International Journal of Engineering and Manufacturing Science.
ISSN 2249-3115 Volume 8, Number 1 (2018) pp. 135-144

http://www.ripublication.com

Implementation of K-Means Clustering Approach for the Identification and Edge Detection of Cotton Leaves Image Processing Technique

Bharathi N.[1] and Santosh K. C.[2*]

[1]*ISE Dept.,.BIET., Davangere, India*
[2] *CSE Dept. BIET., Davangere, India*
[*]E-mail: kcsantoo@gmail.com

Abstract

This paper proposes colour based segmentation method that uses k means clustering technique to identify diseases that are affected on cotton leaves. The identification of leaf types is done using colour histogram and edge histogram approaches. In addition to the environmental parameters like rain, temperature, disease on crop is a major factor which effects production quality and quantity of crop yield so detection must be done at early stage to control the spread of disease.

Keywords: Colour histogram, k means clustering, segmentation, lab colour space model

I. INTRODUCTION

India is an agricultural country where most of the population depends on agriculture. And agriculture is one of the major domain which decides economy of the nation. The quality & quantity of the agricultural production is affected by environmental parameters like rain, temperature & other weather parameters which are beyond control of human beings. Another major parameter which affects productivity of the crop is the disease where human beings can have control to improve the productivity for quality as well as for quantity.

The diseases can be controlled by proper Disease management which is a challenging task. This challenge can be converted to easiest task by using image processing for detecting diseases of leaf, stem, root & fruit. With image processing it is possible to detect the affected area, type of disease & severity of the disease. Mostly diseases are seen on the leaves or stems of the plant.

In addition to environmental parameters like rain, temperature, diseases on leaf is major factor. Hence disease management is key issue in agriculture. For management of disease, it needs to be detected at early stage so as to treat it properly & control spread of the disease. Because of advances in the technologies now days it is possible to use the images of diseased leaf to detect the type of disease. This can be achieved by extracting features from the images which can be further used with classification algorithms or content based image retrieval systems.

Plants [2] form a fundamental part of life on earth, providing us with breathable oxygen, food, fuel, medicine and more besides. Plants also help to regulate the climate, provide habitats and food for insects and other animals and provide a natural way to regulate flooding. A good understanding of plants is necessary to improve agricultural productivity and sustainability, to discover new pharmaceuticals, to plan for and mitigate the worst effects of climate change, and to come to a better understanding of life as a whole.

Each leaf [1] has its own features and carries significant information that can help people to recognize and classify the plant by looking at it. Leaf shape is a prominent feature that most people use to recognize and classify a plant. The diameter, physiological length, physiological width, leaf area and perimeter are basic geometry information. In addition, leaf color, textures and vein are also considered as features. All these features are useful for recognition and classification of leaf image.

2. SYSTEM DESIGN

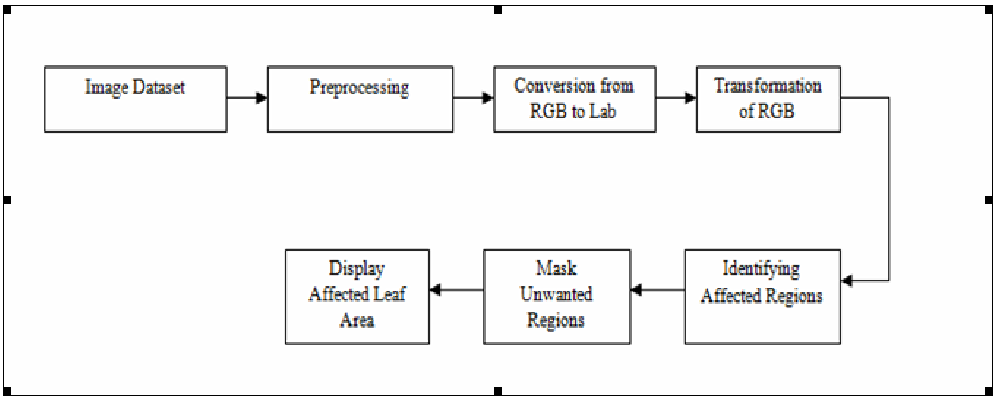


Fig. 1: Methodology for disease Identification

The images of various leaves are acquired using a digital camera[6]. Then image processing techniques are applied to the acquired images to extract useful features that are necessary for further analysis.

2.1 Pseudo Code

Step 1: Read Image.

Step 2: Convert Image from RGB color space to L*a*b Color space.

Step 3: Classify the colors in a*b space using K means Clustering.

Step 4: Label every pixel in the image using the result From K means.

Step 5: Create the images that segment the dominant Colors.

Step 1: Read image

```
he = imread('hestain.png');
imshow (he), title ('H&E image');
```

Step 2: Convert image from RGB colour space to L*a*b* colour space

The L*a*b* color space is derived from the CIE XYZ tristimulus values[5]. The L*a*b* space consists of a luminosity layer 'L*', chromaticity-layer 'a*' indicating where color falls along the red-green axis, and chromaticity-layer 'b*' indicating where the color falls along the blue-yellow axis. All of the color information is in the 'a*' and 'b*' layers. You can measure the difference between two colors using the Euclidean distance metric.

Convert the image to L*a*b* color space using makecform and applycform.

```
cform = makecform('srgb2lab');
lab_he = applycform(he,cform);
```

Step 3: Classify the colour in 'a*b*' space using K-means clustering

Clustering is a way to separate groups of objects. K-means clustering treats each object as having a location in space. It finds partitions such that objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. K-means clustering requires that you specify the number of clusters to be partitioned and a distance metric to quantify how close two objects are to each other.

Since the color information exists in the 'a*b*' space, your objects are pixels with 'a*' and 'b*' values. Use kmeans to cluster the objects into three clusters using the Euclidean distance metric.

Step 4: Label every pixel in the image using the results from kmeans

For every object in your input, kmeans returns an index corresponding to a cluster. Label every pixel in the image with its cluster index.

Step 5: Create images that segment the dominant colour

Using pixel labels, you can separate objects by colour, which will result in three images.

imshow (segmented_images{1}), title('objects in cluster 1');
imshow (segmented images{2}), title('objects in cluster 2');

imshow (segmented images {3}), title ('objects in cluster 3');

2.2 K means Clustering

There are always K clusters. There is always at least one item in each cluster. The clusters are non-hierarchical and they do not overlap. Every member of a cluster is closer to its cluster than any other cluster because closeness does not always involve the 'centre' of clusters. The *k*-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

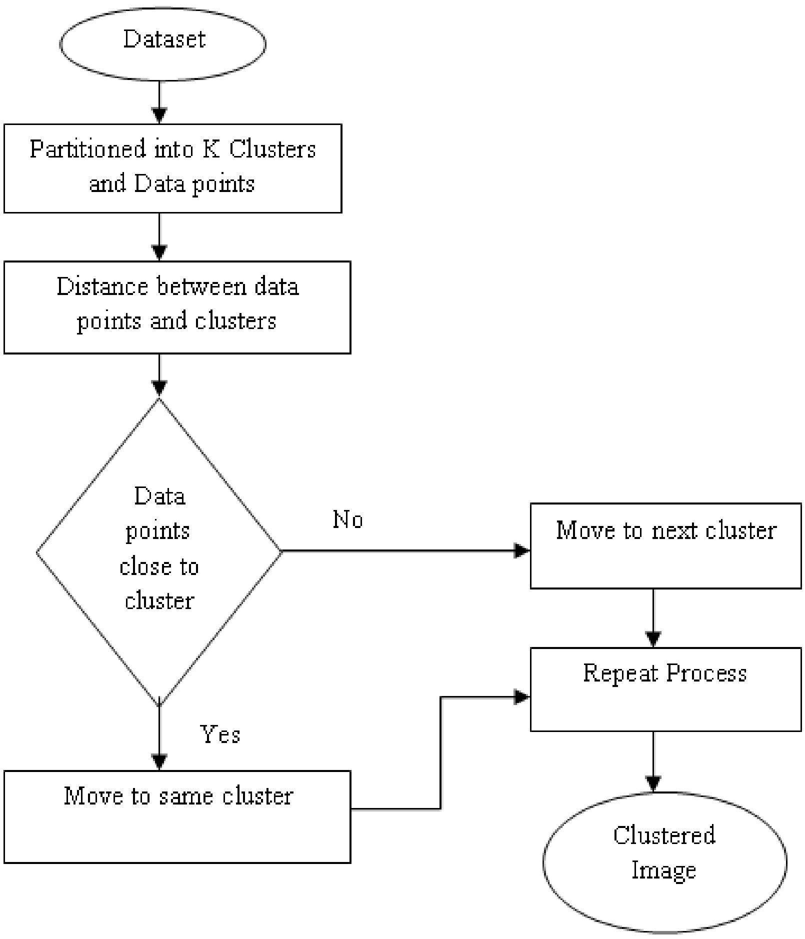


Fig. 2: Flow Chart for K means Algorithm

K-means[4] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters assumes k clusters fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to recalculate k new centroids as barycentre of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated.

As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move anymore. K-Means clustering generates a specific number of disjoint, flat clusters. K-Means method is numerical, unsupervised, non-deterministic and iterative. Hierarchical clustering is also widely employed for image segmentation. The most popular method for image segmentation is k-means clustering.

2.3 Histogram Intersection Algorithm

Step 1: First we input various object Images $_{Oi}$ then Create Bm-Block Matrix. Calculate Mean μ of Block Matrices. All Block Matrices obtained are concatenated.

Step 2:

Then pixel to pixel distances are calculated for each image and averages are found out. Rows denote the number of object in the images. There is a threshold value for which the matching image detected. The matching criterion for the matching of the images is set at 80 percent. This could be set to various desired level of matching.

Step 3:

Now Calculate Euclidean Distance (D)

$$\text{Deucld}\ (r, s) = \sqrt{\sum_{i=1}^{N} (r_i - s_i)^2} \qquad (1)$$

Where r and s represent the average values of feature vectors respectively while $_{Ti==Oi}$ where $_{Ti}$ is the test query image & $_{Oi}$ is the object images. Repeat above procedure for n object images. Now we have ‘N’ object image and its Euclidean distance matrices.

Step 4:

The above said features are combined to match the image. In this work the above similarity is measured are taken then the averages are taken out for the final output.

2.3 FEATURE EXTRACTION

2.3.1 Color Histogram

A color space is defined as a model for representing color in terms of intensity values. Typically, a color space defines a one to four dimensional space. A color component, or a color channel, is one of the dimensions. Color spaces are related to each other by mathematical formulas. Many histogram distances have been used to define the similarity of two color histogram representations. Euclidean distance and its variations are the most commonly used.

2.3.1.1 Color Histogram Definition

An image histogram refers to the probability mass function of the image intensities. This is extended for color images to capture the joint probabilities of the intensities of the three colors Channels.

$$A, B, C\ (a, b, c) = N.\ \text{prob}\ (A=a, B=b, C=c) \tag{2}$$

Where A, B and C represent the three color channels R.G.B or H.S.V. and *N* is the number of pixels in the image. Since the typical efficient image mining computer represents color images with up to 224 colors.

2.3.2 Histogram Euclidian Distance

Let **h** and **g** represent two color histograms. The Euclidean distance between the color Histograms **h** and **g** can be computed as in this distance formula, there is only comparison between the identical bins in the respective histograms.

$$d^2\ (h, g) = \sum_A \sum_B \sum_C (h(a, b, c) - g(a, b, c))^2 \tag{3}$$

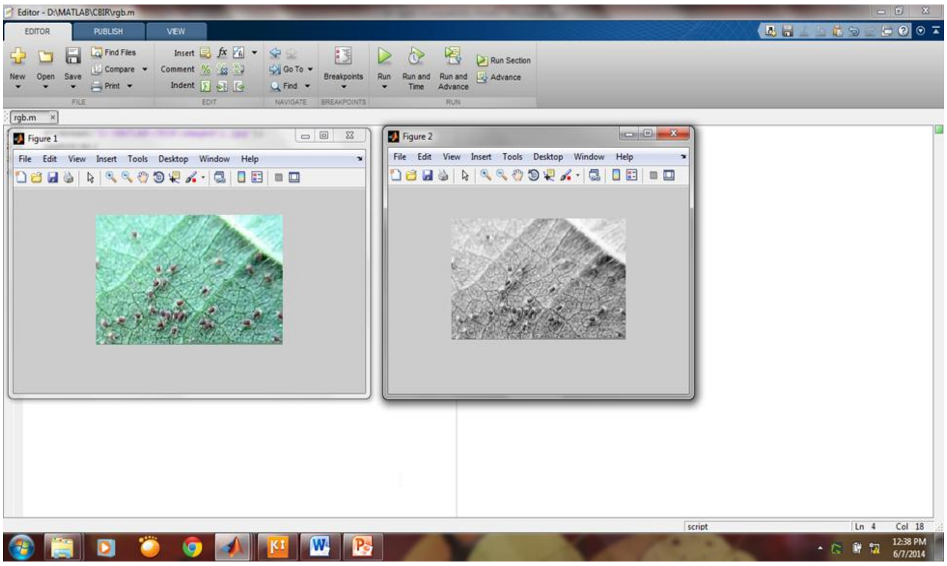
2.3.3 Histogram Intersection Distance

The color histogram intersection was proposed for color image retrieval. The intersection of histograms **h** and **g** is given by

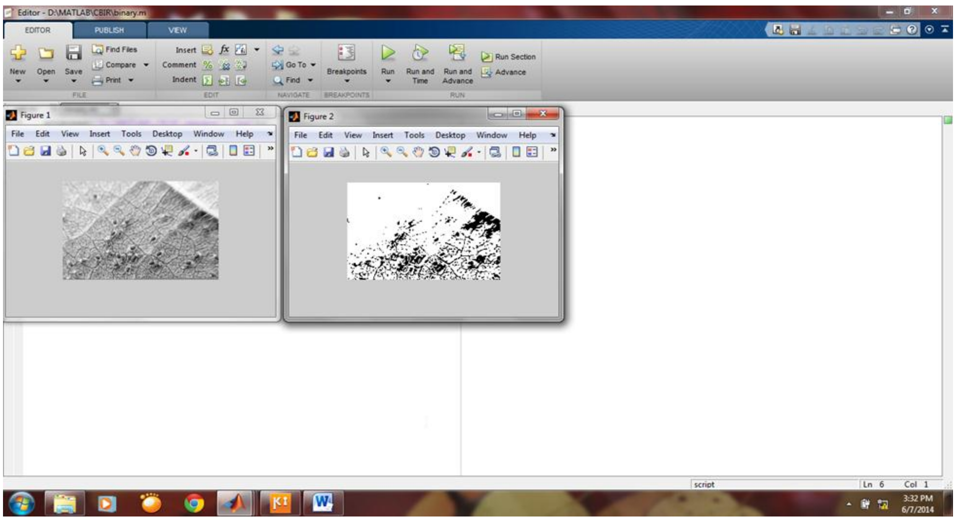
$$d\ (h, g) = \frac{\sum_A \sum_B \sum_C \min(h(a, b, c), g(a, b, c))}{\text{Min}(|h|, |g|)} \tag{4}$$

Where |h| and |g| gives the magnitude of each histogram, which is equal to the number of samples. Colors not present in the user's query image do not contribute to the intersection distance. This reduces the contribution of background colors. The sum is normalized by the histogram with fewest samples.

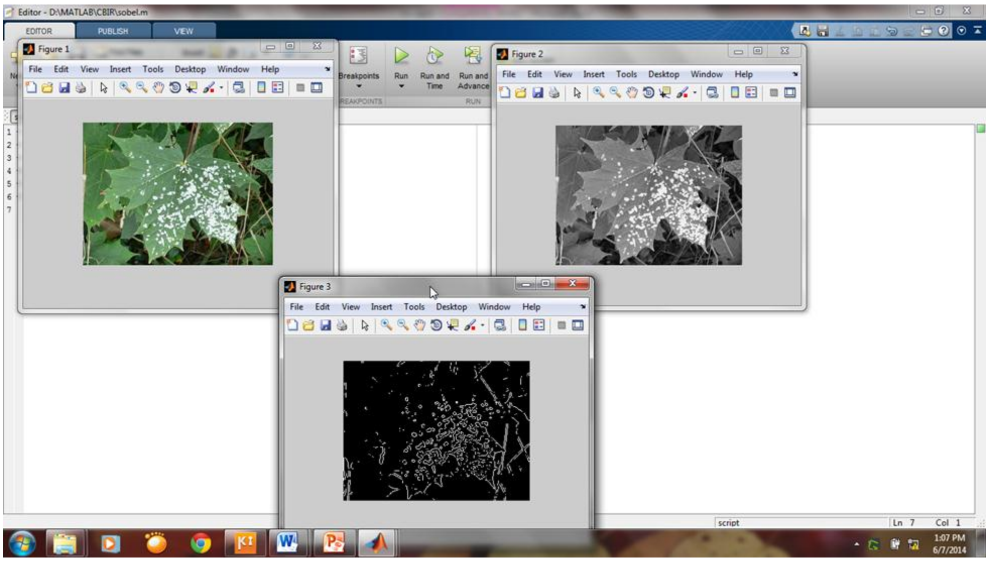
3. RESULTS



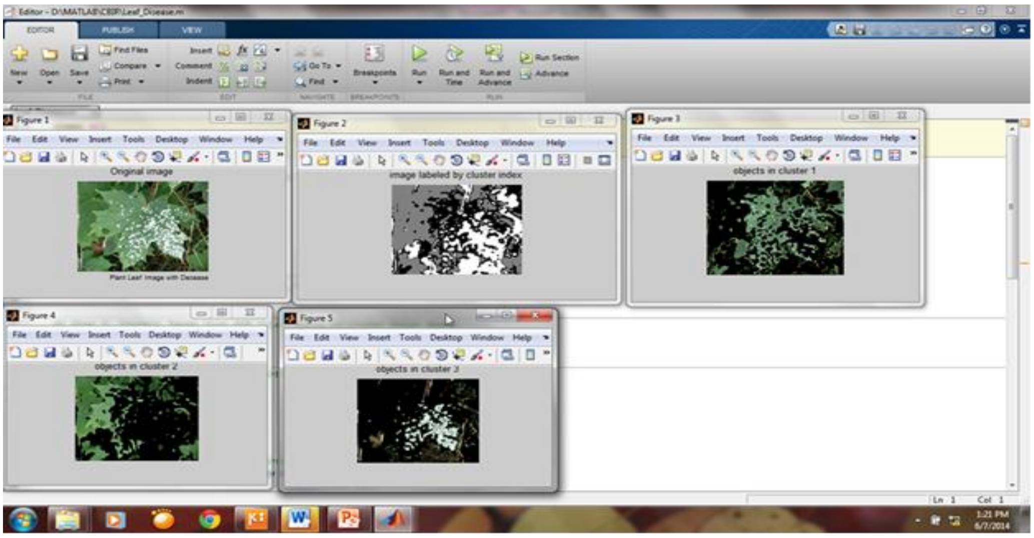
Snapshot 1: RGB to Gray-scale image



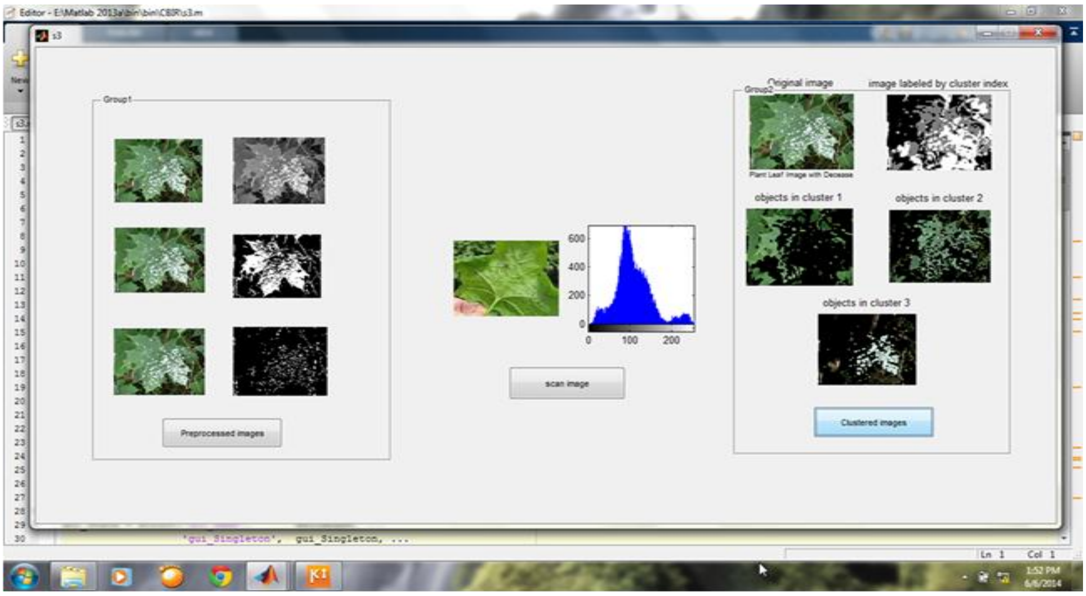
Snapshot 2: RGB to Binary image



Snapshot 3: Edge detection using Sobel method



Snapshot 4: Colour-based Segmentation using Clustering



Snapshot 5: Clustered images

4. CONCLUSION

The proposed system implemented by considering leaf features of color and texture that can be extracted using color histogram and edge histogram methods. The affected parts of cotton leaves have been identified by using K means clustering algorithm and color transformation structure where RGB is converted into Lab color space. For smaller values of k the algorithms give good results. For larger values of k, the segmentation is very coarse many clusters appear in the images at discrete places. Different initial partitions can result in different final clusters .The advantage of K Means algorithm is simple and quite efficient. It works well when clusters are not well separated from each other.

REFERENCES

[1]. **Mohamad Faizal Ab Jabal, Suhardi Hamid, Salehuddin Shuib and Illiasaak Ahmad** “Leaf Features Extraction and Recognition approaches to classify Plant” Faculty of Computer Science and Mathematics, University Teknologi MARA, Malaysia Journal of Computer Science 9 (10) 1295-1304, 2013 ISSN: 1549-3636 © 2013 Science Publications.

[2]. **Ehsanirad, A.**, 2010. “Plant classification based on leaf recognition”. Int. J. Comput. Sci. Inform. Security, 8: 78-81.

[3]. **Kadir, A., L.E. Nugroho, A. Susanto and P.I. Santosa,** 2011b. "Leaf classification using shape, color and texture features". Int. J. Comput. Trends Technol. Seventh Sense Res. Group, 1: 225-230.

[4]. **Lee, C.L. and S.Y. Chen,** 2003. "Classification for leaf images". Proceedings of the 16th IPPR Conference on Computer Vision Graphics and Image Processing, (CVGIP '03), pp: 355-363.

[5]. **J. R. Smith and S.F.Chang**. "Automated image retrieval using color and texture", Technical Report CU/CTR 408-95-14, Columbia University, July 1995.

[6]. **Cope, J.S., D. Corney, J.Y. Clark, P. Remagnino and P. Wilkin,** 2012. "Plant species identification using digital morphometrics" A review. Expert Syst. Applic., 39: 7562-7573. DOI: 10.1016/j.eswa.2012.01.073.

[7]. **Du, J.X., X.F. Wang and G.J. Zhang**, 2007. "Leaf shape based plant species recognition". Applied Mathem. Comput., 185: 883-893. DOI: 10.1016/j.amc.2006.07.072