

Automatic detection of rust disease of *Lentil* by machine learning system using microscopic images

Kuldeep Singh¹, Satish Kumar², Pawan Kaur³

¹Department of Computer Science, Panjab University, India

²Department of Computer Applications, Panjab University SSG Regional Centre, India

³Department of Biotechnology, Chaudhary Devi Lal University, India

Article Info

Article history:

Received Apr 14, 2018

Revised Sep 4, 2018

Accepted Sep 26, 2018

Keywords:

Brightness Bi-Histogram
Equalization

Lentil

Local binary pattern

Microscopic image

Rust disease

ABSTRACT

Accurate and early detection of plant diseases will facilitate mitigate the worldwide losses experienced by the agriculture area. MATLAB image processing provides quick and non-destructive means of rust disease detection. In this paper, microscopic image data of rust disease of *Lentil* was combined with image processing with depth information and developed a machine learning system to detect rust disease at early stage infected with fungus *Uromyces fabae* (Pers) de Bary. A novel feature set was extracted from the image data using local binary pattern (LBP) and HBBP (Brightness Bi-Histogram Equalization) for image enhancement. It was observed that by combining these, the accuracy of detection of the diseased plants at microscopic level was significantly improved. In addition, we showed that our novel feature set was capable of identifying rust disease at haustorium stage without spreading of disease.

Copyright © 2019 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Pawan Kaur,

Department of Biotechnology, Chaudhary Devi Lal University,

Sirsa-125055, Haryana, India.

Email: pawankaurnano@gmail.com

1. INTRODUCTION

India is an agriculture based country and agricultural productivity directly affects the economical growth of a developing country. Plant diseases are the major concern in crop productivity. Automatic detection of disease in plants is an emerging area for contemporary day biocomputing science. Image processing and pattern recognition techniques have paved the way for accurate and early detection of fungal diseases in agriculture [1-3]. MATLAB image processing has been used by various researchers to study the effect of disease on a plant [4-7].

Among various imaging techniques, digital imaging has been shown to be a robust technique for identification of fungal diseased regions on leaves [8, 9]. One amongst the main issue related to digital imaging in plants is the late detection of rust disease because at visual stage rust disease could easily spread with insects and wind to other fields and couldn't be controlled at an early stage. So we need better technique to close the gap between visualization of disease and objective quantitative results to control rust disease at an early stage before spreading to other fields. Microscopic images could be better alternatives to detect rust disease at early i.e. haustorium stage [10, 11]. Seiffert et al. (2005) used pattern recognition software for automatic detection of hyphal growth of powdery mildew disease of barley i.e. *Blumeria graminis* f. sp. *Hordei* [12].

In this paper, we tend to combine microscopic images with image processing techniques to overcome these issues and present a method for automatic detection of rust disease at an early stage using machine learning techniques. This work is a step towards creating a novel approach for automatic detection of rust disease of pea plants caused by fungus *Uromyces fabae* (Pers) de Bary before the appearance of

visible symptoms or before spreading of disease. For depth estimation, we proposed two complimentary approaches Local binary patterns (LBP) to extract a novel feature set and BBHE (Brightness Bi-Histogram Equalization) for image enhancement to show that it is capable of automatic detection of leaves poised to be suffering from rust throughout the experiment.

2. RESEARCH METHOD

2.1. Image acquisition

An experimental setup was designed and developed to simultaneously acquire microscopic images of leaves to distinguish leaves with or without rust disease. The imaging setup consisted of a light microscope with the camera (LIECA). The experiment was carried out on 300 images of leaves were collected for 15 consecutive days and stained with congo red dye to improve the appearance of fungal haustorium. The disease symptoms that developed consisted of haustorium that expanded over time in the form of spores and eventually caused chlorosis and ready for spreading to other plants as well as fields. Figure 1a showed the haustorium of rust after staining.

2.2. Selection of an image

Initially, an image had selected by the user on which the operation was performed. In MATLAB, “uigetfile” function was used to browse an image from everywhere in the computer. The syntax used to get the image was as follows:

```
[File Path]=uigetfile ('.jpg', 'Select An Image');
```

Where the get image was stored in a matrix in the form of the file name and corresponding path.

After selecting an image, a few steps were performed to make it feasible for further procedures.

- a. Read the selected image using “imread” function available in the MATLAB.
- b. Resize the read image in order to reduce the processing time.
- c. Lastly, show the selected image considered as an original image using “imshow” function of the MATLAB

2.3. Application of median filter

Once the image had selected and output had shown, a median filter was applied over the selected image. The application of this filter was smoothening of the actual image while preserving the edges. The median filter uses a window size where one entry immediately preceding and following each entry. In the proposed work, the median filter was applied to the image to smoothening of the image as shown in Figure 1b which had obtained after applying median filter onto the original image.

- a. Allocate the actual size of image i.e. width, height, and layer.
- b. Assign the size of the window where its width and height is initialized.
- c. Set for loop for the number of layers from 1 to total size of the image.
- d. Initialize the loop for rows from 1 to total size of the image rounded down by 2.
- e. Initialize the loop for Columns from 1 to total size of the image rounded down by 2.
 - e.1. Now initialize the for loop x from 1 to 3
 - e.2. Initialize the for loop y from 1 to 3

Shift the window over the output image one by one and increment the location of window by 1.

- f. End of the for loop y
- g. End of the for loop x
- h. Evaluate the median filter of the obtained window.
- i. Demonstrate the acquired smoothened image.

2.4. Apply BBHE enhancement

Once the smoothened image had acquired from the previous step, next step was to enhance the image. For the enhancement purpose, BBHE technique was used. The BBHE i.e. Brightness Bi-Histogram Equalization algorithm preserves the mean brightness of a given image while enhancing the contrast of that image. After apply BBHE for the enhancement purpose over the image, the acquired enhanced output image has shown in Figure 1c and Figure 1d.

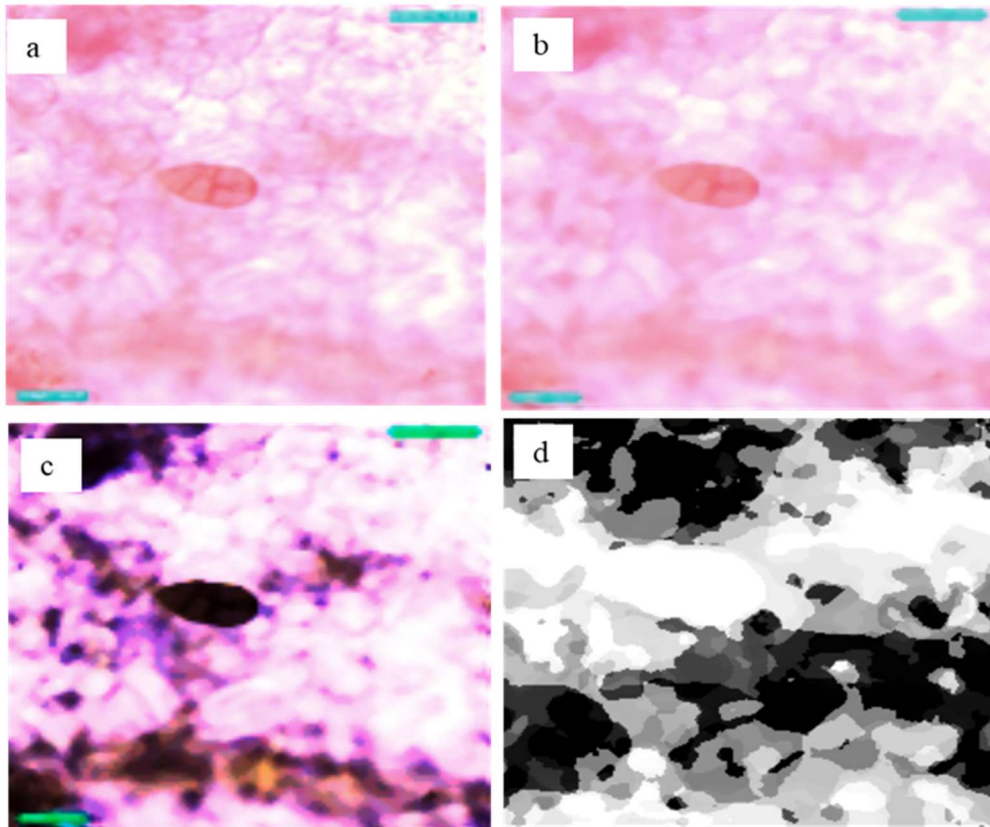


Figure 1. (a) Original Image; (b) Smoothened Image using Median Filter; (c) enhanced image after applying BBHE; (d) Feature extracted from an image using Local Binary pattern

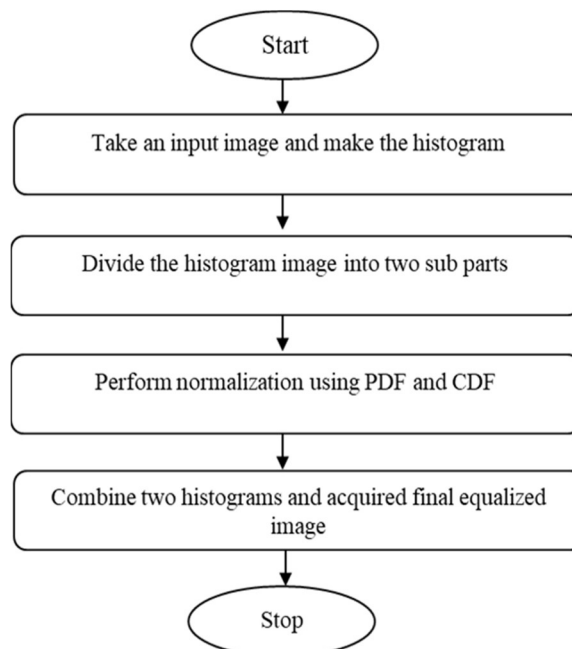


Figure 2. Flowchart of the BBHE algorithm

2.5. Algorithm for BBHE

- a. Evaluate the mean of the smoothened image and considered as a mean threshold value.
- b. Generate two histograms i.e. lower histogram and upper histogram for the evaluation of a number of count of each color.
- c. Set a for loop i for number of rows.
- d. Set a for loop j for number of columns.
- e. Now compare the pixel value of an image with the mean threshold value and decomposed the original image into sub-images as lower_histogram and Upper_histogram.
 1. If the mean threshold value is greater than the pixel's value than it will be lower histogram.
 2. If the mean threshold value is lesser than the pixel's value than it will be upper histogram.
- f. End of the for loop j
- g. End of the for loop i
- h. Calculates the PDF i.e. Probability Density Function of the lower and upper histogram for normalization.
 1. Normalize_lower_histo=histo_lower/sum(histo_lower);
 2. Normalize_upper_histo=histo_upper/sum(histo_upper);
- i. Evaluate the Cumulative Density Function for lower and upper range.
- j. Initiate the for loop K from 2 to 255 and estimates the lower and upper histogram's CDF.
 1. hist_lower_cdf(k)=hist_lower_cdf(k-1) +Normalize_lower_histo (k);
 2. hist_upper_cdf(k)=hist_upper_cdf(k-1) +Normalize_upper_histo (k);
- k. Initiate the loop i from 1 to total number of rows.
- l. Initiate the loop j from 1 to total number of columns.
- m. Define transform function based on the CDF such as:
 1. final_image(i,j)=range_lower(1)+round(((range_lower(2)-range_lower(1))*hist_lower_cdf(original_pixel_value+1)));
 2. final_image(i,j)=range_upper(1)+round(((range_upper(2)-range_upper(1))*hist_upper_cdf(original_pixel_value+1)));
- n. The final attained histogram equalized image will be expressed as:
EnhancedImage(:, :, ij)=Equilzed_BBHE (Ouput Image (:, :, ij));

2.6. Feature extraction using LBP

After enhancing the quality of the input image, features were extracted from that image. For the feature extraction purpose, in the proposed work, Local Binary Patterns operator was used. This operator was basically used to describe the texture and shape of a digital image. The process was done by dividing the whole image into small regions. Further, these regions were used to extract the features as shown in Figure 1(d) and flow chart of LBP in Figure 3. The steps followed by the LBP were:

- a. Initialize the size of the LBP label "L".
- b. Compute the center of the label by dividing it with 2.
 $C = \text{round}(L/2)$
- c. Initialize the size of the window by computing row_max and col_max.
 $\text{row_max} = \text{size}(\text{Input_Image}, 1) - L + 1;$
 $\text{col_max} = \text{size}(\text{Input_Image}, 2) - L + 1;$
- d. Compare each pixel of the cell to its neighbors such as Left-top, Left-middle, left bottom, right-top, right-middle, right-bottom etc. Follow the pixels along a circle such as clockwise or counterclockwise.
- e. Compare the center pixel's value, if greater than "0" write "1". This process will end up with 8-digit binary number.
- f. Convert the generated binary number into decimal using below equation.
 $\text{LBP_Image}(i, j) = (\text{Center}, \text{Left}) + (\text{Left}, \text{Left}) * 2 + (\text{Left}, \text{Center}) * 4 + (\text{Left}, 1) * 8 + (\text{Center}, 1) * 16 + (1, 1) * 32 + (1, \text{Center}) * 64 + (1, \text{Left}) * 128;$
- g. The attained pixel is the LBP output image.

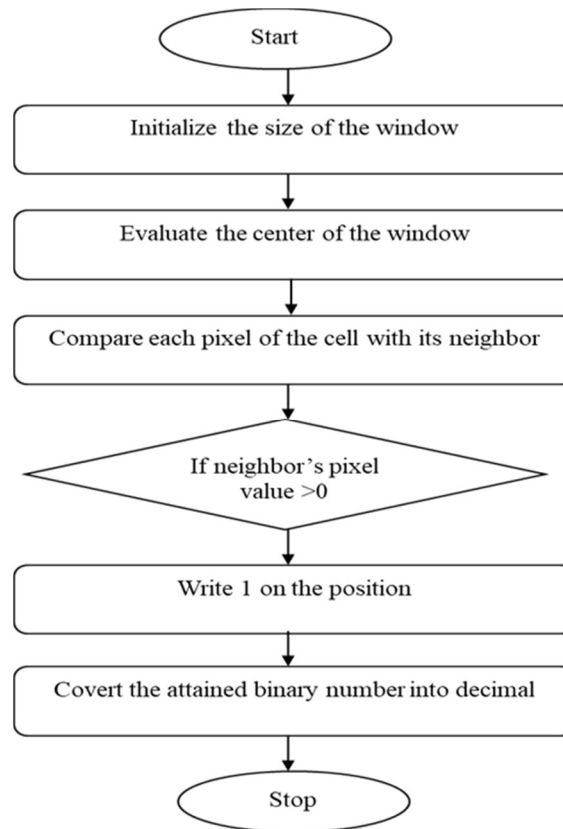


Figure 3. Flowchart of the LBP feature extraction technique

2.7. Evaluate the color value to estimates the intensity value of the input image

In this step, the intensity of the image was evaluated. Thus, the steps followed for the same were:

- Evaluate the size of the image.
- If the numbers of layers are three then extract each layer from an image i.e. red, green and blue.
`r=image(:,:,1); % red color`
`g=image(:,:,2); % green color`
`b=image(:,:,3); % blue color`
- Calculate the mean of each layer individually:
`red=mean(mean(r));`
`green=mean(mean(g));`
`blue=mean(mean(b));`
- Now evaluate the mean of all the layers:
`main_value=(red+green+blue)/3;`
- Resultant value will be the color value.

2.8. Identify the shape value of the image

This step provided the shape value of an image in the form of numeric value. The higher the Value, the chances of having circle shape were also high. Algorithm to evaluate the shape was as follows:

- Take the RGB image.
- Convert the RGB image into gray format.
- Evaluate the threshold value of an image.
`Threshold=graythresh(image);`
- Convert the threshold image into black and white image.
- Remove the noise by eliminating the entire object that contains fewer than 30 pixels.
- Find the corresponding boundaries
 - Display the label matrix and draw each boundary using the for loop k.
`boundary=B{k};`

- g. Determine the round objects by evaluating the area and center using regionprops function.
 1. Initiate a for loop from 1 to total length of the boundaries.
 2. Obtain boundary coordinates.
 3. Compute the perimeter of an object:
 $\text{delta_square} = \text{differences}(\text{boundary})^2$;
 $\text{perimeter} = \text{sum}(\text{square root}(\text{sum}(\text{delta_sq}, 2)))$;
 4. Obtain the total area calculated.
 5. Compute the roundness of the metric.
 $\text{Metric} = 4 * \pi * \text{area} / \text{perimeter}^2$;
 6. End of the for loop
- h. Acquired the shape value of an image.

3. RESULTS AND ANALYSIS

In this work, image processing of rust disease was performed as shown in Figure 4. At the time of acquisition, the image may be corrupted due to noise that degrades the actual quality of the image. So, in order to extract the features, it should be enhanced firstly. In the proposed work, initially, pre-processing steps have followed where the image is filtered and enhanced and then features are extracted from the enhanced image as shown in flowchart of the proposed methodology. The output image that has acquired after apply BBHE algorithm over the smoothened image. From the result, it has been cleared that the output image is more enhanced and bright in comparison with the image which has attained in the previous step as shown in Figure 4. After feature extraction color and shape value of the input image were found 191.5965 and 0.7213, respectively. Previously, automatic detection methods for rust disease of the Wheat crop were reported [13, 14]. Image enhancement and feature extraction were reported by many scientists for the fungal disease of crops [1, 2, 15, 16].

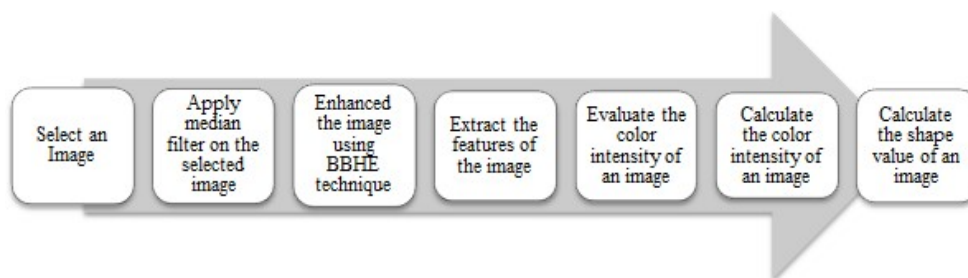


Figure 4. Framework of the proposed model

4. CONCLUSIONS

This paper exhibited the utilization of microscopic images to detect rust disease on lentil leaves. The different lentil varieties did not influence the algorithm's ability to recognize rust disease on leaves. BBHE techniques produced the best image contrast enhancement, without unfortunate antiquities and maintain input mean brightness of images. This investigation endeavored to check the achievability of an image processing technique for rust disease detection and the outcomes introduced are promising. Future work is aimed to carry out a similar experiment to detect rust of other crops. One of the fundamental difficulties looked in this investigation was staining protocol which causes a commotion in images. The primary target will be to modify and improve the algorithm to develop a robust algorithm that can make exact recognition of rust disease in field conditions.

REFERENCES

- [1] M.M. Kamal *et al.*, "Classification of Leaf Disease from Image Processing Technique," *Indonesian Journal of Electrical Engineering and Computer Science*. vol.10, pp. 191-200, 2018
- [2] P. Mitkal *et al.*, "Leaf Disease Detection and Prevention Using Image processing using Matlab," *International Journal of Recent Trends in Engineering & Research*. vol. 02, pp. 26-30, 2016.
- [3] K. Singh *et al.*, "Detection of Powdery Mildew Disease of Beans in India: A Review," *Oriental Journal of Computer science & Technology*. vol. 9, pp. 226-234, 2016.

- [4] Q. He, et al., "Cotton Pests and Diseases Detection based on Image Processing," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol.11, pp. 3445-3450, 2014.
- [5] S.Raut and A. Fulsunge, "Plant Disease Detection in Image Processing Using MATLAB," *International Journal of Innovative Research in Science, Engineering, and Technology*. vol. 6, pp.10373-10381, 2017.
- [6] G.M. Choudhary and V. Gulati, "Advance in Image Processing for Detection of Plant Diseases," *International Journal of Advanced Research in Computer Science and Software Engineering*. vol. 5, pp.1090-1093, 2015.
- [7] N. D. Kartika, et al., "Oil Palm Yield Forecasting Based on Weather Variables Using Artificial Neural Network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 3, pp. 626-633, 2016.
- [8] K. Singh et al., "Local Binary Patterns Based Detection of Rust Disease of Lentils (*Lens culinaris*) using K-NN Classification System," *International Journal of Computer Science Engineering and Information Technology Research*. vol.7, pp. 47-52, 2017.
- [9] K. Singh et al., "Support Vector Machine Classifier Based Detection of Fungal Rust Disease in Pea Plant (*Pisam sativam*)," *International Journal of Information Technology* DOI 10.1007/s41870-018-0134-z.
- [10] T. Baum, et al., "HyphArea-Automated Analysis of Spatiotemporal Fungal Patterns," *Journal of Plant Physiology*. vol.168, pp. 72-78, 2011.
- [11] S. Hartati, et al., "The Digital Microscope and Its Image Processing Utility," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 9, pp. 565-574, 2011.
- [12] U. Seiffert and P. Schweizer, "A Pattern Recognition Tool for Quantitative Analysis of In Planta Hyphal Growth of Powdery Mildew Fungi," *Molecular Plant-Microbe Interactions* vol.18, pp. 906-912, 2005.
- [13] H. Sabrol and S. Kumar, "An Identification of Wheat Rust Diseases in Digital Images: A Review", *International Journal of Computer Science Engineering and Information Technology Research*. vol. 3, pp. 85-94, 2013.
- [14] J. Varghese Pt et al., "Identification of Wheat Rust Disease in Digital Images," *International Journal of Engineering Research in Electronic and Communication Engineering*. vol. 3, pp. 65-68, 2016.
- [15] D.P. Jagadeesh et al., "Neuro-kNN Classification System for Detecting Fungal Disease on Vegetable Crops using Local Binary Patterns," *Agric Eng Int: CIGR Journal*. vol.16, pp. 299-308, 2014.
- [16] H. Sabrol, et al., "Recognition of Tomato Late Blight by using DWT and Component Analysis. *International Journal of Electrical and Computer Engineering (IJECE)*. vol.7, pp. 194-199, 2017

BIOGRAPHIES OF AUTHORS



Mr. Kuldeep Singh is research scholar of Department of Computer Science and Application, Panjab University, Chandigarh, India. He is presently working as assistant professor at FGM Govt. College, Adampur, Hisar, Haryana, India. His research interests include image processing and pattern recognition. He has more than 10 years teaching and research experience in the field of computer applications.



Dr. Satish Kumar, Ph.D., is an Associate Professor at the department of computer science and application, Panjab University, Swami Sarvanand Giri, Regional Centre, Hoshiarpur, Panjab, India. He has received Two R&D projects from UGC. His research interests include image processing and pattern recognition. He has more than 20 years teaching and research experience in the field of computer applications. He has supervised more than 150 project work in due course of MCA curriculum and industrial training.



Dr. Pawan Kaur, Ph.D., is an Assistant Professor (contractual) at the Department of Biotechnology, Chaudhary Devi Lal University, Sirsa, India. She has post Ph.D. research experience on drug-loaded nanoformulations against *trypanosome evansi*, the causal agent of surra disease in equines. Her area of interests includes synthesis of nanoparticles, nanocomposites, antimicrobial activity & cytotoxicity of nanoformulations and application of nanoformulations in agriculture biotechnology against plant diseases.