DE GRUYTER

Creative Commons - Some Rights Reserved

Open Access | Published by **De Gruyter Open Access** | December 8, 2021

Classification of plant diseases using machine and deep learning

Monika Lamba, Yogita Gigras and Anuradha Dhull

From the journal Open Computer Science
https://doi.org/10.1515/comp-2020-0122

Citations 7

Abstract

Detection of plant disease has a crucial role in better understanding the economy of India in terms of agricultural productivity. Early recognition and categorization of diseases in plants are very crucial as it can adversely affect the growth and development of species. Numerous machine learning methods like SVM (support vector machine), random forest, KNN (*k*-nearest neighbor), Naïve Bayes, decision tree, etc., have been exploited for recognition, discovery, and categorization of plant diseases; however, the advancement of machine learning by DL (deep learning) is supposed to possess tremendous potential in enhancing the accuracy. This paper proposed a model comprising of Auto-Color Correlogram as image filter and DL as classifiers with different activation functions for plant disease. This proposed model is implemented on four different datasets to solve binary and multiclass subcategories of plant diseases. Using the proposed model, results achieved are better, obtaining 99.4% accuracy and 99.9% sensitivity for binary class and 99.2% accuracy for multiclass. It is proven that the proposed model outperforms other approaches, namely LibSVM, SMO (sequential minimal optimization), and DL with activation function softmax and softsign in terms of *F*-measure, recall, MCC (Matthews correlation coefficient), specificity and sensitivity.

Keywords: deep learning; plant disease; auto-color correlogram; machine learning algorithm; ACC

1 Introduction

In India, agriculture is a supporting factor for all the existing cultures and 70% of the population depends on the agriculture sector. Agriculturalists have a large variety of choices to select vegetable and fruit crops. Practices like crop rotation, pesticides, irrigation, and fertilizers in agriculture have been practiced for a long time. Farmers face multiple problems in detecting and identifying the diseases in crops [1]. The introduction of seed, soil, chemical-based approaches to agriculture was the output of the production system; however, wise and careful management of every input is needed for the survival of agriculture's complex system. So, advancement is required in the production of plants in a proper manner. Plants are seen everywhere on the Earth. Thus, identifying disease is a very important step in agriculture. Plant diseases and chemical fertilizers are the major issues that can affect the cultivation of rice, tomato, potato, and pepper plants. Hence, a more serious diagnosis and proper handling of crops on time are needed to prevent them from heavy losses. Detecting plant diseases is an important factor as it can affect the life of an animal as well as human beings and cause a lot of changes in the quantity and quality production of crops [1]. Hence, the detection and classification of plant diseases is a vital task. Diseases in tomato, potato, pepper, and rice plants can occur in several parts like roots, stems, leaves, and fruits. Major diseases affecting the rice crop are leaf_smut, brown_spot, and bacterial_leaf. In the case of potatoes, diseases are early_blight and late_blight; pepper diseases can be bacterial, whereas in the case of tomato it can be target_spot, leaf_mold, mosaic_virus, yellow_leaf_curl_virus, bacterial_spot, early_blight, healthy, late_blight, septorial_leaf_spot and spider_mites_two_spotted_spider_mite. The diseases of rice, tomato, potato, and rice are shown in Figure 1. Improvements are obtained using Deep Learning (DL) tactics in image-based detection of diseases developed with autor-color correlogram and DL for image-based automatic diagnosis and detection of plant disease asperity. The model's competence for accurately forecasting the disease severity stage is better than other machine learning algorithms.



(a) Brown spot Leaf Smut Bacterial leaf bright


(b) Bacterial Healthy



(c) Early blight Healthy Late blight



(d) Target_spot Mosaic_virus Yellow_leaf_curl_virus


Bacterial_pot Early_blight



Healthy Late_blight Leaf_mold


Figure 1

Samples of leaf images showing different types of diseases in (a) rice, (b) pepper, (c) potato, and (d) tomato plants.

Our experimental results are decisive steps towards the plant disease severity investigation. The overall flow of the remaining of the paper is as follows: Section 2 reviews the literature survey, Section 3 presents the proposed model in detail, Section 4 describes the auto-color correlogram filter approach, Section 5 describes the DL technique, Section 6 is a performance measure related analysis, Section 7 represents experimental study, Section 8 describes the results and discussion, and lastly, Section 9 gives the conclusion.

2 Literature review

Akhtar et al. presented an approach, where features were trained using SVM, *k*-NN, RNN(recurrent neural network), Naïve Bayes, decision tree, and detection of disease using wavelet-based plant disease and gray-level co-occurrence matrix [2]. Using hyperspectral measurement, Fathima et al. presented techniques to identify diseases in leaf [3]. The asperity of diseases in plant leaves is proposed by Karthik et al. [4]. An automated approach by Semary et al. used the texture as well as color features and SVM for categorization. Gabor wavelet features and GLCM (gray-level co-occurrence matrix), using weighted KNN, features were trained in ref. [5]. Detecting disease using texture and color features [6] is presented by Padol et al. where the infected area was segmented with support of *k*-means and features were trained and used for classification using SVM. Using *k*-means for disease detection and categorization was used in ref. [5]. Identifying the presence of fungal infection in leaves using *k*-means [7] is presented by Mehra et al. Major challenges in clustering algorithm is finding the count of clusters and adjusting the parameters to separate from each cluster. Scale-invariant features were examined for multiple problems in the image [3,4,8]. The use of scale-invariant as an approach for disease identification was presented by Dandawate and Kokare [9] where features were examined with the help of SIFT (scale-invariant transforms are trained using the support vector machine to identify the presence of disease).

To have an effective classification, SIFT features were joined and combined with SB distribution for detecting diseases in tomato leaves [10]. As various factors are responsible for low production of all crops like potato, tomato, and rice, etc., hence, image segmentation in pattern recognition and image analysis is the first step, as it is the necessary component and one of the important tasks in processing the images and analyzing the result. Image segmentation is defined as a method of partitioning the image into disjoints sets with *k*-means clustering methods [11,12,13]. Sannakki and Rajpurohit

used a classification method depending on the backpropagation neural network, explaining the defected area, color and texture in segments used as features [14]. This categorization has achieved an accuracy of 97.3% with only limited crops as input. Rothe presented a pattern recognition technique using snake segmentation. The proposed model assesses the vitality in the infected part of the leaf securing a classification accuracy of 85.52% [15]. Rastogi used *k*-means to segment the affected part and used ANN (artificial neural network) for classification that helps in identifying the hardness of the diseased leaf [16]. Tian used SVM as a classifier for finding the disease in wheat plants [17]. Owomugisha used decision trees, nearest neighbors, naïve Bayes, and random forest for diagnosing diseases in plants [18]. Hall et al. [19] used random forest and CNN on 32 species for classification of leaves and performance achieved is 97.3% classification accuracy. Using image multiscale representation and the leaf number improved counting accuracy by using deep CNN [20,21,22]. Identification of a variety of plants using DL methods [23,24,25], where it was used a modified CNN together with the architecture of AlexNet using accuracy parameter [23,25]. For crop discrimination [26,27], authors introduced CNN and evaluated the performance using accuracy and recall with the help of two datasets. DL was studied for the recognition of plants and successfully achieved a 91.78% success rate [28]. For the classification of crop variety, Kussul et al. [29] mentioned modified CNN according to Mortensen et al. [30] applying VGG 16, enforcing three-unit LSTM [31], and used the RGB histogram method and CNN [30]. Accuracy purely depends on the nature of handcrafted features selected. The performance of the work needs to be updated by large datasets. These shortcomings can be overcome using the model of DL for plant disease discovery [32]. Multiple research articles have summarized the research on agriculture in the field of plant diseases using DL [21,22,23,24,25,28,29,33,34]. Contemplating the advantages of DL, we have chosen them after ACC in our proposed model.

3 Proposed model

The analysis of the methodology of the proposed work has been presented using images in Figure 1. Image datasets are collected from the rice dataset [35] and the Kaggle plant village dataset [36]. These images go through pre-processing called filtering. After the pre-processing step, different texture and color features are collected. The features obtained are fed as an input in DL to correctly classify the disease of a given plant.

3.1 Input image

In the proposed model, the initial step is to gather and capture the image dataset to obtain relevant features, and the obtained features are stored in the database. The rice dataset consists of three classes: bacterial leaf bright (40), brown spot (40) and leaf smut (40). The pepper datasets consist of two classes as healthy (997) and bacterial (997). The potato datasets consist of three classes, namely early-blight (1,000), healthy (152) and late blight (1,000). The tomato datasets consist of ten classes, namely spider_mites_two_spotted_spider_mite (1,676), target_spot (1,404), mosaic_virus (373), yellow_leaf_curl_virus (3,209), bacterial_spot (2,127), septorial_leaf_spot (1,771), early_blight (1,000), healthy (1,591), late_blight (1,909) and leaf_mold (952). The numbers in parenthesis show the count of samples for the category of plant diseases.

3.2 Image database

The next step is creating the database of images. The database of rice consists of 120 images, pepper consists of 1,994 images, potato consists of 2,152 images and tomato consists of 16,072 images. The images present in the database are responsible for good and accurate capability of classification that enhances the strength of the algorithm.

3.3 Image pre-processing

It is defined as an operation on images with an improvement in images to correct distortion and enhance the important features required for processing and analyzing the images. This step does not add any further information to the image and helps in reducing the redundancy in images.

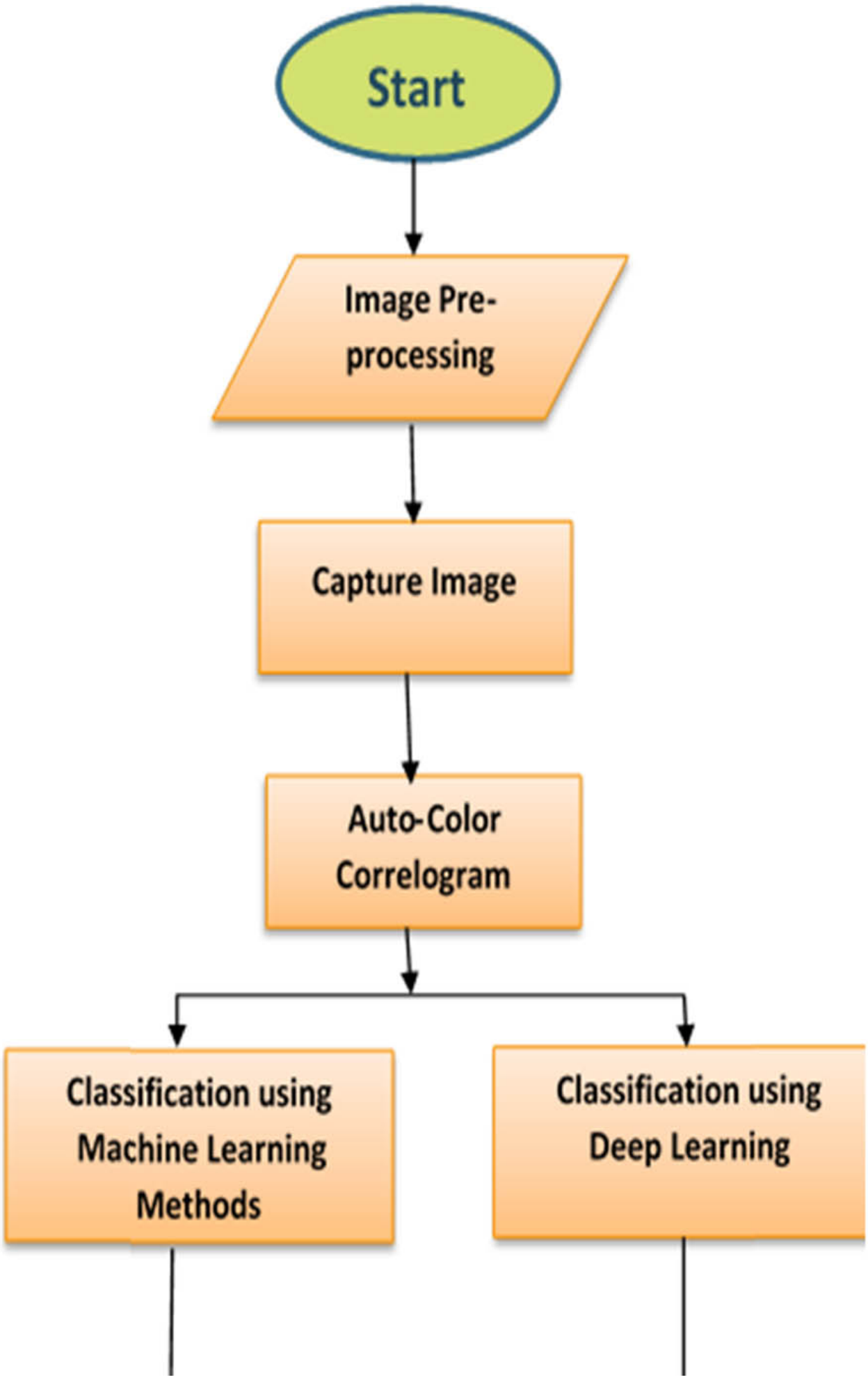
4 Auto-color correlogram (ACC) filter

It is a pre-processing step defined as a filter that helps in calculating a color correlogram from the input image. It encodes the dimensional correlation of colors in images and is effective in changing the viewing position. An ACC is defined as the average of all color pixels of color *Hj* at a gap (length) ith against a pixel of color *Hj* in the image [37] (Figure 2).

$$\mathrm{ACC} = I\ (x, y)\,, \quad x = 1, 2, 3, \ldots A, \ \ y = 1, 2, 3, \ldots B,$$

is explained as

$$\begin{aligned} \mathrm{ACC}(i, j, k) = AHj(k)HiHj = RmHj(k)HiHj(I), \\ GmHj(k)HiHJ(I), BmHj(k)HiHj(I)|Hi \neq Hj. \end{aligned} \tag{1}$$


Start
Image Pre-processing
Capture Image
Auto-Color Correlogram
Classification using Machine Learning Methods
Classification using Deep Learning

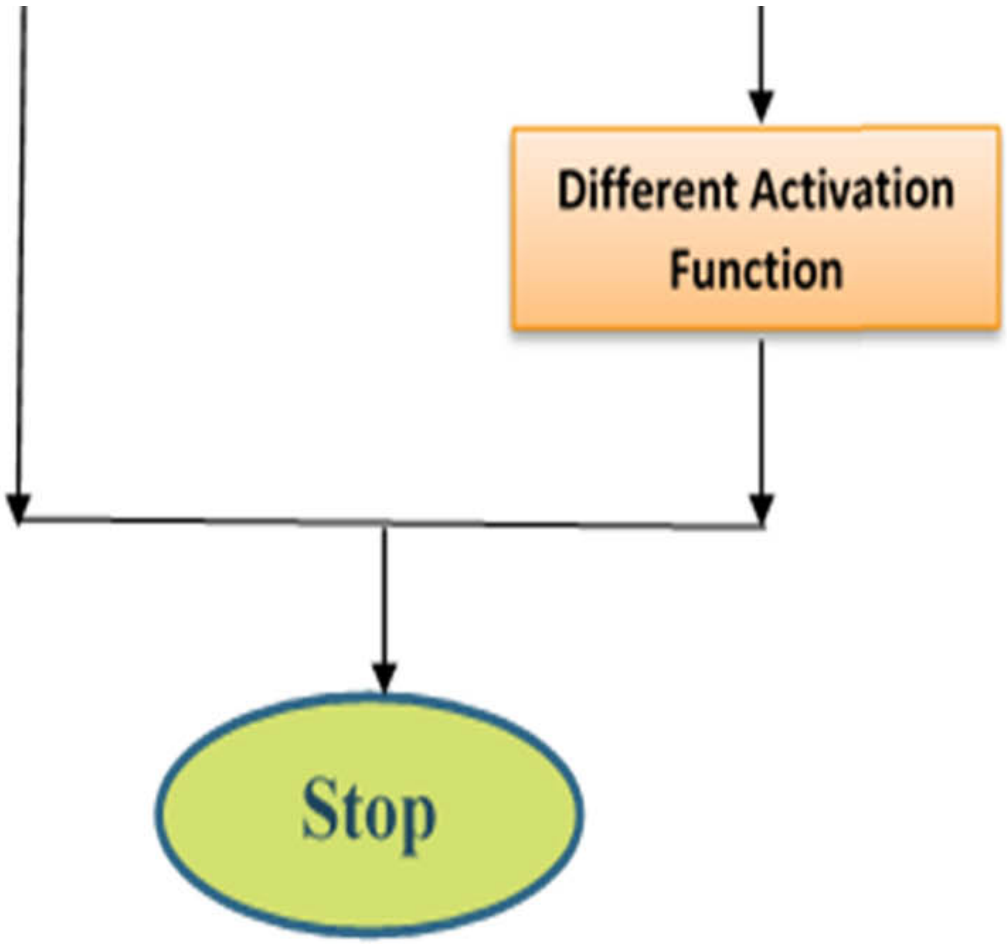


Figure 2

Flow chart of the proposed model.

Image (*i, j*) is defined as quantized to *m* colors *H*1, *H*2, *H*3,..., *Hm*, and the gap between two pixels *k* [min(*A* < *B*)] is fixed considering A *Hj* to be the RGB value of color m in the image. The average colors are defined as

$$RmHj(k)HiHj(I) = (\pi(k)Hi, RHj(I))/(\pi(k)Hi, Hj(I))|Hi \neq Hj, \quad (2)$$

$$GmHj(k)HiHj(I) = (\pi(k)Hi, GHj(I))/(\pi(k)Hi, Hj(I))|Hi \neq Hj, \quad (3)$$

$$BmHj(k)HiHj(I) = (\pi(k)Hi, BHj(I))/(\pi(k)Hi, Hj(I))|Hi \neq Hj \quad (4)$$

.

Here, x represents the accounting color *Hi* against *Hj* at a gap of k . $\pi(k)Hi$, *xHj*(*I*) denominator is the sum of pixel values of color *Hj* at point k from color *Hi* when x is the RGB color space, *Hj* ≠ 0.

This reduces the intensity of the color correlogram 0(m2d) to 0(3md), and the ACC can be used to find the logical spatial correlation among colors to decrease the size of storage and increase the speed of the retrieval size of ACC from 0(3md) to 0(m). With the help of the method, RGB dominant captured the peak values in any color bin. The dominant elements are gathered to reduce the storage speed and the amount of retrieval while processing (similarity) calculation of the two images [38].

5 Deep learning

Contrary to the old machine learning approaches, DL has the potential to implement itself directly on the input data and it does not require any handcrafted features. From the concept of the learning pattern of the human brain, the architecture of neural networks was developed. Having two or more hidden layers lead to DL. As a brain neuron receiving stimuli from the outside world and then processing the input and producing output at the end, multiple neurons are connected to form a complex network that helps in learning complicated patterns from the dataset. The detailed architecture of DL consisting of input layers tracked by multiple hidden layers and output layers using an activation function is shown in Figure 3. Determining the total count of neurons and the hidden layers depends on the trial-and-error rule [39]. Every neuron is characterized by some of its weight, activation function and bias. The images are fed into the DL using the input layer, and the neuron will perform a transformation described by equation (5).

$$X = (\text{weight} * \text{input}) + \text{bias}. \tag{5}$$

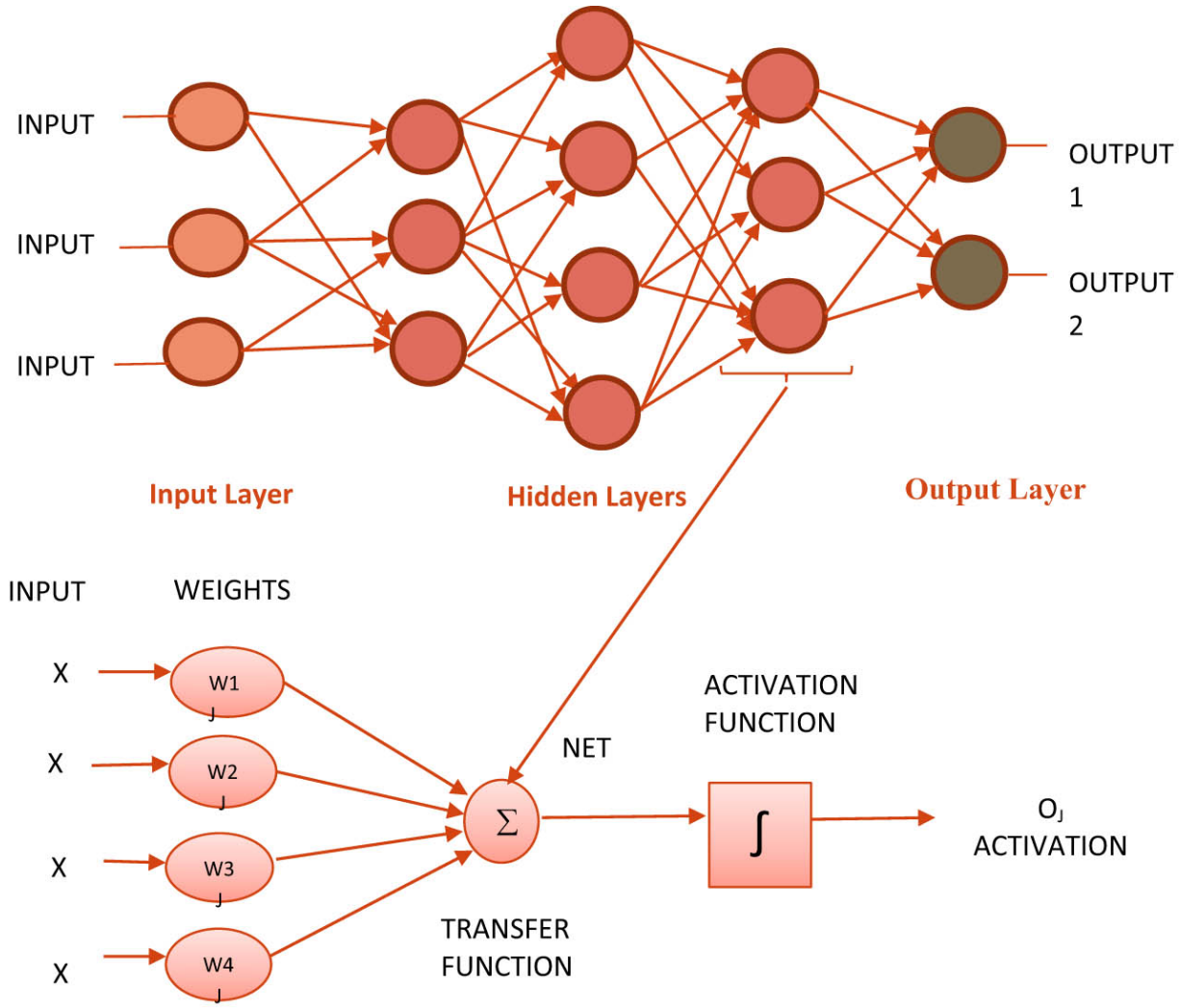


Figure 3

The architecture of deep learning.

Applying the activation function on the linear transformation, the output produced is supplied as an input to the next hidden layer and the same is repeated. The transfer of information is called forward propagation.

$$Y = \text{Activation}((\text{weight} * \text{input}) + \text{bias}). \quad (6)$$

A network without the activation function will be less powerful and will not be suitable for many problems having a complex pattern. Without the activation function, it is known as a linear regression model.

In DL, when the input is given to the neurons along with weights, the weighted sum of input is generated that is further given to the activation function to generate the output. The activation function (AF) is utilized to map the input to the output. Due to AF, the DL learns complicated patterns in the datasets easily. The important uses of AF are as follows:

a. To maintain output limited to a specific range.

b. To include nonlinear functionality in data.

Different AF exists in the literature [45,46,47] and some of them are described in detail as follows.

5.1 Kinds of activation functions

1. Sigmoid: It is defined as a nonlinear function producing values between 0 and 1, and the output produced will be nonlinear of the same sign. This is not symmetric around zero:

$$f(x) = \frac{1}{1 - e^{-x}}. \tag{7}$$

2. tanh: It is like a sigmoid function but it is symmetric around 0 producing values lying between [−1, 1], and the output sign may not be the same:

$$\tanh(x) = 2 * \text{sigmoid}(2x) - 1, \tag{8}$$

$$g(x) = 1 - 2\tanh^2 x. \tag{9}$$

   The output produced in the case of sigmoid and tanh function has lower and upper limits.

3. ReLu: It is nonlinear and stands for a rectified linear unit. ReLu has one advantage over the other activation function as it never initiates all the neurons at the same time. If the value of output is less than 0, the neuron will get deactivated. It is more computationally efficient than the sigmoid and tanh functions:

$$f(x) = \max(0, x). \tag{10}$$

4. Softmax: It consists of multiple sigmoids. The sigmoid returns values between 0 and 1 and treats probabilities of data relating to the class used for the multiclass problem:

$$\pi(z)j = \frac{e^{zj}}{\sum_{k=1}^{K} e^{zj}}, \quad \text{for } j = 1, 2, 3, 4, \ldots k. \tag{11}$$

5. Softplus: It is a kind of traditional function introduced in 2001 and is differentiable and easy to demonstrate due to its derivative.

   It is an alternative to dead ReLu. The output lies between [0 to ∞]:

$$y = \log(1 + e^x). \tag{12}$$

6. Softsign: It is more like the tanh function; the difference is tanh converges exponentially and softsign converges polynomially. The output produced is between [−1, 1]:

$$y = \frac{1}{1 + \frac{1}{x}}. \tag{13}$$

7. Hard sigmoidal: It is a variation of the sigmoidal function and is given by

$$f(x) = \max\left(0, \min\left(1, \frac{(x+1)}{2}\right)\right). \tag{14}$$

Hard sigmoidal offers a lower computation cost [43].

8. Hard tanh: It is a variation of the tanh activation function, and represents a cheaper and more computational efficient version of tanh. It reports both accuracy and speed enhancements [43]. The function values lie between [−1, 1] and are given by

$$f(x) = \begin{cases} -1, & x < -1 \\ x, & -1 \le x \le 1 \\ 1 & x > 1. \end{cases} \tag{15}$$

9. Rational tanh: It is also known as a rational approximation of a hyperbolic tangent [44] and is represented as

$$f(x) = 1.7159 \times \tanh\left(\frac{2}{3}x\right), \tag{16}$$

$$\tanh(y) = \operatorname{sgn}(y)\left(1\frac{1}{1 + |y| + y^2 + 1.41645 \times y^4}\right). \tag{17}$$

A detailed description of different AFs with advantages and disadvantages are given in Table 1.

Table 1

Advantages and disadvantages of activation functions

Types of Activation function	Advantages	Disadvantages
1. Sigmoidal/Logistic [45]	· Smooth gradient, precluding "jumps" in output values	· Vanishing gradient: make network declining to learn further and being very slow to attain a precise prediction
	· Clear predictions	
	· Smooth function	
	· Continuously differentiable	· O/P is not zero centered
	· O/P is nonlinear	· Computationally costly
	· Easy to apply and understand	· Slow convergence
2. Tanh/hyperbolic tangent [45]	· Zero centered	· Like the sigmoid function
	· Continuous and differentiable at every point	· Vanishing gradient problem
	· As the function is nonlinear it can effortlessly backpropagate the faults/errors	· Low gradients
3. ReLU [45]	· Computationally efficient: permits the network to unite very promptly	· The Dying ReLU problem: once inputs get in touch with zero/negative, the gradient turns into zero, resulting in the network unable to perform backpropagation and will not be able to learn
	· Non-linear: though it has a derivative function	
	· Backpropagation allowed	· Unbounded and nondifferentiable at zero
		· Gradients for negative I/P are zero
		· O/P is not zero-centered, hence it does harm the performance of the neural network
		· Activation mean value is nonzero

Types of Activation function	Advantages	Disadvantages
4. Softmax [46]	· Competent to handle various classes, just one class in other activation functions	· It will not work if data are not linearly separable
	· Give the probability of the I/P value being in a particular class	· Does not support null rejection
	· Beneficial for O/P neuron usually	
5. Softplus [45]	· Smooth derivative used in backpropagation	
	· Derivative of the softplus is equal to a sigmoid function [45]	
6. Softsign [45]	· Smoother than the tanh activation function	· More expensive to compute than tanh
	· Grows polynominally rather than exponentially	· Often, gradient produces extremely high/low values
	· Faster and better learning because of the lack of struggling along with the vanishing gradient	
	· Softsign barring neurons from being saturated, which leads to more efficient learning	
7. Hard sigmoidal [47]	· Variant of sigmoidal	
	· Lesser computation cost	

6 Performance measures

The performance of classification of plant diseases is evaluated using an output that can be a binary class or multiclass. The confusion matrix, shown in Figure 4, helps in measuring the performance consisting of actual and predicted values. It also helps in understanding the usefulness of sensitivity as true positive rates and then to analyze the ability to correctly identify the healthy leaf from the diseased leaf, specificity as true negative rates given by equation (20), accuracy as a positive predicted value shown in equation (18), recall known as the probability of detection, defined as the number of truly classified positive outcomes by the sum of total positive outcomes in equation (19). The Mathews correlation

coefficient (MCC) is used to overcome the class imbalance problems. The *F*-measure is the harmonic mean of precision and recall and helps to evaluate recall and precision as in equation (21).

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}, \tag{18}$$

$$\text{Recall/Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})} = \frac{\text{TP}}{\text{P}}, \tag{19}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{N}}, \tag{20}$$

$$F\text{-measure} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Recall} + \text{Precision})}, \tag{21}$$

$$\text{MCC} = \frac{((\text{TN} * \text{TP}) - (\text{FN} * \text{FP}))}{((\text{FP} + \text{TP})(\text{FN} + \text{TP})(\text{FP} + \text{TN})(\text{FN} + \text{TN}))^{0.5}}. \tag{22}$$

True positive (TP) is a positive outcome that is true and prediction comes to be true. True negative (TN) is a negative outcome that is true. False positive (FP) seems to be positive and prediction comes to be false. False negative (FN) is a negative outcome that is false.

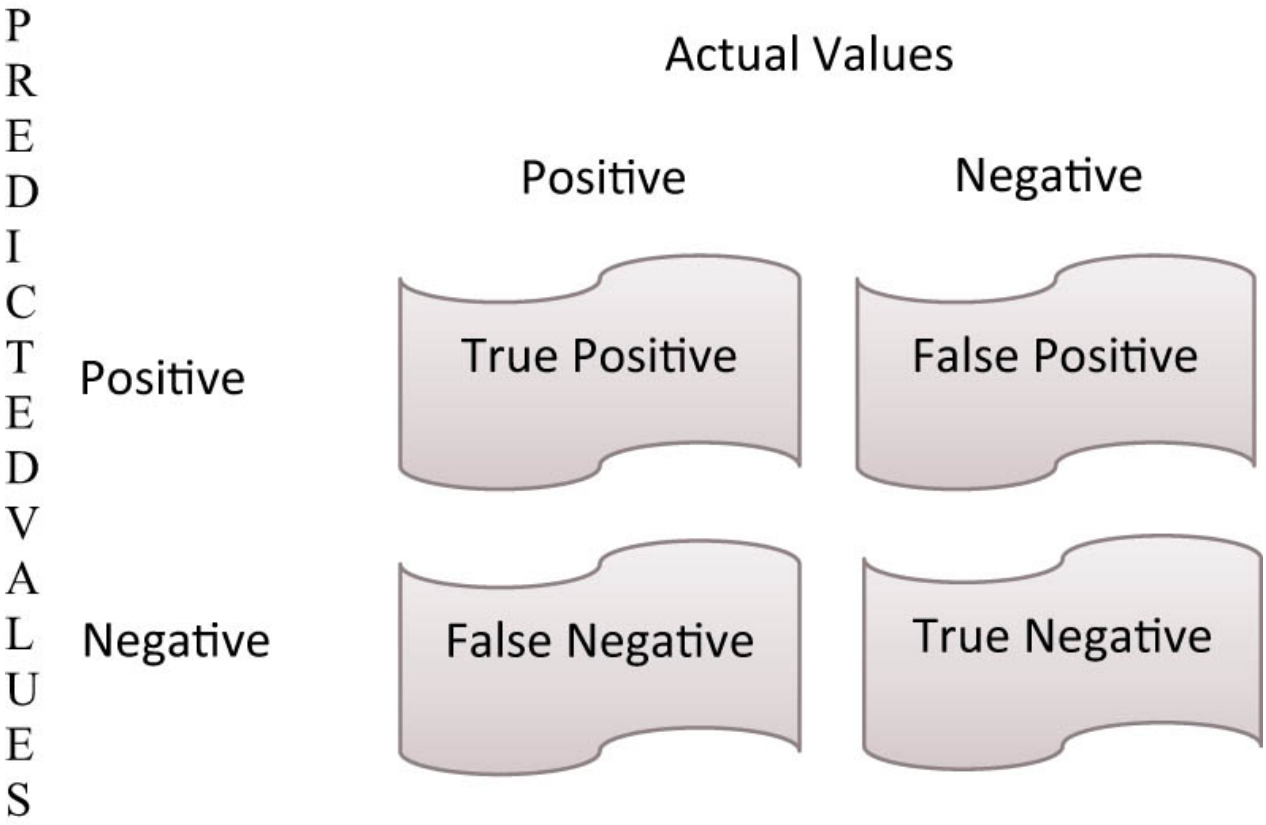


Figure 4

Confusion matrix.

Algorithm of ACC

7 Experimental study

A study centered on the experimental results is designed to determine the influence of the proposed model. Taking the advantages of the four image datasets of plant diseases from Kaggle [37], namely, pepper, tomato, potato, and rice from the UCI repository [38]. The experiments analyze the performance of four datasets-based approaches to six parameters, sensitivity, specificity, accuracy, recall, *F*-measure and MCC, using 15 machine learning algorithms for classification, namely, Bayes Net, Naïve Bayes, LibSVM, SMO, RBF_classifier, RBF_Network, PART, J48, Random Forest, Filtered Classifier, Iterative optimized Classifier, multiclass classifier, simple logistic, decision table and randomized filtered classifier as mentioned in Tables 2, 4, 6, and 8. The performance using DL is also evaluated using different activation functions given in Tables 3, 5, 7, and 9. The experiment was carried out as per the following process:

START

Input: Images of four datasets of plant diseases.

Output: Performance evaluation of 15 machine learning algorithms and DL with different activation functions.

Step 1: Preparing the target image datasets.

Step 2: Creating the image datasets of rice containing 120 images [38], pepper having 1994 images, potato containing 2,152 images and tomato having 16,072 images [37].

Step 3: Image pre-processing.

Step 4: Using unsupervised Image Filter known as Auto Color Correlogram in WEKA 3.9.4 on four datasets.

Step 5: Results obtained from the image filter, used for evaluating the execution of 15 machine learning algorithms in WEKA 3.9.4 [40].

Step 6: Evaluating the performance of DL with nine activation functions.

Step 7: Comparing various machine learning algorithms and DL based on six parameters, sensitivity, specificity, accuracy, recall, *F*-measure and MCC, and finding out which algorithm outshines in each dataset.

Table 2

Effectiveness of machine learning algorithm on the pepper bell dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
BayesNet	0.981	0.017	0.981	0.981	0.981	0.961
Naïve bayes	0.957	0.033	0.96	0.957	0.957	0.914
LibSVM (3)	**0.995**	**0.005**	**0.994**	**0.994**	**0.994**	**0.993**
SMO (2)	**0.996**	**0.005**	**0.996**	**0.996**	**0.996**	**0.992**
RBF_Classifier	0.994	0.007	0.994	0.994	0.994	0.987
RBF_Network	0.97	0.031	0.97	0.97	0.97	0.937
Simple logistic	0.994	0.007	0.994	0.994	0.994	0.987
Decision table	0.977	0.025	0.977	0.977	0.977	0.952
PART	0.983	0.019	0.983	0.983	0.983	0.983
J48	0.975	0.026	0.975	0.975	0.975	0.948
Random forest (1)	**0.997**	**0.004**	**0.997**	**0.997**	**0.997**	**0.993**
Filtered classifier	0.98	0.022	0.98	0.98	0.98	0.959
Iterative optimized classifier	0.988	0.012	0.988	0.988	0.988	0.976
MultiClass classifier	0.989	0.013	0.989	0.989	0.989	0.978
Randomized filtered classifier	0.989	0.013	0.989	0.989	0.989	0.978

The number in parenthesis defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

Table 3

Effectiveness of deep learning with different activation functions on the pepper bell dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
DL_softmax	0.976	0.034	0.977	0.976	0.976	0.951
DL_Relu	0.993	0.008	0.993	0.993	0.993	0.985
DL_tanh (3)	**0.997**	**0.004**	**0.997**	**0.997**	**0.997**	**0.994**
DL_Hard_Sigmoidal	0.978	0.016	0.979	0.978	0.978	0.956
DL_Hard_tanh (2)	**0.998**	**0.003**	**0.998**	**0.998**	**0.998**	**0.995**
DL_rational_tanh	0.98	0.003	0.998	0.998	0.998	0.995
DL_Sigmoidal	0.95	0.042	0.952	0.95	0.95	0.899
DL_Softplus	0.993	0.01	0.993	0.993	0.993	0.985
DL_Softsign (1)	**0.999**	**0.002**	**0.994**	**0.999**	**0.999**	**0.998**

The number in parenthesis defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

A. Mixed Datasets: Four plant disease datasets are selected in the study, out of which pepper is a binary class and the rest, rice, potato and tomato, are multiclass datasets. The data is balanced as per class in both rice and pepper, whereas it is imbalanced in the case of potato and tomato datasets.

B. Fifteen machine learning classification algorithms and DL with nine activation functions. All these classification algorithms are implemented on the four image datasets with the help of WEKA 3.9.4 learning environment using default parameters. In WEKA, the first image data are loaded into WEKA, the class name is converted to nominal. Using imageFilters named ACC, the batch filter is used to compute the color correlogram from the images. ACC converts the spatial color correlation in images and is an efficient feature that is resilient to changes in the viewing position. After that, the classification is operated using 10-fold cross-validation tactics [41].

8 Results and discussion

8.1 Results

Out of the four datasets, two datasets are balanced, namely rice and pepper and the remaining two datasets are unbalanced, namely tomato and potato. Only one data set consists of binary classes, i.e., pepper and the remaining three datasets consist of multiclasses. As per the proposed model, initially, all the datasets are pre-processed by removing the duplicated and distorted images to build the image database consisting of various classes for all four datasets individually. After pre-processing, the image filter called ACC (Auto color correlogram) is applied to compute the color correlogram.

Evaluating the four datasets of binary and multiclass with 15 machine learning algorithms and DL with nine different activation functions based on six parameters are mentioned in Tables 2–9. First, out of the 15 machine learning algorithms, the top three algorithms are highlighted in bold in Tables 2, 4, 6, and 8. Second, rankwise results of DL are highlighted in Tables 3, 5, 7, and 9 with activation functions. Third, the assessment of the results of the top machine learning algorithm with DL is accomplished. The detailed results of the four datasets are described as:

(a) Pepper Dataset

Out of all mentioned machine learning algorithms mentioned in Table 2, the finest accuracy is achieved by Random Forest as 0.997, followed by SMO and LibSVM. The DL with a softsign function shows a good performance with all the six parameters and achieved 0.999 with sensitivity, recall and *F*-measure, highlighted in boldface in Table 3. Specificity is 0.002, MCC is 0.998 and accuracy is 0.994. Ranking wise, DL with softsign followed by DL with hard_tanh and third is DL with tanh.

(b) Rice Dataset

As per the comparison made on the six parameter measures, Randomforest and SMO perform the best when compared with 15 machine learning algorithms as shown in Table 4. The DL with Softmax outshines among all the machine learning algorithms with 0.907 for both sensitivity and recall in Table 5. The accuracy is 0.922, the *F*-measure is 0.904 and MCC is 0.853, highlighted in boldface in Table 5. Rankingwise, second is DL with hardtanh.

(c) Tomato Dataset

LibSVM outshines the other learning algorithms as mentioned in Table 6. LibSVM achieved 0.982 for sensitivity, accuracy, recall and *F*-measure. MCC is 0.976 and sensitivity is 0.007. The performance of DL with softsign is like LibSVM, followed by SMO and DL with tanh, as shown in >Tables 6 and 7.

(d) Potato Dataset

LibSVM outshines the other learning algorithms as mentioned in Table 8. LibSVM achieved 0.993 for sensitivity, accuracy, recall and *F*-measure. MCC is 0.988 and sensitivity is 0.006. Ranking wise, LibSVM, followed by DL with softsign same as DL with hardtanh same as DL with softplus, DL with rational tanh and SMO.

Table 4

Effectiveness of machine learning algorithm on the rice dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
BayesNet	0.733	0.133	0.731	0.733	0.732	0.599
Naïve bayes	0.717	0.142	0.726	0.717	0.716	0.58
LibSVM	0.85	0.075	0.866	0.85	0.846	0.784
SMO (1)	**0.875**	**0.063**	**0.877**	**0.875**	**0.875**	**0.813**
RBF_Classifier (2)	**0.867**	**0.067**	**0.867**	**0.867**	**0.867**	**0.8**
RBF_Network	0.775	0.113	0.789	0.775	0.777	0.669
Simple logistic	0.692	0.154	0.723	0.692	0.674	0.554
Decision table	0.708	0.146	0.716	0.708	0.708	0.566
PART	0.792	0.104	0.791	0.792	0.791	0.687
J48	0.792	0.104	0.793	0.792	0.791	0.688
Random forest (1)	**0.875**	**0.063**	**0.887**	**0.875**	**0.871**	**0.82**
Filtered classifier	0.75	0.125	0.757	0.75	0.75	0.629
Iterative optimized classifier (3)	**0.867**	**0.067**	**0.866**	**0.867**	**0.866**	**0.8**
MultiClass classifier	0.775	0.113	0.776	0.775	0.774	0.663
Randomized filtered classifier	0.742	0.129	0.749	0.742	0.741	0.616

The number in parentheses defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

Table 5

Effectiveness of deep learning with different activation functions on the rice dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
DL_softmax (1)	**0.907**	**0.093**	**0.922**	**0.907**	**0.904**	**0.853**
DL_Relu	0.856	0.123	0.871	0.856	0.84	0.773
DL_tanh (2)	**0.894**	**0.098**	**0.905**	**0.894**	**0.89**	**0.829**
DL_Hard_Sigmoidal	0.706	0.294	0.815	0.706	0.653	0.535
DL_Hard_tanh (3)	**0.894**	**0.094**	**0.903**	**0.894**	**0.888**	**0.831**
DL_rational_tanh	0.9	0.094	0.909	0.9	0.896	0.839
DL_Sigmoidal	0.725	0.275	0.823	0.725	0.672	0.563
DL_Softplus	0.838	0.146	0.853	0.838	0.823	0.727
DL_Softsign	0.869	0.119	0.884	0.869	0.858	0.791

The number in parenthesis defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

Table 6

Effectiveness of the machine learning algorithm on the tomato dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
BayesNet	0.916	0.021	0.917	0.916	0.916	0.895
Naïve bayes	0.845	0.042	0.85	0.845	0.844	0.806
LibSVM (1)	**0.982**	**0.007**	**0.982**	**0.982**	**0.982**	**0.976**
SMO (2)	**0.98**	**0.008**	**0.98**	**0.98**	**0.98**	**0.973**
RBF_Classifier (3)	**0.977**	**0.01**	**0.977**	**0.97**	**0.973**	**0.96**
RBF_Network	0.808	0.076	0.808	0.808	0.808	0.8
Simple Logistic	0.889	0.036	0.894	0.889	0.889	0.85
Decision table	0.751	0.101	0.745	0.751	0.734	0.657
PART	0.925	0.025	0.925	0.925	0.925	0.9
J48	0.9	0.036	0.899	0.9	0.899	0.864
Random forest (3)	0.965	0.03	0.764	0.765	0.764	0.734
Filtered classifier	0.871	0.047	0.87	0.871	0.87	0.827
Iterative optimized classifier	0.925	0.029	0.925	0.925	0.924	0.898
MultiClass classifier	0.928	0.034	0.929	0.928	0.927	0.901
Randomized filtered classifier	0.764	0.085	0.761	0.764	0.762	0.678

The number in parenthesis defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

Table 7

Effectiveness of deep learning on different activation functions on the tomato dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
DL_softmax	0.891	0.047	0.888	0.891	0.871	0.844
DL_Relu	0.96	0.015	0.96	0.96	0.96	0.946
DL_tanh (2)	**0.977**	**0.008**	**0.977**	**0.977**	**0.977**	**0.969**
DL_Hard_Sigmoidal	0.961	0.009	0.961	0.961	0.961	0.95
DL_Hard_tanh	0.972	0.006	0.974	0.972	0.973	0.965
DL_rational_tanh (3)	**0.976**	**0.006**	**0.977**	**0.976**	**0.976**	**0.976**
DL_Sigmoidal	0.967	0.009	0.967	0.967	0.967	0.958
DL_Softplus	0.967	0.009	0.965	0.967	0.966	0.954
DL_Softsign (1)	**0.981**	**0.005**	**0.982**	**0.981**	**0.981**	**0.976**

The number in parenthesis defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

Table 8

Effectiveness of the machine learning algorithm on the potato dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
BayesNet	0.971	0.024	0.971	0.971	0.971	0.949
Naïve bayes	0.955	0.038	0.956	0.955	0.955	0.919
LibSVM (1)	**0.993**	**0.006**	**0.993**	**0.993**	**0.993**	**0.988**
SMO (2)	**0.99**	**0.007**	**0.99**	**0.99**	**0.99**	**0.983**
RBF_Classifier (3)	**0.987**	**0.011**	**0.987**	**0.987**	**0.987**	**0.976**
RBF_Network	0.921	0.069	0.921	0.921	0.921	0.921
Simple logistic	0.941	0.047	0.943	0.941	0.941	0.898
Decision table	0.934	0.051	0.934	0.934	0.934	0.885
PART	0.965	0.028	0.965	0.965	0.965	0.938
J48	0.964	0.028	0.964	0.964	0.964	0.937
Random forest (3)	**0.987**	**0.012**	**0.987**	**0.987**	**0.986**	**0.977**
Filtered classifier	0.95	0.038	0.95	0.95	0.95	0.913
Iterative optimized classifier	0.974	0.02	0.974	0.974	0.974	0.954
MultiClass classifier	0.983	0.009	0.983	0.983	0.983	0.973
Randomized filtered classifier	0.868	0.099	0.868	0.868	0.868	0.769

The number in parenthesis defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

Table 9

Effectiveness of deep learning on different activation functions on the potato dataset

Machine learning methods	Sensitivity	Specificity	Accuracy	Recall	*F*-measure	MCC
DL_softmax	0.942	0.05	0.944	0.942	0.944	0.906
DL_Relu	0.983	0.012	0.983	0.984	0.984	0.973
DL_tanh (3)	**0.989**	**0.009**	**0.989**	**0.989**	**0.989**	**0.981**
DL_Hard_Sigmoidal	0.966	0.024	0.968	0.968	0.942	0.942
DL_Hard_tanh (1)	**0.992**	**0.007**	**0.992**	**0.992**	**0.992**	**0.986**
DL_rational_tanh (2)	**0.99**	**0.008**	**0.991**	**0.99**	**0.99**	**0.984**
DL_Sigmoidal	0.986	0.008	0.987	0.986	0.986	0.977
DL_Softplus (1)	**0.992**	**0.007**	**0.992**	**0.992**	**0.992**	**0.986**
DL_Softsign (1)	**0.992**	**0.007**	**0.992**	**0.992**	**0.992**	**0.986**

The number in parentheses defines the ranking with the machine learning algorithm.

Bold is used to high-light the ranking of top 3 classifiers among machine learning algorithms.

8.2 Discussion

Images of healthy and unhealthy (consisting of some diseases) plants, as computer processes the images, neurons present in the deep neural network try to adjust the weight and bias to recognize fewer images wrongly depending on the input, i.e., number of images with each dataset. This needs a small change of weights (or/and bias) producing a small change in the output. The DL cannot show this small change as a behavior. It requires a perceptron, i.e., 0 or 1 to produce a big change. So, there is a need for a function that progressively changes with continuity from 0 to 1 [42]. Utilizing the different activation functions at the output layer of the DL to uncover the change in performances of six parameters.

In the case of a balanced number of classes, the DL with softsign with 0.999 accuracy outshines in pepper datasets (binary class) and DL with softmax performs best with the rice dataset (multiclass) securing accuracy of 0.922. Whereas in unbalanced classes, DL with softsign performs best in the tomato dataset achieving an accuracy of 0.982 and DL with hard tanh, softplus and softsign shows the same performance obtaining an accuracy of 0.992 in the potato dataset. With DL, on observing the difference in performances of various activation functions with different parameter measures, the significant difference can be viewed from the graphs shown in Figures 5–10 on the four image datasets.

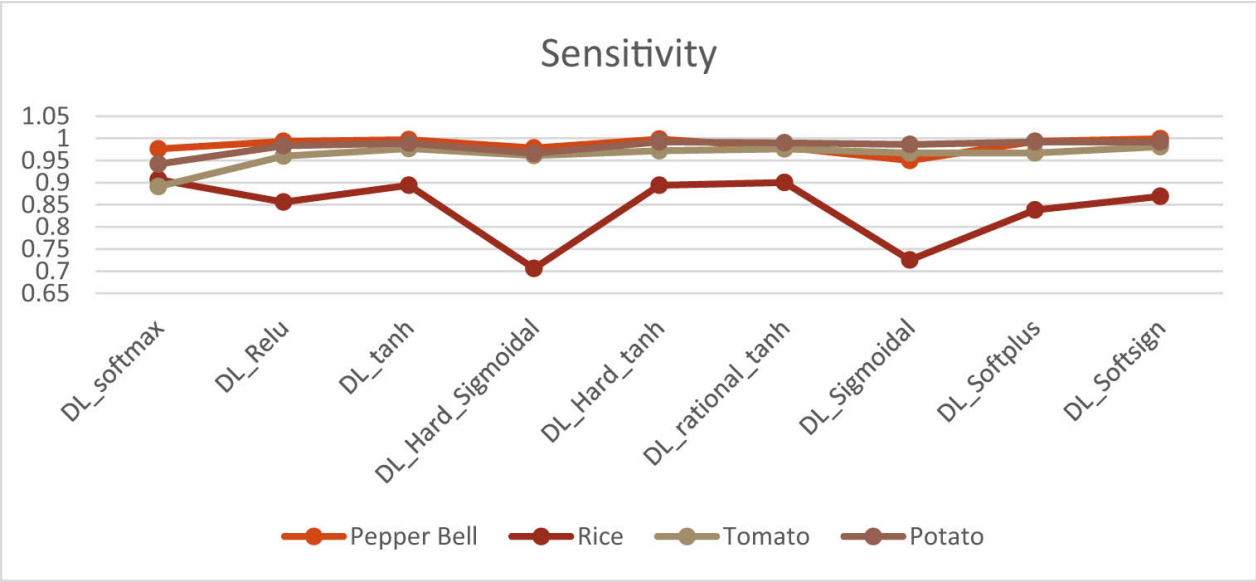


Figure 5

Performance of sensitivity using deep learning with different activation functions on the four datasets.

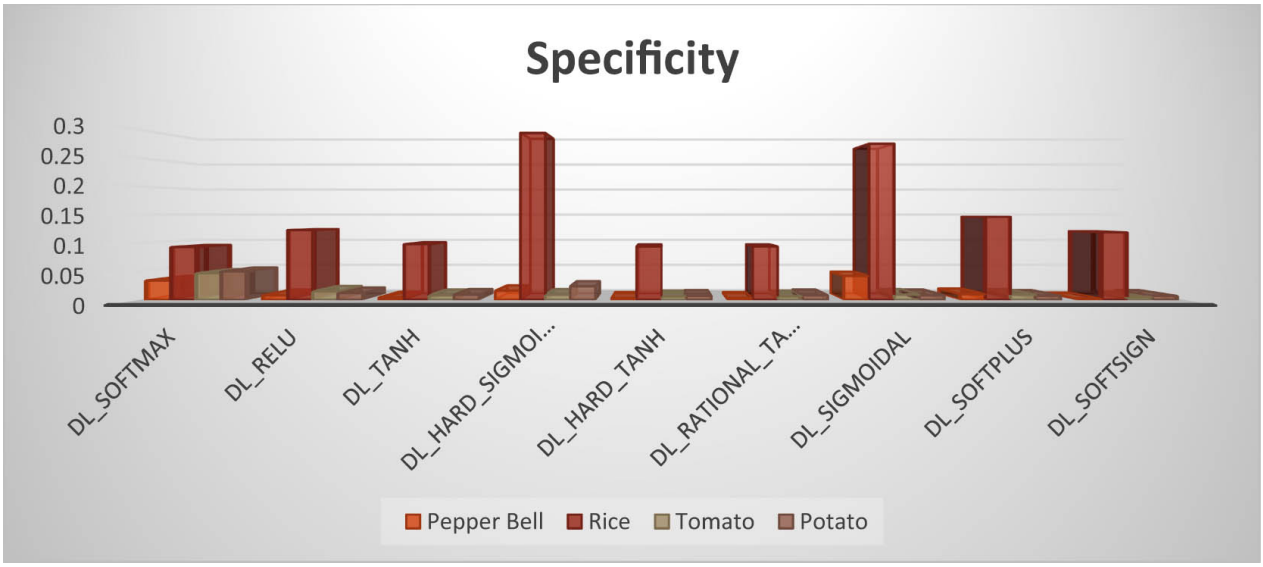


Figure 6

Performance of specificity using deep learning with different activation functions on the four datasets.

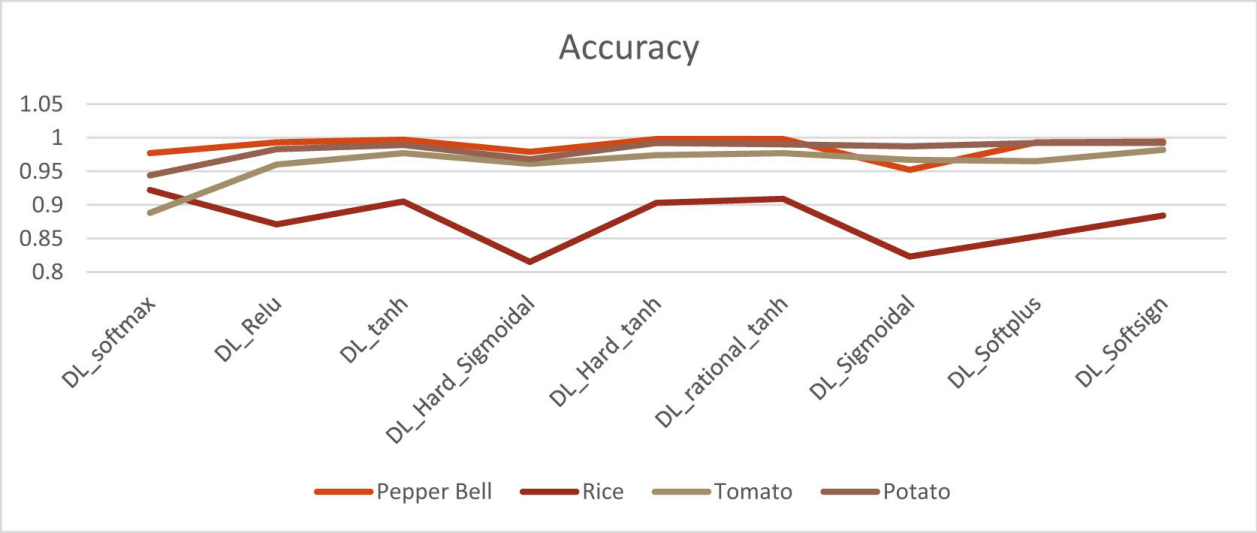


Figure 7

Performance of accuracy using deep learning with different activation functions on the four datasets.

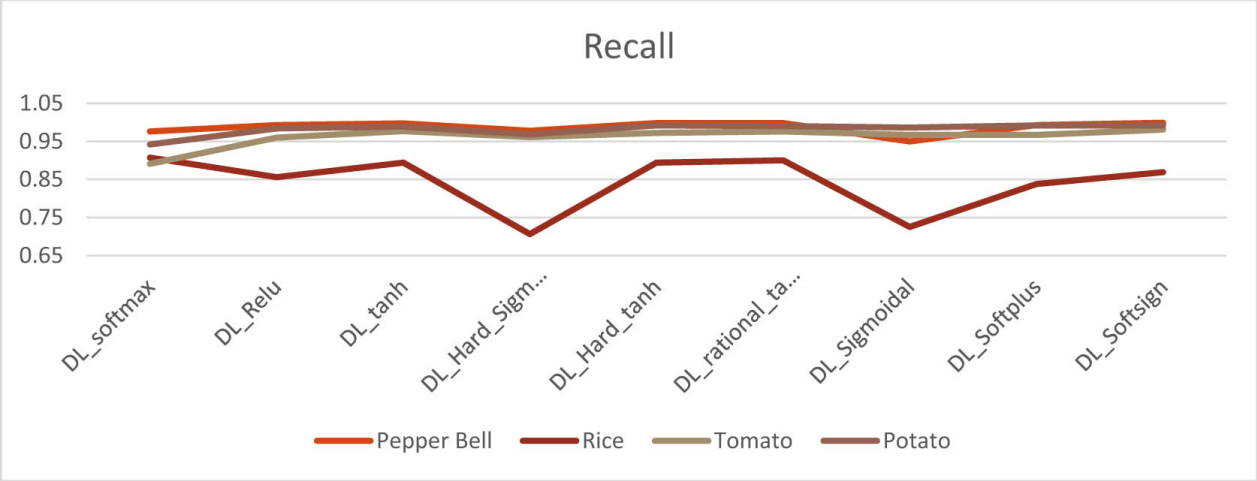


Figure 8

Performance of recall using deep learning with different activation functions on the four datasets.

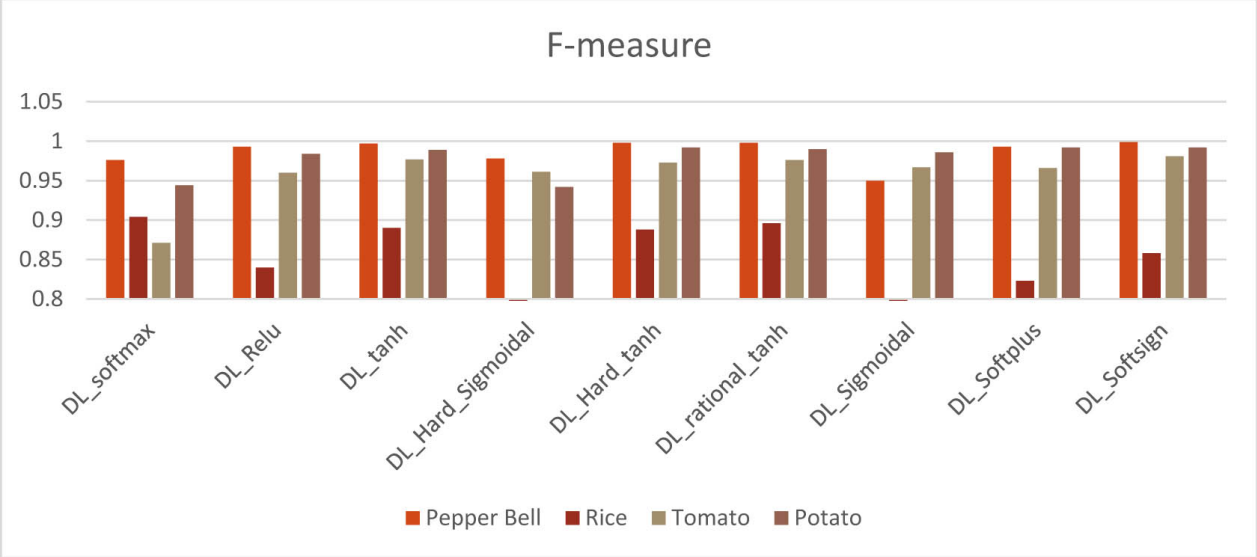


Figure 9

Performance of *F*-measure using deep learning with different activation functions on four datasets.

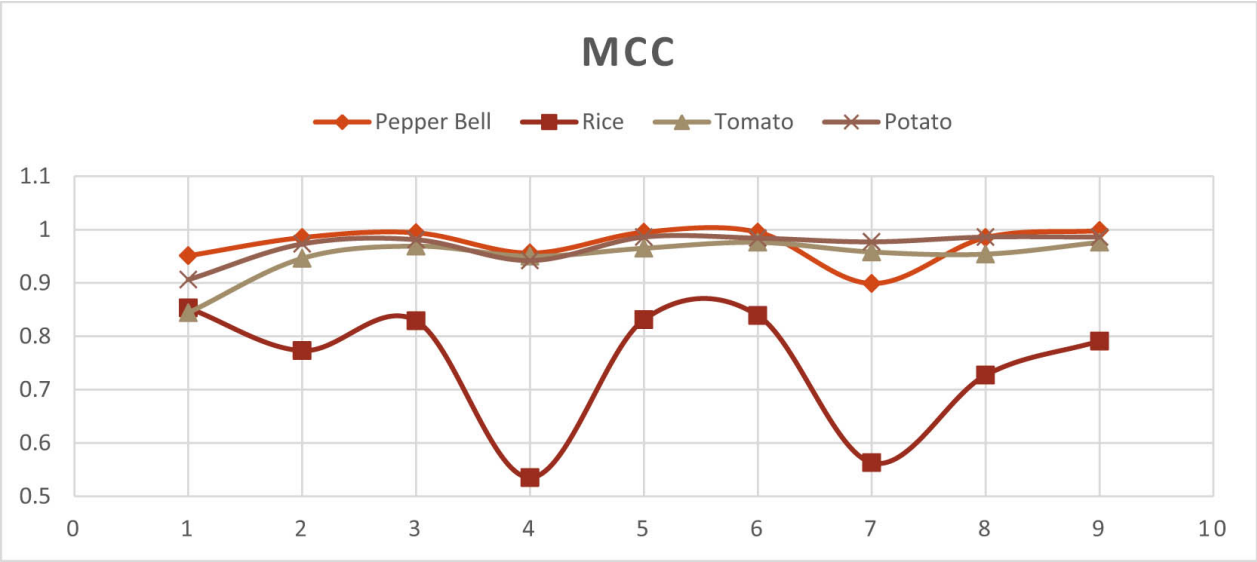


Figure 10

Performance of MCC using deep learning with different activation functions on the four datasets.

This proposed model helps to differentiate between healthy and diseased leaves. The work has been done on four datasets consisting of different numbers of images with different numbers of classes. The experimental result works better for binary classification in pepper bell rather than multiclass in rice, tomato, and potato.

DL is advantageous over 15 machine learning algorithms due to the following reasons:

(a) DL comes up with the expertise to recognize themselves and produce output that is not constrained to an input supplied.

(b) Concerning the case of large data size, the implementation of DL is quite good and makes up the highest possible consumption of unstructured information.

(c) DL has high-end infrastructure for instructing large data in a reasonable time.

(d) The performance of DL is stronger even in issues of complex dilemmas.

The effectiveness of DL is extremely focused on the activation function; it performs as a statistical "gate" sandwiched in the middle of an input feeding the present neuron as well as its output moving on to the next layer. It could be as straightforward as a step function that transforms the neuron output off and on, varying upon a threshold or law. It plays a major role due to the amalgamation of the arbitrary linear model. When it comes to the design solution of complex problems, the use of different activation functions can enhance the performance.

8.3 Experimental comparison of machine learning and DL

On comparing the machine learning methods with DL, the DL key benefits over other machine learning algorithms are its ability to accomplish features engineering on its own. DL will scan the information and data to seek features that associate and merge them to facilitate speedier learning without being explicitly stated to do so. AF establishes the

output of a DL, its computational efficiency and the accuracy of training a model. Utilizing the nonlinear AFs that can facilitate the model to discover complex data, handle and learn providing correct predictions. Using different AFs can result in different results. Out of four datasets, DL with softsign has gained 0.99 sensitivity, *F*-measure and recall in pepper bell. The accuracy achieved is 0.922 in rice datasets with DL softmax. There is an increase of 4.5% in accuracy achieved by DL than the ML algorithm named SMO. But, the performance of LibSVM and SMO is outstanding in the case of potato and tomato datasets with respect to all the parameters. The details are shown by graphs in Figures 11–14.

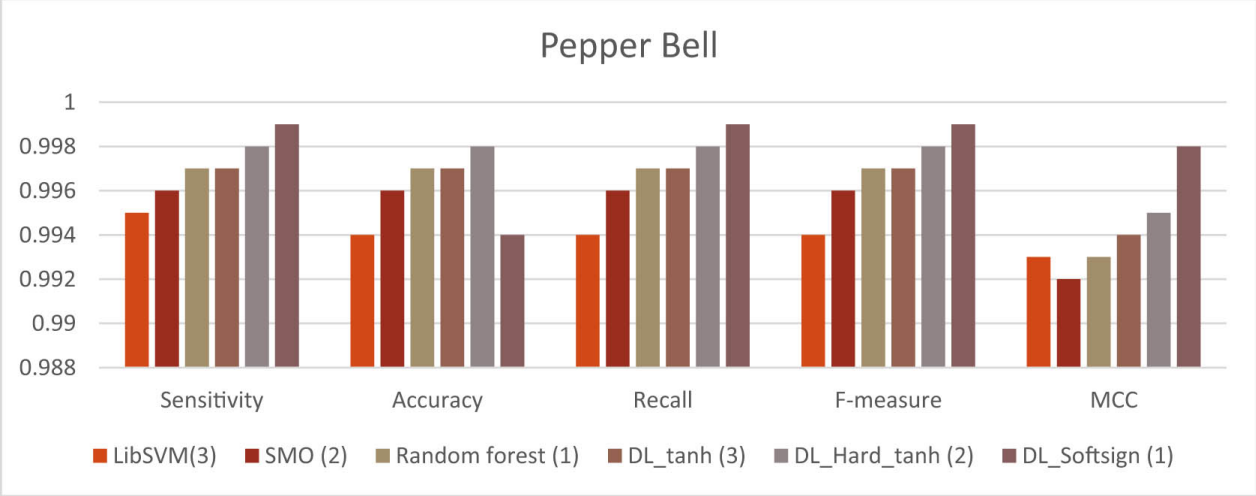


Figure 11

Comparison of machine and deep learning on pepper bell.

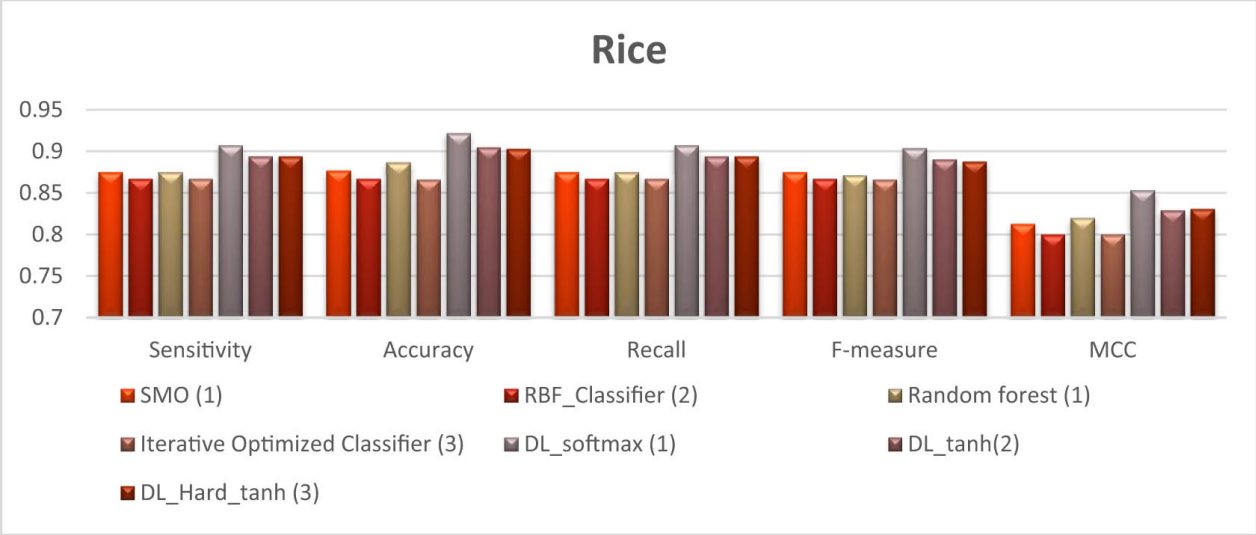


Figure 12

Comparison of machine and deep learning on rice.

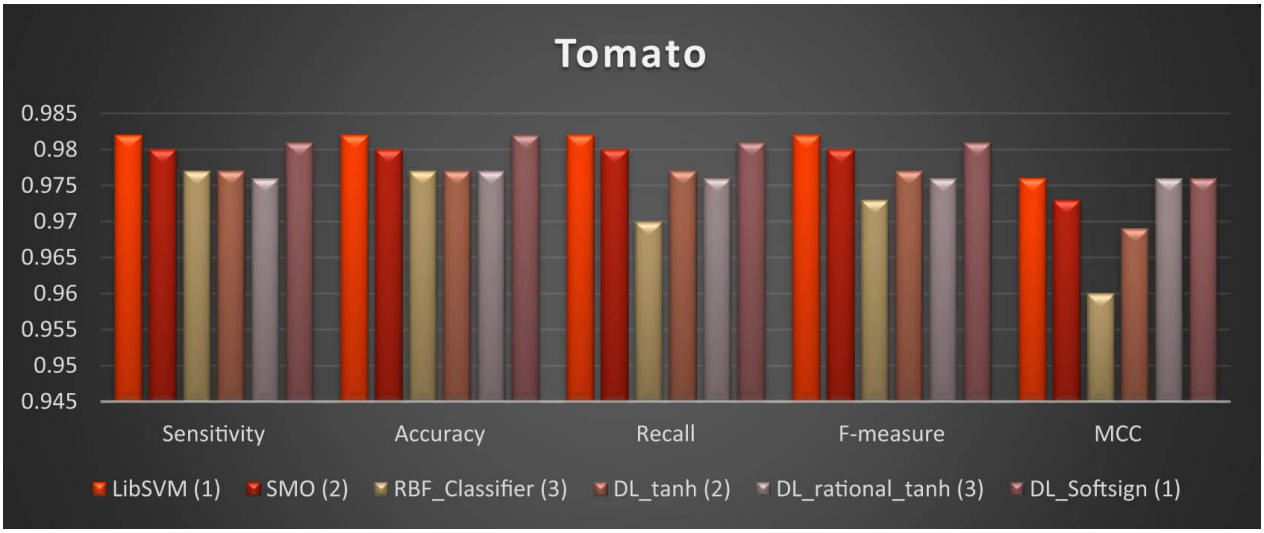


Figure 13

Comparison of machine and deep learning on tomato.

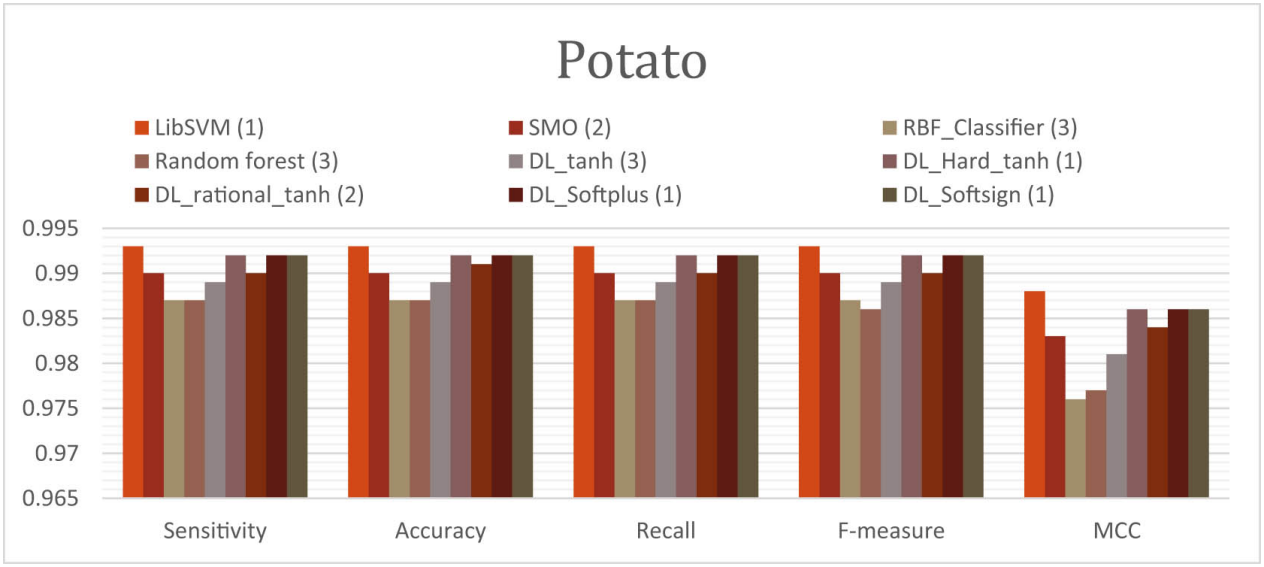


Figure 14

Comparison of machine and deep learning on potato.

9 Conclusions

This article mainly focuses on extracting data from images using auto-color correlogram filter and classifying the diseases in plants. The model highlights the importance of image pre-processing and extracts information that helps in the classification task, which can then support disease detection in plants. DL helps in easy learning of difficult complex patterns by making use of the unseen layers between inputs and outputs. It is required to represent mediator representations of the data that other machine learning methods cannot easily do. More data are needed to train in comparison to other learning methods by DL as it has more parameters for estimation. This proposed model will boost the performance and recognize the complex patterns in various plant datasets.

Experimental results show that the proposed model has achieved an accuracy of 99.4% with binary class and 99.2% with multiclass.

Furthermore, future work will involve increasing the performance of DL in the detection of plant diseases having a multiclass subcategory. As the severity of the disease is supposed to be changing with time, so the DL should be improved to enhance detection and classification of the diseases throughout the whole development cycle of plant leaves.
Conflict of interest: Authors state no conflict of interest.

References

[1] V. A. Gulhane and A. A. Gurjar, "Detection of diseases on cotton leaves and its possible diagnosis," Int. J. Image Process. (IJIP), vol. 5, no. 5. pp. 590–598, 2011.

[2] A. Akhtar, A. Khanum, S. A. Khan, and A. Shaukat, "Automated plant disease analysis (APDA): Performance comparison of machine learning techniques," in 2013 11th International Conference on Frontiers of Information Technology, IEEE, 2013, December, pp. 60–65.
10.1109/FIT.2013.19 (https://doi.org/10.1109/FIT.2013.19)

[3] A. A. Fathima, R. Karthik, and V. Vaidehi, "Image stitching with combined moment invariants and sift features," in ANT/SEIT, Elsevier, 2013, December, pp. 420–427.
10.1016/j.procs.2013.06.057 (https://doi.org/10.1016/j.procs.2013.06.057)

[4] R. Karthik, A. AnnisFathima, and V. Vaidehi, "Panoramic view creation using invariant momentsand SURF features," in 2013 International Conference on Recent Trends in Information Technology (ICRTIT), IEEE, 2013, July, pp. 376–382.
10.1109/ICRTIT.2013.6844233 (https://doi.org/10.1109/ICRTIT.2013.6844233)

[5] S. Kaur, S. Pandey, and S. Goel, "Semiautomatic leaf disease detection and classification system for soybean culture," IET Image Process., vol. 12, no. 6. pp. 1038–1048, 2018.

[6] P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," in 2016 Conference on Advances in Signal Processing (CASP), IEEE, 2016, June, pp. 175–179.
10.1109/CASP.2016.7746160 (https://doi.org/10.1109/CASP.2016.7746160)

[7] T. Mehra, V. Kumar, and P. Gupta, "Maturity and disease detection in tomato using computer vision," in 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, 2016, December, pp. 399–403.
10.1109/PDGC.2016.7913228 (https://doi.org/10.1109/PDGC.2016.7913228)

[8] R. Menaka and R. Karthik, "A novel feature extraction scheme for visualisation of 3D anatomical structures," Int. J. Biomed. Eng. Technol., vol. 21, no. 1. pp. 49–66, 2016.
10.1504/IJBET.2016.076732 (https://doi.org/10.1504/IJBET.2016.076732)

[9] Y. Dandawate and R. Kokare, "An automated approach for classification of plant diseases towards development of futuristic decision support system in Indian perspective," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2015, August, pp. 794–799.
10.1109/ICACCI.2015.7275707 (https://doi.org/10.1109/ICACCI.2015.7275707)

[10] C. S. Hlaing and S. M. M. Zaw, "Tomato plant diseases classification using statistical texture feature and color feature," 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), IEEE, 2018, June, pp. 439–444. 10.1109/ICIS.2018.8466483 (https://doi.org/10.1109/ICIS.2018.8466483)

[11] M. Zhang, L. I. Ludas, M. T. Morgan, G. W. Krutz, and C. J. Precetti, "Applications of color machine vision in the agricultural and food industries," in Precision Agriculture and Biological Quality, Vol. 3543, International Society for Optics and Photonics, Bellingham WA, SPIE, 1999, January, pp. 208–219.

[12] S. Bashir and N. Sharma, "Remote area plant disease detection using image processing," IOSR J. Electron. Commun. Eng., vol. 2, no. 6. pp. 31–34, 2012. 10.9790/2834-0263134 (https://doi.org/10.9790/2834-0263134)

[13] L. Wojnar, Image Analysis: Applications in Materials Engineering, CRC Press, 1998.

[14] S. S. Sannakki and V. S. Rajpurohit, "Classification of pomegranate diseases based on back propagation neural network," Int. Res. J. Eng. Technol. (IRJET), vol. 2, no. 2, 2015.

[15] P. R. Rothe and R. V. Kshirsagar, "Cotton leaf disease identification using pattern recognition techniques," in 2015 International Conference on Pervasive Computing (ICPC), IEEE, 2015, January, pp. 1–6. 10.1109/PERVASIVE.2015.7086983 (https://doi.org/10.1109/PERVASIVE.2015.7086983)

[16] A. Rastogi, R. Arora, and S. Sharma, "Leaf disease detection and grading using computer vision technology fuzzy logic," in 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, 2015, February, pp. 500–505. 10.1109/SPIN.2015.7095350 (https://doi.org/10.1109/SPIN.2015.7095350)

[17] C. Zhao, S. Lu, and X. Guo, "SVM-based multiple classifier system for recognition of wheat leaf diseases," in Proceedings of 2010 Conference on Dependable Computing (CDC '2010), 2010, November.

[18] G. Owomugisha, J. A. Quinn, E. Mwebaze, and J. Lwasa, "Automated vision-based diagnosis of banana bacterial wilt disease and black sigatoka disease," in International Conference on the Use of Mobile ICT in Africa, 2014, pp. 1–5.

[19] D. Hall, C. McCool, F. Dayoub, N. Sunderhauf, and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions," in 2015 IEEE Winter Conference on Applications of Computer Vision, IEEE, 2015, January, pp. 797–804. 10.1109/WACV.2015.111 (https://doi.org/10.1109/WACV.2015.111)

[20] Y. Itzhaky, G. Farjon, F. Khoroshevsky, A. Shpigler, and A. Bar-Hillel, "Leaf counting: Multiple scale regression and detection using deep CNNs," BMVC, 2018, September, p. 328.

[21] J. Ubbens, M. Cieslak, P. Prusinkiewicz, and I. Stavness, "The use of plant models in deep learning: an application to leaf counting in rosette plants," Plant. Methods, vol. 14, no. 1. p. 6, 2018. 10.1186/s13007-018-0273-z (https://doi.org/10.1186/s13007-018-0273-z)

PubMed (https://pubmed.ncbi.nlm.nih.gov/29375647/)
PubMed Central (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5773030/)

[22] M. Rahnemoonfar and C. Sheppard, "Deep count: fruit counting based on deep simulated learning," Sensors, vol. 17, no. 4. p. 905, 2017.
10.3390/s17040905 (https://doi.org/10.3390/s17040905)

PubMed (https://pubmed.ncbi.nlm.nih.gov/28425947/)
PubMed Central (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5426829/)

[23] G. L. Grinblat, L. C. Uzal, M. G. Larese, and P. M. Granitto, "Deep learning for plant identification using vein morphological patterns," Comput. Electron. Agric., vol. 127, pp. 418–424, 2016.
10.1016/j.compag.2016.07.003 (https://doi.org/10.1016/j.compag.2016.07.003)

[24] S. H. Lee, C. S. Chan, P. Wilkin, and P. Remagnino, "Deep-plant: Plant identification with convolutional neural networks," in 2015 IEEE international conference on image processing (ICIP), IEEE, 2015, September, pp. 452–456.
10.1109/ICIP.2015.7350839 (https://doi.org/10.1109/ICIP.2015.7350839)

[25] M. P. Pound, J. A. Atkinson, A. J. Townsend, M. H. Wilson, M. Griffiths, A. S. Jackson, et al., "Deep machine learning provides state-of-the-art performance in image-based plant phenotyping," Gigascience, vol. 6, no. 10. p. gix083, 2017.
10.1093/gigascience/gix083 (https://doi.org/10.1093/gigascience/gix083)

PubMed (https://pubmed.ncbi.nlm.nih.gov/29020747/)
PubMed Central (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5632296/)

[26] A. Milioto, P. Lottes, and C. Stachniss, "Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks," in ISPRS Ann. Photogram, Remote. Sens. Spat. Inf. Sci., vol. 4, p. 41, 2017.
10.5194/isprs-annals-IV-2-W3-41-2017 (https://doi.org/10.5194/isprs-annals-IV-2-W3-41-2017)

[27] C. Potena, D. Nardi, and A. Pretto, "Fast and accurate crop and weed identification with summarized train sets for precision agriculture," International Conference on Intelligent Autonomous Systems, Cham, Springer, 2016, July, pp. 105–121.
10.1007/978-3-319-48036-7_9 (https://doi.org/10.1007/978-3-319-48036-7_9)

[28] Y. Sun, Y. Liu, G. Wang, and H. Zhang, "Deep learning for plant identification in natural environment," Comput. Intell. Neurosci., vol. 2017, p. 6, 2017.
10.1155/2017/7361042 (https://doi.org/10.1155/2017/7361042)

PubMed (https://pubmed.ncbi.nlm.nih.gov/28611840/)
PubMed Central (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5458433/)

[29] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," IEEE Geosci. Remote. Sens. Lett., vol. 14, no. 5. pp. 778–782, 2017.

10.1109/LGRS.2017.2681128 (https://doi.org/10.1109/LGRS.2017.2681128)

[30] A. K. Mortensen, M. Dyrmann, H. Karstoft, R. N. Jørgensen and R. Gislum, "Semantic segmentation of mixed crops using deep convolutional neural network," in Proceedings of the International Conf. of Agricultural Engineering (CIGR), 2016, June.

[31] J. Rebetez, H. F. Satizábal, M. Mota, D. Noll, L. Büchi, M. Wendling, et al., "Augmenting a convolutional neural network with local histograms-A case study in crop classification from high-resolution UAV imagery," in ESANN, Belgique Ciaco, Louvain-la-Neuve, 2016, April.

[32] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," Plants, vol. 8, no. 11. p. 468, 2019.
10.3390/plants8110468 (https://doi.org/10.3390/plants8110468)

PubMed (https://pubmed.ncbi.nlm.nih.gov/31683734/)
PubMed Central (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6918394/)

[33] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Comput. Electron. Agric., vol. 147, pp. 70–90, 2018.
10.1016/j.compag.2018.02.016 (https://doi.org/10.1016/j.compag.2018.02.016)

[34] A. K. Singh, B. Ganapathysubramanian, S. Sarkar, and A. Singh, "Deep learning for plant stress phenotyping: trends and future perspectives," Trends plant. Sci., vol. 23, no. 10. pp. 883–898, 2018.
10.1016/j.tplants.2018.07.004 (https://doi.org/10.1016/j.tplants.2018.07.004)

PubMed (https://pubmed.ncbi.nlm.nih.gov/30104148/)

[35] UCI Machine Learning Repository: Centre for Machine Learning and Intelligent Systems. Available Link: https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases

[36] Kaggle, PlantVillage Datasets. Available Link: https://www.kaggle.com/emmarex/plantdisease.

[37] A. Tungkasthan, S. Intarasema, and W. Premchaiswadi, "Spatial color indexing using ACC algorithm," in 2009 7th International Conference on ICT and Knowledge Engineering, IEEE, 2009, December, pp. 113–117.
10.1109/ICTKE.2009.5397321 (https://doi.org/10.1109/ICTKE.2009.5397321)

[38] G. A. Montazer and D. Giveki, "Content based image retrieval system using clustered scale invariant feature transforms," Optik, vol. 126, no. 18. pp. 1695–1699, 2015.
10.1016/j.ijleo.2015.05.002 (https://doi.org/10.1016/j.ijleo.2015.05.002)

[39] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," Math. Probl. Eng., vol. 2013, p. 11, 2013.
10.1155/2013/425740 (https://doi.org/10.1155/2013/425740)

[40] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," ACM SIGKDD Expl. Newsl., vol. 11, no. 1. pp. 10–18, 2009.
10.1145/1656274.1656278 (https://doi.org/10.1145/1656274.1656278)

[41] T. Fushiki, "Estimation of prediction error by using K-fold cross-validation," Stat. Comput., vol. 21, no. 2. pp. 137–146, 2011.
10.1007/s11222-009-9153-8 (https://doi.org/10.1007/s11222-009-9153-8)

[42] Towards Data Science Available Link: https://towardsdatascience.com/complete-guideof-activation-functions-34076e95d044

[43] M. Courbariaux, Y. Bengio, and J. P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," arXiv Prepr. arXiv:151100363, 2015.

[44] I. Kalinovskii and V. Spitsyn, "Compact convolutional neural network cascade for face detection," arXiv Prepr. arXiv:150801292, 2015.

[45] T. Szandała, "Review and comparison of commonly used activation functions for deep neural networks," Bio-inspired Neurocomputing, Singapore, Springer, 2021, pp. 203–224.
10.1007/978-981-15-5495-7_11 (https://doi.org/10.1007/978-981-15-5495-7_11)

[46] M. A. Mercioni and S. Holban, "The most used activation functions: classic versus current," in 2020 International Conference on Development and Application Systems (DAS), IEEE, 2020, May, pp. 141–145.
10.1109/DAS49615.2020.9108942 (https://doi.org/10.1109/DAS49615.2020.9108942)

[47] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," arXiv Prepr. arXiv:1811033, 2018.

Received: 2020-03-31
Revised: 2021-06-10
Accepted: 2021-06-28
Published Online: 2021-12-08

© 2021 Monika Lamba *et al.*, published by De Gruyter

This work is licensed under the Creative Commons Attribution 4.0 International License.

From the journal

Open Computer Science
Volume 11 Issue 1

Articles in the same Issue

Towards a Formal Specification of Production Processes Suitable for Automatic Execution

Diacritics restoration based on word n-grams for Slovak texts

A Strategic Approach for implementing A Smart Pedestrian Network (SPN) System

Downloaded on 11.8.2023 from
https://www.degruyter.com/document/doi/10.1515/comp-2020-0122/html