

---

## International Journal of Intellectual Advancements and Research in Engineering Computations

---

### Prediction of Betal Leaf Disease using image processing techniques

<sup>1</sup> T.Jayachandran, <sup>2</sup>D.Ramya, <sup>2</sup>T.Santhiya, <sup>2</sup>P.Jagadheesh, <sup>2</sup>V.Praveen kumar<sup>1</sup>Assistant professor,  
Department of Electronics and Communication Engineering,

Nandha Engineering college, Erode, TamilNadu, India.

<sup>2</sup>UG students, Department of Electronics and Communication Engineering,

Nandha Engineering college, Erode, TamilNadu, India.

thamilarasiece@gmail.com

#### Abstract:

Agriculture plays a vital role to development of economy of India. Farmers have difficulty to select suitable fruit and vegetable crop. Disease management by manually is a challenging task. Most of the diseases are seen on the leaves or stems of the plant. Hence agriculturist needs to find out the efficient techniques. In this paper we have combined two techniques to predict the disease. The goal of proposed work to capture the leaf image from the camera. Apply preprocessing steps to remove noise from the image using Adaptive Median Filter Technique. K means segmentation is used to segment the affected part in the leaves. After segmentation Process extracts the features using color and

Gabor and finally classify the disease based on Neural Network.

#### Introduction:

India is well known for agricultural country; wherein about 70% of the population depends on agriculture. Farmers have wide range of multiplicity to select suitable crops for their farm. However, the cultivation of these crops for optimum yield and quality produce is mostly technical. It can be improved by the aid of technological support. The management of perennial crops requires close controlling especially for the management of diseases that can affect production significantly and after

words the post-harvest life. The image processing is best technique used in agricultural applications for following purposes. Predict plant disease from image of plants. The leaf disease diagnosis is limited by human visual capabilities because most of the first symptoms are microscopic. This process is tedious, time consuming. There is need for design system that automatically recognizes, classifies and quantitatively detects plant disease symptoms. In case of plant disease the disease is known as any impairment of normal physiological function of plants, producing characteristic symptoms. A symptom is a reality accompanying something and is observed as evidence of its existence. Disease is caused by pathogen which is any agent causing disease. Disease management is a challenging task. Mostly diseases are seen on the leaves on plants or stems of the plant. Precise quantification of these visually observed diseases, pests, traits has not studied yet because of the complication of visual patterns. In most of the cases diseases are seen on the leaves or stems of the plant. Therefore recognition of plants, leaves and finding out the diseases, symptoms of the disease attack, plays a important role in successful cultivation of crops. Hence developing a computer vision system to detect, recognize, and classify disease affected on crops which will avoid human interference and hence lead to précised unbiased decision about disease infection and its further valuation. The development of an automated system also helps

farmers avoid consulting divine. Automatic detection of leaf diseases is most important research topic as it may prove gain in monitoring large fields of crops, and thus automatically detect the diseases from the symptoms that present on the plant leaves. This enables machine vision that is to provide image based here image processing plays important Role. The system provides the facility to Capture image, process it and get result through image processing. In the real world, farmers visually carry out inspection of crops such as fruits, vegetables and the like affected by the different disease for recognition and classification. In recent literature, the image processing techniques are being widely and efficiently used in agricultural field for disease detection and classification. In this paper consists of two phases to identify the affected part of the disease. Initially Edge detection based Image segmentation is done, and at lastly image analysis and classification of diseases is carried out using our proposed Homogeneous Pixel Counting Technique for Cotton Diseases Detection (HPCCDD) Algorithm. The goal of this research work is identify the disease affected part of cotton leaf sport by using the image analysis technique. In [4], present paper to detection of leaf diseases. In this used method is threefold:

- 1) Identifying the infected object based upon k-means clustering;

2) Extracting the features set of the infected objects using color co occurrence methodology for texture analysis;

3) Detecting and classifying the type of disease using NNs, moreover, the presented scheme classifies the plant leaves into infected and not-infected classes.

In, the process of image segmentation was analyzed and leaf region was segmented by using Otsu method. In the HSI color system, H component was chosen to segment disease spot to reduce the disturbance of illumination changes and the vein. Then disease spot regions were segmented by using Sobel operator to examine disease spot edges. Finally plant diseases are graded by calculating the quotient of disease spot and leaf areas.

## EXISTING METHOD

### Image thresholding

Thresholding is a image processing method used to convert a grey scale image (value of pixels ranging from 0-255) into binary image (value of pixels can have only 2 values: 0 or 1). Thresholding techniques are mainly used in segmentation. The simplest thresholding methods replace each pixel in an image with a black pixel if the pixel intensity is less than some fixed constant  $T$ , else it is replace with a white pixel.

There are two basic types of thresholding methods:

Static image thresholding

Dynamic image thresholding

### OTSU Thresholding:

This method is named after its inventor Nobuyuki Otsu and is one of the many binarization algorithms.

The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal, or equivalently (because the sum of pairwise squared distances is constant), so that their inter-class variance is maximal. Consequently, Otsu's method is roughly a one-dimensional, discrete analog of Fisher's Discriminant Analysis. Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum.

### Algorithm steps:

1. Compute histogram and probabilities of each intensity level.
2. Set up initial class probability and initial class means.

3. Step through all possible thresholds maximum intensity.
4. Update  $q_i$  and  $\mu_i$ .
5. Compute between class variance.
6. Desired threshold corresponds to the maximum value of between class variance.

In optimal thresholding, a criterion function is devised that yields some measure of separation between regions. A criterion function is calculated for each intensity and that which maximizes this function is chosen as the threshold.

- Otsu's thresholding chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels.

Formulated as discriminant analysis: a particular criterion function is used as a measure of statistical separation.

Based on a very simple idea:

- Find the threshold that minimizes the weighted within-class variance.
- This turns out to be the same as maximizing the between-class variance.
- Operates directly on the gray level histogram [e.g. 256 numbers,  $P(i)$ ], so it's fast (once the histogram is computed).

- I've used it with considerable success in "murky" situations.

- Histogram (and the image) are bimodal.

- No use of spatial coherence, nor any other notion of object structure.

- Assumes stationary statistics, but can be modified to be locally adaptive. (Exercises).

- Assumes uniform illumination (implicitly), so the bimodal brightness behavior arises from object appearance differences only.

The weighted within-class variance is:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Where the class probabilities are estimated as:

$$q_1(t) = \sum_{i=1}^t P(i) \quad q_2(t) = \sum_{i=t+1}^l P(i)$$

And the class means are given by:

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad \mu_2(t) = \sum_{i=t+1}^l \frac{iP(i)}{q_2(t)}$$

Finally, the individual class variances are:

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^l [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Now, we could actually stop here. All we need to do is just run through the full range of t values that minimizes.

But the relationship between the within-class and between class variances can be exploited to

that permits a much

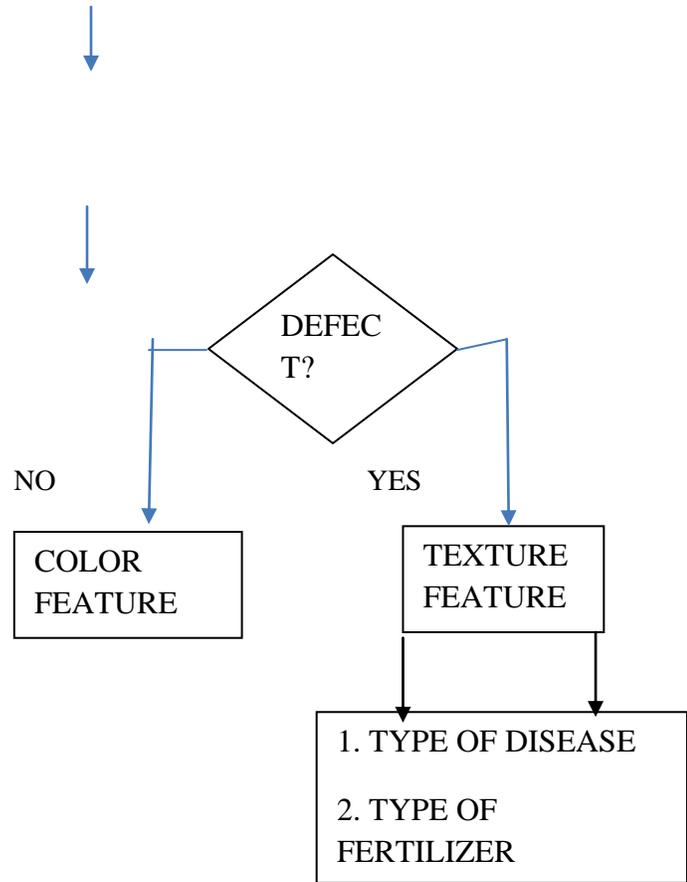
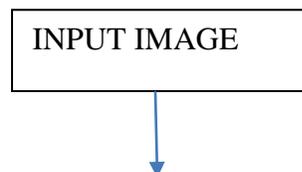
Between/Within/Total Variance

- The book gives the details, but the basic idea is that the total variance does not depend on threshold (obviously).
- For any given threshold, the total variance is the sum of the within-class variances (weighted) and the between class variance, which is the sum of weighted squared distances between the class means and the grand mean.

Since the total is constant and independent of t, the effect of changing the threshold is merely to move the contributions of the two terms back and forth. So, minimizing the within-class variance is the same as maximizing the between-class variance. The nice thing about this is that we can compute the quantities in recursively as we run through the range of t values.

**PROPOSED METHOD**

**BLOCK DIAGRAM**



**RGB TO LAB CONVERSION**

**CIE**

There are two CIE based colour spaces, CIELuv and CIELab. They are nearly linear with visual perception, or at least as close as any colour space is expected to sensibly get. Since they are based on the CIE system of colour measurement, which is itself based on human vision, CIELab and CIELuv are device

independent but suffers from being quite unintuitive despite the L parameter having a good correlation with perceived lightness. To make them more user friendly, the CIE defined two analogous spaces - CIELhs or CIELhc where h stands for hue, s for saturation and c for chroma. In addition CIELuv has an associated two-dimensional chromaticity chart which is useful for showing additive colour mixtures, making CIELuv useful in applications using CRT displays. CIELab has no associated two dimensional chromaticity diagrams and no correlate of saturation. CIELhs can therefore not be defined.

#### **CIE L\*a\*b\*:**

This is based directly on CIE XYZ and is another attempt to linearise the perceptibility of unit vector colour differences. Again, it is non-linear, and the conversions are still reversible. Colouring information is referred to the colour of the white point of the system, subscript n. The non-linear relationships for L\* a\* and b\* are the same as for CIELUV and are intended to mimic the logarithmic response of the

Again, L\* scales from 0 to 100. Again, there are polar parameters that more closely match the visual experience of colours.

Hue is an angle in four quadrants, and there is no saturation term in this system. When determining CIEL\*a\*b\* or CIEL\*u\*v\* values

for CRT displayed colours it is usual to use the CRT's white point as the reference white.

#### **K means Segmentation:**

The k-means algorithm is an evolutionary algorithm that gains its name from its method of operation. The algorithm clusters observations into k groups, where k is provided as an input parameter. It then assigns each observation to clusters based upon the observation's proximity to the mean of the cluster. The cluster's mean is then recomputed and the process begins again. Here's how the algorithm works:

Step 1: The algorithm arbitrarily selects k points as the initial cluster centers ("means").

Step2: Each point in the dataset is assigned to the closest cluster, based upon the Euclidean distance between each point and each cluster center.

Step 3: Each cluster center is recomputed as the average of the points in that cluster.

Step 4: Step 2 and 3 repeat until the clusters converge. Convergence may be defined differently depending upon the implementation.

#### **Description:**

Original K-means algorithm choose k points as initial clustering centers, different points may obtain different solutions. In order to diminish the sensitivity of initial point choice,

we employ a mediod [11], which is the most centrally located object in a cluster, to obtain better initial centres. The demand of stochastic sampling is naturally bias the sample to nearly represent the original dataset, that is to say, samples drawn from dataset can't cause distortion and can reflect original data's distribution. In order to lessen the influence of sample on choosing initial starting points, following procedures are employed. First, drawing multiple sub-samples (say J) from original dataset (the size of each sub-sample is not more than the capability of the memory, and the sum for the size of J sub-samples is as close as possible to the size of original dataset). Second, use K-means for each sub-sample and producing a group of mediods respectively. Finally, comparing J solutions and choosing one group having minimal value of square error function as the refined initial points. To avoid dividing one big cluster into two or more ones for adopting square-error criterion, we assume the number of clustering is  $K'$  ( $K > K'$ ,  $K'$  depends on the balance of clustering quality and time). In general, bigger  $K'$  can expand searching area of solution space, and reduce the situation that there are not any initial values near some extremum. Subsequently, re-clustering the dataset through Kmeans with the chosen initial conditions would produce  $K'$  mediods, then merging  $K'$  clusters (which are nearest clusters) until the number of clusters reduced to k.

## COLOR FEATURE

The Color feature is the most general feature extracted from an image. The color images are having the standard Color model known as RGB. Color histogram is a commonly used method for the extraction of color information in CBIR system. Feature refers to the attribute of an object. Feature extraction results in reducing the dimensionality of an object, that form the significant part in image processing.

### Gabor filters (wavelet):

For a given image  $I(x, y)$  with size  $P \times Q$ , its discrete Gabor wavelet transform is given by a convolution:

where,  $s$  and  $t$  are the filter mask size variables, and  $\psi^*_{mn}$  is the complex conjugate of  $\psi_{mn}$  which is a class of self-similar functions generated from dilation and rotation of the following mother wavelet:

$$\psi(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cdot \exp(j2\pi Wx)$$

Where  $W$  is called the modulation frequency. The self-similar Gabor wavelets are obtained through the generating function:

$$\psi_{mn}(x, y) = a^{-m} \psi(\tilde{x}, \tilde{y})$$

Where  $m$  and  $n$  specify the scale and orientation of the wavelet respectively, with  $m = 0, 1, \dots, M-1$ ,  $n = 0, 1, \dots, N-1$ , and

$$\begin{aligned}\tilde{x} &= a^{-m}(x \cos \theta + y \sin \theta) \\ \tilde{y} &= a^{-m}(-x \sin \theta + y \cos \theta)\end{aligned}$$

Where  $a > 1$  and  $\theta = n\pi/N$ . The variables in the above equations are defined as follows:

$$\begin{aligned}a &= (U_h/U_l)^{\frac{1}{M-1}}, \\ W_{m,n} &= a^m U_l \\ \sigma_{x,m,n} &= \frac{(a+1)\sqrt{2 \ln 2}}{2\pi a^m (a-1)U_l}, \\ \sigma_{y,m,n} &= \frac{1}{2\pi \tan\left(\frac{\pi}{2N}\right) \sqrt{\frac{U_h^2}{2 \ln 2} - \left(\frac{1}{2\pi \sigma_{x,m,n}}\right)^2}}\end{aligned}$$

### Texture representation

After applying Gabor filters on the image with different orientation at different scale, we obtain an array of magnitudes:

$$E(m, n) = \sum \sum |G_{mn}(x, y)|,$$

$m = 0, 1, \dots, M-1$ ;  $n = 0, 1, \dots, N-1$  These magnitudes represent the energy content at

different scale and orientation of the image. The main purpose of texture-based retrieval is to find images or regions with similar texture. It is assumed that we are interested in images or regions that have homogenous texture, therefore the following mean  $\mu_{mn}$  and standard deviation  $\sigma_{mn}$  of the magnitude of the transformed coefficients are used to represent the homogenous texture feature of the region:

$$\begin{aligned}\mu_{mn} &= \frac{E(m, n)}{P \times Q} \\ \sigma_{mn} &= \frac{\sqrt{\sum_x \sum_y (|G_{mn}(x, y)| - \mu_{mn})^2}}{P \times Q}\end{aligned}$$

A feature vector  $f$  (texture representation) is created using  $\mu_{mn}$  and  $\sigma_{mn}$  as the feature components [5, 10]. Five scales and 6 orientations are used in common implementation and the feature vector is given by:

$$\mathbf{f} = (\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{01}, \dots, \mu_{45}, \sigma_{45}).$$

### Neural Network Characteristics:

The word network in Neural Network refers to the interconnection between neurons present in various layers of a system. Every system is basically a 3 layered system, which are Input layer, Hidden Layer and Output Layer. The

input layer has input neurons which transfer data via synapses to the hidden layer, and similarly the hidden layer transfers this data to the output layer via more synapses.

Neural Network (NN) can be represented using a directed graph  $G$ , an ordered 2-tuple  $(V, E)$  consisting of a set  $V$  of vertices and  $E$  of edges with vertices  $V = \{1, 2, n\}$  and arcs  $A = \{i \rightarrow j, j \leq n\}$ , having the following restrictions:

- $V$  is partitioned into a set of input nodes  $V_I$ , hidden nodes,  $V_H$ , and output Nodes  $V_O$ .
- The vertices are also partitioned into layers.
- Any arc must have node  $i$  in layer  $h-1$  and node  $j$  in layer  $h$ .
- Arc is labeled with a numeric value  $w_{ij}$ .
- Node  $i$  is labeled with a function  $f_i$ .

When each edge is assigned an orientation, the graph is directed and is called a directed graph or a diagraph. A feed forward network has directed acyclic graph. Diagraphs are important in neural network theory since signals in NN systems are restricted to flow in particular directions. The vertices of the graph represent neurons (input/output) and the edges, the synaptic links. The edges are labeled by the weights attached to the synaptic links.

Neural networks essentially comprise three pieces: the architecture or model; the learning algorithm; and the activation functions. Neural networks are programmed or “trained” to “. . . store, recognize, and associatively retrieve patterns or database entries; to solve combinatorial optimization problems; to filter noise from measurement data; to control ill-defined problems; in summary, to estimate sampled functions when we do not know the form of the functions.” It is precisely these two abilities (pattern recognition and function estimation) which make artificial neural networks (ANN) so prevalent a utility in data mining. As data sets grow to massive sizes, the need for automated processing becomes clear. With their “model-free” estimators and their dual nature, neural networks serve data mining in a myriad of ways.

## CONCLUSION

Plant Disease management is a challenging task. In that mostly diseases are seen on the leaves of the plant. Basically there are three main types of Leaf disease, they are Bacterial, Fungal and Viral. There is main characteristics of disease detection are speed and accuracy. Hence working on development of automatic, efficient,

fast and accurate which is use for detection disease leaf and stem. Work can be extended for development of machine vision system that automatically recognizes, classify and quantitatively detects leaf disease symptoms. The objective of this work is the detection, classification of leaf diseases affecting on the fruits crops, stem, leaves of the plants and what type of fertilizer is used using image processing tools and all information about the disease