Classifier Results

Color of misclassified points represents: True class

Type: Decision Tree
Preset: Simple Tree
Data Transformation: None

Fig 8: Scatter plot for GLCM features based Cactus classification

The above screenshot (Fig 8) is a scatter plot that demonstrates a classifier created using GLCM features of 75% of the 500 diseased images and 75% of the 72 healthy images. It also shows that the model is tested by 25% of each of the image categories using simple tree, medium tree, linear SVM, cubic SVM, fine KNN, coarse KNN, bagged trees and Subspace Discriminant techniques as it can be seen from the scatter plot. Of these techniques, in this case, a simple tree is found to have with good accuracy (87.4%). The correctly classified and incorrectly classified images are shown by dot (.) and cross(x) respectively.



Data Browser

History

Tree
Simple Tree 87.4%

Tree
Medium Tree 86.7%

SVM
Linear SVM 87.4%

SVM
Cubic SVM 86.7%

KNN
Fine KNN 81.1%

KNN
Coarse KNN 87.4%

Ensemble
Boosted Trees 87.4%

Ensemble
Subspace Discriminant 86.7%

Current model

Type: Decision Tree
Preset: Simple Tree
Data Transformation: None

Overall Accuracy
87.4%

Overall Error
12.6%

Summarize

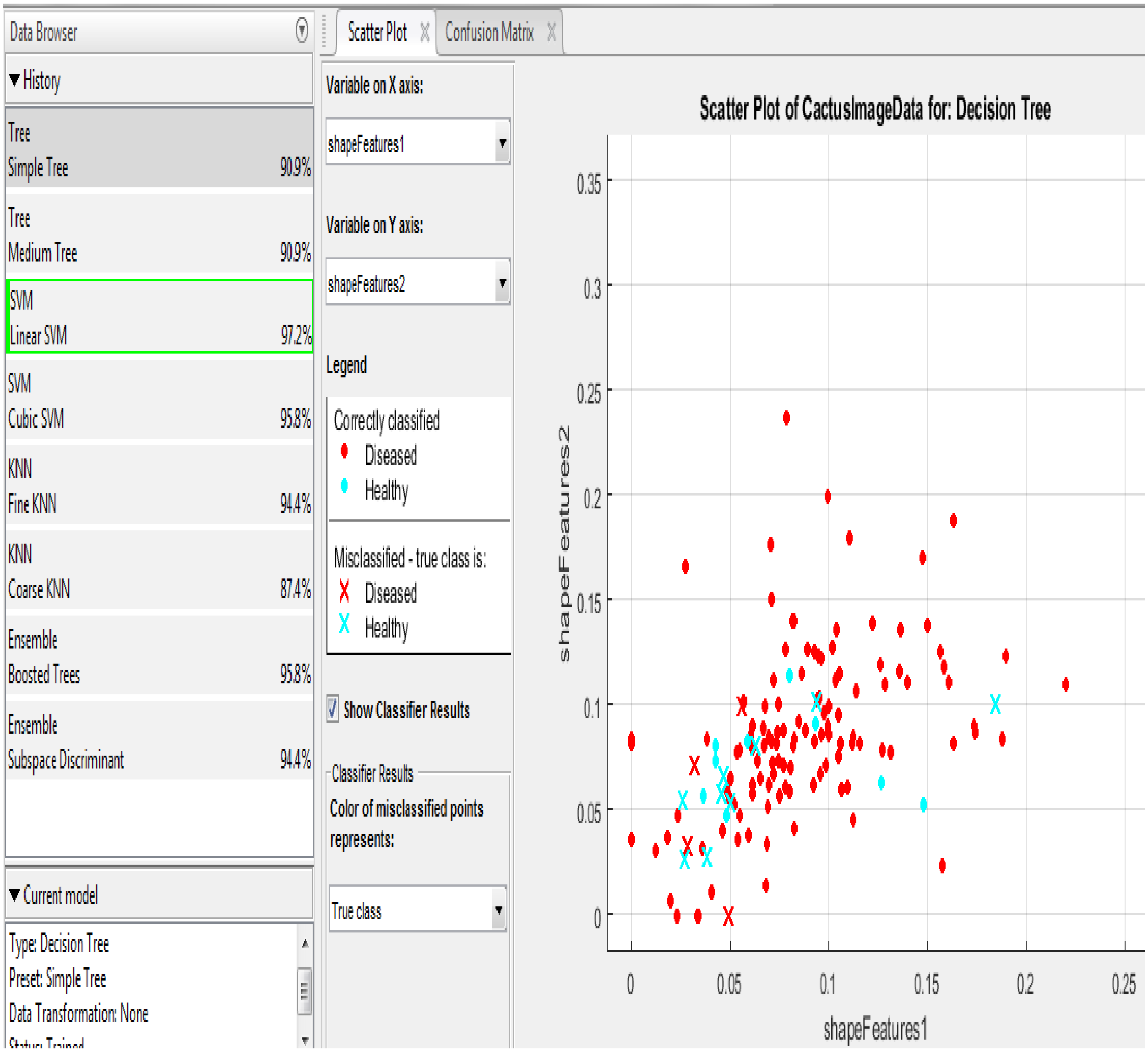
● Per true class
View percentages per true class including True Positive Rates (TPR) and False Negative Rates (FNR).

○ Per predicted class
View percentages per predicted class including Positive Predictive Values (PPV) and False Discovery Rates (FDR).

○ Overall
View percentages over the entire confusion matrix.

Fig 9: Confusion matrix for GLCM features based Cactus classification

Fig 9 is a confusion matrix that depicts a classifier that correctly categorized 124 diseased images into their predicted class and misclassified one image into ‘healthy’ class. It also correctly classified one image into its predicted class (Healthy) and misclassified 17 healthy images into ‘Diseased’ class. Therefore, the model has the accuracy of 99.2 and 5.6 to correctly classify the used images into ‘Diseased’ and ‘Healthy’ classes respectively. To train and test the model simple tree, medium tree, linear SVM, cubic SVM, fine KNN, coarse KNN, boosted trees and Subspace Discriminant are used and have different accuracies as it can be seen from the figure. Of these techniques, in this case, a simple tree is found to have with good accuracy (87.4%).



Data Browser

▼ History

Tree Simple Tree	90.9%
Tree Medium Tree	90.9%
SVM Linear SVM	97.2%
SVM Cubic SVM	95.8%
KNN Fine KNN	94.4%
KNN Coarse KNN	87.4%
Ensemble Boosted Trees	95.8%
Ensemble Subspace Discriminant	94.4%

▼ Current model

Type: Decision Tree
Preset: Simple Tree
Data Transformation: None
Status: Trained

Variable on X axis: shapeFeatures1

Variable on Y axis: shapeFeatures2

Legend

Correctly classified
● Diseased
● Healthy

Misclassified - true class is:
X Diseased
X Healthy

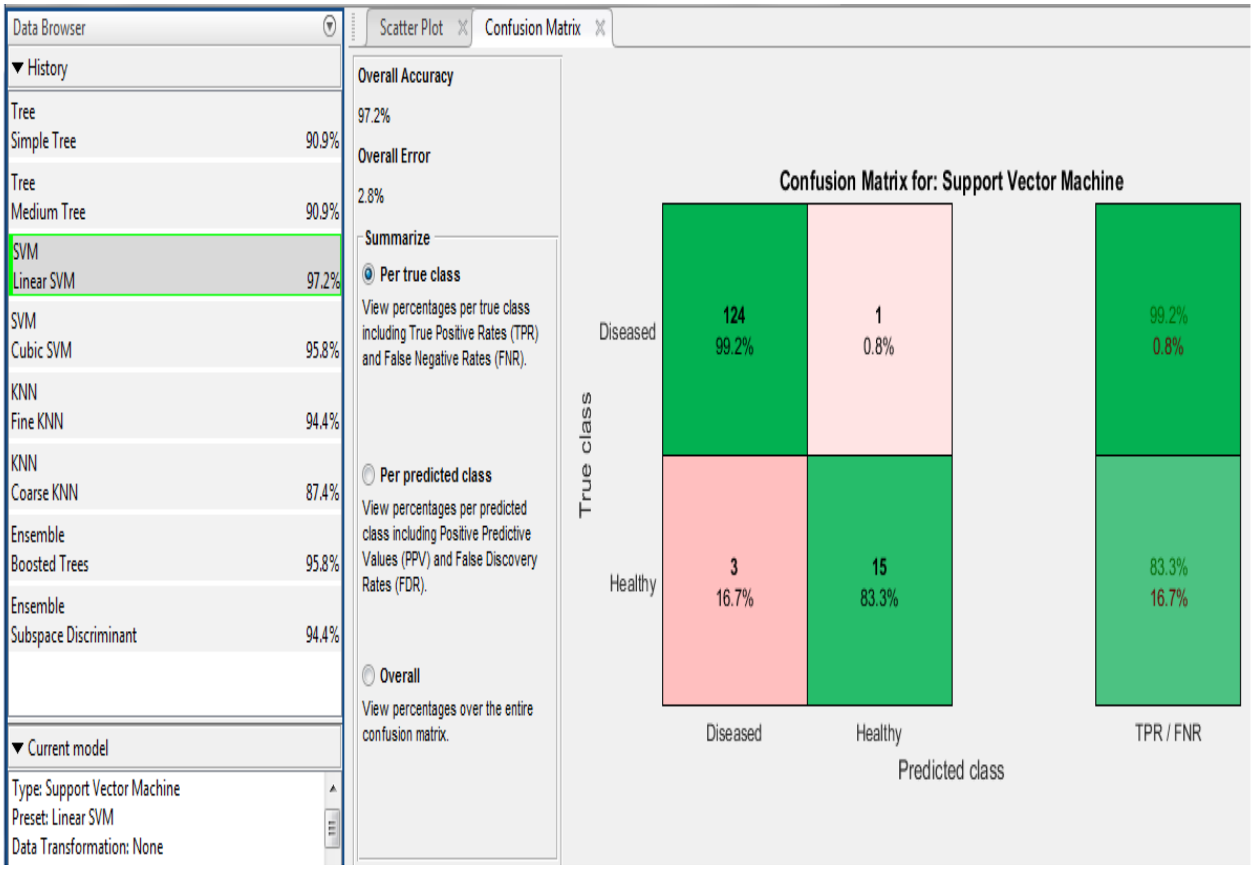
☑ Show Classifier Results

Classifier Results

Color of misclassified points represents: True class

Fig 10: Scatter plot for bag of features based Cactus classification

The above scatter plot (Fig 10) demonstrates that the classifier is created using bag of features of 75% of the 500 diseased images and 75% of the 72 healthy images. It also shows that the model is tested by 25% of each of the image categories using simple tree, medium tree, linear SVM, cubic SVM, fine KNN, coarse KNN, bagged trees and Subspace Discriminant techniques as it can be seen from the scatter plot. Of these techniques, in this case, a *linear SVM* is found to have with good accuracy (97.2%). The correctly classified and incorrectly classified images are shown by dot (.) and cross(x) respectively.



Tree Simple Tree	90.9%
Tree Medium Tree	90.9%
SVM Linear SVM	97.2%
SVM Cubic SVM	95.8%
KNN Fine KNN	94.4%
KNN Coarse KNN	87.4%
Ensemble Boosted Trees	95.8%
Ensemble Subspace Discriminant	94.4%

Type: Support Vector Machine
Preset: Linear SVM
Data Transformation: None

Overall Accuracy
97.2%
Overall Error
2.8%

Summarize

● Per true class
View percentages per true class including True Positive Rates (TPR) and False Negative Rates (FNR).

○ Per predicted class
View percentages per predicted class including Positive Predictive Values (PPV) and False Discovery Rates (FDR).

○ Overall
View percentages over the entire confusion matrix.

Confusion Matrix for: Support Vector Machine

True class \ Predicted class	Diseased	Healthy	TPR / FNR
Diseased	124 99.2%	1 0.8%	99.2% 0.8%
Healthy	3 16.7%	15 83.3%	83.3% 16.7%

Fig 11: Confusion matrix for bag of features based Cactus classification

Fig 11 shows that bag of features are used to train and test the model. The overall accuracy of the model using linear SVM is found to be 97.2% and the accuracy of the model to correctly classify the 'Diseased' images into the predicted class is 99.2% and that of the 'Healthy' images is 83.3%. This is to mean that, of the 125 unhealthy images, 124 are correctly classified into their predicted class ('Diseased') and one unhealthy image is incorrectly classified into 'Healthy' class. Of the 18 healthy images, 15 images are correctly classified into 'Healthy' class and 3 images are misclassified as unhealthy images. Therefore, the accuracy of the model to correctly classify into 'Diseased' and 'Healthy' classes is 99.2% and 83.3% respectively using linear SVM. To train and test the model simple tree, medium tree, linear SVM, cubic SVM, fine KNN, coarse KNN, boosted trees and Subspace Discriminant are used and have different accuracies as it can be seen from the figure.

Summary:

In this scenario (figures 6-11), average accuracies of 86.5, 93.4 and 86.4 are found for color, bag of features and texture (GLCM) features respectively. The misclassification of the images into unpredicted classes is also high while applying color and GLCM image features. As a result, it is found that bag of features have good classification power than the other two features. Besides, if we look at the confusion matrix of each classifier (in the screenshots above), the images are better classified into their predicted classes in the model created by bag of features. Therefore, it is concluded that bag of features have good classifying power than the other features using the matlab app. Of the applied classifiers for training and testing the model using these features, linear support vector machine (linear SVM) was found to be the best classifier with the overall accuracy of 97.2% as it can be seen from figures 10 & 11 and the following line graph.

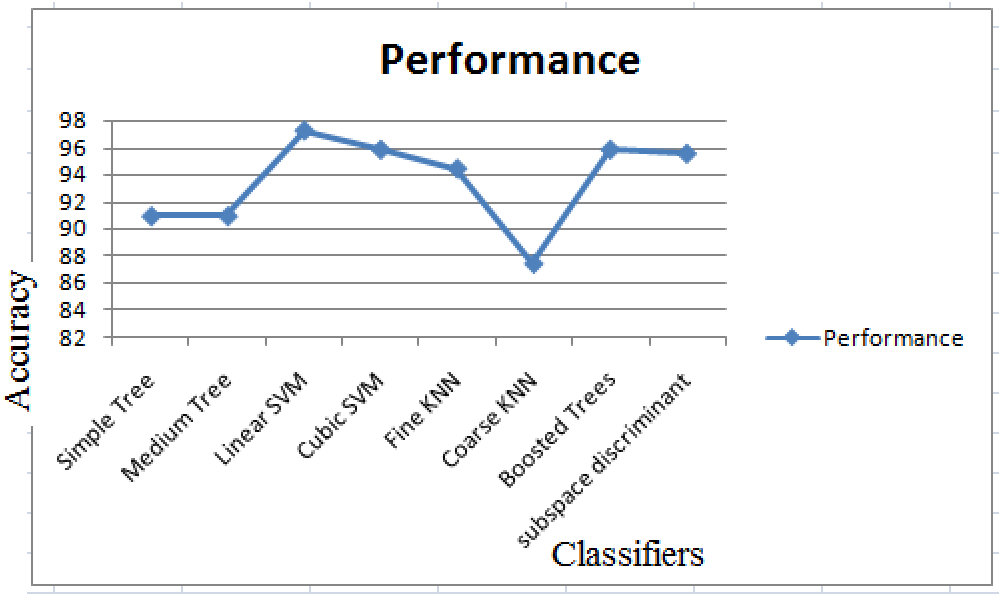
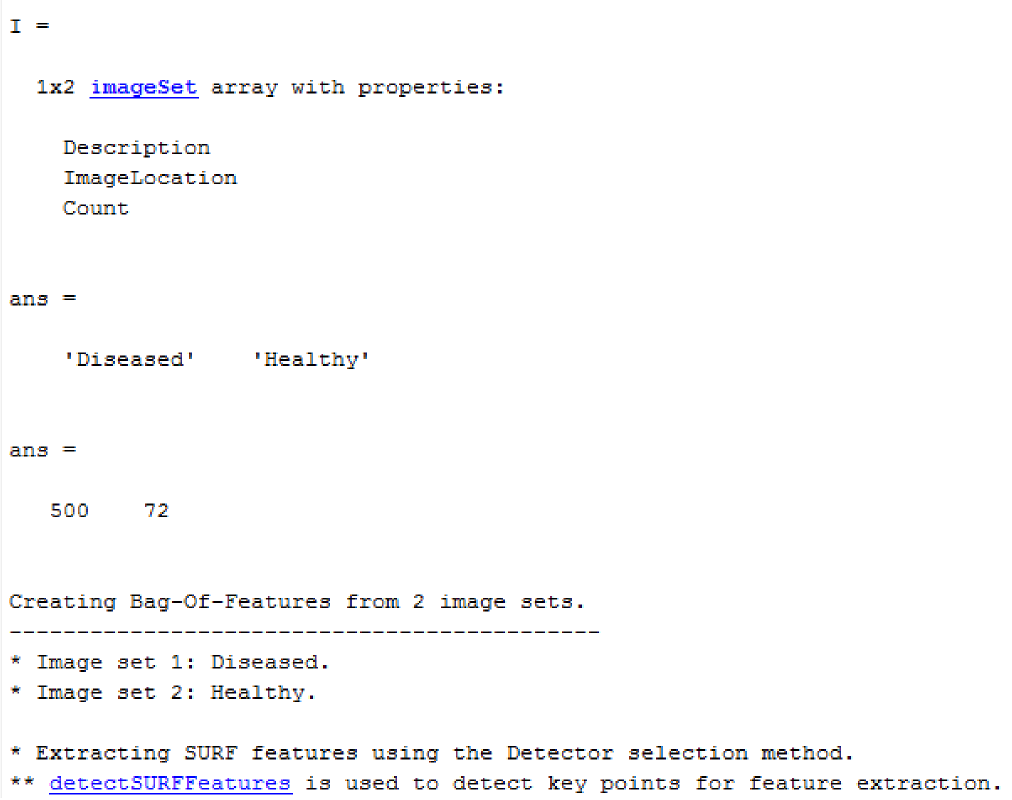
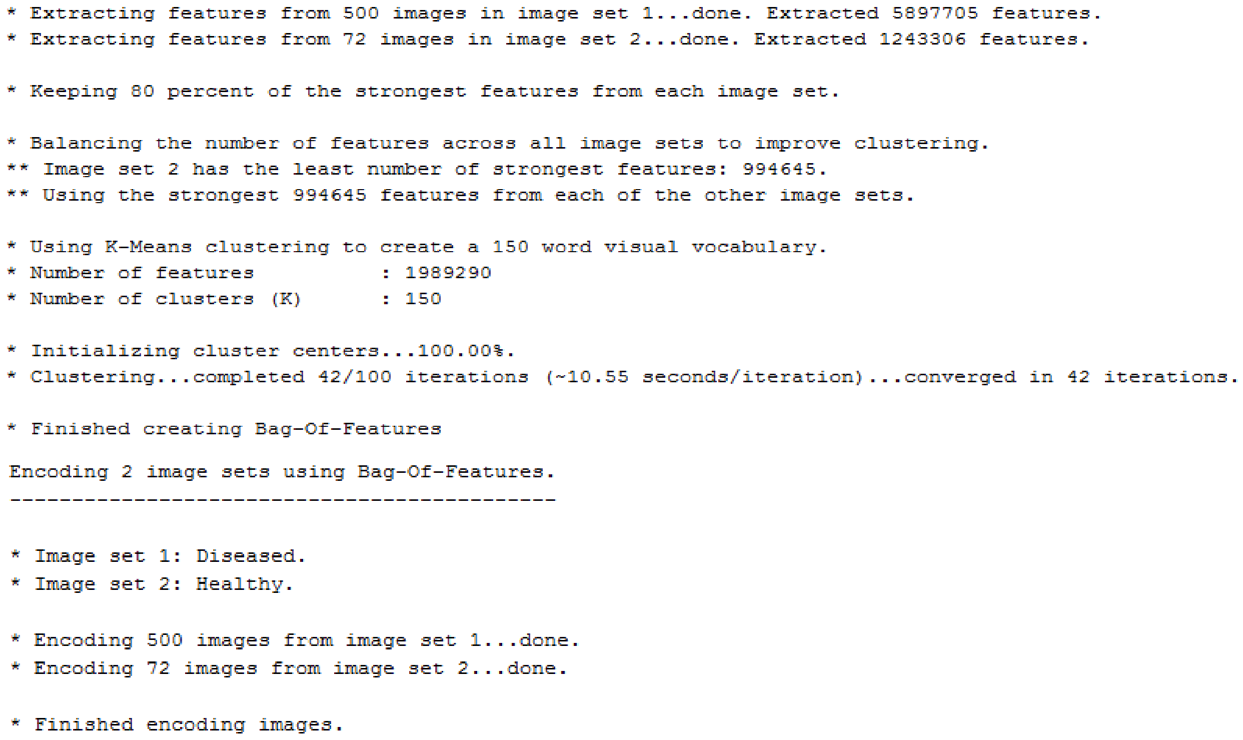


Fig 12: Performance of classifiers

After we identified that *linear SVM* has better classification performance than the other algorithms used using the matlab app, we have extracted the features and labels of each image using bag of features technique and 80% of the features were used for training and testing the model as it can be seen in the following screenshot.



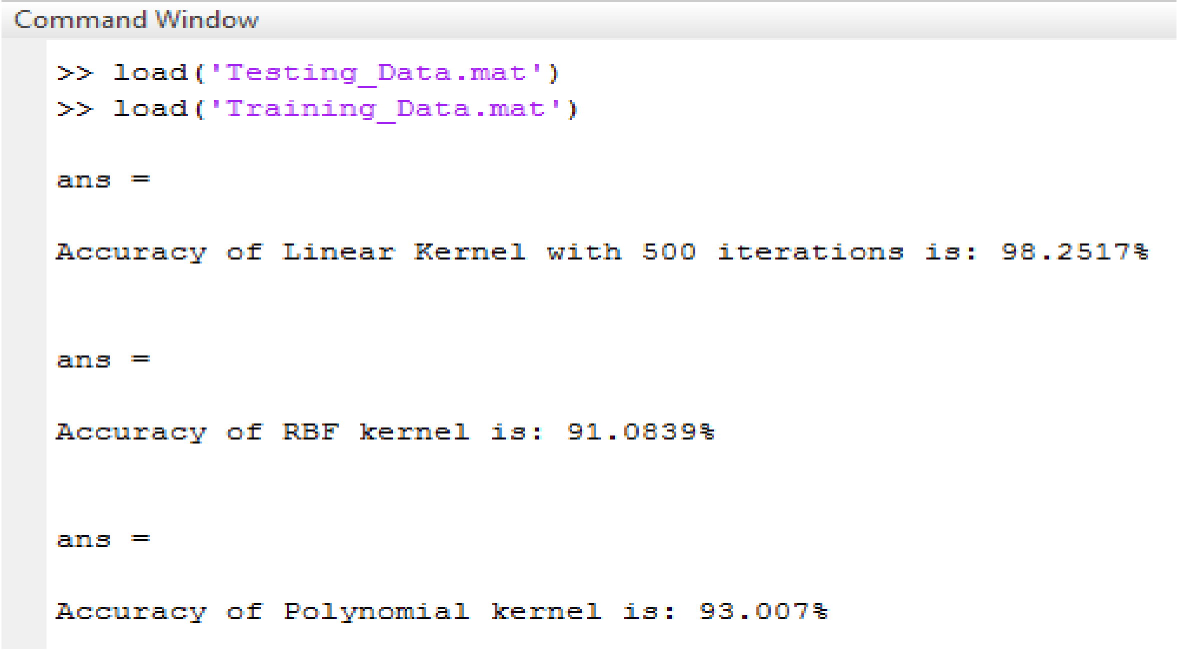
```
I =

  1x2 imageSet array with properties:

    Description
    ImageLocation
    Count


ans =

    'Diseased'    'Healthy'


ans =

   500    72


Creating Bag-Of-Features from 2 image sets.
--------------------------------------------
* Image set 1: Diseased.
* Image set 2: Healthy.

* Extracting SURF features using the Detector selection method.
** detectSURFFeatures is used to detect key points for feature extraction.
```



```
* Extracting features from 500 images in image set 1...done. Extracted 5897705 features.
* Extracting features from 72 images in image set 2...done. Extracted 1243306 features.

* Keeping 80 percent of the strongest features from each image set.

* Balancing the number of features across all image sets to improve clustering.
** Image set 2 has the least number of strongest features: 994645.
** Using the strongest 994645 features from each of the other image sets.

* Using K-Means clustering to create a 150 word visual vocabulary.
* Number of features          : 1989290
* Number of clusters (K)      : 150

* Initializing cluster centers...100.00%.
* Clustering...completed 42/100 iterations (~10.55 seconds/iteration)...converged in 42 iterations.

* Finished creating Bag-Of-Features

Encoding 2 image sets using Bag-Of-Features.
--------------------------------------------

* Image set 1: Diseased.
* Image set 2: Healthy.

* Encoding 500 images from image set 1...done.
* Encoding 72 images from image set 2...done.

* Finished encoding images.
```

Fig 13: Feature extraction using bag of features

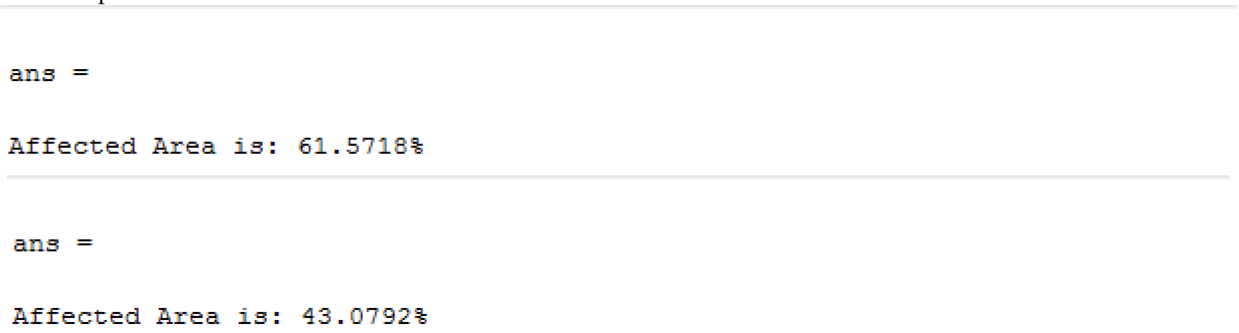
After the image features are extracted, we have written a program that created training (Training_Data.mat) and testing (Testing_Data.mat) sets having image features and labels each. The model is trained using Training_Data.mat (75% of the features) being in the matlab command window using linear SVM. The created model was again tested using Testing_Data.mat (25% of the features) without using the matlab app using linear SVM. To classify the testing dataset into predicted classes and see how much the model meets the goal, we applied linear, RBF and polynomial kernels. As it can be seen from Fig 14, the accuracy of the each method is different. However, linear kernel was found to be with a good accuracy (98.2517%).



```
>> load('Testing_Data.mat')
>> load('Training_Data.mat')

ans =

Accuracy of Linear Kernel with 500 iterations is: 98.2517%


ans =

Accuracy of RBF kernel is: 91.0839%


ans =

Accuracy of Polynomial kernel is: 93.007%
```

Fig 14: Performance of the model

Besides classification, the model can also determine the percentage of the diseased portion of the image (plant) as it can be seen in the sample screenshot below.



```
ans =

Affected Area is: 61.5718%
```

```
ans =

Affected Area is: 43.0792%
```

Fig 15: Determining affected portion of the unhealthy cactus

4. Concluding Remarks

Major challenge in machine learning is to have well processed (quality) data. Therefore, in this work, image processing (brightness enhancement, de-noising and segmentation) is intensively done using different techniques to select the best technique. Hence, imadjust (for image enhancement), guided filter (for noise filtering) and color based K-means (for image segmentation) image processing techniques were selected based on their better performance and used in the model creation and testing. As part of the model creation and testing process, color, texture and bag of features were extracted using color histogram, GLCM and Bag of features techniques respectively to extract each feature. To determine which features are most important for creating and testing a model for this specific data (image), model creation and testing using different classifying algorithms was done in matlab app. Hence, bag of features were found with better model creation performance when the model is tested. As a result, bag of features were preferable for creating and testing our model and we used them by dividing them as training and testing features in our model. Since it has better performance (97.2%), linear SVM was selected as a classifier to create and test the model. Finally, the similarity for classification was checked using linear kernel, RBF kernel and Polynomial kernel and an average accuracy of 94% was achieved though linear kernel is the best classifying method. To summarize*:*

- The used images were enhanced to have good brightness.
- Noise removal techniques are implemented and compared.
- Different image segmentation techniques are implemented and compared.
- Image features are extracted and compared to select the best ones.
- We created our color histogram to extract color features.
- GLCM features are extracted and used for our model.
- It is found that bag of features have best classifying power.
- A comparison is done among the results of the different features.

References

[1]. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani: "An Introduction to Statistical Learning with Applications in R, Springer Texts in Statistics, 2013.

[2]. http://www.cs.princeton.edu/~schapire/talks/picasso-minicourse.pdf:"Machine Learning Algorithms for Classification", accessed on 22-04-2019 at 2:00 PM.

[3]. https://www.tutorialspoint.com/software_architecture_design/architecture_models.htm: "Software Architecture and Design Architecture Models", accessed on 03-03-2019 at 8:00 PM.

[4]. R. William Lewis, Computing Consultant, "Introduction to Scientific Computing with MATLAB", SAW Training Course, MATLAB R2015a.

[5]. Anna Fabijańska, Dominik Sankowski: "Image Noise Removal – The New Approach", International Journal of Computer Aided Design System in Microelectronics, February 2007.

[6]. R. Srinivas, Satarupa Panda: "Performance Analysis of Various Filters for Image Noise Removal in Different Noise Environment", International Journal of Advanced Computer Research, December 2013.

[7]. D. Palani, K. Venkatalakshmi E. Venkatraman: "Implementation & Comparison of Different Segmentation Algorithms for Medical Imaging", International Journal of Innovative Research in Science, Engineering and Technology, March 2014.

[8]. Er. Anjna, Er.Rajandeep Kaur: "Review of Image Segmentation Technique", International Journal of Advanced Research in Computer Science, May 2017.

[9]. Gajendra Singh Chandel, Ravindra Kumar, Deepika Khare, Sumita Verma: "Analysis of Image Segmentation Algorithms Using MATLAB", International Journal of Engineering Innovation & Research, 2012.

[10]. Zoltan Kato, Ting-Chuen Pong: "A Markov random field image segmentation model for color textured images", Image and Vision Computing, March 2006.

[11]. Daniel Zemene Mequanint: "Automatic Malaria Detection from Images of Microscopic Thin Blood Films", unpublished MSc thesis, March 2016.

[12]. Stephen O'hara and Bruce A. Draper: "Introduction to the Bag of Features Paradigm for Image Classification and Retrieval", First Edition, Jan 17, 2011.

[13]. www.ens-lyon.fr/LIP/Arenaire/ERVision/bof_classification_winter.pdf:"Bag-of-features for image classification", accessed on 08-05-2019 at 6:14 PM.

[14]. shodhganga.inflibnet.ac.in/bitstream/10603/44194/8/08_chapter3.pdf: "Color feature extraction", accessed on 04-05-2019 at 11:00 AM.

[15]. Fritz Albregtsen: "Statistical Texture Measures Computed from Gray Level Coocurrence Matrices", First Edition, November 5, 2008.

[16]. Hailay Beyene Berhe, Narayan A. Joshi: "Image Processing Techniques for Cactus (Beles) Diseases detection (implementation and analysis)", February 2019.

Authors' Profile

Hailay Beyene

Hailay Beyene has pursued his bachelor degree in Mekelle University in 2008 in the department of Computer Science. He has also studied his MSC degree in Computer Science in Addis Ababa University. He is now working as a lecturer in Aksum University and Pursuing his PhD in computer Science in Parul University India. His research interests are *Information retrieval, Machine learning and image processing.*