IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 3, Ser. II (May - June 2019), PP 41-49
www.iosrjournals.org

Recognition of Plant leaf Diseases Using Learning Vector Quantization Neural Network Classifier

Sachin Jadhav[1], Dr. Vishwanath Udupi[2],Dr. Sanjay Patil[3]
[1](Department of Electronics and Telecommunication, Bharati Vidyapeeth college of Engineering, (Ph.D Scholar, V.T.U Belagavi, India)
[2](Electronics and Communication, Maratha Mandal Engineering College Belagavi, India,)
[3](Electronics and Telecommunication, Rajgad Dnyanpeeth Technical Campus, Bhor, Pune, India)
Corresponding Author:Sachin Jadhav

Abstract:*In this work, we applied digital image preprocessing techniques with the Learning Vector Quantization Neural Network (LVQ) to implement auto-diagnosis system for 4 different soybean leaf category classifications. Out of these, 3 categories define diseased soybean leaves database which is affected by diseases such as blight, frogeye leaf spot, and Septoria brown spot and 1 category define healthy leaves database. At first 4 different databases is created ac-cording to defined soybean leaf categories. Then total of 8 training and testing leaf features are extracted out of which 6 are color features and 2 are texture features. These extracted features define as training and testing feature vector set and given as input vector to the LVQ. Further leaf category assigned as target group vector to the LVQ. The probabilistic neural network is trained by230 leaves and classifies 4 types of soybean leaves with performance accuracy of 93%. Compared with other approaches, our algorithm is an accurate artificial intelligence approach which is fast in execution and easy in implementation.*

Date of Submission: 06-05-2019 Date of acceptance: 20-05-2019

I. Introduction

Plant disease has become a major threat to global food security [1]. Plant diseases contribute up to 20% losses in the global harvest of [2]. According to a report [3], our world population is increasing rapidly. Therefore, agricultural production needs to be increased up to 75% to fulfill the food requirements growing population. Plant disease affects both the quality and quantity of the fruits, vegetables etc. and its impact is major losses in production [4, 6]. Most of the plant diseases are caused by fungal. Other severe diseases of plants are caused due to bacteria, viruses [7]. Infected plants are required to be identified by their external foliar and fruit characteristics symptoms on fruits and leaves prior to examination in the laboratory. In most cases, these visible symptoms typically manifest in the middle to later stages of the infection [1]. However, morphological identification of diseases is not reliable. An appropriate method is needed for detection of the causal agent. Traditionally, fungi were identified morphologically followed by isolating and culturing. While biochemical tests were employed to detect bacteria, and viruses were identified based on genetic material, transmission assays and their host range [4]. Recently, the advancements in the field of biotechnology and molecular biology have revolutionized the field of plant disease detection. A plant disease can be detected with the onset of the symptoms by these laboratory techniques. These techniques are also referred to as molecular marker or destructive techniques. They involve destructive leaf sampling followed by chemical treatment.

Gharge S, Singh P [2]propose an algorithm consist of NNs are mathematical models that have been used in data mining. Fundamentally, NNs are an interconnected network of nodes, parallel to the vast network of neurons in the human brain. In an Artificial Neural Network (ANN), each node assigned to the network represents a neuron. Generally, neurons receive the signals from other similar neurons via synapse connection. A neuron typically connects to an individual processing element, which is called perceptron. In a network, the neurons play an important role, they accept and process the inputs and create the outputs [10,11]. Generally, the connection between two neurons carries the weights in which the electrical information is encoded implicitly. Then electrical information simulates with specific values stored in those weights that enable the networks to have capabilities like learning, generalization, imagination and creating the relationship within the network [12].

Major types of NNs This section provides a brief description of the major types of NNs which are Single-Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), Radial-Basis Function (RBF) networks, Kohonen's Self-Organising Map (SOM) networks, Probabilistic Neural Network (PNN), and Convolutional Neural Network (CNN). Multi-Layer Perceptron (MLP) MLP consists more than one hidden layer of perceptron in a network. A common set of layers in an MLP has input, output, and hidden layers. In an ANN, the input layer is the first passive layer acts a conduit for entering the data. The second layer is a hidden layer. Paola and Schowengerdt [13] emphasized the importance of hidden layer in a network to increase the network's ability and for modeling the complex problems. The last layer is the output layer that produces the output signals at the network. Since the SLP is not of practical utility these days, the MPL is most suitable for analyzing hyperspectral data specifically in the context of non-destructive disease detection for high performance classification [13,14]

Moshou et al. [15] used MLP architecture in order to detect yellow rust in wheat crop. The MLP architecture was designed for input layer having neurons equal to the number of processed spectral bands, one hidden layer with different numbers of neurons varying from 5 to 25, and output layer consisting of two neurons, each for healthy and diseased crop. They used a handheld spectrograph (460–900 nm) for capturing the images in wheat field. In this work, four optimal spectral bands were selected. They tested different quantities of neurons, then most efficient neurons were selected for final MLP architecture. The MLP architecture produced over 98% classification accuracy for the healthy plants and over 99% classification accuracy for diseased plants. Recent researches have demonstrated that the MLP is a highly applicable network. Most of the MLP networks are trained with the back-propagation algorithms. Therefore, MLP is a very popular choice among researchers [16]. Back propagation algorithms employ a supervised learning paradigm in MPL, which minimizes errors between the desired outputs and the calculated outputs driven from the inputs and network learning [17]. NN models Different types of NNs are implemented on the basis of specific neural architectures and learning algorithms which in combination are called NN models. The most important NN models are discussed in this following section.

Feed-Forward Neural Network (FFNN) several studies [18-20] have attempted to explain FFNN as a transformation network that transforms input layers to output layers in the forward direction. FFNN is most useful when an end user is interested in input and output layers and not in the hidden layers. Therefore, FFNNs have been increasingly used in non-parametric data analysis. FFNN is an alternative to classic pattern classification and clustering techniques. Hawkins and Bode´n [19] explored the relationship between input and hidden layers in a standard FFNN. They highlighted that one set of connections could be fully connected from the input layer to the hidden layer. These are some of the challenges in NN classifiers to classify different plant disease on the basis of combination of optimal parameters such as texture, color, and shape in a normal camera image [13].

The general objectives of this research are: 1. To discuss applicability of NNs to the analysis of soybean leaf disease for early disease detection 2. To implement Linear Vector Quantization (LVQ) model for detection of defined soybean plant diseases using NN classifiers

II. Material and Methods

Proposed Method

The proposed work involves four modules: Image collection, Feature extraction, diseased plant leaves classification and Performance evaluation. The step by step process is given below.

1. Diseased soybean leaves collected from crop fields.
2. Data set of each disease was created and categorized in to training and test dataset
3. Extract the color and texture feature of soybean diseased leaf.
4. Soybean disease Classification done by using Feed forward neural network algorithm and Learning vector quantization algorithm.
5. Evaluation Matrix: The performance metrics are calculated to evaluate the performance such as Accuracy, Precision etc.

A. Data set: The proposed pre-trained GoogleNet and AlexNet deep CNNs are then used to classify defined test images from test database. Data of soybean images were collected from soybean fields, Kolhapur district, Maharashtra, India. In this study, 80 testing data sets of soybean leaf images were used. The training data sets which included 150 images with blight disease, 150 images with Frogeye leaf spot disease, 100 images with brown spot disease and 150 images with non-disease(healthy). Fig. 1 depicts the leaves samples of the testing data. The summary of our dataset are provided in Table 1. The total number of sample images in our dataset is 550 fragmented into 3 disease category and 1 non disease category.

Bacterial Blight	**Frogeye Leaf Spot**	**Brown Spot**
		

Fig 1: Soybean leaf disease database samples.Three different soybean fungal diseases from left to right show the Bacterial Blight disease, Cercospora Leaf Spot, and Septoria Brown Spot disease.

B. Feed-forward neural networks

Feed-forward neural networks (Rumelhart et al., 1986) provide a general framework for representing non-linear functional mappings between a set of input and output variables. For training, Bayesian regularization (Bishop, 1995) was used.

A validation set was used to test the generalization performance of the neural network. The number of neurons in the input layer is 6 which are vector input of extracted features of diseased leaf training samples. The output layer had 1neurons representing the leaf disease type which is categorized in to 4 classes such as:

1. Blight disease
2. Frogeye leaf spot disease
3. Brown spot disease and
4. Healthy leaf.

The numbers of neurons in the hidden layer were set to 10. However, using 10-hidden neurons improves the classification result while the results when using five-hidden neurons were inferior compared to the 10-hidden neuron network. The network consisted of three layers and two mapping functions for hidden and output nodes as shown in Fig.2.

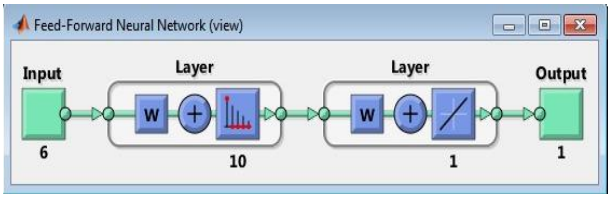

Fig.2. Feed Forward Network view

Equally, adding more neurons in the hidden layer does not improve the generalization result when Bayesian regularization is used as the training algorithm because weights that are not used remain small, therefore, the extra hidden neurons become inactive. Hence, the preferred neural network configuration was the one with the lowest complexity from the validated ones that did not compromise the performance.

Feed forward network training

An operation of updating reference vectors repeats till classification rate is achieved or maximum number of epochs is reached. When the network weights and biases are initialized, the network is ready for training. The multilayer feed forward network can be trained for function approximation (nonlinear regression) or pattern recognition. The training process requires a set of examples of proper network behavior—network inputs p and target outputs t.

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \quad (1)$$

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance, as defined by the network performance function net. Perform function. The default performance function for feed forward networks is mean square error MSE—the average squared error between the networks outputs and the target outputs t shown in eq (1).

C. Learning Vector Quantization neural networks:

Learning Vector Quantization, proposed by Kohonen [11], is a neural network that combines competitive learning with supervised learning. It is a powerful and heuristic algorithm for solving classification problems. Due to its simple topology and adaptive model, LVQ has been widely used in many applications. It classifies the given data in a fixed number of classes [12]. It consists of three layers with input, Kohonen (competition) and output layer. Input layer neurons collect the values of the input variables. Each neuron in the output layer represents a class of input. The input and Kohonen layers are fully connected, while the Kohonen and output layers are partially connected. The learning occurs in Kohonen layer. The classified results are passed to the linear output layer [13]. The LVQ network architecture is shown in Fig. 3. LVQ (learning vector quantization) neural networks consist of two layers.

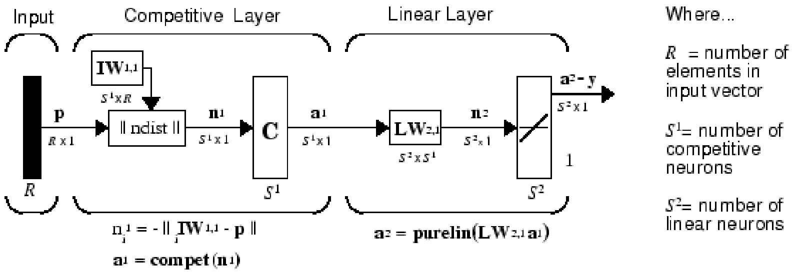

Fig.3: LVQ network architecture

In LVQ, the reference vectors consisting of weights are used to represent the classes for learning. The learning is based on the similarity between the input vector and reference vectors. Only one of the outputs takes the value 1 and the others take the value 0. The reference vector that receives the value 1 gives the class of the input vector. LVQ model works according to the “winner-takes-all” approach and only the weights of the winning reference vector which is closest to the input vector are updated at every iteration. The winning reference vector is found via calculating the Euclidean distance from the input vector to each of the reference vectors by (2);

d = arg mini {x − wi} (2) where x is input vector, wi is i. reference vector. Reference vectors are updated by (3) if classification is correct, otherwise updated by (2):

wi(t + 1) = wi(t) + η(t)(x − wi(t)) (3) wi(t + 1) = wi(t) − η(t)(x − wi(t))

(2)

where η ∈ (0, 1). η is the learning rate, and this rate is decreased monotonically with time. If the reference vector and input vector classes are matched, the reference vector is moved towards input vector. Otherwise, it is moved away from the input vector [14].

In this research color and texture features both are extracted to get better accuracy. For feature extraction the methods computed values mean, standard deviation of RGB color channels are used in color feature extraction and for texture feature extraction GLCM method is used [20]. Based on the GLCM-kutoisis and skewness, total eight statistical parameters including color features are computed with eight feature vectors below:

D. Feature Extraction:

i) Mean

$$\text{Mean:}\mathrm{M}' = \frac{1}{\mathrm{N}}\mathrm{M_i} \qquad (3)$$

Where M_i is the pixel intensity and N is the total number of pixels. Here mean is considered as one of the main features for RGB each channel (R-mean, G-mean, B-mean).

ii) Standard Deviation

Standard Deviation is the square root of the variance of the distribution. It is calculated for RGB each channel (R-std, G- std, B- std) using following formula:

$$\text{Standard Deviation } (\sigma) = \sqrt{\frac{1}{N}}\sum_{i=1}^{N}(Mi - M')^2 \qquad (4)$$

iii) Skewness

The skewness is used to judge the image surface. It is used to detect edges in dark objects on white background, having a sign change at luminance changes in images based on degree 3 and 4 moment, so these are termed higher order statistics [22].

$$\text{Skewness}=\frac{\frac{1}{N}\sum_{i=1}^{N}(\mathrm{Mi}-\mathrm{M}')^{3}}{\left(\frac{1}{N}\sum_{i=1}^{N}(Mi-M')^{2}\right)^{3/2}} \quad (5)$$

III. Result and Discussion

The classification of soybean diseased plant leaves performance of various neural network techniques which have been analyzed for the 243 input leaf images. The performance evaluated for the neural network techniques which have been used in this paper from the confusion matrix of their respective classifier. The Figure-3 shows the confusion matrix for feed forward neural network classification results for 118 input leaf samples.

A. Confusion matrix

The diagonal cells show the number of cases that were correctly classified, and the off-diagonal cells show the misclassified cases. The blue cell in the bottom right shows the total percent of correctly classified cases (in green) and the total percent of misclassified cases (in red). The results show very good recognition.

Fig. 4: Confusion matrix

Overall Confusion Matrix: matrix of output and target class; trained result gives accuracy of 98.8% where class 1 is one time misclassified as class 4 by 1.3%. Blue box indicates accuracy result.

Fig. 5 depicts the overall confusion matrix of LVQ which shows summary of results of soybean leaf disease classification for the 4 classes. The accuracy is improved and reaches to 80.4% which is a good amount as shown in diagonal blue box. It clearly shows that class 1 is 2 time misclassified as class 2 by 1.3%, class2 is one time misclassified as class 1 by 1.3%, and class 3 is one time misclassified as class 1by 1.3% but class 4 is correctly classified as class 4 by 100%.

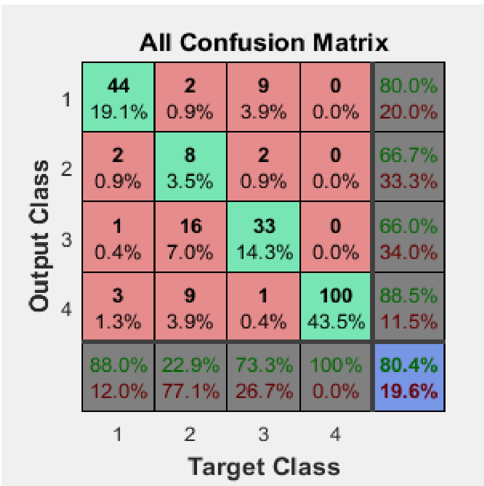

Fig. 5 Overall confusion matrix of LVQ

B. Receiver Operating Characteristic (ROC) curve

The colored lines in each axis represent the ROC curves. The *ROC curve* is a plot of the true positive rate (sensitivity) versus the false positive rate (1 - specificity) as the threshold is varied shown in Fig. 6 and 7 respectively. A perfect test would show points in the upper-left corner, with 100% sensitivity and 100% specificity. For this problem, the network performs very well.

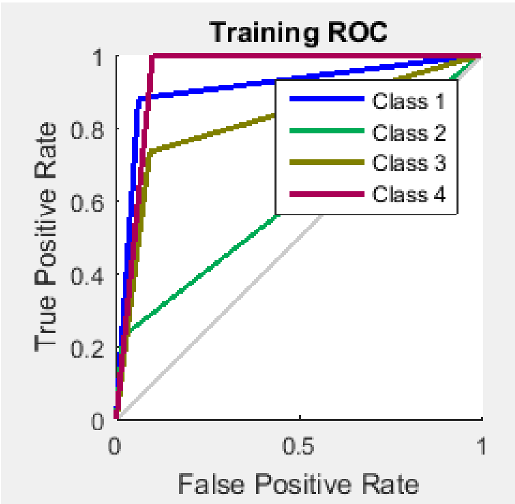

Fig. 6. ROC curve for sensitivity

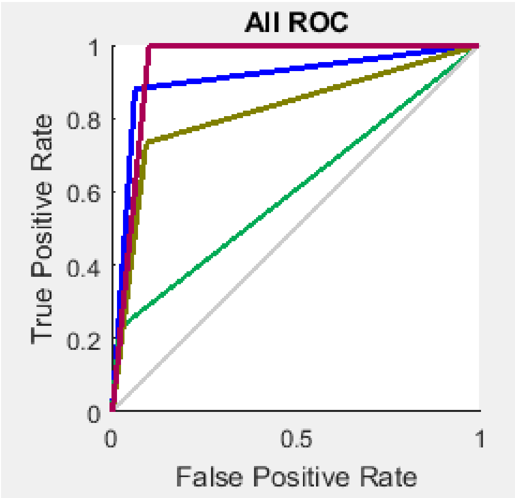

Fig. 7.ROC curve for specificity.

C. Training of Neural Network

The recognition system is implemented using MATLAB. The image is taken as dataset input and feed-forward architecture is used. Firstly, neural network is trained using known dataset of different plant leaves. After training, system is tested using several unknown dataset and result is obtained. The desired performance goal is achieved in 7 epochs shown in Fig. 8. Fig. 9 shows performance graph. It specifies the gradient at epoch 7, validation checks and also learning rate.

Recognition rate = Number of leaves identified correctly/Total number of leaves∗100

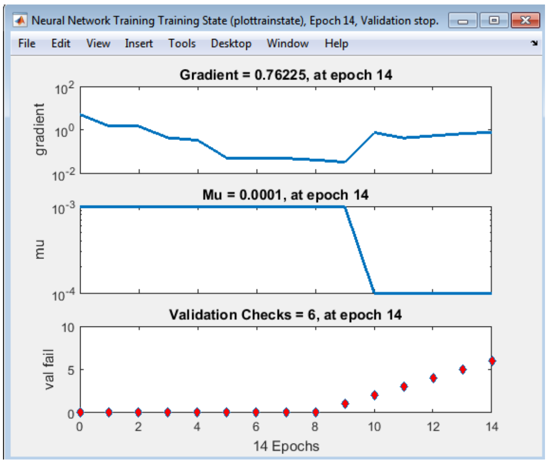

Fig. 8. Neural network training

An input vector is created by placing first 60 values out of 80 of segmented leaf images. 8 leaf images are selected for validation. For testing phase, 12 images are taken randomly. The following parameters are used for creating network for training: No. of neurons in input layer: 10 No. of layers: 2 No. of epochs: 13 Transfer Function: “transig” Adaptive Learning Function: “LEARNGDM” Validation checks = 6 Performance Function: “MSE” as shown in Fig. 9 and Fig.10 respectively.

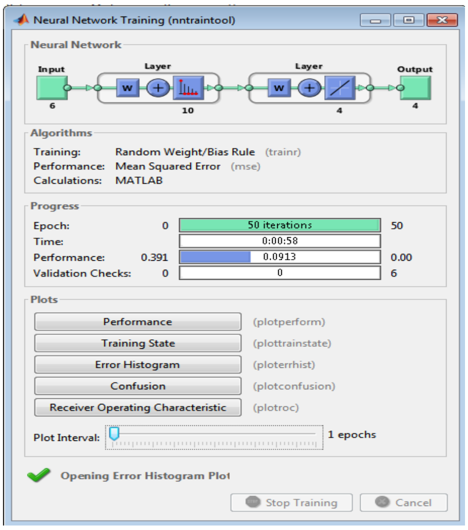

Fig. 9. Neural network testing

D. Performance Analysis

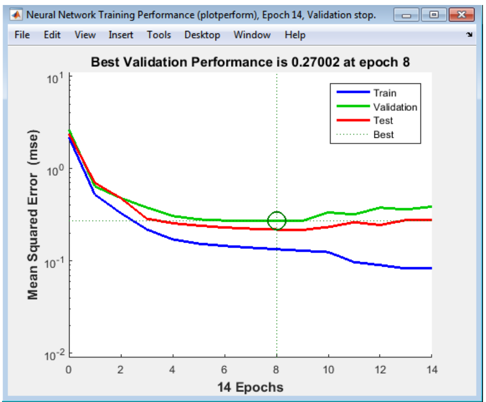

Fig. 10. Performance analysis Graph:

The graph of mean square error vs. epochs; blue line indicates the performance of trained data, green for validation data, red for test data. Best Validation Performance is 0.003035 at epoch 7.

E. Error Histogram

Fig. 11 shows the error histogram to obtain additional verification of network performance. The blue bars represent training data, the green bars represent validation data, and the red bars represent testing data. The histogram can give you an indication of outliers, which are data points where the fit is significantly worse than the majority of data. In this case, you can see that while most errors fall between -5 and 5, there is a training point with an error of 17 and validation points with errors of 12 and 13. These outliers are also visible on the testing regression plot. The first corresponds to the point with a target of 50 and output near 33. It is a good idea to check the outliers to determine if the data is bad, or if those data points are different than the rest of the data set. If the outliers are valid data points, but are unlike the rest of the data, then the network is extrapolating for these points. You should collect more data that looks like the outlier points, and retrain the network.

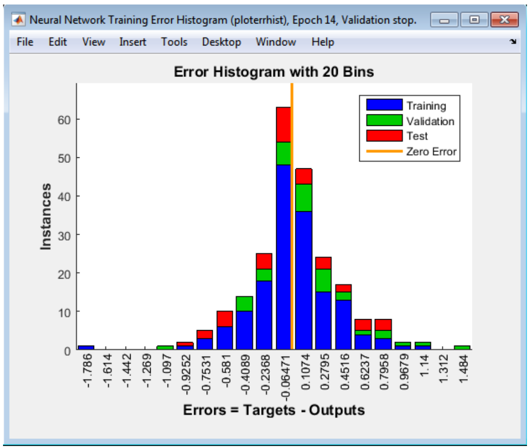

Fig. 11. Error histogram

IV. Conclusion

In this paper the neural network algorithm is proposed for diseased plant leaf classification. The neural network techniques such as feed forward neural network (FFNN), learning vector quantization (LVQ) and radial basis function network (RBF) were tested for two different diseased leaf image classifications such as bean and bitter gourd leaves. The performance is measured using evaluation matrix with various parameters the performance is analyzed and based on the analysis the LVQ classification approach provides better result.

Acknowledgment

This work is supported by Dr.V.S. Patil, Plant Pathology, Zonal Agricultural Research Station, Kolhapur under Mahatma Phule Krishi Vidyapeeth, Rahuri, Maharashtra India; Dr. Mahajan, Assistant Professor, College of Agriculture Kolhapur, Maharashtra, India.

References

[1]. Strange RN, Scott PR. Plant disease: a threat to global food security. Annu Rev Phytopathol 2005;43(1):83-116.
[2]. Society for General Microbiology. Combating plant diseases is key for sustainablecrops.Link: www.sciencedaily.com/releases/2011/04/110411194819.htm. 2013.
[3]. Food and Agriculture Organization (FAO). Declaration of the World Summit onFood Security. Rome: Food and Agriculture Organization. 2009.
[4]. Sharma P, Sharma S. Paradigm shift in plant disease diagnostics: a journey fromconventional diagnostics to nano-diagnostics. In: Kumar P, Gupta KV, Tiwari KA, Kamle M, editors. Current trends in plant disease diagnostics and management practices. Switzerland: Springer International Publishing AG; 2016. p. 237-64.
[5]. Arivazhagan S., Newlin Shebia R. 2013. Detection ofunhealthy region of plant leaves and classification of plant leaf diseases using texture features. Agricultural Engineering Institute: CIGR journal. Volume 15, No.1.
[6]. Desai M, Jain AK, Jain NK, Jethwa K. Detection and classification of fruit disease: a review. Int Res J Eng Technol 2016;3(3):727-29.
[7]. Kamlesh Golhani, Siva K Balasundram, Ganesan Vadamalai, Biswajeet Pradhan, "A Review of Neural Networks in Plant Disease Detection using Hyperspectral Data." Information Processing in Agriculture(2018), doi: https://doi.org/10.1016/j.inpa.2018.05.002
[8]. Bauer S. D., F. Korc., W. Forstner. 2011. Thepotential of automaticmethods of classification toidentify leaf diseases from multi spectral images'Precision Agriculture. Vol 12, pp 361-377.
[9]. Lu D. and Weng Q. 2007. A Survey of ImageClassification Methods and Techniques for Improving Classification Performance. International Journal of Remote Sensing. pp 823–870.A.
[10]. Shiffman D. Neural Networks. The Nature of Code: Simulating natural systems with processing (Online). Link: http://natureofcode.com/book/chapter-10-neuralnetworks/. 2012.
[11]. Karray FO, De Silva CW. Soft computing and intelligent systems design: theory, tools, and applications. London: Pearson Education; 2004.
[12]. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 1943;5(4):115-33.

[13]. Abdulridha J, Ehsani R, de Castro A. Detection and differentiation between Laurel wilt disease, phytophthora disease, and salinity damage using a hyperspectralsensing technique. Agriculture 2016;6(4):56.

[14]. Paola JD, Schowengerdt RA. A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. Int J Remote Sens 1995;16(16):3033-58.

[15]. Abdulridha J, Ehsani R, de Castro A. Detection and differentiation between Laurel wilt disease, phytophthora disease, and salinity damage using a hyperspectral sensing technique. Agriculture 2016;6(4):56.

[16]. Moshou D, Bravo C, West J, Wahlen S, McCartney A, Ramon H. Automatic detection of “yellow rust” in wheat using reflectance measurements and neural networks. Comput Electron Agric 2004;44(3):173-88.

[17]. Ghaiwat SN, Arora P. Detection and classification of plant leaf diseases using image processing techniques: a review. Int J Recent Adv Eng Technol 2014;2:2347-812.

[18]. Lu Y, Yi S, Zeng N, Liu Y, Zhang Y. Identification of rice diseases using deep convolutional neural networks. Neurocomputing 2017;267:378-84.

[19]. Plate TA, Bert J, Grace J, Band P. Visualizing the function computed by a feedforward neural network. Neural Comput 2000;12(6):1337-53.

[20]. Hawkins J, Boden M. The applicability of recurrent neural networks for biological sequence analysis. IEEE/ACM Trans Comput Biol Bioinforma 2005;2(3):243-53.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Sachin Jadhav. "Recognition of Plant leaf Diseases Using Learning Vector Quantization Neural Network Classifier.” IOSR Journal of Computer Engineering (IOSR-JCE) 21.3 (2019): 41-49.