



Detection of paddy disease using image processing

Md Sohag Babu^{1✉}, Md Mosharof Hossain², Md Tanvir Hasan Sagar³, Raziun Uddin Ahmed Chowdhury⁴

¹Department of computer science and Engineering, Jahangirnagar University, Dhaka, Bangladesh

²Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh

³Department of Electrical and Electronic Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh

⁴Department of Electrical and Electronic Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh

✉Corresponding author

Md Sohag Babu

Department of computer science and Engineering

Jahangirnagar University,

Dhaka, Bangladesh

Email: shahidul.uni@gmail.com

Article History

Received: 04 October 2020

Accepted: 07 November 2020

Published: November 2020

Citation

Md Sohag Babu, Md Mosharof Hossain, Md Tanvir Hasan Sagar, Raziun Uddin Ahmed Chowdhury. Detection of paddy disease using image processing. *Indian Journal of Engineering*, 2020, 17(48), 551-568

Publication License



© The Author(s) 2020. Open Access. This article is licensed under a [Creative Commons Attribution License 4.0 \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).

General Note



Article is recommended to print as color digital version in recycled paper.

ABSTRACT

Paddy is considered one of the top food items across the world. So, it usually gets more priority to save it from various diseases to prevent the loss of the farmer. Only proper identification can be a key tool to protect crop, increasing yield and reduce the losses. Technical tactics the use of machine getting to know and computer vision are actively researched to obtain intelligence farming through early detection of paddy ailment. A mobile software is manifestly ideal to resource the farmers in diagnosing what kinds of

illnesses a paddy has. Although some similar applications exist, most of them achieve the function by submitting the image to a team of plant pathologists or expert garden advisors to get possible identification results and some advice. This have a look at presents the research, design, and implementation of a mobile software that could mechanically pick out the paddy sicknesses based on its leaf look with some pc imaginative and prescient and system getting to know approach. Many experiments and reviews on exceptional segmentations, characteristic extractions, and classification methods had been performed to locate the simplest method. The target institution of the consumer is individuals who request an unfastened and brief analysis of common paddy disorder at any time of the day.

Keywords: Rice disease, machine learning, agricultural device, digital agriculture, smart farming

1. INTRODUCTION

Rice, referred as *Oryza sativa* in botanical classification, is the main food for more than half of the sector's population (M. D. Il Islam, et al. 2020). Rice is the staple food of about 165 million people of Bangladesh (Haider et al. 2015). It provides almost 48% of rural employment, about -thirds of general calorie deliver and about one-half of the full protein intake of a human (Sarker et al. 2020). Rice zone contributes one-half of the rural GDP and one-sixth of the country wide profits in Bangladesh (Alam et al. 2019). About 75% of the overall cropped region and over 80% of the entire irrigated place is planted to rice (Md Saiful Islam et al., 2020). Consequently, rice performs a critical position in the livelihood of the people of Bangladesh (Sarker, 2016b). But, there are numerous elements that make paddy rice production turn out to be sluggish and less efficient (Sarker & Jie, 2017). One of the fundamental elements is a paddy sickness. A bizarre circumstance that injures the plant or leads it to feature improperly is known as a sickness (Nasrin, Sarker, & Huda, 2019). Sicknesses are readily diagnosed by way of their signs and symptoms. There are quite a few paddy disorders types which can be pink disorder virus, brown spot disease and many greater. Image processing and information mining are very beneficial to the agricultural enterprise. They may be greater potential and more important to many areas in agricultural technology.

As an agricultural country, Bangladesh gets its one-sixth of national income from rice (Sarker, Islam, et al., 2019). About 10.5 million hectares' lands produce 25.0 million tons' rice every year. Now govt.'s target is to produce another 30 million over the next 20 years. Paddy can be harvest twice in a year. Most of the paddy farmer faces many issues to reap their paddy due to the fact they were attacked through many illnesses (Sarker, Islam, Murmu, & Rozario, 2020). While the paddy have been attacked, the others areas were exposed to be inflamed (Sarker, Wu, et al., 2019). Thus, it will lower paddy farmer's earnings and cause giant losses to the farmer (Sarker, Yang, Lv, Enamul, & Kamruzzaman, 2020). Currently, the paddy farmer determines the kind of ailment manually. The errors would possibly arise so that you can determine the sort of sicknesses. Paddy farmer additionally has to spend a number of time to discover the sort of sickness (Akher, Sarker, & Naznin, 2018). It also takes a time as the paddy farmers manually test the ailment for the reason that paddy discipline is in a wide vicinity (M. S. Islam, Khanam, & Sarker, 2018). It is possible to identify those diseases with the color of paddy leaves (Prodhan, Sarker, Sultana, & Islam, 2017). The common research questions in this thesis are which segmentation technique is best for this situation? How to reduce the image into an appropriate size? And how can we detect paddy leaf diseases?

There are numerous elements that make paddy rice manufacturing come to be slow and much less efficient (Sarker, 2016a). One of the predominant factors is a paddy disease. Previous studies only focus on 3 kinds of diseases like leaf Blast, brown spot, and Bacterial leaf blight (Haider et al., 2015). The development of the application has met its specification successfully. For the given leaves with three categories of disease, the application can recognize all them. From existing research and gathering knowledge we fixed our scopes range. Existing research most of them using one segmentation technique. In this research, we have used two different segmentation technique and four Classification technique. After the procedure, it is possible to detect paddy leaf diseases more perfect and accurate.

2. PROPOSED METHODOLOGY DESIGN

2.1. Architecture design

In order to ease the implementation, verification, and documentation, the system could be divided into three different components based on its functionalities (Ngugi, Abelwahab, & Abo-Zahhad, 2020). Figure 1 shows the architecture of the program. The first part is the desktop application as a client which provide a user interface to interact with the functions of the application (Mahmud Sultan et al., 2020). The user can select an image from the gallery of the mobile phone as the leaf image to be diagnosed. Once the users decide which leaf picture, he/she wants to use, the Android application will send the image to the second part. The second part is

the server implemented with java servlet which aims to communicate between client and main functions (Tichkule & Gawali, 2017). The servlet will open a socket on a port for each connection and receive an input image and call the main function on the third part, finally sent the resulting image back to the client. The third part is the main component of the application which is used to recognize the category of the disease and implemented by MATLAB, and it is also the most difficult and central program (Barbedo, 2016). The diagnosis program design would be roughly introduced in the next section. After the recognition, the resulting image would be saved in the server for returning to the client (Figure 1).

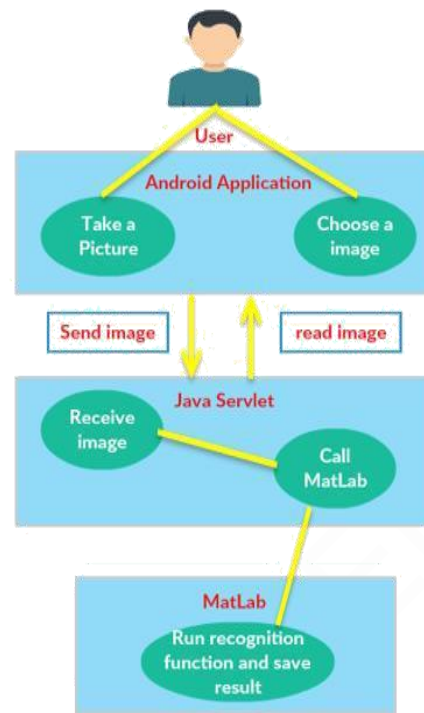


Figure 1. User Interaction Flow Diagram

2.2. User interface design

Figure 1 shows the graphical user interface draft of the application in the android mobile phone. As the difficulty of the project focused on the implementation of the diagnosis algorithm, the function buttons and provided operation are limited. There are two functional buttons at the bottom of the screen which allows the user to choose whether they want to use an existing leaf image in the gallery or take a new photo of a leaf (Gharat, 2012). Above the buttons, the user can see the diagnosis result in this area which contains category labels for every leaf.

2.3. Design for main component

This section will focus on the design of the main component development plan on the server sides. As the recognition program programmed with MATLAB is the central problem-solving part, it is necessary to introduce how the program designed to be followed.

3. PROPOSED METHODOLOGY

3.1. Leaf segmentation

The EM color segmentation segments the image into different images based on the iterative EM algorithm. An RGB image consists of many pixels, and every pixel consists of three elements which represent for red, green and blue respectively. It can also be viewed as red, green and blue three planes combined together (Tichkule & Gawali, 2017). According to the Expectation Maximization algorithm, an algorithm was implemented to estimate the prior possibility of the mixture of Gaussians gave a set of pixel data, and then each pixel can be classified into the corresponding category by possibility comparison (Chaware et al. 2017). To visualize the process, the function was tested by giving a leaf picture and 3 categories, and the classification result during each iteration was plotted. Since the quality of the given pixel is too high, we have to resize the image into a smaller one for further classification (Bakar et al., 2018). For a quick and clear plot, the image was resized into the one width 100.

Visualization of the changes of clustering for each pixel during the EM process, which was implemented with MatLab. The red, green points represent the pixel data clustered into three kinds of categories. In order to save the segmentation time, stopping criteria should be introduced. If the error rate is less than a certain value (0.1), the iteration will stop. After that, we need to partition the color image into some colored segments with the segmentation labels for each pixel (Khairnar & Dagade, 2014).

3.2. Post processing

Although the EM segmentation can perform well for most images, there are still some serious problems for some particular images. Therefore, some post-processing the technique can be taken for a better segmentation result (Khirade & Patil, 2015). Look at Figure 2, (a) is one of segmentation result images, however, some coins and blue edges can still not be removed. For some defects in the image, we can use the color threshold to get rid of some defects. Thus a keeping green and yellow method were added to post-process the image by a color threshold. Picture (b) is the result of the method, which looks good. Next, we want to get and separate each leaf out of the image, so we use connected neighborhood method to label each connected pixel as an object. However as shown in Figure 2 (a), there are lots of small pixels which are isolated which will be still labeled. To improve the efficiency and save the execution time, the small connected component in a binary image would be removed that have fewer than 50 pixels. The results of the operation are shown in (Figure 2).

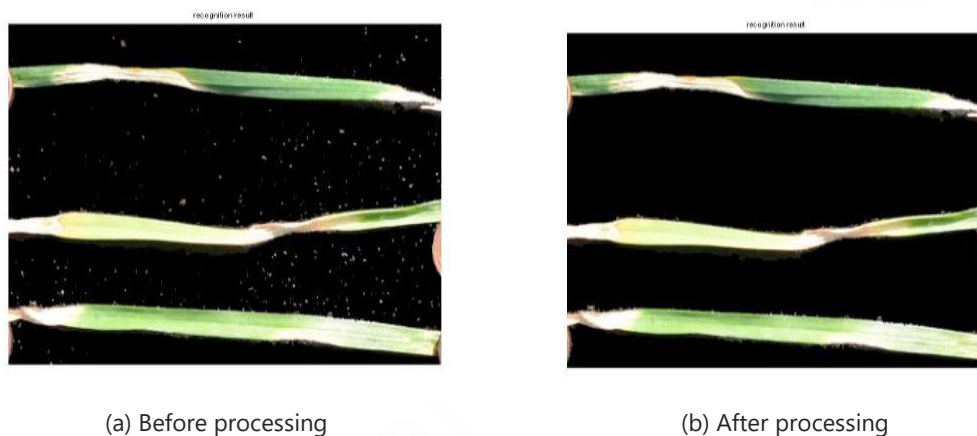


Figure 2. Processing segmented image with a function to get rid of some irrelevant color

For each label connected component, our goal is to sieve the leaf out. First, we transfer it into RGB color image and then the sum of the RGB value of each pixel was calculated, which denoted S . Via experiments, it was found that the ratio of the sum of red value by green value should be less than 1.1 and less than 0.855 for the ratio by blue by green. After that, the program calculates the maximum and minimum boundary of the object and then extract it out (Barbedo, 2016).

3.3. Disease segmentation

The aims of segmentation are for getting useful and discriminate features to train a classifier. Both leaf features and disease features were used to do an experiment to compare which classification result is better (Maity et al., 2018). As we discussed in the last section, an EM segmentation algorithm has been implemented, which can be used to segment the leaf disease. In some ideal cases, EM works very well for disease segmentation. However, some fatal problems still exist as an automatic disease recognition smart phone application (Bapat, Sabut, & Vizhi, 2020). In terms of the disease segmentation, the EM segmentation function was modified a little by partitioning the leaf into two elements which are leaf and disease without considering the black background. For a given clear image, some segmentation results shown in (Figure 3). From the result, it can easily be seen that the segmentation is extremely good.

However, there are two kinds of fatal limitations which make EM algorithm for disease segmentation impractical within a smartphone application (Larijani et al. 2019). One problem is that the segmentation time is too long to get a result from an interactive application. Another one is that the original leaf image must be very clear and contain no defects (Khirade & Patil, 2015). If not, the segmentation result would be definitely wrong. For some unclear image, the color of the disease is similar to the leaf which means the distance between each pixel is very short, so the EM algorithm may not cluster the disease very well (A. & L. N., 2017). Some fail EM disease segmentation examples shown in Figure 4.

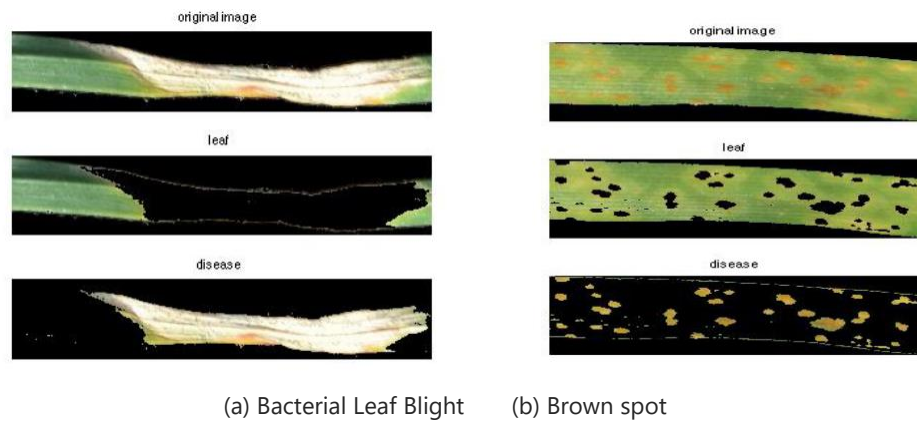


Figure 3. Successful examples by EM segmentation



Figure 4. Failure examples by EM segmentation

Thus a simpler and quicker segmentation method Otsu segmentation was used to segment the disease out based on the threshold which maximizes the ratio of between class variances to the overall variance (Lin et al., 2020). By using this threshold segmentation method, it is very quick to get a relatively better result for almost all leaf pictures with generalized features although the result is not outstandingly accurate (Vinoth Kumar & Jayasankar, 2019). Some segmentation result examples are shown in Figure 4. The average time of disease segmentation by Otsu's method is around 0.4 second (Shi et al. 2015).

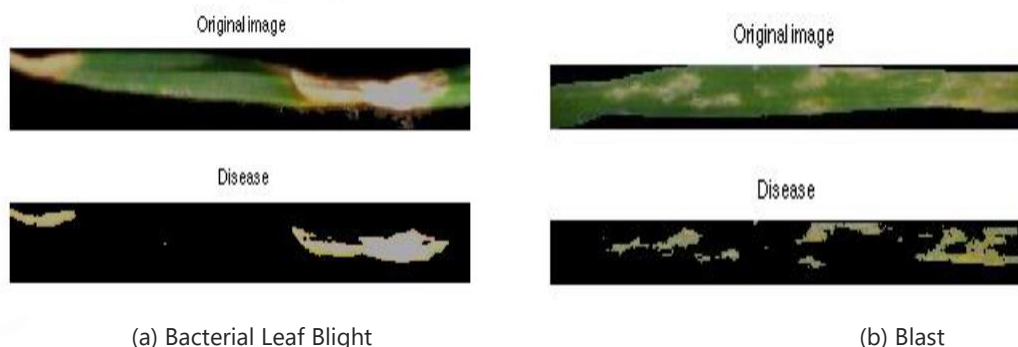


Figure 4. Disease examples segmented by Otsu

3.4. Feature extraction

In order to identify the leaf disease, we should choose some appropriate features as a descriptor to distinguish different kinds of leaf disease (Sharma, Verma, & Goel, 2020). Inappropriate or excessive features would lead the classification to over fit and long search time (Meena Prakash, Saraswathy, Ramalakshmi, Mangaleswari, & Kaviya, 2018). Hence it is critically important to choose a good descriptor from the various diseases (Vinoth Kumar & Jayasankar, 2019). Feature extraction is a kind of dimension reduction which can effectively represent the interesting part as a compact feature vector (Khairnar & Dagade, 2014). After lots of experiments, it was found that the combination of color histogram and Tamura Texture can be a good feature descriptor (Mitkal et al., 2016).

3.4.1. Color histogram

Color is often the most expressive among all the visual features. The different disease has different color distribution, and the same disease should have a similar color (Dubey & Jalal, 2015). Thus, we choose color histogram as a descriptor to discriminate the leaf diseases. We choose to use HSV (Hue, Saturation, and Value) color space to calculate the histogram. As the figures shown in Figure 5, they are the results of the color histograms of the same examples. It can be easily found that the color histogram of the leaves with the same disease is similar, but different diseases are totally different. Hence HSV color histogram should be an appropriate descriptor (Khirade & Patil, 2015).

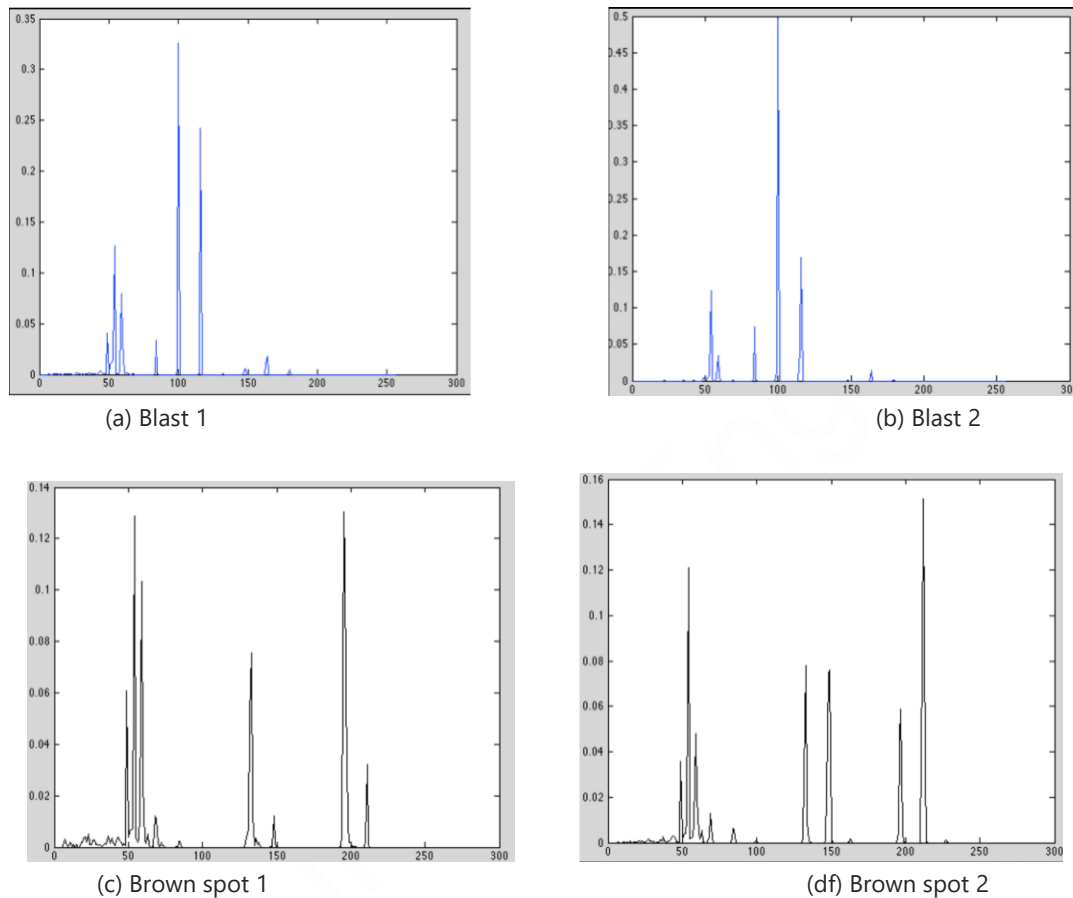


Figure 5. This figure shows the color histogram of some examples of the three kinds of disease.

Figure (a) and (b) in blue line are the color histograms of two different leaves with the Blast disease. Similarly, (c) and (d) in red line are the histograms of Bacterial Leaf Blight, and (e) and (f) in black line of the Brown spot.

3.4.2. Tamura's texture features

Texture feature is another commonly used and very important feature in many image analysis and computer vision applications (Ngugi et al., 2020). Image texture represents the spatial arrangement of intensities and color in the image (Gharat, 2012). Characterizing the textures falls into two approaches which are statistical and structural. For this project, one statistical method calls Tamura texture was used as discriminate to classify the diseases (Mushtaq Adnan, Karol Ali, & Drushti, 2019). Tamura Features are designed based on the texture of human perception according to psychological study. Tree properties of Tamura's texture were implemented in this project which is Coarseness, Contrast, and Directionality (Pavithra, Priyadharshini, Praveena, & Monika4, 2015). Finally, we store the color histogram and Tamura texture features in a $1 * M$ vector for each segmented image, where M is the number of features. Then we put the features vectors together in an $N * M$ matrix, where M represents a segmented disease images features and N represents the total number of image examples. There is also an $N * 1$ vector which represents the labels of the corresponding disease examples (Maity et al., 2018). These labels are numbered from 1 to 4, which represents disease free leaf, blast leaf, bacterial leaf blight leaf, brown spot leaf respectively (Khirade & Patil, 2015).

3.5. Learning and classification

In order to ease the implementation, verification, and documentation, the system could be divided into three different components based on its functionalities (Bapat et al., 2020). The first part is the Android application as a client which provide a user interface to interact with the functions of the application (Lin et al., 2020). The user can choose to take a picture or select an image from the gallery of the mobile phone as the leaf image to be diagnosed (Ramcharan et al., 2017). Once the users decide which leaf picture, he/she wants to use, the Android application will send the image to the second part. The second part is the server implemented with java servlet which aims to communicate between client and main functions (Ngugi et al., 2020). The servlet will open a socket on a port for each connection and receive an input image and call the main function on the third part, finally sent the resulting image back to the client (Sladojevic, Arsenovic, Anderla, Culibrk, & Stefanovic, 2016). The third part is the main component of the application which is used to recognize the category of the disease and implemented by MATLAB (Mitkal et al., 2016) and it is also the most difficult and central program (Ngugi et al., 2020). The diagnosis program design would be roughly introduced in the next section. After the recognition, the resulting image would be saved in the server for returning to the client (Sharma et al., 2020).

3.5.1. Decision tree

At first, the 'decision tree learning (X, label)' function was created to initialize the tree structure and then call the 'createTree (T, X, label)' function to do the recursive learning for the subtrees (Ngugi et al., 2020). At the start of the createTree function, stopping criteria was performed first. If all the input label have the same value, then the same value (0 or 1) will be marked as the trees class and meanwhile returning the tree (Barbedo, 2016). Suppose the matrix data x has n rows of leaf disease data and m columns of features. Then the program finds a maximum and minimum value for each leaf data feature. And then generate a vector of 10 linearly equally spaced points as a set of the threshold between each maximum and minimum value of each feature (Bapat et al., 2020). Try splitting the data x into two matrixes by each threshold in the vectors and then calculate the Gain every time until the iteration stops. The threshold leads to the best gain should be the decision attribute. After that, divide the leaf disease data into two matrixes by the best attribute. And then call createTree function for each of the two divided leaf disease data to get two kid trees. 'learn Disease (x,y)' is a function to learn the six-leaf disease decision trees for a given set of leaf disease data and labels and returns a list of six trees. decideTree (x, T)' is a function to decide whether a disease feature data belongs to a leaf disease by the leaf diseases decision tree. Disease Decision (x, trees)' is a function to decide which disease of a row disease features belongs to, given a set of disease decision trees. The function tries to classify by each binary decision tree and then get a list of binary results (S., S., B., & K., 2018). First, find the indexes of the positive results. If the positive index list is empty, then return a random result between those four leaf diseases, else random get one from the positive index list and translate it to the leaf disease and return (Sharma et al., 2020). The binary decision trees built on the four kinds of leaf disease are shown in Figure 6.

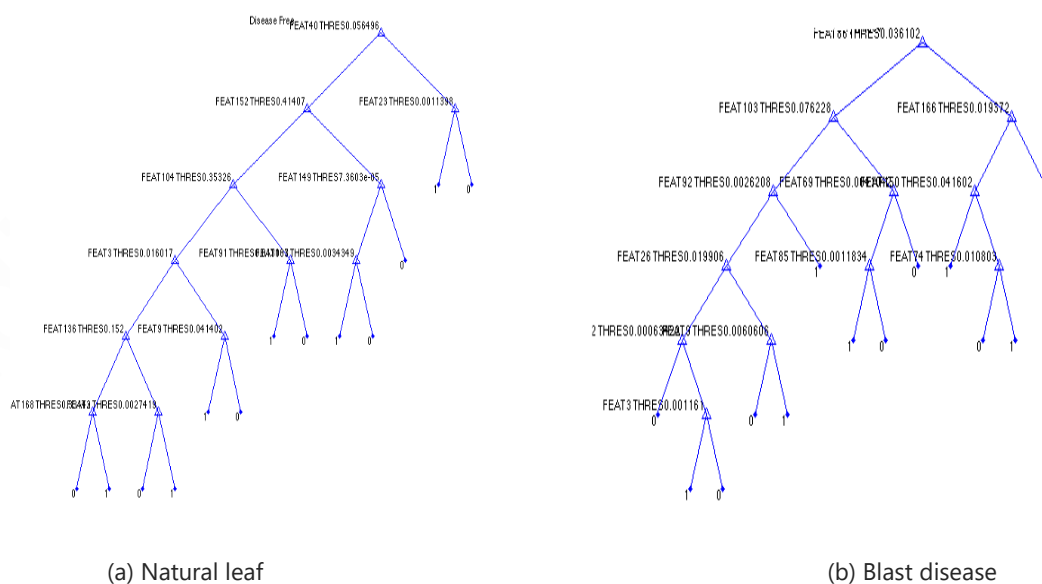


Figure 6. The decision trees of these four categories

3.5.2. ANN

In neural networks, in order to fit the training set best, gradient descent is used to tune network parameters by back-propagation (Sladojevic et al., 2016). Back-propagation over multilayer networks is sure to converge toward some local minimum error between the training example values and the output values (Liang et al. 2019). However, it is not guaranteed that the algorithm will find the global minimum error. The number of hidden unit layers defines exactly the expressive power of the network, and also the complexity of the decision boundary (Chaware et al., 2017). Multiple experiments were produced in order to find an appropriate model for reducing the error, and keeping a performance graph in which the way training data is processed is as accurate as possible (Mahmud Sultan et al., 2020). The structure is formed with input data, hidden layers, and 4 outputs. We implement a data function to transform the data into the corresponding input format which can be accepted by the ANN tools, we have used one hidden layer of sigmoid units containing 20 hidden units (neurons) with a number of epochs of 100 (Bapat et al., 2020).

3.5.3. Multiple class classification

The pairwise classification was used by the support vector machines. In terms of the training process, each two of the leaf disease will create an SVM struct (Sharma et al., 2020). For this project, we know that there are four leaf disease classes, so we should build $(4-1)*4/2 = 6$ SVM struct. When we test a leaf disease data, we shall evaluate the 6 SVM and then classify according to which of the classes gets the highest numbers of the vote (Halder, Sarkar, & Bahar, 2018). A vote means a classifier put a pattern into the class (Pavithra et al., 2015).

3.5.4 Random forest

The Random Forest classifier was implemented via using the TreeBagger function in the MATLAB statistic toolbox (Mitkal et al., 2016). The TreeBagger function can create an ensemble of bagged decision trees (Dhakad et al., 2017). The parameters were adjusted by the given features. An ensemble of 50 trees was created to train the features. The final error nearly equals 0.05 as the number of grown trees increasing to the 50, which is a quite satisfactory result.

3.6. Disease detection

After the implementation of the classifier, it is not difficult to build a recognition program to classify which category every leaf belongs to (Khirade & Patil, 2015). Firstly segmentation function should be used to segment the meaningful area out and then feature extraction function was called to get the distinguished features from the segment storing in a vector (Mushtaq Adnan et al., 2019). Successively the feature vector should be put into the classifier to predict its category. Finally, the predicted label will be drawn on the leaf (Pavithra et al., 2015). Figure 7 shows some sample results of the leaves over these four categories.



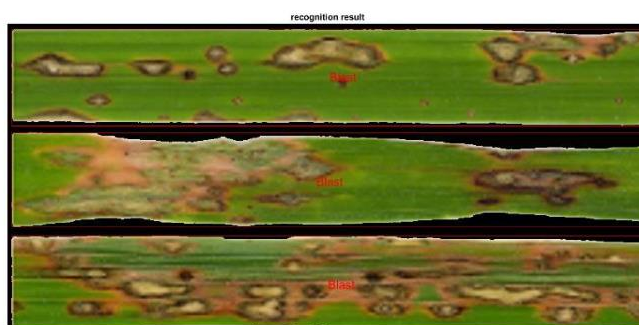
(a) Disease Free



(b) Disease Free



(c) Blast



(d) Blast



(e) Bacterial Leaf Blight



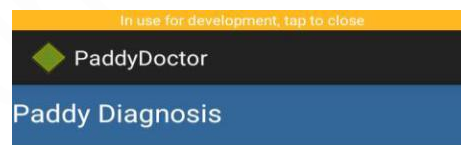
(f) Bacterial Leaf Blight



(g) Brown spot

Figure 7. Some sample recognition results over the leaves in these four categories

Sometimes wrong pictures for example without a big enough leaf in the image may be entered into the program, it is necessary to handle this situation at that time (Chaware et al., 2017). Figure 7 shows the two kinds of catch methods used in this program for this kind of exception. After the EM segmentation, if none of the segmented partition has an overwhelming green proportion (Khirade & Patil, 2015), the result as sample 1 will be returned with showing 'no leaf in the picture' at the top of the image (Bapat et al., 2020). If the segmented part is eligible but the pixels of the objects in the partition don't satisfy certain leaf condition, this kinds of the object will be marked as no leaf in yellow (Bakar et al., 2018).



Load Image

Take Image

Figure 8. The Application Graphical User Interface

3.7. Android Development

For now, the central program has been finished, however, we still need a user interface to interact with mobile phone users (Reddy, Pawar, Rasane, & Kadam, 2015). A simple android application has been implemented which is shown in Figure 8. The program can choose to select an image from the image from the mobile phone gallery by clicking the button 'Load Image' or take a new leaf photograph (Ramcharan et al., 2017). Then the image will be automatically sent to the server programmed with java servlet in NetBeans, and the server will call the MATLAB function by using the 'matlabcontrol' library (Figure 8).

3.8. Additional computer-based software development

To extend the usability of the program, an additional computer-based software was developed to aid the computer user to operate the function more comfortably. The user can choose an image from the disk on the computer and do some operations on the resulting image which will be introduced below. The user can choose to open, save or exit the software. If the user clicks the view button, two options are provided which make the user show or hide the panel and tool bar by their preference as shown in Figure 9.

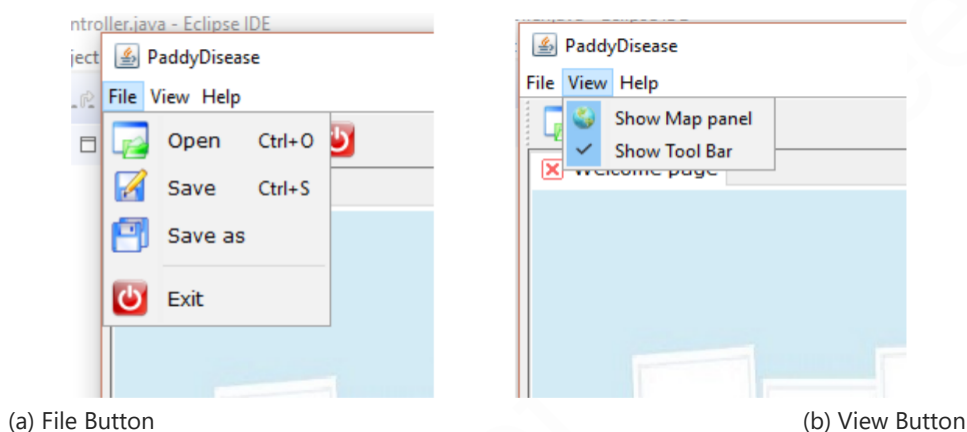


Figure 9. Menu Bar

The toolbar is also available shown in Figure 10 which is very convenient to the user and contain all the main functions. If the user doesn't want to show the toolbar, he can make it invisible by the setting of view preference.



Figure 10. The Application Graphical User Interface



Figure 11. The application graphical user interface

At the start of using this software, the user can click the button in the menu, or tool bar or type Ctrl + O to open an image from the computer. Once an image is chosen, the MATLAB program will start to diagnose what disease the leaf in the image has. And the result will be shown in a new tab panel with leaving the previous ones as Figure 11. The user can also save or close any tab panel of them at any time.

The user can use the mouse to click and drag the resulting image to any position they want, and can even use mouse wheel to zoom in and out the image to the bigger or smaller size. When user zoom in or zoom out the map, four red rectangle arcs will show around the zoom area and the zoom center will be around the mouse position which is shown in Figure 12.

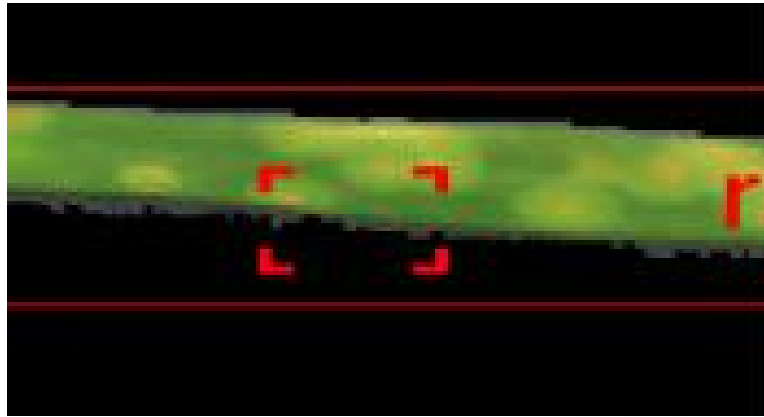


Figure 12. The application graphical user interface

If the user clicks the right button of the mouse, a pop menu will occur to allow the user to zoom or center the image in another manner.

4. RESULTS AND DISCUSSION

4.1. Experimental analysis

The experiments and related analysis are done in this section. The experiments and analysis processes are done on a computer with Core-i5 processor having 4 cores with each core having 2.5GHz Speed. Also, the system had 8GB of RAM, and 2GB of internal intel HD video memory. For software, MATLAB Version 9.5.0.944444 (Release R2018b), Android Studio IDE, NetBeans IDE 8.2 and GlassFish Server Open Source Edition 4.1.

4.1.1. Segmentation test



Figure 13.1. Leaf segmentation results of Category 4

To obtain the valid result of segmentation method on general leaves, it is vitally important to test the segmentation program on other kinds of leaves. All the segmentations presented above are operated on only one kind of leaf with different diseases. If the

program cannot work well on other kinds of disease, the application will lose the generalization property. To this end, five other categories of leaves were chosen to be tested by the EM segmentation program to show the generalization. Four of five categories leaf examples are shown in Figure 13.1 to 13.4 which are all in a relatively complex background. After the segmentation, we got the segmented leaves which are shown in Figure 13.1, 13.2, 13.3 and 13.4. From the segmentation images, it can be obviously seen that the results are extremely good with no defect. The result images are complete and no mixed with other objects.

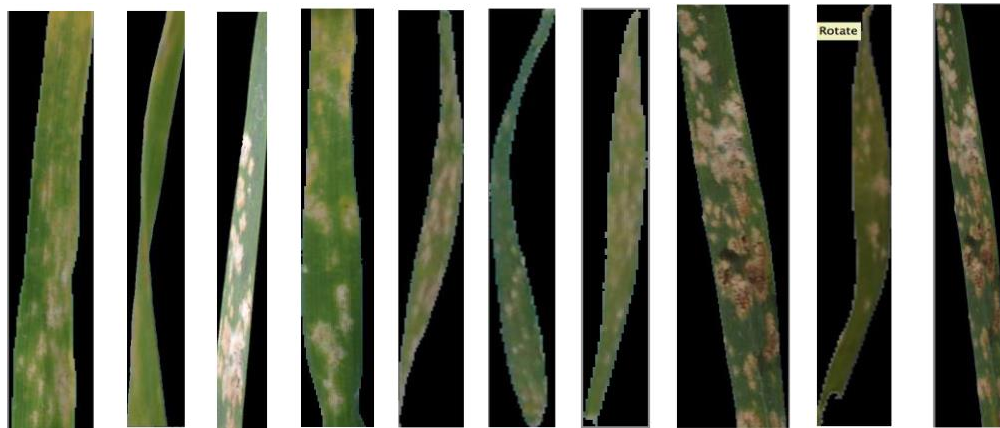


Figure 13.2. Leaf segmentation results of Category 4



Figure 13.3. Leaf segmentation results of Category 4

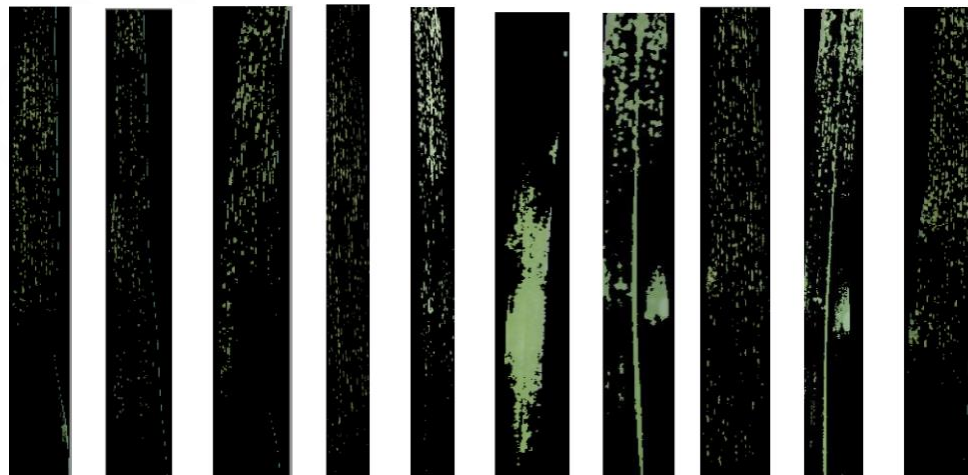


Figure 13.4. Leaf segmentation results of Category 4

4.1.2. Evaluation on different learning algorithms

Cross-validation is one of the most widely used methods to improve the generalization of the results and avoid over fitting. Usually, a learning algorithm is trained for a set of data with the corresponding target until it got a satisfied result to predict the output for other input data. However, if the learning is performed too long, it will be particularly adjusted to some points or features which are highly correlative with the target function. The training error will decrease continuously, but the validation error for new input may start to increase. Thus, the learning will be over fitting and lose the generalization to other new input. To solve this problem, cross-validation is often used. N-fold Cross-validation is a statistical practice of portioning the data into N different subsets such that initially a single subset is used to be the validation set which aims to estimate the generalization error and the other subset are used to be the training set together for adjusting the parameters of the classifiers. After that, the subsequent subset will be used as a validation set and the others to be the training set and so on. The process shows in Figure 14.

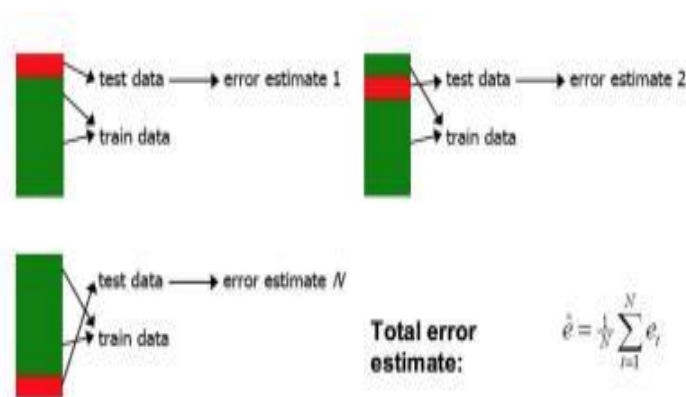


Figure 14. N-Fold Cross-Validation

4.2. Decision trees

4.2.1. Leaf Features Training

Next, we will use the evaluation concepts introduced in the last section to measure the performance of different learning methods. First, we should measure the decision trees algorithm. Decision trees are trained on leaves features and disease spot features respectively with 10-fold cross-validation and the confusion matrix of leaves features. It shows that the recognition rate is 84:4% and there are some confusions between Blast and Bacterial Leaf Blight, where 23:2% are misclassified on the Blast and 29:0% are misclassified on Bacterial Leaf Blight, while other diseases are predicted pretty well. The average recall, precision, and F1 Measure are shown in Table 1. The average precision of Blast and average recall of Bacterial Leaf Blight are relatively bad which are less than 80%. The F1 measures of these are also obviously worse than the other two.

Table 1. Decision trees evaluation on leaf

	Disease Free#1	Blast #2	Bacterial Leaf Blight#3	Brown spot#4
Average recall	0.9750	0.8321	0.7048	0.8467
Average Precision	0.8954	0.7915	0.8964	0.9124
Average F1 Measure	0.9289	0.8003	0.7551	0.8669

4.2.2. Disease spot features training

We will look at the evaluation of decision trees training on disease spot features. The confusion matrix of the cross-validation result is shown in Figure 5.10. The overall recognition rate is 85:0% which is similar to the results of decision trees training on leaf features. The confusion matrix shows the training has a problem in distinguishing among the three diseases but classify natural leaf without disease very well. The average recall, precision, and F1 Measure are shown in Table 2. The average precision of Blast and average recall of Bacterial Leaf Blight are relatively bad which are less than 82%. The F1 measures of these are also obviously worse than the other two. This experiment result is similar to the leaf features training with decision tree above.

Table 2. Decision trees evaluation on disease spot

	Disease Free#1	Blast #2	Bacterial Leaf Blight#3	Brown spot#4
Average recall	0.9875	0.8411	0.7548	0.7933

Average precision	0.9172	0.8010	0.8161	0.9417
Average F1 Measure	0.9482	0.8154	0.7775	0.8449

4.3. Artificial Neural Network

4.3.1. Leaf features training

The results show the evaluation of artificial neural network training on leaf features. The overall recognition rate is 87:3% and confusion matrix shows the distinguishing ability of the classifier is nearly balanced without outstanding performance for every category. The average recall, precision, and F1 Measure are shown in Table 3. The average precision of Blast and Bacterial Leaf Blight are relatively not very good which are less than 80%, but their recall is better. The F1 measures of three kinds of disease are not very good around 80%, the result of disease-free leaves is much better around 90%.

Table 3. Artificial neural network average evaluation on leaf

	Disease Free#1	Blast #2	Bacterial Leaf Blight#3	Brown spot#4
Average recall	0.9750	0.7857	0.7976	0.8400
Average precision	0.8547	0.8335	0.8839	0.8238
Average F1 Measure	0.9027	0.7900	0.8173	0.8146

4.3.2. Disease spot features training

The results show the evaluation of artificial neural network training on disease spot features. The overall recognition rate is 78:1% which are much worse than the results of artificial neural network training on leaf features. The confusion matrix shows the distinguishing ability of the classifier is nearly balanced without outstanding performance for every category except the disease-free is better. The average recall, precision, and F1 Measure are shown in Table 4. Both of the average precision and recall of Blast and Bacterial Leaf Blight are relatively bad that are less than 80%. The F1 measures of three kinds of disease are bad around 70%. Overall, the experiment result on disease spot features is much worse than the leaf features training with Artificial Neural Network above.

Table 4. Artificial neural network average evaluation on disease spot

	Disease Free#1	Blast #2	Bacterial Leaf Blight#3	Brown spot#4
Average recall	0.8464	0.7411	0.6667	0.8867
Average precision	0.8578	0.7606	0.7261	0.7938
Average F1 Measure	0.8492	0.7353	0.6768	0.8007

4.4. Support Vector Machine

4.4.1. Leaf features training

The results show the evaluation of Support Vector Machine training on leaf features. The confusion matrix of the cross-validation result is shown in Figure 5.13. The overall recognition rate is 94:2% and confusion matrix shows the distinguishing ability of the classifier is nearly balanced with outstanding performance for every category. The average recall, precision, and F1 Measure are shown in Table 5. The average precision of Blast and Bacterial Leaf Blight are relatively not low which are around 80%, and average recall of disease-free leaves and ones with Brown spot are relatively low around 93%. The F1 measures of all these four class are good around 94%, overall, the performance of the method is very good which is better than the previous methods.

Table 5. Support vector machine average evaluation on leaf

	Disease Free#1	Blast #2	Bacterial Leaf Blight#3	Brown spot#4
Average recall	0.9750	0.9232	0.9286	0.9400
Average precision	0.9328	0.9621	0.9653	0.9357
Average F1 Measure	0.9513	0.9389	0.9411	0.9317

4.4.2. Disease Spot Features Training

The results show the evaluation of Support Vector Machine on disease spot features. The confusion matrix of the cross-validation result is shown in Figure 5.14. The overall recognition rate is 90:9% which are a little worse than the results of Support Vector Machine training on leaf features. The confusion matrix shows the distinguishing problem between Blast and Bacterial Leaf

Blight, whose misclassification rate are obviously higher than the other two. The average recall, precision, and F1 Measure are shown in Table 6. Both of the average precision and recall of Blast and Bacterial Leaf Blight are relatively worse which are less than 90% but still good. The F1 measures of Blast and Bacterial Leaf Blight are relatively worse than the other two around 85%. Overall, the experiment result on disease spot features is a little worse than the leaf features training with Support Vector Machine above.

Table 6. Support vector machine average evaluation on disease

	Disease Free#1	Blast #2	Bacterial Leaf Blight#3	Brown spot#4
Average recall	0.9625	0.8554	0.8571	0.9800
Average precision	0.9750	0.8732	0.8488	0.9633
Average F1 Measure	0.9683	0.8607	0.8472	0.9709

4.5. Random Forest

The results blew shows the evaluation of Random Forest training on leaf features. The overall recognition rate is 96:4% which is the best result of all the others. The confusion matrix shows the distinguishing ability of the classifier is nearly balanced with an extremely outstanding performance for every category. The average recall, precision, and F1 Measure are shown in Table 7. The average precision and recall of all of these categories are over 90% and even over 95%. The F1 measures of all these four class are extremely outstanding over 94% and average 96%, overall, the performance of the method is extremely good that is the best method.

Table 7. Random forest average evaluation on leaf

	Disease Free#1	Blast #2	Bacterial Leaf Blight#3	Brown spot#4
Average recall	1.0000	0.9589	0.9286	0.9600
Average precision	0.9675	0.9453	1.0000	0.9714
Average F1 Measure	0.9822	0.9484	0.9603	0.9624

Table 8. Overall learning evaluation

Learning Method	Measure	Disease Free	Blast	Bacterial Blight	Brown spot
DT on leaf	Recall	0.9750	0.8321	0.7048	0.8467
	Precision	0.8954	0.7915	0.8964	0.9124
	F1 Measure	0.9289	0.8003	0.7551	0.8669
DT on disease	Recall	0.9875	0.8411	0.7548	0.7933
	Precision	0.9172	0.8010	0.8161	0.9417
	F1 Measure	0.9482	0.8154	0.7775	0.8449
ANN on leaf	Recall	0.9750	0.7857	0.7976	0.8400
	Precision	0.8547	0.8335	0.8839	0.8238
	F1 Measure	0.9027	0.7900	0.8173	0.8146
ANN on disease	Recall	0.8464	0.7411	0.6667	0.8867
	Precision	0.8578	0.7606	0.7261	0.7938
	F1 Measure	0.8492	0.7353	0.6768	0.8007
SVM on leaf	Recall	0.9750	0.9232	0.9286	0.9400
	Precision	0.9328	0.9621	0.9653	0.9357
	F1 Measure	0.9513	0.9389	0.9411	0.9317
SVM on disease	Recall	0.9625	0.8554	0.8571	0.9800
	Precision	0.9750	0.8732	0.8488	0.9633
	F1 Measure	0.9683	0.8607	0.8472	0.9709
RF on leaf	Recall	1.0000	0.9589	0.9286	0.9600
	Precision	0.9675	0.9453	1.0000	0.9714
	F1 Measure	0.9822	0.9484	0.9603	0.9624

5. CONCLUSION

The development of the mobile application has met its specification successfully. For the given leaves with three categories of disease, the application can recognize all them. The segmentation program can successfully segment the leaves and diseases spot out with few defects. The machine learning methods based on the extracted features work very well. The cross-validation was done which aims to prove the generalization of the learning method. Both the leaf features and disease spot features are trained via four different kinds of popular machine learning methods, and the best result is got from leaf features training with Random Forest whose average recognition rate is 96.4%. The high cross-validation results of recognition show the application is generalized for the other untrained leaves with these three kinds of diseases. The result can demonstrate that both features extracted and the classification method are appropriate for this application. Furthermore, the result will be much higher if more training samples are given in the future. In addition, the implementation of the Java software is also tested with JUnit Test and all of them passed. The operation is easy for the user to start to use the application. The user evaluation results show the high user-friendly. Overall, the project is successful for identifying the plant's disease based on its leaf appearance. It can be a central prototype to be used by the agricultural industry to improve the production of crops or by gardeners to recognize the disease and avoid it next time.

Although the system has many positive aspects and successfully achieved all the goals on the specification, it could still be enhanced to be a publishable application. In terms of the segmentation, the leaves cannot be segmented into very good result if the background is too complex that is nearly an unavoidable problem. However, the application can add a new function to allow the user to cut their interest part out, then the result should be acceptable. The application can also make the user input the EM segmentation parameter if they would like to. For example, if the user chooses to partition the image into two sub-images, but the result is not very good, then the user can try to partition it into three. For the feature extraction, the more complicated feature can be tried in experiments. However, implementing all the candidate algorithms is no practical given a limited amount of time. The system only trained three categories disease of the same plant with a limited number of images which cannot lead generally used in reality. But the project is just a research in this research field, the recognition could be more generalized and better if more images are given and trained in the future.

Many extensional functions can be added to the application which could make it is a real application in a Google Store or Apple Store. Other types of software such as website could also be implemented to make the program more conveniently to attain. The application should allow the user to cut the leaf region out if the background is too big or too complex, which would very helpful to segment the leaf out successfully. More other advanced features can be implemented and tested to optimize the algorithm to get a better result. Other categories of leaves with different diseases should be trained via experiments to make the application applicable to as many categories of plants as possible. The application software can be built in some additional functions such as viewing histories, saving locations and so on.

Conflict of interests

There is no conflict of interest among the authors.

Funding

There is no direct funding for this project.

Peer-review

External peer-review was done through double-blind method.

Data and materials availability

All data associated with this study are present in the paper.

REFERENCE

1. A., I., & L. N., O. (2017). Digital Image Processing for Detecting and Classifying Plant Diseases. *Circulation in Computer Science*, 2(11), 1–7. <https://doi.org/10.22632/ccs-2017-252-66>
2. Akher, S. A., Sarker, M. N. I., & Naznin, S. (2018). Salt stress mitigation by salicylic acid in wheat for food security in coastal area of Bangladesh. *Journal of Plant Stress Physiology*, 4, 07–16.
3. Alam, G. M. M., Khatun, M. N., & Sarker, M. N. I. (2019). Vulnerability to food security due to riverbank erosion in Bangladesh. In M. Hossain, Q. K. Ahmad, & M. Islam (Eds.), *Climate Adaptation for a Sustainable Economy: Lessons from Bangladesh, an Emerging Tiger of Asia* (pp. 1–20).
4. Bakar, M. N. A., Abdullah, A. H., Abdul Rahim, N., Yazid, H.,

- Misman, S. N., & Masnan, M. J. (2018). Rice leaf blast disease detection using multi-level colour image thresholdin. *Journal of Telecommunication, Electronic and Computer Engineering*, 10(1–15), 1–6.
5. Bapat, A., Sabut, S., & Vizhi, K. (2020). Plant leaf disease detection using deep learning. *International Journal of Advanced Science and Technology*, 29(6), 3599–3605.
 6. Barbedo, J. G. A. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, 144, 52–60.
 7. Chaware, R., Karpe, R., Pakhale, P., & Desai, S. (2017). Detection and Recognition of Leaf Disease Using Image Processing. *International Journal of Engineering Science and Computing*, 7(5), 11964–11967.
 8. Dhakad, N. G., Yewale, U. S., Patil, T. Y., Deore, G. A., Science, C., College, L. G. N. S., ... Akhil, G. U. (2017). Leaf Disease Detection Using Image Processing for Pesticides Spraying. *International Journal of Advance Engineering and Research Development*, 4(04), 1268–1274.
 9. Dubey, S. R., & Jalal, A. S. (2015). Application of Image Processing in Fruit and Vegetable Analysis: A Review. *Journal of Intelligent Systems*, 24(4), 405–424.
 10. Gharat. (2012). Leaf Disease Detection Using Image Processing. *International Journal of Innovative Research in Science, Engineering and Technology*, 1(6), 1268–1274.
 11. Haider, M. K., Islam, M. S., Islam, S. S., & Sarker, M. N. I. (2015). Determination of crop coefficient for transplanted Aman rice. *International Journal of Natural and Social Sciences*, 2(23), 34–40.
 12. Halder, M., Sarkar, A., & Bahar, H. (2018). Plant Disease Detection By Image Processing: a Literature Review. *SDRP Journal of Food Science & Technology*, 3(6), 534–538.
 13. Islam, M. D. Il, Rahman, A., Sarker, M. N. I., Sarker, M. S., & Jianchao, L. (2020). Factors Influencing Rice Farmers' Risk Attitudes and Perceptions in Bangladesh amid Environmental and Climatic Issues. *Polish Journal of Environmental Studies*, 30(1), 1–11.
 14. Islam, M. S., Khanam, M. S., & Sarker, N. I. (2018). Health risk assessment of metals transfer from soil to the edible part of some vegetables grown in Patuakhali province of Bangladesh. *Archives of Agriculture and Environmental Science*, 3(2), 187–197.
 15. Islam, Md Saiful, Proshad, R., Asadul Haque, M., Hoque, M. F., Hossin, M. S., & Islam Sarker, M. N. (2020). Assessment of heavy metals in foods around the industrial areas: health hazard inference in Bangladesh. *Geocarto International*, 35(3), 280–295.
 16. Khairnar, K., & Dagade, R. (2014). Disease Detection and Diagnosis on Plant using Image Processing A Review. *International Journal of Computer Applications*, 108(13), 36–38.
 17. Khirade, S. D., & Patil, A. B. (2015). Plant disease detection using image processing. *Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015*, (3), 768–771.
 18. Larijani, M. R., Asli-Ardeh, E. A., Kozegar, E., & Loni, R. (2019). Evaluation of image processing technique in identifying rice blast disease in field conditions based on KNN algorithm improvement by K-means. *Food Science and Nutrition*, 7(12), 3922–3930.
 19. Liang, W. jie, Zhang, H., Zhang, G. feng, & Cao, H. xin. (2019). Rice Blast Disease Recognition Using a Deep Convolutional Neural Network. *Scientific Reports*, 9(1), 1–10.
 20. Lin, S., Li, Z., Fu, B., Chen, S., Li, X., Wang, Y., ... Zheng, Z. (2020). Feasibility of using deep learning to detect coronary artery disease based on facial photo. *European Heart Journal*, 1–12.
 21. Mahmud Sultan, M., Zaman, Q. U., Esau, T. J., Chang, Y. K., Price, G. W., & Prithiviraj, B. (2020). Real-time detection of strawberry powdery mildew disease using a mobile machine vision system. *Agronomy*, 10(7).
 22. Maity, S., Sarkar, S., Vinaba Tapadar, A., Dutta, A., Biswas, S., Nayek, S., & Saha, P. (2018). Fault Area Detection in Leaf Diseases Using K-Means Clustering. *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018*, 1538–1542.
 23. Meena Prakash, R., Saraswathy, G. P., Ramalakshmi, G., Mangaleswari, K. H., & Kaviya, T. (2018). Leaf Disease Detection, Quantification and Classification Using Digital Image Processing. *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017, 2018-Janua(July)*, 1–4.
 24. Mitkal, P., Pawar, P., Nagane, M., Bhosale, P., Padwal, M., & Nagane, P. (2016). *Leaf Disease Detection and Prevention Using Image Processing using Matlab*. 26–30.
 25. Mushtaq Adnan, Karol Ali, & Drushti, G. (2019). Plant Disease Detection using CNN & Remedy. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 8(3), 622–626.
 26. Nasrin, M., Sarker, M. N. I., & Huda, N. (2019). Determinants of health care seeking behavior of pregnant slums dwellers in Bangladesh. *Medical Science*, 23(95), 35–41.
 27. Ngugi, L. C., Abelwahab, M., & Abo-Zahhad, M. (2020). Recent advances in image processing techniques for automated leaf pest and disease recognition – A review. *Information Processing in Agriculture*, 1–25.
 28. Pavithra, S., Priyadharshini, A., Praveena, V., & Monika4, T. (2015). Paddy Leaf Disease Detection Using Svm. *International Journal of Communication and Computer Technologies*, 3(1), 16–20.
 29. Prodhana, A. S., Sarker, M. N. I., Sultana, A., & Islam, M. S. (2017). Knowledge, adoption and attitude on banana

- cultivation technology of the banana growers of Bangladesh. *International Journal of Horticultural Science and Ornamental Plants*, 3(1), 47–52.
30. Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8(October), 1–7.
 31. Reddy, S., Pawar, A., Rasane, S., & Kadam, S. (2015). *A Survey on Crop Disease Detection and Prevention using Android Application*. 2(4), 621–626.
 32. S., R., S., P., B., J., & K., P. (2018). Skin Disease Diagnosis System using Image Processing and Data Mining. *International Journal of Computer Applications*, 179(16), 38–40.
 33. Sarker, M. N. I. (2016a). Knowledge, Adoption and Constraint analysis of Chilli Technology in Char Area of Bangladesh. *International Journal of Ecology and Development Research*, 1(1), 16–18.
 34. Sarker, M. N. I. (2016b). Role of Banks on Agricultural Development in Bangladesh. *International Journal of Ecology and Development Research*, 1(1), 10–15.
 35. Sarker, M. N. I., Ahmad, M. S., Islam, M. S., Syed, M. M. M. A., & Memon, N. H. (2020). Potential food safety risk in fruit production from the extensive use of fluorine-containing agrochemicals. *Fluoride*, 53(3), 1–22.
 36. Sarker, M. N. I., Islam, M. S., Ali, M. A., Islam, M. S., Salam, M. A., & Mahmud, S. M. H. (2019). Promoting digital agriculture through big data for sustainable farm management. *International Journal of Innovation and Applied Studies*, 25(4), 1235–1240.
 37. Sarker, M. N. I., Islam, M. S., Murmu, H., & Rozario, E. (2020). Role of Big Data on Digital Farming. *International Journal of Scientific & Technology Research*, 9(04), 1222–1225.
 38. Sarker, M. N. I., & Jie, Z. (2017). Social Security for Vulnerable Groups in Bangladesh on Government Perspective: Contribution of Research Leader. *Journal of Public Policy and Administration*, 1(1), 1–9.
 39. Sarker, M. N. I., Wu, M., Chanthamith, B., Yusufzada, S., Li, D., & Zhang, J. (2019). Big Data Driven Smart Agriculture: Pathway for Sustainable Development. *2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 60–65.
 40. Sarker, M. N. I., Yang, B., Lv, Y., Enamul, M., & Kamruzzaman, M. M. (2020). Climate Change Adaptation and Resilience through Big Data. *International Journal of Advanced Computer Science and Applications*, 11(3), 533–539.
 41. Sharma, V., Verma, A., & Goel, N. (2020). Classification Techniques for Plant Disease Detection. *International Journal of Recent Technology and Engineering*, 8(6), 5423–5430.
 42. Shi, Y., Wang, X. F., Zhang, S. W., & Zhang, C. L. (2015). PNN based crop disease recognition with leaf image features and meteorological data. *International Journal of Agricultural and Biological Engineering*, 8(4), 60–68.
 43. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience*, 2016.
 44. Tichkule, S. K., & Gawali, D. H. (2017). Plant diseases detection using image processing techniques. *Proceedings of 2016 Online International Conference on Green Engineering and Technologies, IC-GET 2016*, 1–6.
 45. Vinoth Kumar, K., & Jayasankar, T. (2019). An identification of crop disease using image segmentation. *Int J Pharm Sci Res*, 10(3), 1054–1064.