International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 4, Issue 12, pp 6-12, December-2020

https://doi.org/10.47001/IRJIET/2020.412002

Classification of Leaf Disease using Image Processing and Machine Learning

[1]Koushik Bhattacharyya, [2]Ram Lal

[1]Computer Science and Engineering, Dream Institute of Technology, Kolkata, India
[2]Computer Services Centre, IIT Delhi, Delhi, India

Abstract **- A successful implementation of education, health and agriculture programs are the main concerns of developing countries like Ethiopia for sustainable development of the countries and ease life of their citizens. Farmers and agriculture experts visually carry out examination of crops. However, this evaluation process is tedious, time consuming, and less accurate, which can cause high risk of loss later. Image processing and machine learning have been widely used in various disease diagnosis approaches. It has been applied to both images captured from cameras of visible light and from equipment that captures information in invisible wavelength, assisting experts to select the right measure and treatment. In this research article, a digital camera captured image is used as input and enhanced with various preprocessing techniques followed by color-based segmentation method to separate the regions of interest then features are extracted using Gray Level Co-occurrence Matrix. Classification of the input image is performed at the final stage taking four different supervised learning algorithms to classify in to two different classes called 'healthy' and 'infected'.**

Keywords: Confusion Matrix, Deep Learning, k-Nearest Neighbor, Naïve Bayes, Random Forest, Support Vector Machines.

I. INTRODUCTION

Early stage detection of plant leaf diseases will give strength to overcome and treat the effects appropriately by providing the details to the farmer or expert, so that the desired prevention action should be taken. Currently, farmers and agriculture experts visually carry out examination of agriculture crops such as cereals, commercial crops, fruits, vegetables and the like affected by different disease for recognition and classification. This evaluation process is however, tedious, time consuming, less accurate and moreover subjective.

The decision-making capability of human inspector also depends on his/her physical condition, such as fatigue and eyesight, mental state caused by biases, work load and pressure, work conditions like uncomfortable and unstable weather condition [6]. Significance progress in the area of artificial intelligence and image processing has led to a good number of real-world applications that include industrial process, business implementation, medical science, biological science, material science and the like.

The development in certain disciplines of computer science like image processing, machine learning, pattern recognition, deep learning etc. promise the required technological support to tackle the various issues in computer vision.

Image processing technology has been and being applied for different applications, agriculture is one of these applications. Image processing is applied for weed detection, for which the plants growing in wrong place in farm which compete with crop for water, light, nutrients and space, causing reduction in yield and effective use of machinery. Weed control was important from agriculture point of view; so many researchers developed various methods based on image processing.

Weed detection techniques used algorithms based on edge detection, color detection, and classification. Image processing is also applied for fruit grading, need of accurate sorting of fruits and foods or agriculture products arises because of increased expectations in quality food and safety standards.

II. RELATED WORK

In this section discussing the related work of image processing and machine learning over agriculture which is related to leaf disease.

Several researchers have proposed and/or developed systems related to the diagnostics and classification of plant diseases using different techniques, among them the literatures in specific to image processing and machine learning techniques are reviewed as follows.

III. PROPOSED SYSTEM

These sections give detailed information about the architecture of the system, concepts and algorithms used evaluation metrics and their representation.

Table 1: Name of the Table that justify the values

S.No	Related work on leaf disease				
	Author(s) and Year	**Techniques used**	**Application (Focus area)**	**Observed weakness**	**Results and remarks**
1.	Anand R. Et al. [19] (2016)	K-means clustering with ANN	Detection of disease on brinjal leaf	Tested with single classifier	84% accuracy
2.	H.Sabrol and K. Satish [20] (2016)	Otsu's segmentation, color, shape and texture feature extraction with classification tree	Tomato plant disease classification	Ready-made image background was used	97.3% classification accuracy
3.	Aasha Nandhini et al. [21] (2017)	GLCM, SVM	Web based plant leaf disease detection	Tested with single classifier	98.4% classification and 98.5% detection accuracies
4.	Peifeng Xu. et. al. [22] (2017)	Embedded image processing with ARM9 microcontroller	Wheat leaf rust detection and grading	Image was taken at a specific and prepared condition	96.2% recognition with 92.3% accuracy
5.	Pooja V. et al. [23] (2017)	Otsu's detection, RGB to HIS conversion, SVM classifier	Plant leaf diseases identification	Proper evaluation technique missed	92.4% rate of recognition on five different leaf diseases types
6.	Artzai Picon et al. [24] (2018)	Image processing with deep convolutional neural network	Wheat crop disease classification	All samples were taken at the same angle of acquisition	96% accuracy

3.1 System Architecture

In the following figure the architecture for an implementation of the proposed system is depicted. Generally, this architecture tells the overall process followed to classify a particular input image in either of two classes. As shown in the figure, there are two separate phases in the system. The training phase and the testing phase with a slight difference in appearance.

These two phases are treated separately, after the training phase come to an end the testing phase follows. The training phase starts from importing a batch of input images, which means several images are started to be processed at a given time in one after the other manner, or independently. In the testing phase a single image is imported for the process. After image is imported both phases pass through the same processes that are providing the same purpose.

The preprocessing, segmentation and future extraction functionalities are the same for both phases (a detailed design and explanation is found in the coming sections). After the feature extraction both phases follow different paths, the training phase provides feature vector with a label input for the model to train and the result is stored in the knowledge base.

The testing phase provides a feature vector to the model and expects for label return classifier returns that label from knowledgebase that is trained previously.

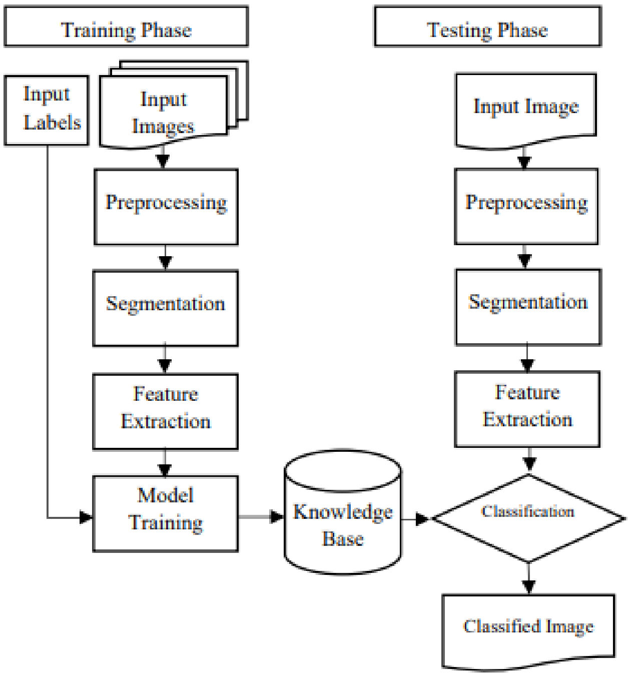


Figure 1: Proposed system architecture

3.2 Dataset Preparation

Data preparation is needed to train and to test the model. From the collected image data manually classified and labeled images in training set and randomly selected, unclassified and unlabeled image data in testing set are prepared. The images in testing set are different from the images that are included in the training set. From the collected total 270 images 189 (70%) samples for training and 81 (30%) samples are used.

3.3 Image Processing Tasks

In the preprocessing task input image passes through different filtering techniques for the sake of getting better image. The preprocessing task is the fundamental step in any image processing application which can help to get a more meaningful interpretation of an image. In addition, preprocessing tasks can affect the overall performance of a classifier since they can reveal hidden or unclear information in an image. This task has number of sub-tasks like image type conversion for ease of computation, image enhancement for visual easiness, noise removal for process consistency, image scaling and resizing or shrinking for complexity minimization quick computation, using filter functions and other important techniques that are necessary to improve the quality of the input leaf image. These sub-tasks can be used one or more time during the process of the implementation, and their order of appearance might not be necessarily the same as it is shown in Figure 2 Image resizing is applied for ease of computation in some cases image files with large size can slow down the overall performance of a model.

This resizing task is applied in way which cannot abuse the information that is found in the original image. Histogram equalization is another sub-task under the pre-processing task that is applied for contrast enhancement which enables to get detailed information from input image. Input image has RGB/BGR color space originally, but for additional purposes this color image can be converted to different color spaces. To grayscale conversion helps to work fine with many image processing algorithms grayscale images are easy to process than RGB images. Because of the number of possible pixel values in different color spaces varies, most algorithms treat input images differently. Additionally, HSV/HIS images are easy for the machine to identify the dominant color in a given image than the RGB one. Here, sometimes it is also called as post processing technique; image dilation technique is applied to recover missed pixels due to techniques that are applied before, mostly due to segmentation. Dilation helps to fill up these missed pixels so that we can get the desired unbiased information, but dilation not only recovers missed pixels it might also add unnecessary pixels accidentally these additives are sometimes found to be noises. So, for these noises to be removed some denoising techniques like median filtering are applied.

3.4 Color-based Segmentation

Segmentation task is also another essential part of the image processing stage. It helps to only focus on the desired region out of the whole leaf image, typically the resultant image from preprocessing task. The color-based segmentation is one of the stochastic types that works on the discrete pixel values of the image. Foreground and background regions are separated based on the color range they found in. Wheat leaf by nature has a range of color between light green and dark green. This can also be due to nature of the crop or the imaging conditions of particular environment and particular acquisition tool. But if the leaf is infected with any disease, some other type of color can be seen. For instance, if the leaf is infected with ‘Septoria tritici’ a lesion colored between yellow and brown is seen. It is easy for human eye to detect and recognize such changes, but for a machine it could be a bit difficult. For this reason, image background and foreground regions should be separated first. After having the desired foreground region, the background region is eliminated (colored black) since it is not needed in the coming processes. In the foreground image one color range for healthy leaf and two different color ranges for infected leaf are expected. Earlier these all ranges were joined and treated as a single range of color in order to eliminate the background. Furthermore, the foreground image is separated in to healthy and infected regions with the same technique for the same reason. Now things are getting easy, the relevant part of the input image is selected out and segmentation task is ended here feature extraction comes next.

3.5 Naïve Bayes Classifier

This is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability. The classification approach is described as follows [37]. Assume that there are N classes of patterns C1, C2 , . . . , CN, and an unknown pattern x in a d dimensional feature space x= [x1, x2,. . ., xd] . Hence the pattern is characterized by d number of features. The problem of pattern classification is to compute the probability of belongingness of the pattern x to each class Ci, i = 1, 2, . . . N. The pattern is classified to the class Ck if probability of its belongingness to Ck is a maximum. While classifying a pattern based on Bayesian classification, we distinguish two kinds of probabilities. These are prior probability and posterior probability. The prior probability indicates the probability that the pattern should belong to a class, say Ck, based on the prior belief or evidence or knowledge. This probability is chosen even before making any measurements, i.e., even before selection or extraction of a feature. Sometimes this probability may be modeled using Gaussian distribution, if the previous evidence suggests it. In cases where there exists no prior knowledge about the class membership of the pattern, usually a uniform distribution is used to model it. For example, in a

ISSN (online): 2581-3048
Volume 4, Issue 12, pp 6-12, December-2020
https://doi.org/10.47001/IRJIET/2020.412002

four-class problem, we may choose the prior probability as 0.25, assuming that the pattern is equally likely to belong to any of the four classes. The posterior probability P(Ci|x), on the other hand, indicates the final probability of belongingness of the pattern x to a class Ci. The posterior probability is computed based on the feature vector of the pattern, class conditional probability density functions P(x|Ci) for each class Ci and prior probability P(Ci) of each class Ci. Naïve Bayes classification states that the posterior probability of a pattern belonging to a pattern class Ck is given by:

$$\mathrm{P}\ (C_k|\mathrm{x}) = \frac{\mathrm{P}\ (x|C_k)\ \mathrm{P}\ (C_k)}{\sum_{i=1}^{N}(\mathrm{P}\ (x|C_i)\mathrm{P}\ (C_i))}$$

The denominator ∑ (P $(x|Ci)$P (Ci)) N i=1 = P (x) is a scaling term which yields the normalized value of the posterior probability that the pattern x belongs to class Ci. Hence, x belongs to class Cp where:

P(Cp|x) = max{P(C1|x), P(C2|x), . . . , P(CN|x)}

3.6 k-Nearest Neighbor Classifier

k-Nearest Neighbor is considered to be simple classifier in the supervised learning algorithms where the classification is achieved by identifying the nearest neighbors to an input and then make use of those neighbors for determination of the class of the input. In K-NN the classification i.e., to which class the given point is belongs is based on the calculation of the minimum distance between the given point and other points. It is not applicable in case of large number of training examples as it is not robust to noisy data. For the leaf classification, the Euclidean distance between the test samples and training samples is 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. In binary (two class) classification problems, it is helpful to choose k to be an odd number as this avoids tied votes.

3.7 Support Vector Machines Classifier

SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The main idea here is to find a maximum marginal hyperplane that best divides the dataset into classes. The classifier separates data points using a hyperplane with the largest amount of margin. The SVM algorithm is implemented in practice using a kernel. A kernel transforms an input data space into the required form this helps to build a more accurate classifier. For this implementation linear kernel is applied, A linear kernel can be used as normal dot product of any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values.

3.8 Random Forest Classifier

Random forest is a type of supervised machine learning algorithm based on ensemble learning where different types of algorithms or same algorithm are joined multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e., multiple decision trees, resulting in a forest of trees, hence the name "Random Forest".

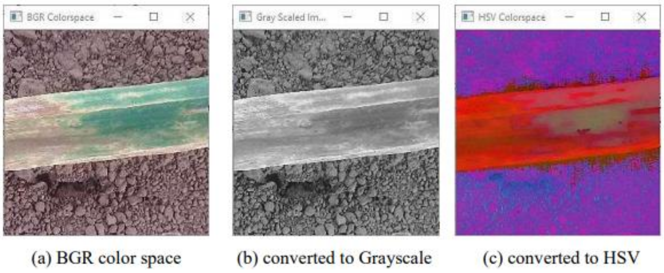
3.9 Performance Evaluation Metrics

The most common metrics used in many machine learning classification applications are the accuracy, specificity and sensitivity. These metrics are based on the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) samples. The reference samples (ground truth) are separated in the group of the positive ones to a disease and the ones that are negative to that disease. This separation can be performed manually by an expert. A number of the negative samples are recognized by an application as true negative (TN), but some of them will be recognized as false positive (FP). On the other hand, some of the positive samples will be correctly recognized as true positive (TP) while some of them will be recognized as false negative (FN).

IV. PROPOSED SYSTEM

As explained early, two separate folders named as ‘HEALTHY’ and ‘INFECTED’ are created for purpose of training. These two folders hold leaf image data (input data) that are manually classified by guidance of an expert. In the training phase multiple images should be imported from the dataset at a time, due to this demand we applied a batch processing with the help of looping the following detailed tasks. The loop starts with image reading task. After image data is collected and dataset is prepared, input images get ready for processing. Images are first treated with preprocessing techniques for better result in the coming steps. The very first step in our code is to read an image with OpenCV Python method imread (‘img_name.jpg’) , here JPG format is used for a reason of its popularity and wide access through many image acquisition tools, all most all digital cameras and mobile phones use the JPG image format to produce image files. Conversion for image format extension will not be applied for any input image with format other than .JPG, keeping in mind that ‘all input images for both training and testing phases must be .JPG images’ is very important.

Unless Python might fail to recognize an image and could terminate the execution before further process.



(a) BGR color space (b) converted to Grayscale (c) converted to HSV

Figure 2: Different color conversions

As OpenCV method resize(img,(256,256)) is applied for input image to resize the image to 256x256 pixels for the ease of computation. After having a resized image cvtColor(img, cv2.COLOR_BGR2HSV) is used to convert the original RGB color space (BGR in case of OpenCV) to HSV color space. Then the in Range() method brings out the most relevant color ranges which are HSV color points between (36,0,0) and (100,255,255) for green and the second range for yellow-brownish that are HSV color points between (8,0,0) and (36,255,255). These color ranges represent the possible actual color of a certain wheat leaf image, the green range for the healthy part and the yellow-brownish for the infected part, since the target disease septoria is characterized by the yellow-brownish color ('Septoria' is a fungal disease causes tan, elongated lesions on wheat leaves. Lesions have a yellow margin, with brown body). These two ranges are then assigned to two separate variables. Here the main purpose of the color ranges is to filter out the leaf part from the whole input image. In other words, to segment the interesting region by separating foreground and background colors. To have a meaningful foreground, parts of an image that have similar color value with the previous separate color ranges should come together. For this reason, method bitwise_or() is applied using the variables that hold the two color ranges as input. Now by applying bitwise_or() once again, but with different number of arguments, foreground of the input image (parts of the image that holds the leaf) is segmented.

4.1 Testing Phase

As specified in the architecture of the system in Chapter 3, here also in the experiment the testing phase followed the same function calls as the training phase does. An input image passes through the same techniques and algorithms like preprocessing, segmentation and feature extraction tasks. The only difference here is, the feature vectors in the testing phase are input for the model without any label. The classifier is expected to return the predicted label ('HEALTHY' or 'INFECTED') based on the provided feature vector. In the testing phase, two different approaches are followed for implementation unlike in the methodology explanation. Two separate programs are written for each approach. Both the programs run the same code with slight difference at the testing time. The first one is an independent test for a single image where all classification models are called one after the other for manual testing, in this approach a single input image is inserted for the testing module and the classification models will put their predictions on the top-left corner of the original image. For this approach to conclude Figure 4.4 shows the individual output of the main image processing tasks with the prediction and Figure 4.5 shows the prediction of each classifier according to the provided feature vector.

4.2 Performance Evaluation Results

All classification models are evaluated by the help of 2x2 confusion matrix. This confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the target values in the data. by calculating the number of test samples under four categories (TP, TN, FN, FP). These categories are discussed in detail under Section 3.7, here in this section an interpretation of the metrics that are calculated based on these four categories is presented. Accuracy: overall, how often is the classifier correct? How correctly the classifier predicts for healthy leaves as 'HEALTHY' and as 'INFECTED' for leaves infected with septoria disease. Classification error / misclassification: tells how often the classifier is incorrect to predict the expected class of an input. Sensitivity: also known as true positive rate or recall, when the actual value is positive, how often is the prediction correct? Or how sensitive is the classifier in detecting positive instances? In our case, how correct is the classifier to classify all available infected leaves as 'INFECTED'. Specificity: when the actual value is negative, how often is the prediction correct? How specific or selective is the classifier in predicting positive instances? How correct enough is the classifier to recognize the healthy samples. Precision: when a positive value is predicted, how often is the prediction correct? How precise is the classifier in predicting positive instances? For samples predicted as 'INFECTED' how many of them are infected for real. False positive rate: when the actual value is negative, how often is the prediction incorrect? In our case how likely a model misclassifies the healthy samples as 'INFECTED'. Based on the evaluation metrics different classification models has the following results. In Table 2, the result of the generated 2x2 confusion matrix is populated and calculated for a share out of the total testing sampled taken. The first columns under each classification models indicate the samples that are found from the generated confusion matrix while the second columns under each model represent the percentage of each sample size in relation to the total samples taken by the splitter.

In general, Random Forest and SVM takes more training time than Naive Bayes and k-NN but the prediction is faster. In some cases of the experiment Naive Bayes and K-NN outperforms SVM. However, the training time and performance totally depends on the scenario taken and the dataset recruited.

Table 2: Name of the Table that justify the values

Samples	Classifier							
	Naïve Bayes		*K-NN*		*SVM*		*RF*	
True Positive	39	47.9 %	45	55.3 %	45	55.3 %	47	57.7 %
True Negative	28	34.3 %	34	41.7 %	35	42.9 %	35	42.9 %
False Positive	7	8.4%	0	0%	0	0%	0	0%
False Ne gative	11	13.3 %	6	5.9%	5	4.7%	3	2.23 %
Total test samples	85		85		85		85	

V. CONCLUSION

Different types of crops can be infected with several kinds of diseases. The naked eye observation of experts is the main approach used in practice for detection and identification of these diseases. But this needs continuous monitoring of experts. This evaluation process is however, tedious, time consuming, and less accurate. The main aim of this research work is to design and implement a model for a classification of wheat leaf images with septoria tritici disease using different image processing and machine learning techniques in combination. The machine learning algorithms and techniques was generally applied at the classification stage while the image processing algorithms and techniques was applied in prior stages of the whole process.

REFERENCES

[1] Solomon Mulugeta Kassa, in Girimite SciTech, Addis Ababa, *Rohobot Publishers*, 2018, p. 258.

[2] Jegadeesh, D. Pujari; Rajesh, Yakkundimath; Abdulmunaf, S. Byadgi, "Image Processing Based Detection of Fungal Diseases in Plants," *in International Conference on Information and Communication Technologies*, 2015.

[3] Anup, Vibhute; S, K Bodhe, "Applications of Image Processing in Agriculture:A Survey," *International Journal of Computer Applications*, vol. 52, no. 2, pp. 34-40, 2012.

[4] Arya, M S; Anjali, K; Divya, Uni, DETECTION OF UNHEALTHY PLANT LEAVES USING IMAGE PROCESSING AND GENETIC ALGORITHM WITH ARDUINO, *IEEE*, 2018.

[5] Bhumika, S.Prajapati; Vipul K, Dabhi; Harshadkumar B, Prajapati, "A Survey on Detection and Classification of Cotton Leaf Disease," *International Conference on Electrical, Electronics, and Optimization Techniques*, pp. 2499-2506, 2016.

[6] VAISHALI, SHARMA; BRAHMDUTT, BOHRA; SAYAR SINGH, SHEKHAWAT, "A REVIEW ON VARIOUS IMAGE SEGMENTATION TECHNIQUES," *International Journal of Computer Science and Mobile Applications*, vol. VI, no. 4, pp. 119-124, 2018.

[7] Nikos Petrellis, A Review of Image Processing Techniques Common in Human and Plant Disease Diagnosis, 2018.

[8] R. Nikita and J. S. Gill, "An Overview on Detection and Classification of Plant Diseases in Image Processing," *International Journal of Scientific Engineering and Research (IJSER)*, vol. V, no. 5, pp. 114-117, 2014.

[9] Enquhone, Alehegn, "Maize Leaf Diseases Recognition and Classification Based on Imaging and Machine Learning Techniques," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 12, 2017.

[10] S. Aasha, Nandhini; R, Hemalatha; S., Radha; K., Indumathi, "Web Enabled Plant Disease Detection System for Agricultural Applications Using WMSN," *Springer Journals*, 2017.

[11] Kangshun, Li; Lu, Doing; Dongbo, Zhang; Zhengping, Ling; Yu, Xue, "The Research of Disease Spots Extraction Based on Evolutionary Algorithm," *Hindawi Journal of Optimization*, 2017.

[12] Peifeng, Xu; Gangshan, Wu; Yijia, Guo; Xianoyin, Chen; Heating, Yang; Rongbiao, Zhang, "Automatic Wheat Leaf Rust Detection and Grading Diagnosis via Embedded Image Processing System," *International Congress of Information and Communication Technology*, no. 107, pp. 836-841, 2017.

[13] S. Aasha, Nandhini; R, Hemalatha; S., Radha; K., Indumathi, "Web Enabled Plant Disease Detection System for Agricultural Applications Using WMSN," *Springer Journals*, 2017.

[14] Pranjali B, Padol; S D, Sawant, "Fusion Classification Technique Used to Detect Downy and Powdery Mildew Grape Leaf Diseases," *International Conference on Global Trends in Signal Processing, Information Computing and Communication*, pp. 298-301, 2016.

[15] Samuel Gebreselassie, Mekbib G. Haile, Mathias Kalkuhl, The Wheat Sector in Ethiopia: Current Status and Key Challenges for Future Value Chain Development, *Bonn: Zef, Center for Development Research, University of Bonn*, 2017.

[16] Michael Beyeler, Machine Learning for OpenCV, *Birmingham: Packt Publishing Ltd.*, 2017.

[17] Gabriel Garrido , Prateek Joshi, OpenCV 3.x with Python By Example Second Edition, *Birmingham: Packt Publishing Ltd.*, 2018.

[18] Nikos Petrellis, A Review of Image Processing Techniques Common in Human and Plant Disease Diagnosis, *Computer Science and Engineering Department, Technological Educational Institute of Thessaly*, 2018.

Citation of this Article:

Koushik Bhattacharyya, Ram Lal, "Classification of Leaf Disease using Image Processing and Machine Learning" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 4, Issue 12, pp 6-12, December 2020. DOI of article https://doi.org/10.47001/IRJIET/2020.412002

****