International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020

Experimental Selection of Machine learning Techniques and Image features to Detect "Cactus" Diseases

Hailay Beyene, Narayan A.Joshi.

Abstract: ***Image is a very important data in machine learning. In order to select better features, feature extraction techniques and classifiers, intensive experiments are taken place using data. In this work, best feature, feature extraction technique and machine learning classifier are selected experimentally. Hence, bag of features were the best features experimentally out of color, texture and bag of features. Of color histogram, bag of features and GLCM (Gray-level co-occurrence matrix), bag of features extraction technique is found to be the best one experimentally. Of the machine learning classifiers shown in the scatter plot and confusion matrix, linear support vector machine is selected and the achieved accuracy is 97.2%.***

Keywords: Cactus, bag of features, GLCM, Color histogram, Confusion matrix

I. INTRODUCTION

The main purpose of this work is to experimentally select machine learning and feature extraction techniques to correctly classify whether the cactus image is healthy or unhealthy. To do this, features with good classification power are also selected experimentally so that the classification will be accurate or with better performance. Before doing all these, images were acquired, enhanced (their brightness was improved); noises were removed from every image and segmentation of image pixels was performed in both phases.

II. ARCHITECTURAL DESIGN

The intention of this work is to propose a machine learning model that detects cactus plants as healthy or unhealthy to maintain the quality and quantity of the plant to get the usual benefits. To achieve an accurate classification result, different steps were performed as it is depicted in Fig 1. The proposed architecture consists of two phases, namely, Model creation and Testing. In the model creation phase, data (cactus image) is acquired, images are enhanced, important features are extracted and the model is created by training it by the extracted features. In the second phase, the same activities that are done in the model creation phase are done except the training step. In this phase, the extracted features of the new input image are compared with the features in which the model is created and classification is done if there is matching of the features.

Revised Manuscript Received on February 05, 2020.
* Correspondence Author
Hailay Beyene*. lecturer department of Computer Science Aksum University, Ethiopia
Dr. Narayan A Joshi, Professor & Head, Dharmsinh Desai University, Nadiad.

III. ALGORITHM

Although a number of algorithms were used for every activity in the architecture in Fig1, the following algorithm was used to extract bag of features and create the classifier because bag of features were found with better classification power than color and texture features.

```
Input: Segmented image
Output: Bag of features
Steps:
1.put the images in step 4 of  Image
segmentation in ImageSet of their respective
directories (subdirectories)
2.for each image I in the directory
   if I is RGB image
           extract bag of features (apply bag
of
           feature extraction technique)
     else if I is not RGB image
         change I into RGB image
    else
      continue reading an RGB image
end
3.put the features in a tabular array with
each images' labels
4.divide the features as training and
testing set in which each set contains
features and their labels.
5.use  SVM and train the model by training
set
6.end
```

Algorithm 1: Bag of features feature extraction and Model creation

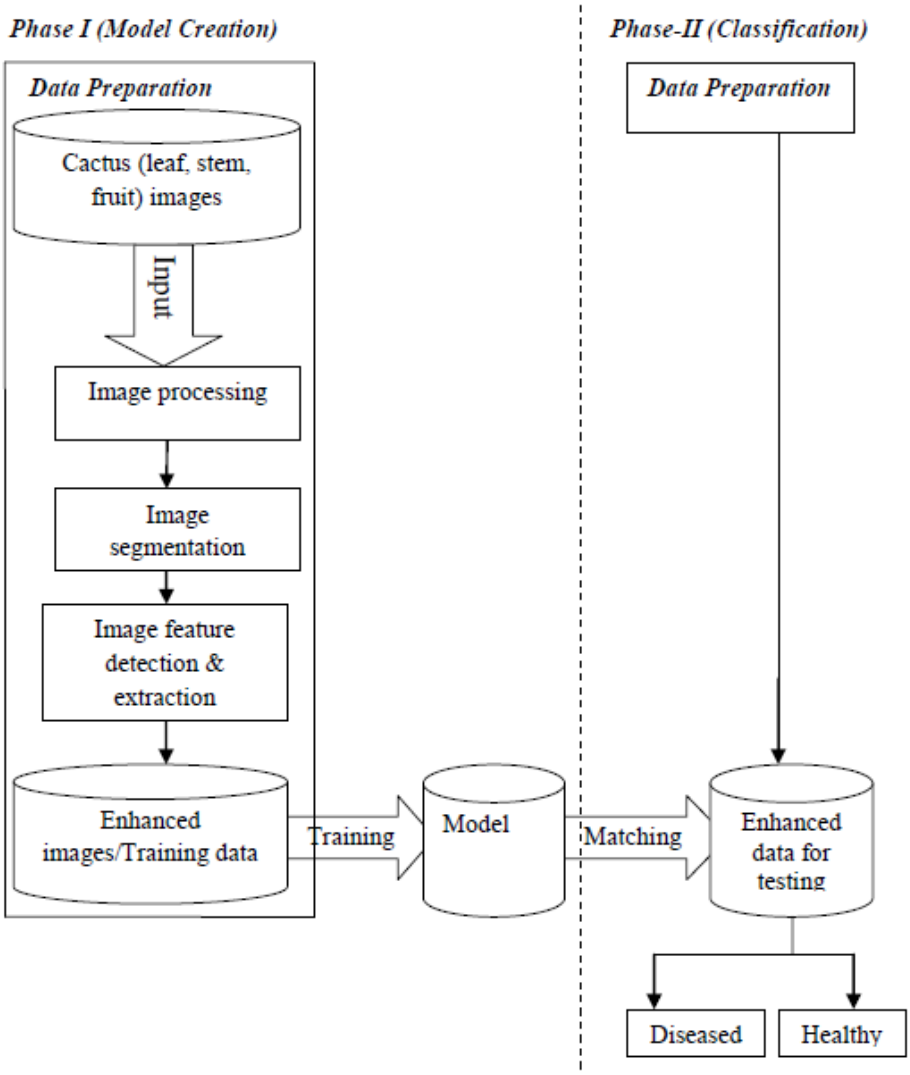

Fig 1: Architectural Design of the System

IV. EXPERIMENTAL RESULTS

In this section, the experimental results are shown to select the most important machine learning technique and cactus image feature that can create a better performing classifier. The experimental results were shown using scatter plot and confusion matrix. Therefore, the results of the confusion matrix are expressed in a tabular form.

As the experimental results were depicted using scatter plot and confusion matrix, it is better to discuss about scatter plot and confusion matrix before directly going to the results. *Scatter plot* is a matlab feature that shows the relationship between two quantitative variables that measure the same individual [1]. The values of the variables are put on the vertical and horizontal axes. In all the scatter plots we have in this document, the values of the selected features (two variables) are put one of them in the X axis and the other on the Y axis. The class of each feature is also shown by dot for correctly classified and x for incorrectly classified ones, whereas confusion matrix shows the predicted and actual classification of the classifier [2]. In the screenshots in this document, confusion matrix depicts the true and predicted classes of each image (feature) or it shows the number of images that are correctly classified versus the number of images that are misclassified. It also tells the exact number of the correctly classified and misclassified number of images as it can be seen from each result.

The main purpose of this work is to correctly classify cactus images into two classes, namely, 'Diseased' and 'Healthy' using the image features. 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 color histogram, GLCM and bag of features extraction techniques respectively as it can be seen from the following screenshots.

In each confusion matrices below, the most right hand side shows true positive and false negative ratios (TPR/FNR). Given a classifier and an instance, there are four possible outcomes, namely, true positive, false negative, true negative and false positive [3]. If the instance is positive and if it is classified as positive, it is called true positive, unless it is false negative. However, if the instance is negative and if it is classified as negative it is called true negative, unless it is called false positive. The ratio of each category is represented as the following formulae.

$$\text{False Positive rate} = FP/(FP + TN)$$

$$\text{True Positives rate} = \text{TP}/(FN + TP)$$

True negative rate = TN=/ (TN + FP).
False negative rate = FN/(FN + TP).

Therefore, every ratio that is put at the most right hand side of each confusion matrix is calculated by any one of these formulae and it shows the classification accuracy of each model.

Fig 2 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.

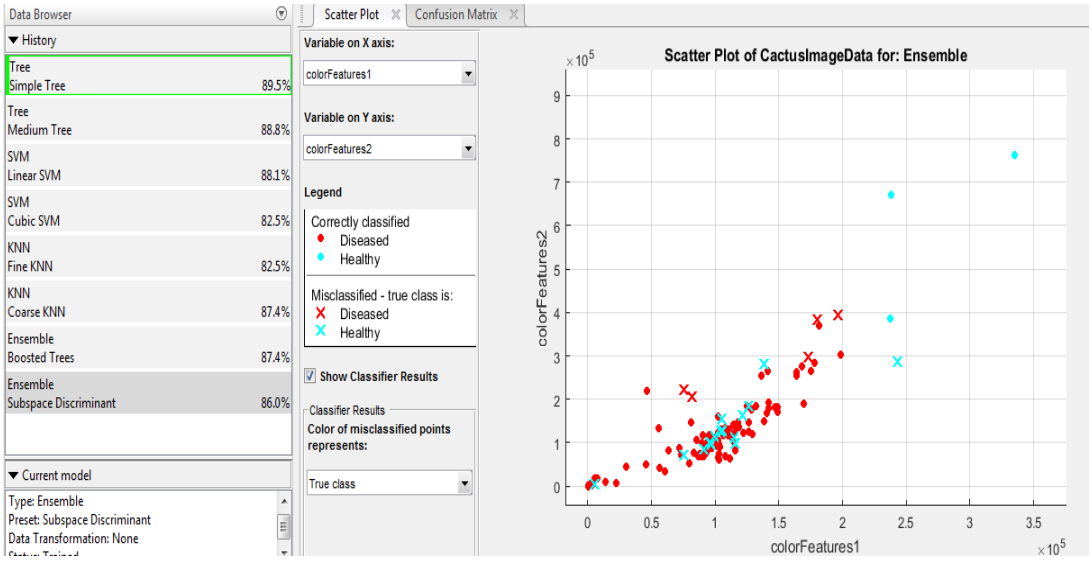

Fig 2: Scatter plot for color features based Cactus classification [4]

Table 1 shows a result of a confusion matrix of a simple tree that was used for training and testing the model using color features. For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model. The overall accuracy of the classifier (simple tree) was 89.5%. This model has correctly classified 123 (98.4%) diseased images into their predicted class and failed to correctly classify 2 (1.6%) diseased images into their predicted class. It also correctly classified 5 (27.8%) healthy images into their predicted class and failed to correctly classify 13 (72.2%) healthy images into their predicted class. Therefore, this shows that majority of the healthy images were misclassified and this model has less performance.

Table 2 shows the result of the confusion matrix of a medium tree that was used for training and testing the model using color features. For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model. The overall accuracy of the classifier (medium tree) was 88.8%. This model has correctly classified 118 (94.4%) diseased images into their predicted class and failed to correctly classify 7 (5.6%) diseased images into their predicted class. It also correctly classified 9 (50%) healthy images into their predicted class and failed to correctly classify 9 (50%) healthy images into their predicted class. Therefore, this shows that half of the healthy images were misclassified and this model has less performance.

Table 1: Result of Confusion Matrix for Simple Tree Using Color based Cactus classification [4]

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Color	2	123	98.4
18 (Diseased)		13	5	27.8

Table 2: Result of Confusion Matrix for Medium Tree Using Color based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Color	7	118	94.4
18 (Diseased)		9	9	50

Table 3 shows a result of a confusion matrix of a linear support vector machine that was used for training and testing the model using color features.

For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model.

[1] colorFeatures2 = Color features in the second column; colorFeatures1 = Color features in the first column

The overall accuracy of the classifier (linear SVM) was 88.1%. This model has correctly classified 125 (100%) diseased images into their predicted class and did not fail to correctly classify the diseased images into their predicted class. It also correctly classified only one (1) (5.6%) healthy images into their predicted class and failed to correctly classify 17 (94.4%) healthy images into their predicted class. Therefore, this shows that majority of the healthy images were misclassified and this model has less performance.

Table 3: Result of Confusion Matrix for linear SVM Using Color based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Color	0	125	100
18 (Diseased)		17	1	5.6

Table 4 shows a result of a confusion matrix of a cubic support vector machine that was used for training and testing the model using color features. For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model. The overall accuracy of the classifier (cubic SVM) was 82.5%. This model has correctly classified 113 (90.4%) diseased images into their predicted class and failed to correctly classify 12 (9.6%) diseased images into their predicted class. It also correctly classified 5 (27.8%) healthy images into their predicted class and failed to correctly classify 13 (72.2%) healthy images into their predicted class. Therefore, this shows that majority of the healthy images were misclassified and this model has less performance.

Table 4: Result of Confusion Matrix for Cubic SVM Using Color based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Color	12	113	90.4
18 (Diseased)		13	5	27.8

Table 5 shows the result of a confusion matrix of a Fine K-Nearest Neighbor that was used for training and testing the model using color features. For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model. The overall accuracy of the classifier (Fine KNN) was 82.5%. This model has correctly classified 109 (87.2%) diseased images into their predicted class and failed to correctly classify 16 (12.8%) diseased images into their predicted class. It also correctly classified 9 (50%) healthy images into their predicted class and failed to correctly classify 9 (50%) healthy images into their predicted class. Therefore, this shows that half of the healthy images were misclassified and this model has less performance.

Table 5: Result of Confusion Matrix for Fine KNN Using Color based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy %)*
125 (Healthy)	Color	16	109	87.2
18 (Diseased)		9	9	50

Table 6 shows a result of a confusion matrix of a coarse k-Nearest Neighbor that was used for training and testing the model using color features. For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model. The overall accuracy of the classifier (coarse KNN) was 87.4%. This model has correctly classified 125 (100%) diseased images into their predicted class and did not fail to correctly classify the diseased images into their predicted class. It also misclassified all (18) (100%) healthy images and failed to correctly classify even one (0%) healthy image into its predicted class. Therefore, this shows that the model has totally failed to classify the healthy images into their predicted class and this model has less performance.

Table 6: Result of Confusion Matrix for Coarse KNN Using Color based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Color	0	125	100
18 (Diseased)		18	0	0

Table 7 shows a result of a confusion matrix of Boosted trees that was used for training and testing the model using color features. For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model. The overall accuracy of the classifier (Boosted trees) was 87.4%. This model has correctly classified 125 (100%) diseased images into their predicted class and did not fail to correctly classify the diseased images into their predicted class. It also misclassified all (18) (100%) healthy images and failed to correctly classify even one (0%) healthy image into its predicted class. Therefore, this shows that the model has totally failed to classify the healthy images into their predicted class and this model has less performance.

Table 7: Result of Confusion Matrix for Boosted Trees Using Color based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Color	0	125	100
18 (Diseased)		18	0	0

Table 8 shows a result of a confusion matrix of a subspace discriminant that was used for training and testing the model using color features. For creating the model, 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) were used. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model. The overall accuracy of the classifier (subspace discriminant) was 86%. This model has correctly classified 120 (96%) diseased images into their predicted class and failed to correctly classify 5 (4%) diseased images into their predicted class. It also correctly classified 3 (16.7%) healthy images into their predicted class and failed to correctly classify 15 (83.3%) healthy images into their predicted class. Therefore, this shows that majority of the healthy images were misclassified and this model has less performance.

Table 8: Result of Confusion Matrix for Subspace Discriminant Using Color based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Color	5	120	96
18 (Diseased)		15	3	16.7

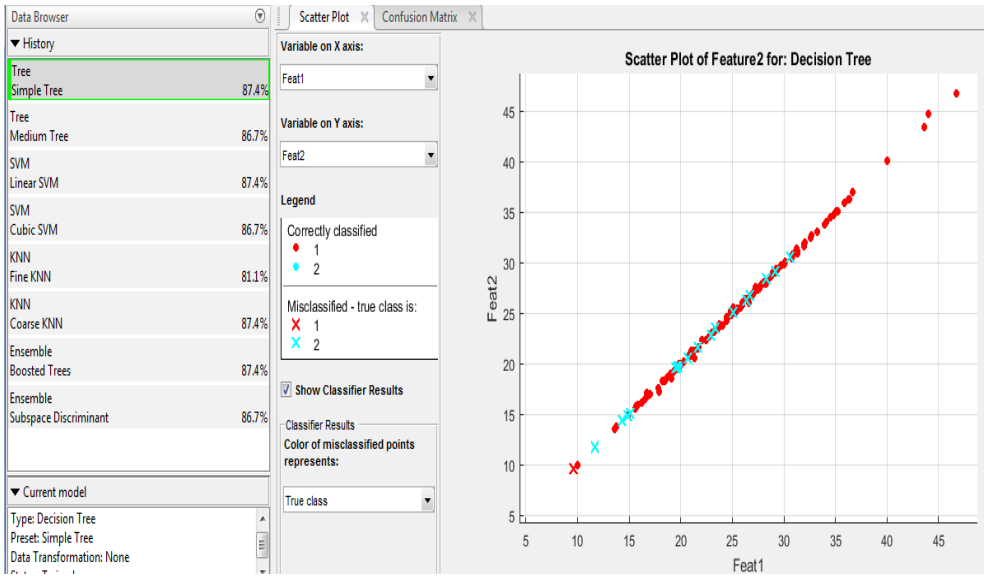

Fig 3: Scatter plot for GLCM features based Cactus classification [4]

The above screenshot (Fig 3) 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.

The result of the confusion matrix in table 9 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a simple tree as a classifier and texture features. The overall accuracy of the model using texture features is 87.4%. During testing, the model has correctly classified 124 (99.2%) diseased images into their predicted class and misclassified one (1) (0.8%) diseased image. It has also correctly classified one (1) (5.6%) healthy image into its predicted class and misclassified 17 (94.4%) healthy images. Therefore, it is probable to say that the model has misclassified the healthy images and it has low performance.

Table 9: Result of Confusion Matrix for Simple Tree using Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	1	124	99.2
18 (Diseased)		17	1	5.6

The result of the confusion matrix table 10 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a medium tree as a classifier and texture features. The overall accuracy of the model using this texture features is 86.7%. During testing, the model has correctly classified 122 (97.6%) diseased images into their predicted class and misclassified 3 (2.4%) diseased images. It has also correctly classified 2 (11.1%) healthy images into their predicted class and misclassified 16 (88.9%) healthy images. Therefore, it is probable to say that the model has misclassified the healthy images and it has low performance. As it can be seen in table 10, the model has correctly classified 122 (97.6%) diseased images into their predicted class and misclassified 3 (2.4%) diseased images.

[2] Feat2 = GLCM features in the second column, Feat1 = GLCM features in the first column

It has also correctly classified 2 (11.1%) healthy images into their predicted class and misclassified 16 (88.9%) healthy images. Therefore, it is probable to say that the model has misclassified the healthy images and it has low performance.

Table 10: Result of Confusion Matrix for Medium Tree sing Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	3	122	97.6
18 (Diseased)		16	2	11.1

The result of the confusion matrix in table 11 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *linear support vector machine* as a classifier and texture features. The overall accuracy of the model using this texture features is 87.4%. During testing, the model has correctly classified 125 (100%) diseased images into their predicted class and did not fail to correctly classify (0%) diseased images into their predicted classes. It has also failed to correctly classify (0%) healthy images into their predicted class and misclassified all 18 (100%) healthy images. Therefore, it misclassified all the healthy images into 'Diseased' class and it has low performance.

Table 11: Result of Confusion Matrix for linear SVM using Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	0	125	100
18 (Diseased)		18	0	0

The result of the confusion matrix in table 12 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *cubic support vector machine* as a classifier and texture features. The overall accuracy of the model using this texture features is 86.7%. During testing, the model has correctly classified 118 (94.4%) diseased images into their predicted class and misclassified 7 (5.6%) diseased images. It has also correctly classified 6 (33.3%) healthy images into their predicted class and failed to correctly classify 12 (66.7%) healthy images into their predicted class. Therefore, it misclassified the majority of the healthy images into 'Diseased' class and it has low performance.

Table 12: Result of Confusion Matrix for Cubic SVM using Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	7	118	94.4
18 (Diseased)		12	6	33.3

The result of the confusion matrix in table 13 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *Fine K-Nearest Neighbor* as a classifier and texture features. The overall accuracy of the model using this texture features is 81.1%. During testing, the model has correctly classified 114 (91.2%) diseased images into their predicted class and misclassified 11 (8.8%) diseased images. It has also correctly classified 2 (11.1%) healthy images into their predicted class and failed to correctly classify 16 (88.9%) healthy images into their predicted class. Therefore, it misclassified the majority of the healthy images into 'Diseased' class and it has low performance.

Table 13: Result of Confusion Matrix for Fine KNN using Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	11	114	91.2
18 (Diseased)		16	2	11.1

The result of the confusion matrix in table 14 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *Coarse K-Nearest Neighbor* as a classifier and texture features. The overall accuracy of the model using this texture features is 87.4%. During testing, the model has correctly classified 125 (100%) diseased images into their predicted class and misclassified zero (0%) diseased images. It has also correctly classified 0 (0%) healthy images into their predicted class and failed to correctly classify 18 (100%) healthy images into their predicted class. Therefore, it misclassified all of the healthy images into 'Diseased' class and it has low performance.

Table 14: Confusion Matrix for Coarse KNN using Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	0	125	100
18 (Diseased)		18	0	0

The result of the confusion matrix in table 15 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased 7and 18 (25% of the healthy images) healthy images were also used for testing the model using *Boosted Trees* as a classifier and texture features. The overall accuracy of the model using this texture features is 87.4%. During testing, the model has correctly classified 125 (100%) diseased images into their predicted class and misclassified zero (0%) diseased images.

It has also correctly classified 0 (0%) healthy images into their predicted class and failed to correctly classify 18 (100%) healthy images into their predicted class. Therefore, it misclassified all of the healthy images into 'Diseased' class and it has low performance.

Table 15: Result of Confusion Matrix for Boosted Trees using Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	0	125	100
18 (Diseased)		18	0	0

The result of the confusion matrix in table 16 shows that a model was created using 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *Subspace Discriminant* as a classifier and texture features. The overall accuracy of the model using this texture features is 86.7%. During testing, the model has correctly classified 123 (98.4%) diseased images into their predicted class and misclassified 2 (1.6%) diseased images. It has also correctly classified 1 (5.6%) healthy image into its predicted class and failed to correctly classify 17 (94.4%) healthy images into their predicted class. Therefore, it misclassified the majority of the healthy images into 'Diseased' class and it has low performance.

Table 16: Result of Confusion Matrix for Subspace Discriminant using Texture based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	Texture	2	123	98.4
18 (Diseased)		17	1	5.6

Fig 4 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.

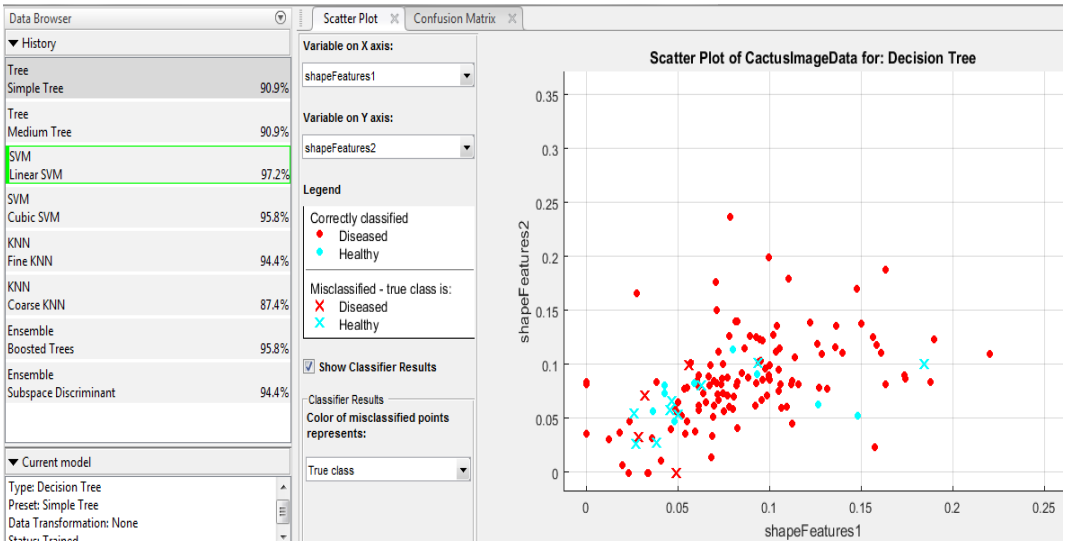

Fig 4: Scatter plot for bag of features based Cactus classification [4]

Table 17 is the result of the confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a simple tree as a classifier. The overall accuracy of the classifier using bag of features was 90.9%. Of the images used for testing the model, 121 (96.8%) diseased images were correctly classified into their predicted classes and 4 (3.2%) diseased images were incorrectly classified. It also shows that 9 (50%) healthy images were correctly classified into their predicted class and 9 (50%) healthy images were misclassified into 'Diseased' class. This means that half of the healthy images were misclassified and there is low performance.

Table 17: Result of Confusion Matrix for Simple Tree using bag of features based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	bag of features	4	121	96.8
18 (Diseased)		9	9	50

[3] shapeFeatures2 = bag of features in the second column; shapeFeatures1 = bag of features in the first column

Table 18 is the result of the confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a medium tree as a classifier. The overall accuracy of the classifier using bag of features was 90.9%. Of the images used for testing the model, 120 (96%) diseased images were correctly classified into their predicted class and 5 (4%) diseased images were incorrectly classified. It also shows that 10 (55.6%) healthy images were correctly classified into their predicted class and 8 (44.4%) healthy images were misclassified into 'Diseased' class. This means that although more than half of the healthy images were correctly classified, there is low performance.

Table 18: Result of Confusion Matrix for Medium Tree using bag of features based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	bag of features	5	120	96
18 (Diseased)		8	10	55.6

Table 19 is the result of the confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *linear support vector machine* as a classifier. The overall accuracy of the classifier using bag of features was 97.2%. Of the images used for testing the model, 124 (99.2%) diseased images were correctly classified into their predicted class and 1 (0.8%) diseased image was incorrectly classified. It also shows that 15 (83.3%) healthy images were correctly classified into their predicted class and 3 (16.7%) healthy images were misclassified into 'Diseased' class. This means that the classifier performs well and more than half of the images were correctly classified.

Table 19: Confusion Matrix for linear SVM using bag of features based Cactus classification [4]

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	bag of features	1	124	99.2
18 (Diseased)		3	15	83.3

Table 20 is the result of the confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a cubic support vector machine as a classifier. The overall accuracy of the classifier using bag of features was 95.8%. Of the images used for testing the model, 123 (98.4%) diseased images were correctly classified into their predicted class and 2 (1.6%) diseased images were incorrectly classified. It also shows that 14 (77.8%) healthy images were correctly classified into their predicted class and 4 (22.2%) healthy images were misclassified into 'Diseased' class. This means that the classifier performs better and more than half of the images were correctly classified although it is not as better as linear SVM.

Table 20: Result of Confusion Matrix for Cubic SVM using bag of features based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	bag of features	2	123	98.4
18 (Diseased)		4	14	77.8

Table 21 is the result of the confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *Fine K-Nearest Neighbor* as a classifier. The overall accuracy of the classifier using bag of features was 94.4%. Of the images used for testing the model, 122 (97.6%) diseased images were correctly classified into their predicted class and 3 (2.4%) diseased images were incorrectly classified. It also shows that 13 (72.2%) healthy images were correctly classified into their predicted class and 5 (27.8%) healthy images were misclassified into 'Diseased' class. This means that the classifier performs better and more than half of the images were correctly classified although it is not as better as linear SVM.

Table 21: Result of Confusion Matrix for Fine KNN using bag of features based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	bag of features	3	122	97.6
18 (Diseased)		5	13	72.2

Table 22 is the result of the confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using a *coarse K-Nearest Neighbor* as a classifier. The overall accuracy of the classifier using bag of features was 87.4%. Of the images used for testing the model, 125 (100%) diseased images were correctly classified into their predicted class and 0 (0%) diseased images were incorrectly classified. It also shows that 0 (0%) healthy images were correctly classified into their predicted class and 18 (100%) healthy images were misclassified into 'Diseased' class.

This means that the classifier performs lower and misclassified all healthy the images.

Table 22: Result of Confusion Matrix for Coarse KNN using bag of features based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	bag of features	0	125	100
18 (Diseased)		18	0	0

Table 23 is the result of confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using *Boosted trees* as a classifier. The overall accuracy of the classifier using bag of features was 95.8%. Of the images used for testing the model, 124 (99.2%) diseased images were correctly classified into their predicted class and 1 (0.8%) diseased image was incorrectly classified. It also shows that 13 (72.2%) healthy images were correctly classified into their predicted class and 5 (27.8%) healthy images were misclassified into 'Diseased' class. This means that the classifier performs better and more than half of the images were correctly classified although it is not as better as linear SVM.

Table 23: Result of Confusion Matrix for Boosted Trees using bag of features based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	*bag of features*	*1*	*124*	*99.2*
18 (Diseased)		*5*	*13*	*72.2*

Table 24 is the result of the confusion matrix that shows the creation and testing of the model using image bag of features. 375 (75% of the total dataset of the unhealthy images) diseased and 54 healthy (75% of the total dataset of the healthy images) images were used to create the model. 125 (25% of the unhealthy images) diseased and 18 (25% of the healthy images) healthy images were also used for testing the model using *Subspace Discriminant* as a classifier. The overall accuracy of the classifier using bag of features was 94.4%. Of the images used for testing the model, 122 (97.6%) diseased images were correctly classified into their predicted class and 3 (2.4%) diseased images were incorrectly classified. It also shows that 13 (72.2%) healthy images were correctly classified into their predicted class and 5 (27.8%) healthy images were misclassified into 'Diseased' class. This means that the classifier performs better and more than half of the images were correctly classified although it is not as better as linear SVM.

Table 24: Result of Confusion Matrix for Subspace Discriminant using bag of features based Cactus classification

No. of Cactus images used	*Feature Used*	*Misclassified*	*Correctly Classified*	*Accuracy (%)*
125 (Healthy)	bag of features	3	122	97.6
18 (Diseased)		5	13	72.2

To summarize, table 19 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 table 19.

The sample summary of the correctly and incorrectly classifying of the images (as in tables 1, 9 and 19) is presented in the following table.

Table 25: Model classification Summary

S/No.	*Feature Used*	*Misclassified Healthy*	*Correctly Classified Healthy*	*Misclassified Diseased*	*Correctly Classified Diseased*
1	Color	13	5	2	123
2	Texture	17	1	1	124

The above table shows the classification power of the model. The values are taken from the confusion matrices of each classification using the three extracted features (color, texture and bag of features) separately. As it can be seen from the table, of the 18 healthy images, 13 images are incorrectly classified and 5 images are correctly classified while using color features. Of the 125 diseased images, 123 images are correctly classified and 2 images are misclassified using the same feature (color). On the other hand, while using texture (GLCM features), 17 healthy images are misclassified and one (1) image is correctly classified and 123 diseased images are also correctly classified and 2 are misclassified. Lastly, while using bag of features, of the 18 healthy images, 15 are correctly classified and 3 images are misclassified. Using the same feature, 124 diseased images are classified correctly one (1) image is misclassified of the 125 diseased cactus images.

V. RESULT ANALYSIS

In this scenario (tables 1-24), 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 results of the (tables) confusion matrix of each classifier (in the tables 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. 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 learning technique with the overall accuracy of 97.2% as it can be seen from the following line graph.

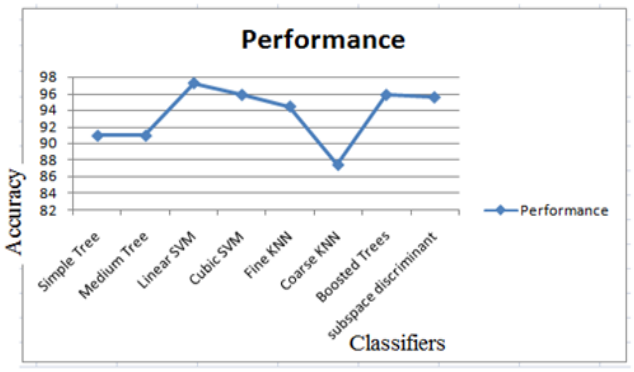

Fig 5: Performance of classifiers [4]

VI. CONCLUSIONS

This research was done to experimentally select better performing features and a learning technique to identify cactus images whether they are healthy or unhealthy. Conducting intensive experiments on cactus images' features and the classification results, bag of features are found to be the best performing features than color and texture features for cactus images. It is also found that bag of features technique is found to be the most important technique than color histogram and GLCM techniques. Using these features (color, texture and bag of features), eight learning techniques (simple tree, medium tree, linear SVM, cubic SVM, fine KNN, coarse KNN, boosted trees and Subspace Discriminant) were used and linear SVM was found the best classifier (learning technique) for these data with 99.2% and 83.3 % to correctly classify the 'Healthy' and 'Diseased' cactus images respectively.

REFERENCES:

1. Moore, D. S., Notz, W. I, & Flinger, M. A: "The basic practice of statistics", Sixth Edition, 2013.
2. Sofia Visa, Brian Ramsay, Anca Ralescu, Esther van der Knaap: "Confusion Matrix-based Feature Selection", unpublished article.
3. Tom Fawcett: "ROC Graphs: Notes and Practical Considerations for Data Mining Researchers", unpublished article, January 2003.
4. Hailay Beyene Berhe and Narayan A. Joshi: "Classification of Healthy and Diseased Cactus plants using SVM", International Journal of Computer Sciences and Engineering, May 2019.

AUTHORS' PROFILE

Hailay Beyene, has pursued his Bachelor degree in Mekelle University in 2007 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

Dr. Narayan A Joshi, is working as a professor & Head at Dharmsinh Desai University, Nadiad. He was felicitated with prestigious 'Drona Award' by IBM-India.