

Application of Image Processing in Diagnosing Guava Leaf Diseases

M.Thilagavathi¹, S.Abirami²

¹Research Scholar, Department of Computer Science, Annamalai University, Chidambaram -608002, India,

²Assistant professor, Department of Computer Science and Engineering, Annamalai University, Chidambaram – 608002, India,

Abstract

Plant and leaf diseases in Guava result in poor plant growth and reduced fruit yields. Leaves are indicators of the health and growth of the Guava shrub/short tree that has its origin in tropical and subtropical regions. Precise diagnosing of diseases is vital, as remedies rely on it. Image processing in the place of manual/visual detection of Guava leaf diseases relieves from difficulties experienced, time consumed and inaccuracy resulted.

In the present work resized leaf images with improved contrasts are subject to region growing segmentation, colour transformation (YCbCr, CIELAB), and Scale Invariant Feature Transform (SIFT). Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN) classifiers have been evaluated for their disease-wise classifying accuracies. 125 leaf samples at 25 per disease class and 128 texture features per sample were used in the study. Though both SVM and k-NN perform reasonably well, the former is slightly superior in terms of accuracy.

Keywords: Region growing segmentation, YCbCr, CIELAB, SIFT feature extraction, SVM classifier, k-NN classifier.

1 INTRODUCTION

Both the fruits and leaves of Guava have high-profile nutrients such as vitamin A, BComplex, C, E and K; Minerals: Copper, Iron, Magnesium, Manganese, Potassium, Sodium and Zinc. The health benefits for the mankind include oral health, heart health, thyroid health, skin health, lower risk of cancers, improved immunity, improved vision, control of diabetes, treating diarrhea and stimulation of cognitive function. Guava leaf supplements in the form of capsules and leaf tea are common due to their medicinal values. Plant and leaf diseases are challenges for the productivity, trading and economical viability and hence cultivation. Precise and fast diagnosis of the diseases followed by remedial measures are essential for sustaining Agro economy and indirectly the human healthcare.

Manual detection of plant/leaf diseases and classification is difficult, slow and inaccurate. Hence the present work attempts to classify using image processing, the leaf samples according to five classes considered namely Algal leaf spot, rust, curl, powdery mildew and viburnum chindo.

2 METHODOLOGY

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

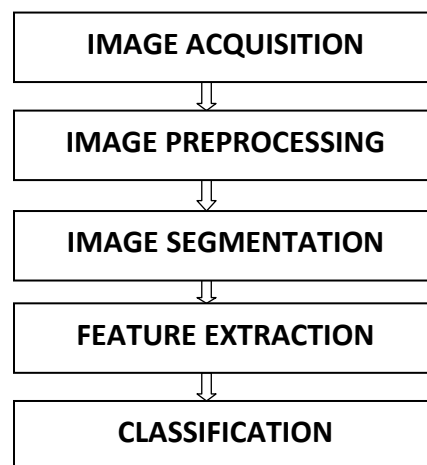


Fig. 1 Basic steps for plant disease detection and classification

2.1 IMAGE ACQUISITION

Guava leaf images have been acquired using a camera. The infected guava leaf is placed on a white background, without any light reflection. The camera is held horizontally to the plane of the leaf. The photograph distance is neither too close nor too far; it is adjusted in such a way that the photograph covers only background and the leaf. A few samples have been downloaded from www.apsnet.org, mrgoutham.blogspot.in, seedbuzz.com. The diseased leaf

images are shown in Fig. 2. The collected image is further applied for preprocessing.



Fig. 2 Sample Guava Leaf Images in the Dataset

2.2 IMAGE PRE-PROCESSING

The captured images undergo preprocessing. In this approach the captured images are resized and their contrast is improved.

2.3 IMAGE SEGMENTATION

Segmentation means partitioning of image into various parts of same features or some similarity. The segmentation can be done using various methods. Here Region growing segmentation is employed to segment the disease affected part of the color image.

2.3.1 Region growing segmentation

The main goal of segmentation is to partition an image into regions. Some segmentation methods such as thresholding achieve this goal by looking for the boundaries between regions based on discontinuities in color properties. Region-based segmentation is a technique for determining the region directly.

Region growing is a region-based image segmentation method, also classified as a pixel-based image segmentation method since it involves the selection of initial seed points. This approach to segmentation examines neighboring pixels of initial seed points and determines whether the pixel neighbors should be added to the region. The process is iterated on, in the same manner as general data clustering algorithms.

2.3.1.1 Basic concept of seed points

The first step in region growing is to select a set of seed points based on some user criterion. The initial region begins as the exact location of these seeds. The regions are then grown from these seed points to adjacent points

depending on a region membership criterion. The criterion could be, for example, pixel intensity, grayscale texture, or color. Since the regions are grown on the basis of the criterion, the image information itself is important. For example, if the criterion were a pixel intensity threshold value, knowledge of the histogram of the image would be of use, as one could use it to determine a suitable threshold value for the region membership criterion. Fig. 3 illustrates the process of region growing. Region growing methods often give very good segmentation that correspond well to the observed edges.

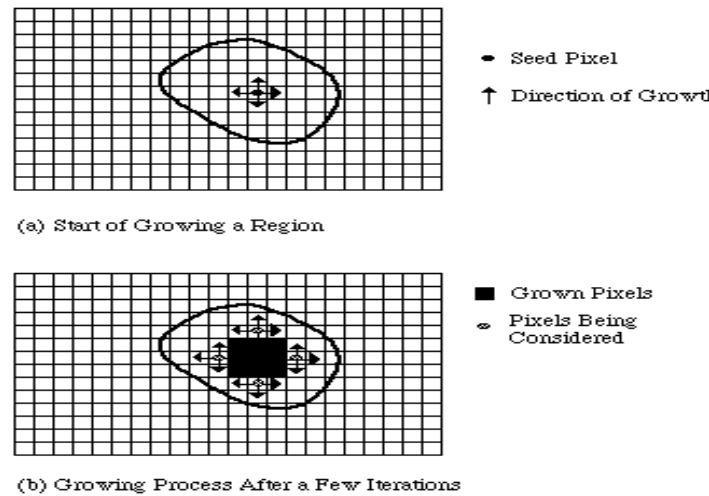


Fig. 3 Regiongrowing illustrated

The region is iteratively grown by comparing all unallocated neighboring pixels to the region. The difference between a pixel's intensity value and the region's mean, is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process stops when the intensity difference between region mean and new pixel become larger than a certain threshold (t). Fig. 4 depicts the diseased guava leaf before and after segmentation.





(b)

Fig. 4 (a) Diseased leaf before Region Growing segmentation (b) Diseased leaf after Region Growing Segmentation

2.4 COLOR TRANSFORMATION

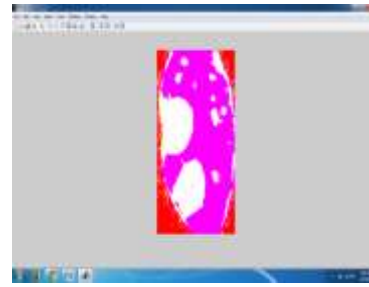
Like segmentation the diseased part of the color image may be detected by the transformation of the color image in YCbCr, and CIELAB. By these transformations the disease affected parts of the leaf will be detected clearly. Fig. 5 shows the original diseased leaf and their transformations respectively.



(a)



(b)



c)

Fig. 5 a) Diseased leaf b) YCbCr conversion c) CIELAB conversion

So from the above steps, the infected portion of the leaf is detected. The infected region is then cropped into a number of patches of equal size. In this approach patch size of 55 x 55 is taken.

The size of the patch is chosen in such a way that the significant information is not lost. Not all segments contain significant amount of information. The patches which are having more than fifty percent of the information are taken into account for the further analysis.

2.5 FEATURE EXTRACTION

For any object there are many features, interesting points on the object, that can be extracted to provide a "feature" description of the object. This description can then be used when attempting to locate the object in an image containing many other objects. SIFT image features provide a set of features of the cropped leaf image that are not affected by many of the complications experienced in other methods, such as object scaling and rotation. These texture features differentiate the diseased leaf from normal leaf.

2.5.1 SIFT - Scale Invariant Feature Transform

The SIFT approach, for image feature generation, takes an image and transforms it into a large collection of local feature vectors. Each of these feature vectors is invariant to any scaling, rotation or translation of the image. To aid the extraction of these features the SIFT algorithm applies a 4 stage filtering approach:

2.5.1.1 Scale-Space Extrema Detection

This stage of the filtering attempts to identify those locations and scales that are identifiable from different views of the same object. This can be efficiently achieved using a "scale

space" function. It has been shown under reasonable assumptions that it must be based on the Gaussian function.

The scale space is defined by the function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

Where * is the convolution operator, $G(x, y, \sigma)$ is a variable-scale Gaussian and $I(x, y)$ is the input image.

Various techniques can then be used to detect stable keypoint locations in the scale-space. Difference of Gaussians is one such technique, locating scale-space extrema, $D(x, y, \sigma)$ by computing the difference between two images, one with scale k times the other. $D(x, y, \sigma)$ is then given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbors at the same scale, and its 9 neighbors up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

2.5.1.2 Keypoint Localization

This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localized on an edge. This is achieved by calculating the Laplacian value for each keypoint found in stage 1. The location of extremum, Z , is given by:

$$Z = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial X}$$

If the function value at \mathbf{z} is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localization, it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the keypoint, the keypoint is rejected.

2.5.1.3 Orientation Assignment

This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, described below, can then be represented relative to this orientation, achieving invariance to rotation. The approach taken to find an orientation is:

- Use the keypoints scale to select the Gaussian smoothed image L , from above
- Compute gradient magnitude, m

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

- Compute orientation, θ

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right)$$

- Form an orientation histogram from gradient orientations of sample points
- Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation
- Some points will be assigned multiple orientations
- Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position

2.5.1.4 Keypoint Descriptor

The local gradient data, used earlier, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of $1.5 * \text{keypoint scale}$. This data is then used to create a set of histograms over a window centered on the keypoint. Keypoint descriptors typically use a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This results in a feature vector containing 128 elements. These resulting vectors are known as SIFT keys and are used in a nearest-neighbors approach to identify possible objects in an image.

2.6 CLASSIFICATION

The features extracted are used for classification of the diseases. SVM and k-NN classifiers are used for classifying.

2.6.1 SVM Classification

More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification,

regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

The vectors defining the hyperplanes can be chosen to be linear combinations with parameters of images of feature vectors that occur in the data base. With this choice of a hyperplane, the points in the feature space mapped into the hyperplane are defined by the relation: it becomes small as grows further away from , each term in the sum measures the degree of closeness of the test point to the corresponding data base point . In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated.

2.6.2 k-Nearest Neighbor

k-Nearest Neighbor is a simple classifier in the machine learning techniques where the classification is achieved by identifying the nearest neighbors to query examples and then making use of those neighbors for determination of the class of the query. In k-NN the classification i. e. to which class the given point belongs is based on the calculation of the minimum distance between the given point and other points. For the plant leaf classification the Euclidean distances between the test samples and training samples are calculated. In this way it finds out similar measures and accordingly the class for test samples. A sample is classified based on the highest number of votes from the k neighbors, with the sample being assigned to the class most common amongst its k nearest neighbors. k is a positive integer, typically small. If $k = 1$, then the sample is simply assigned to the class of its nearest neighbor. Nearest neighbor method is easy to implement, also provide quite good results if the features are chosen carefully.

3 RESULTS AND DISCUSSION

The segmentation is based on two principles-discontinuity and similarity. Discontinuity extracts the regions having different properties like intensity, colour and texture . Similarity groups the image pixels with some predefined

criteria. Based on pixel similarity with the neighbouring pixel, the algorithm used is region based. In leaf disease identification, segmentation is used to identify the diseased area. From this, features of a region are computed. Fig. 6 shows the results of the segmentation and color transformation of the diseased guava leaf.

The authors in the present work have generated training texture feature data file and testing texture feature data file for a total of 125 Guava leaf samples (25 per class). The diseases classes considered are algal leaf spot, rust, curl, powdery mildew and viburnum chindo. A MATLAB program, trains the classifier using training data and classifies using the test data. The results of the study are encouraging, justifying the approach and methodology adopted. The accuracies for the SVM and k-NN classifiers exceed 87 percent. Of the two, SVM is moderately superior. Fig. 6 and Table 1 illustrate the results of the study carried out.

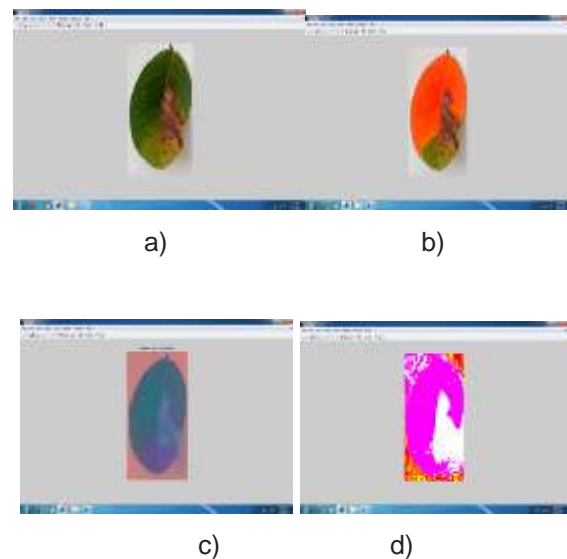


Fig. 6 a) **Viburnum Chindo** diseased guava leaf b) leaf image after segmentation c) leaf image after **YCbCr** color transformation d) leaf image after **CIELAB** color transformation.

Table 1 Classification results per class for proposed approach

Leaf Disease	Accuracy with k-NN (%)	Accuracy with SVM (%)
Algal leaf spot	88	98.2
Rust	96	98.2
Curl	92	95.45
Powdery mildew	92	95.45
Viburnum chindo	92	98.2

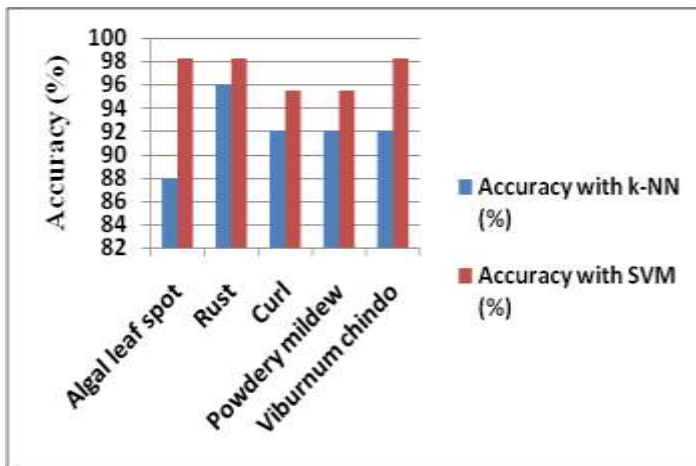


Fig. 7 Classification results as per class for proposed approach

CONCLUSION

The proposed methodology involving region growing segmentation, colour transformation ($YCbCr$, $CIELAB$), SIFT, and MATLAB works well with both the classifiers and the data sets. SVM is slightly superior to k-NN. The evaluation of the classifiers for large datasets, using the present methodology can be attempted in future.

References

- [1] Savita N. Ghaiwat, Parul Arora, Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review, International Journal of Recent Advances in Engineering &

Technology 2347 - 2812, Volume-2, Issue - 3, 2014.

- [2] Manoj Mukherjee, Titan Pal and Debabrata Samanta, Damaged Paddy Leaf Detection Using Image Processing, Journal of Global Research in Computer Science Volume 3, No. 10, October 2012.
- [3] Prof. Sanjay B. Dhaygude, Mr.Nitin P.Kumbhar, Agricultural plant Leaf Disease Detection Using Image Processing, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 1, January 2013.
- [4] Arti N. Rathod, Bhavesh Tanawal, Vatsal Shah, Image Processing Techniques for Detection of Leaf Disease, International Journal of Advanced Research in Computer Science and Software Engineering, 2277 128X, Volume 3, Issue 11, November 2013.
- [5] Jagan Mohan K., Balasubramanian M., Palanivel S, Detection and Recognition of Diseases from Paddy Plant Leaf Images, International Journal of Computer Applications – 8887), Volume 144 – No.12, June 2016.
- [6] Surender Kumar, Rupinder Kaur, Plant Disease Detection using Image Processing- A Review, International Journal of Computer Applications (0975 – 8887) Volume 124 – No.16, August 2015.
- [7] Sujeet Varshney, Tarun Dalal, Plant Disease Prediction Using Image Processing Techniques- A Review, International Journal of Computer Science and Mobile Computing, 2320–088X, Vol. 5, Issue. 5, May 2016,
- [8] Sushil R. Kamlapurkar, Detection of Plant Leaf Disease Using Image Processing Approach, International Journal of Scientific and Research Publications, 2250-3153, Volume 6, Issue 2, February 2016.
- [9] Rajat Kanti Sarkar, Ankita Paramanik, Segmentation of plant Disease Spots Using Automatic SRG Algorithm : A Look Up Table

Approach, International Conference on Advances in Computer Engineering and Applications(ICACEA), 2015.

- [10] Prakash M. Mainkar, Shreekant Ghorpade, Mayur Adawadkar, Plant Leaf Disease Detection and Classification Using Image Processing Techniques, International Journal of Innovative and Emerging Research in Engineering Volume 2, Issue 4, 2015.
- [11] Jinho Kim, Byung-Soo Kim, Silvio Savarese, Comparing Image Classification Methods:K-Nearest-Neighbor and Support-Vector-Machines, Applied Mathematics in Electrical and Computer Engineering, ISBN: 978-1-61804-064-0.
- [12] Shilpa Kamdi, R.K.Krishna, Image Segmentation and Region Growing Algorithm, International Journal of Computer Technology and Electronics Engineering (IJCTEE), Volume 2, Issue 1.
- [13] Jian Wu¹, Zhiming Cui¹, Victor S. Sheng, Pengpeng Zhao¹, Dongliang Su¹, Shengrong Gong¹, A Comparative Study of SIFT and its Variants, Measurement Science Review, Volume 13, No. 3, 2013.
- [14] Kewal Krishan, Sukhjit Singh, Color Image Segmentation Using Improved Region Growing and K-Means Method, IOSR Journal of Engineering (IOSRJEN), Vol. 04, Issue 05, May 2014.
- [15] Kinjal M. Gandhi & Prof. R. H. Kulkarni, Sift Algorithm for Iris Feature Extraction, Global Journal of Computer Science and Technology: F Graphics & Vision Volume 14 Issue 3 Version 1.0 Year 2014.