Received: April 5, 2023. Revised: May 20, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Red Deer Optimization with Deep Learning Enabled Agricultural Plant Disease Detection and Classification Model

D. Raja[1]* **M. Karthikeyan[1]**

[1]*Department of Computer and Information Science, Faculty of Science, Annamalai University, Tamil Nadu, India*
* Corresponding author's Email: agdrajaphd@gmail.com

Abstract: Plant diseases were responsible for crop losses which directly affect global and national food production methods, ensuing in economic losses. Effectual plant disease analysis includes early-season plant disease detection, detection of several diseases from various crops and various simultaneous diseases, evaluation of the severity of diseases, an estimate of the suitable volume of pesticides for execution, and valuable stages for taking to manage diseases for limiting their spread. Plant diseases and pests were significant features defining crop and plant quality. Plant diseases and pest recognition are applied for digital image processing. Recently, deep learning (DL) is developing enhancement in the field of digital image processing, far higher than standard approaches. This article introduces a red deer optimization with deep learning enabled agricultural plant disease detection and classification (RDODL-APDC) technique. The presented RDODL-APDC technique exploits the DL technique for recognizing and categorizing plant diseases. Initially, the RDODL-APDC technique segments the plant leaf regions using the NestNet background removal process. In addition, multi-level thresholding segmentation segments the infected portions of the images of the plant leaf. For feature extraction, the RDO with the MobileNet-v3 model is exploited. Finally, the extreme gradient boosting (XGBoost) and neural network (NN) models are used for the detection process with particle swarm optimization (PSO) as a parameter tuning technique. For exhibiting the betterment of the RDODL-APDC technique, an extensive range of simulations are executed. The experimental values described the improvement of the RDODL-APDC technique over other current algorithms.

Keywords: Agriculture, Machine learning, Plant disease detection, Computer vision, Image classification.

1. Introduction

Pest detection and Plant diseases will be considered crucial research content in the machine vision (MV) domain [1]. MV can be referred to as technology that employs MV tools for obtaining images to determine if there are pests and diseases from the plant images [2]. Currently, the MV-oriented pest detection and plant diseases tool was implemented in the agricultural field and has substituted conventional naked eye detection to some extent [3]. Timely and adequate detection of diseases comprising early impediments has been very essential in this changing atmosphere. Plant pathology is identified in several ways. For conventional machines, vision-related pest detection and plant diseases approach, manual design of features, or traditional image processing methods along with that classifiers were employed often [4]. This technique generally uses distinct properties of pests and plant diseases for devising the imaging method and selects suitable shooting angles and light sources that can be useful to acquire images with uniform illumination [5].

Though carefully built imaging approaches minimize the complexity of traditional technique design however rises the implementation cost [6]. Simultaneously, in a natural atmosphere, it is often impractical to assume the traditional techniques devised to remove the effect of scene changes on the detection outcomes [7]. Enhancements in computer vision (CV) present an opportunity to strengthen and increase the practice of precise plant safety and to widen the market for accurate agriculture CV

applications [8]. To classify and detect plant diseases, the use of common technologies in digital image processing like threshold and colour detection is used [9]. Various methods of deep learning (DL) were now employed for detecting plant diseases. DL is a new trend in ML, with existing outcomes in many research areas, which include CV, bioinformatics, and pharmacy [10]. DL benefits from the capacity to employ raw data directly without using handcrafts. Recently, metaheuristic algorithms can be applied for the hyperparameter tuning of the DL models. Some of the recent metaheuristics are artificial bee colony (ACO) algorithm, whale optimization algorithm (WOA), guided Pelican algorithm [6], Stochastic Komodo algorithm [7], extended Stochastic coati optimizer [8], fixed step average and Subtraction based optimizer [9], Puzzle optimization algorithm [10], etc.

This article introduces a red deer optimization with deep learning enabled agricultural plant disease detection and classification (RDODL-APDC) model. The presented RDODL-APDC technique segments the plant leaf regions using the NestNet background removal process. In addition, multi-level thresholding segmentation segments the diseased portions of the plant leaf imaging. For feature extraction, the RDO with the MobileNet-v3 model is exploited. Finally, the extreme gradient boosting (XGBoost) and neural network (NN) models are used for the detection process with particle swarm optimization (PSO) as a parameter tuning technique. For exhibiting the betterment of the RDODL-APDC technique, an extensive range of simulations are executed.

2. Related works

Annrose et al. [11] presented a hybrid DL approach with Archimedes optimization algorithm (HDL-AOA) for bean ailment classifier. The HDL approach is the integration of LSTM and wavelet packet decomposition (WPD). Firstly, the WPD decays the inputted image into 4 sub-sequences. In the bean disease classifier, an AOA improves the classifier accuracy to several single LSTM networks. Guo et al. [12] examine a scientific procedure of plant ailment identification and classification established on DL that precipitates generality, precision, and training efficacy. Primarily, the RPN was employed for localization and recognization of the leaves from difficult atmospheres.

Nandhini and Ashokkumar [13] introduce an effective mutation-based Henry gas solubility optimization (MHGSO) technique for optimizing the hyperparameter of the DenseNet121 structure. This stage supports the MHGSO optimizing DenseNet121 infrastructure for achieving a superior classifier precision to classify distinct plant images in the PlantVillage database. Aqel et al. [14] present a new system for the automated recognition and categorizing of ailments of the plant leaf dependent upon utilizing the ELM in the DL technique. The BDA-optimized system was utilized for FS, and finally, the ELM technique was utilized for the plant leaf diseases classifier. Gadekallu et al. [15] concentrated on implementing an ML approach to classifier tomato disease image database for effectively taking crucial stages for combating an agricultural crisis. Essential features were extracted in the database utilizing the hybrid-PCA–WOA.

Cristin et al. [16] present an efficient image-processing approach for plant ailment detection. During this case, the inputted image was exposed to the pre-processed step to remove the noise and artefacts existing in the images. Afterwards attaining the preprocessing image, it can be exposed to the segmentation stage to attain the segment utilizing piecewise FCM (piFCM). Pavithra and Kalpana [17] present an intellectual plant disease detection utilizing deep transference learning related EfficientNet with KELM, termed as DTLEN-KELM approach. In addition, the DTLEN-KELM approach comprises an EfficientNetB0-related extracting feature for deriving an optimum feature vector that is next classification utilizing the KELM technique.

3. The proposed model

In this research, an automated plant disease detection approach, called RDODL-APDC algorithm has been introduced, which exploited the DL model for recognizing and categorizing plant ailments. The presented RDODL-APDC technique encompasses NestNet background removal, Kapur's thresholding-based segmentation, MobileNet-v3 feature extracting process, RDO hyperparameter optimization, classification (XGBoost and NN), and PSO-based parameter tuning. Fig. 1 defines the comprehensive process of the RDODL-APDC technique.

3.1 NestNet background removal

Principally, the RDODL-APDC technique segmented the plant leaf regions using the NestNet background removal process. The method is collected of 2 parts: an initial is the encoder element [18]. In 2 distinct periods, images were utilized as the input to the encoder element for performing extracting features. This element comprises 4 hierarchical ranks, and feature tensors at similar ranks

Received: April 5, 2023. Revised: May 20, 2023.

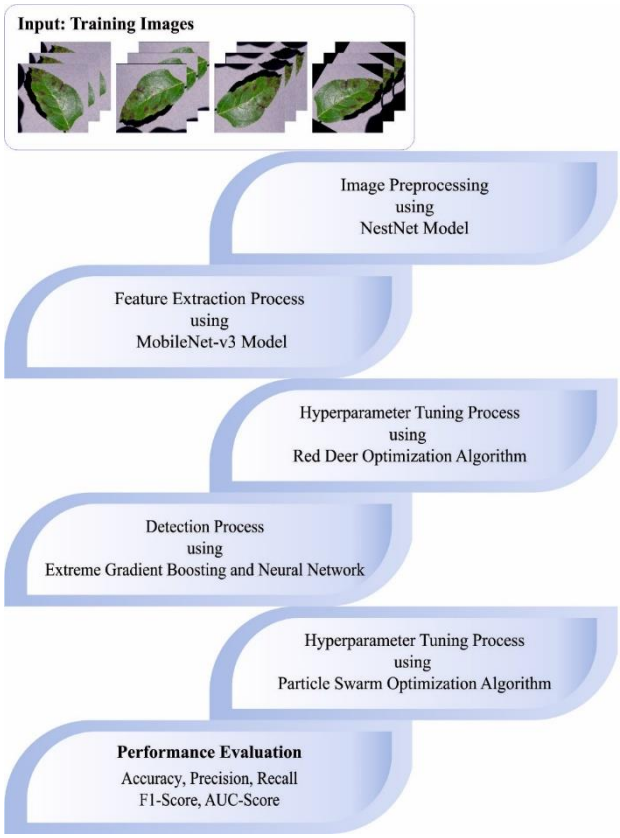


Figure. 1 Overall process of RDODL-APDC system

are similar in height and width. But the channels were distinct. The tensor at a similar position in 2 images is utilized as the input of ADO. ADO as an essential activation function is effectually distinguishing the feature variances betwixt 2 images. The NestNet approach establishes 2 parallel elements for processing images of distinct periods. Enhancing the count of channels results in a superior computational cost to process for extracting image data.

3.2 Image segmentation process

Here, the multi-level thresholding segmentation segmented the diseased portions of the plant leaf imagery. The Kapur is a multi-level thresholding technique employed to apply the rule of segments [19]. This technique selects the best threshold value based on the maximized entropy:

$$F_{kapur}(th) = H_1 + H_2 \quad (1)$$

In Eq. (1), H_1 and H_2 entropies are computed by the following equation:

$$H_1 = \sum_{i=1}^{th} \frac{Ph_i}{\omega_0} ln\left(\frac{Ph_i}{\omega_0}\right) and\ H_2 = \sum_{i=th+1}^{L} \frac{Ph_i}{\omega_1} ln\left(\frac{Ph_i}{\omega_1}\right) \quad (2)$$

In Eq. (2), Ph_i denotes the probability distribution of intensity level, $\omega_0(th)$ and $\omega_1(th)$ represent the possibility distribution in the classes C_1 and C_2. $\ln(\cdot)$ denotes the natural logarithm. The entropy-depended approach was adjusted to multiple threshold values like the Otsu approach. It can be crucial for diving the images as to N class implementing $N-1$ threshold:

$$F_{kapur}(TH) = \sum_{i=1}^{N} H_i \quad (3)$$

In Eq. (3), a vector including various thresholds is $TH = [th1, th2, \dots, thN-1]$. All the entropy are computed discretely with the equivalent values (th) as follows:

$$H_N^c = \sum_{i=th_{N+1}}^{L} \frac{Ph_i}{\omega_{N-1}} ln\left(\frac{Ph_i}{\omega_{N-1}}\right) \quad (4)$$

The probability value $(\omega_0^c, \omega_1, \dots, \omega_{N-1})$ of $the\ N$ class is obtained and the probability distribution Ph_i.

3.3 Feature extraction

In this research, the MobileNet-v3 technique is exploited to derive feature vectors. According to the infrastructure which utilizes depthwise separable convolutional for creating lightweight DNNs, calculation, and size trade-offs of lessening the MobileNetV2 structure by width multiplier [20]. Typical convolutional is at a high price as every function was multiplication. Let there be $an\ N$ filter or kernel of sizes $Dk \times Dk \times M$. When the normal convolutional function was complete, afterwards the resultant size is demonstrated in Eq. (5). Therefore, for a normal convolutional function, the entire count of multiplications is depicted in Eq. (6). A single convolutional function needs $Dk \times Dk$ multiplications. Next, the entire count of multiplications is equivalent to as depicted in Eq. (7).

$$Total\ no\ of\ multiplications = M \times D_k^2 \times D_p^2 D_p \times D_p \times N, \quad (5)$$

$$N \times D_p^2 \times D_k^2 \times M, \quad (6)$$

$$M \times D_p \times D_p \times D_k \times D_k. \quad (7)$$

$$Mult\ sonce = D_K^2 \times M, \quad (8)$$

$$MultsperKernel = D_G^2 \times D_k^2 \times M, \quad (9)$$

$$MultsNKernels = N \times D_G^2 \times D_k^2 \times M. \quad (10)$$

Therefore, the MobileNet decreases the multiplication function which occurs in normal

convolutional, as MobileNet follows 2 important steps which are the depthwise convolutional step and pointwise convolution step rather than typical convolutional.

For optimal hyperparameter tuning process, the RDO algorithm is used here. "Red Deer" relates to an applicable solution X in the solution spaces [21]. The optimization problems are formulated below:

$$Value = f\,(Red\ Deer) = f\left(X_1, X_2, X_3, \ldots., X_{N_{var}}\right) \quad (11)$$

In Eq. (11), X_{Nvar} indicates the size of the array, and X_1, X_2, and X_3 indicate the array component. Now, X_{Nvar} is obtained as 50. For all the array components (viz., RD), a position is allocated according to problem size. Meanwhile, there exist three parameters to 7-level CHB, parameters are allocated for every RD. The location of the $i\text{-}th$ deer was described by the following expression:

$$X_i = [\theta_1, \theta_2, \theta_3] = for i = 1,2,3, \ldots, X_{N_{var}} \quad (12)$$

RDO acts on a population of solutions to a specific instead of an individual solution. Hence, it begins at random. Concerning the resolution spacing from the roar phase, the male RDs have neighbours, and every male RD is allowable for changing their location. Consequently, any of the preceding and succeeding locations has the better fitness function (FF) values chosen. The subsequent formula is applied for updating the location of male RDs.

$$male_{new} = \begin{Bmatrix} male_{old} + a_1 \times (((UB - LB) * a_2) + LB) \\ if \quad a_3 \geq 0.5 \\ male_{old} - a_1 \times (((UB - LB) * a_2) + LB) \\ if \quad a_3 < 0.5 \end{Bmatrix} \quad (13)$$

In Eq. (13), $male_{new}$ characterizes the novel location of male deer, $male_{old}$ its older location, and UB and LB correspondingly represent the upper as well as lower restrictions of the searching region. a_1, a_2, and a_3 are arbitrarily selected with a uniformly distributed number within [0, 1]. These 2 mathematical equations can be given regarding the fighting process in the following:

$$New\,1 = \frac{(Com+Stag)}{2} + b_1 \times \left(((UB - LB) * b_2) + LB\right) \quad (14)$$

$$New2 = \frac{(Com+\mathrm{Stag})}{2} - b_1 \times \left(((UB - LB) * b_2) + LB\right) \quad (15)$$

b_1 and b_2 are randomly selected with a regularly dispersed number within [0,1]. Afterwards, the roar and fight stages derive creating the harems. A harem comprises an ensemble of female deer (hinds) and a male commanding authority. The power of commanders can be determined according to the FF values:

$$N.harem_n = round\,\{P_n.N_{hind}\} \quad (16)$$

In Eq. (16) $N.harem_n$ represent the hind number in the n - th harems, N_{hind} represents the hind numbers, and P_n shows the commanding authority's power in the $n\text{-}th$ harems. In every case, offspring are generated using the following equation.

$$off\,s = \frac{(Com+\mathrm{Hind})}{2} + (UB - LB) \times c \quad (17)$$

Whereas, offs represent the novel offspring RD; i.e., it characterizes a novel solution. c is arbitrarily produced within [0,1]. Two distinct approaches are followed for selecting the upcoming generation. Initially, the best male RDs (Stags and Commanders) are storage (a specific best resolution percentage). Then, hinds are carefully chosen for the upcoming generation from amongst each hind and offspring generated in the mating stage, using the roulette wheel or fitness tournament process. The upcoming generation was generated with the preferred female and male RDs. The ending criteria could be the optimum solution ever found, a time interval, or the iteration count.

3.4 Image classification process

Finally, the PSO with XGBoost and NN models is used for the detection process. An elaborated working of these models is detailed as follows.

3.4.1. XGBoost model

XGBoost is an enhanced structure of the GBRT approach which is a boosting method comprising a sequence of regression trees with a sequential ensemble algorithm [22]. It adaptively adds additional trees to extend the capacity of the model:

$$\hat{y}^{\mathrm{m}} = \hat{y}^{\mathrm{m-1}} + \alpha f_{\mathrm{m}}(X, \theta_{\mathrm{m}}) = \alpha \sum_{j=1}^{m} f_i\left(X, \theta_j\right) \quad (18)$$

In Eq. (18), m represents the regression tree numbers for boosting; θ_j indicates the variable to control the architecture of $j\text{-}th$ trees; α represents the learning rate or shrinkage factor of separate regression tree; X shows the predictor and $\hat{y}$ represents the predictive of $j\text{-}th$ regression trees; and $f_j(X, \theta_j)$ indicates the output of $j\text{-}th$ regression trees based on the architecture of θ without shrinkage, where the predictor X and the residual $y - j$ are applied as input. Subsequently, the residual reduces with the augmented amount of regression trees. XGBoost aims to discover the optimum θ_j and construct $f_j(X, \theta_j)$ at $j\text{-}th$ steps to minimalize the objective function as follows:

$$L = \sum_i l\,(\hat{y}_i, y_i) = \sum_i l\left[\hat{y}_i^{j-1} + \alpha f_i\big(X_i, \theta_j\big), y_i\right] \quad (19)$$

In Eq. (20), l represents the loss function that generally applies the squared errors amongst the ground truth y and the prediction value $\hat{y}$. In comparison with the traditional GBRT approach, a regularization term has been presented to the traditional loss function in XGBoost for penalizing the complication of the model and avoid the model from over-fitting.

$$L = \sum_i l\,(\hat{y}_i, y_i) + \sum_j \Omega\left(\theta_j\right) = \sum_i l\left[\hat{y}_i^{j-1} \alpha f_i\big(X_i, \theta_j\big), y_i\right] + \sum_j \Omega\left(\theta_j\right) \quad (20)$$

In Eq. (20), $\Omega(\theta_j)$ indicates the regularization item on $j\text{-}th$ regression trees to avoid over-fitting:

$$\Omega\big(\theta_j\big) = \gamma T_j + \frac{1}{2}\lambda\|w_k\| = \gamma T_j + \frac{1}{2}\lambda\sum_{k=1}^{T_j}[\,w_{\mathrm{k}}^{(\mathrm{j})}]^2 \quad (21)$$

In Eq. (21), T_j represents the number of leaves in $j\text{-}th$ regression trees, γ represents the minimal loss reduction required to additional node partitions in regression trees, λ shows the regularization term on weighted of leaves in regression trees, and $w_{1_(}^{0)}$ indicates the weight of $k\text{-}th$ leaf in $j\text{-}th$ regression trees. Additional leaves (large T_j) would be penalized by the large factors γ for minimizing the objective function.

3.4.2. NN model

NN is commonly used for nonlinear and nonparametric problems with complicated mapping in input to output. It contributes for recognizing the relationships between unknown parameters and known variables. MLP network has one type of NN, which is a most common and typical feedforward network that obtains the benefits of robustness and non-linearity concerning its mapping procedure from input to output. The backpropagation and signal forward of errors enable the weight to be upgraded efficiently based on the learning rule. Like the biological nervous system, MLP includes a large number of neurons that communicate with the respective connection amongst one another. The neurons are arranged in the form of layers that are classified into output, input, and hidden layers. The eventual outcomes emerge by incorporating the solution in every resultant layer with $the\ f$ activation function. Therefore, combining output with the input X_i of the forwarding process is represented with weight w_{ij}

$$O_j(k) = f\left(b_i + \sum_i w_{ij}\, X_i\right) \quad (22)$$

PSO based Parameter Tuning

To adjust the parameters related to the XGBoost and NN models, the PSO algorithm is exploited here. PSO algorithm is stimulated by the swarming of N individuals named particles, each signifying a solution to the problem with $the\ D$ dimension [23]. Its genotype comprises $2D$ parameters, with the first D parameters signifying the coordinate of particle location and the last D parameter as its velocity component in the D -dimension problem space, correspondingly. In addition to the two fundamental properties, the succeeding properties are the $gbest$ global best location of the entire swarm and the $best$ personal optimum position of all the particles in the search spaces. The FF corresponds to the problem utilized for evaluating the (goodness" of all the particles):

$$v_i^{(t+1)} = \omega v_i^{(t)} + c_1 r_1\left(pbest_i^{(t)} - p_i^{(t)}\right) + c_2 r_2\left(gbest^{(t)} - p_i^{(t)}\right) \quad (23)$$

$$p_i^{(t+1)} = p_i^{(t)} + v_i^{(t+1)}, \quad (24)$$

In Eq. (23), $p_i^{(t)}$ and $v_i^{(t)}$ indicates the position and velocity of $i\text{-}th$ particles at $t\text{-}th$ iterations, correspondingly. c_1 and c_2 positive factors are called the cognitive and social coefficients, controlling the contribution of the $best$ (cognitive component) best local solution and $gbest$ (social component) global best solution, correspondingly. r_1 and r_2 show the two independent random variables ranging from zero to one. The inertia-weighted factor ω is utilized for

Table 1. Details of datasets

Apple Plant Disease Data set		Grape Plant Disease Data set	
Disease	**No. of Images**	**Disease**	**No. of Images**
Scab	2016	Black Measles	1920
Black Rot	1987	Black Rot	1888
Cedar Apple Rust	1760	Leaf Blight	1722
Healthy	2008	Healthy	1692
Total Images	7771	Total Images	7222

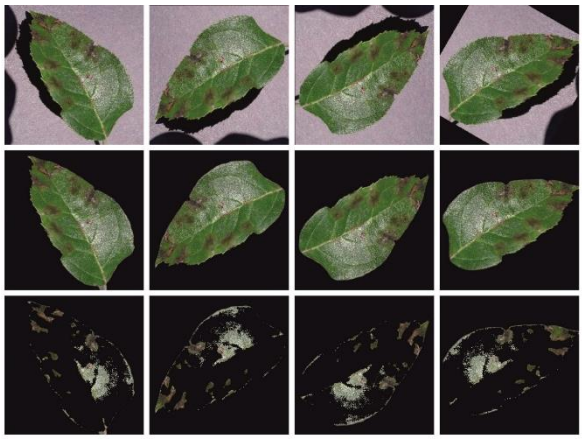


Figure. 2 Original images with pre-processed and segmented versions

controlling the swarm convergence. In the presented method, a non-linear altering inertia factor for PSO and SSPSO can be applied as follows:

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min})t}{T_{\max}}, \quad (25)$$

In Eq. (25), $T_{\max}$ indicates the maximal iteration count and t represents the existing iteration count. Noticeably, in the iteration, each dimension of the velocity can be determined within $[-V_{\max}, V_{\max}]$ for limiting the maximal distance that particles will move.

The PSO system introduces an FF for realizing greater classifier performances. It describes a positive integer for illustrating the good effectiveness of candidate results. During this work, the minimizing of classifier error rating was supposed that FF is formulated as in Eq. (26).

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{no.of\ misclassified\ instances}{Total\ no.of\ instances} \times 100 \quad (26)$$

4. Results and discussion

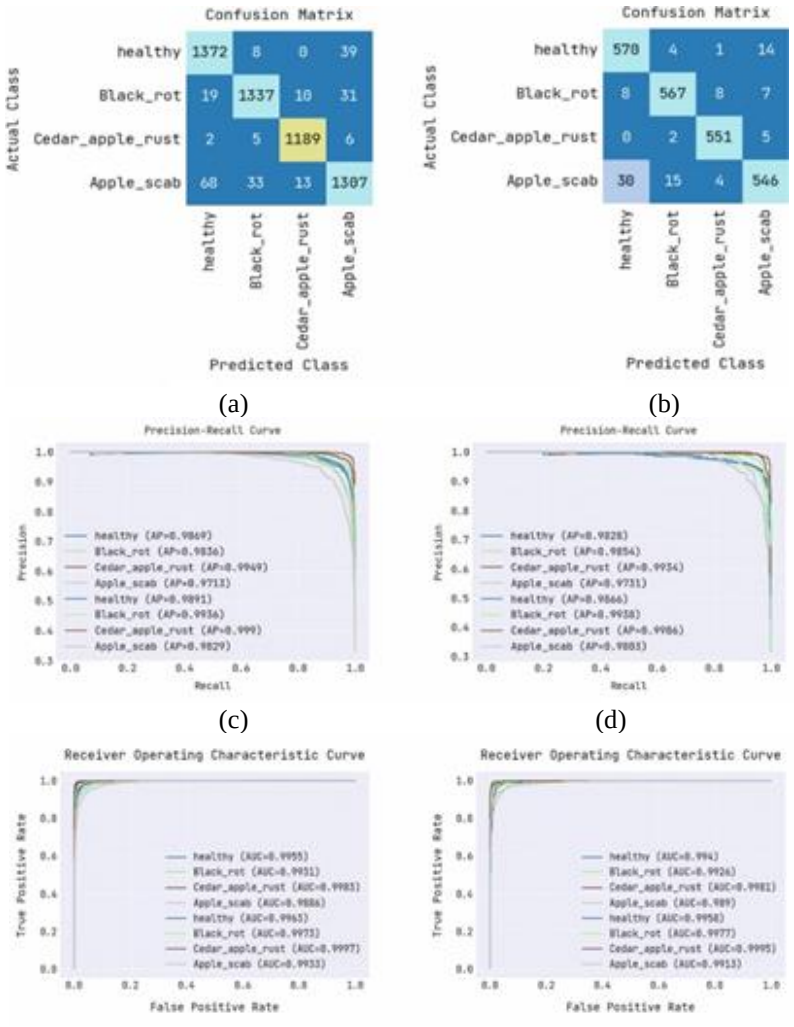
In this segment, the experimental validation of the RDODL-APDC technique is examined by employing two datasets [24]: apple plant disease and datasets of grape plant ailments. The apple plant disease dataset comprises 7771 samples and the grape plant disease dataset holds 7222 images as defined in Table 1.

Fig. 2 shows the sample set of imageries offered by the RDODL-APDC model. The original images and the pre-processed versions are displayed in first and second rows. Then, the final row images showcase the segmented images.

4.1 Results on apple plant disease dataset

Fig. 3 depicts a set of outcomes attained by the NN system with TR/TS data under the Apple dataset. Fig. 3 (a) shows the confusion matrix of the NN system under TR data. The figure pointed out that the NN system has identified 1372 instances in HY class, 1337 instances in the BR class, 1189 instances in the CAR class, and 1307 instances in AS class. Next, Fig. 3b denotes the confusion matrix of the NN system under TS data. The figure stated that the NN model has categorized 570 instances into HY class, 567 instances into the BR class, 551 instances into the CAR class, and 546 instances into AS class. Then, Figs. 3 (c-d) shows the precision-recall curves of the NN methodology under TR/TS data. The figures revealed that the NN algorithm has attained higher performance under distinct classes. Similarly, Figs. 3 (e-f) demonstrates the ROC investigation of the NN technique under the TR/TS dataset. The outcomes designated that the NN approach has attained maximum ROC values.

Fig. 4 depicts a set of outcomes acquired by the XGBoost system with TR/TS data under the Apple dataset. Fig. 4 (a) indicates the confusion matrix of the XGBoost approach under TR data. The figure referred that the XGBoost system has identified 1381 instances in HY class, 1301 instances in the BR class, 1177 instances in the CAR class, and 1282 instances in AS class. Likewise, Fig. 4 (b) represents the confusion matrix of the XGBoost system under TS data. The figure referred that the XGBoost methodology has categorized 573 instances into HY class, 548 instances into the BR class, 544 instances into the CAR class, and 543 instances into AS class. Followed by, Figs. 4 (c-d) shows the precision-recall curves of the XGBoost model under TR/TS data. The figures reported that the XGBoost model has gained enhanced performance under distinct classes. Eventually, Figs. 4 (e-f) indicates the ROC examination of the XGBoost model under the TR/TS dataset. The outputs indicated that the XGBoost approach has reached maximal ROC values. Table 2



(a) (b) (c) (d) (e) (f)

Figure. 3 Neural network classifier on apple dataset: (a) Training confusion matrices, (b) Testing confusion matrices, (c) Training PR-Curve, (d) Testing PR-Curve, (e) Training ROC Analysis, and (f) Testing ROC analysis

Table 2. Result analysis of NN and XGBoost methods under the apple dataset

Measures	Neural Network		XGBoost	
	Training Set	Testing Set	Training Set	Testing Set
Accuracy	95.70	95.80	94.52	94.68
Precision	95.80	95.83	94.60	94.79
Recall	95.82	95.85	94.65	94.73
F1-Score	95.80	95.82	94.60	94.73
AUC Score	99.67	99.61	99.39	99.34

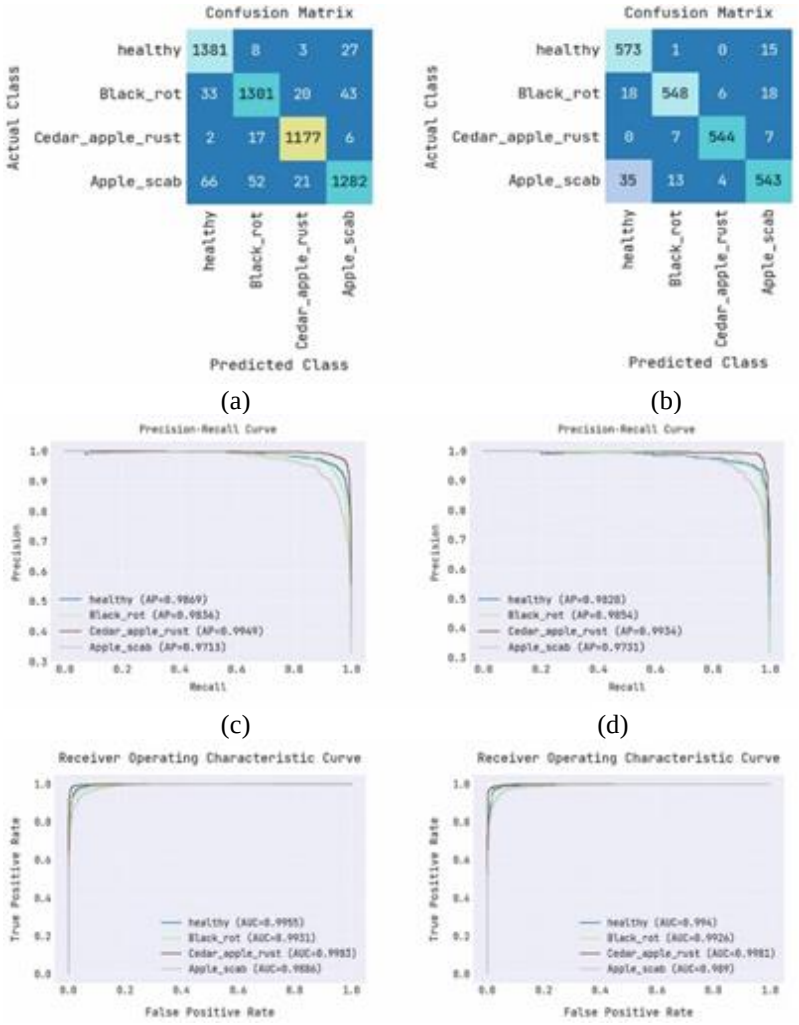
Table 3. Relative evaluation of RDODL-APDC system with other models under Apple database

Apple Dataset	
Method	**Accuracy (%)**
RDODL-APDC	95.80
SVM	68.73
BP Algorithm	54.63
AlexNet	91.19
CNN-MobileNet	73.50
InceptionV3	75.59
ResNet152	77.65

offers a brief plant disease detection results of the NN and XGBoost classifiers on the apple dataset.

4.2 Results on grape plant disease dataset

Fig. 7 demonstrates the overall outputs of the RDODL-APDC model on the testing set of the Grape dataset. The outputs implied that the NN classifier has exhibited higher outcomes over the XGBoost algorithm. The NN classifier has attained maximal $accu_y$ of 97.19%, $prec_n$ of 97.35%, $reca_l$ of 97.24%, $F1_{score}$ of 97.27%, and AUC_{score} of 99.88%.



(a) (b) (c) (d) (e) (f)

Figure. 4 XGBoost classifier on apple dataset: (a) Training confusion matrices, (b) Testing confusion matrices, (c) Training PR-Curve, (d) Testing PR-Curve, (e) Training ROC analysis, and (f) Testing ROC analysis

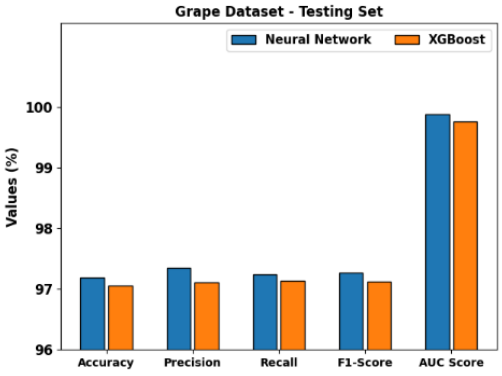


Figure. 7 Result analysis of NN and XGBoost methods with training set under Grape dataset

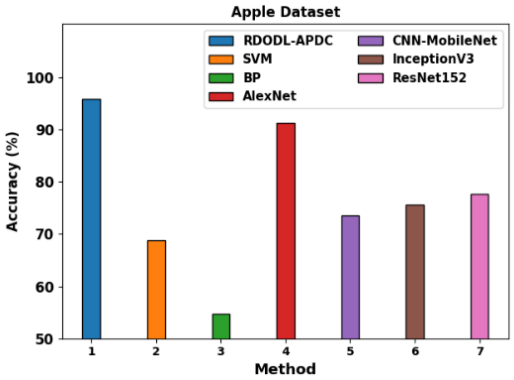


Figure. 8 Relative evaluation of RDODL-APDC system under apple database

4.3 Discussion

Table 3 and Fig. 8 offer relative outputs of the RDODL-APDC technique on the Apple dataset [25-27]. The obtained outputs show that the SVM and BP models have reported lower $accu_y$ values of 68.73% and 54.63% respectively. Next, the CNN-MobileNet, Inception v3 and ResNet152 models have shown slightly enhanced $accu_y$ values of 73.50%, 75.59%, and 77.65% respectively. Although the AlexNet model has reached a reasonable $accu_y$ of 91.19%,

the RDODL-APDC model has shown a maximum $accu_y$ of 95.80%.

5. Conclusion

In this research, an automated plant disease detection approach, namely the RDODL-APDC algorithm has been introduced, which exploited the DL model for recognizing and categorizing plant ailments. The RDODL-APDC technique comprises a series of subprocesses namely NestNet background removal, Multi-level Thresholding Segmentation, MobileNet feature extractor, RDO based hyperaprameter tuning, and PSO with XGBoost and NN models for the detection process. For exhibiting the betterment of the RDODL-APDC technique, a far reaching simulations was accomplished. The experimental values reported the improvement of the RDODL-APDC technique over other existing methodologies 95.80% and 97.19% on the Apple and grape plant disease dataset respectively. In future, the deep instance segmentation process can be applied to accomplish enhanced classification outcomes.

Conflict of interest:

The authors declare no conflict of interest.

Author contributions:

Conceptualization, Raja and Karthikeyan; methodology, Raja; software, Raja; validation, Karthikeyan; formal analysis, Karthikeyan; investigation, Raja and Karthikeyan; resources, Raja; data curation, Raja; writing-original draft preparation, Raja; writing-review and editing, Karthikeyan; visualization, Raja; supervision, Karthikeyan; project administration, Karthikeyan; funding acquisition, Karthikeyan. All authors have read and approved the final manuscript.

References

[1] M. Prabu and B. J. Chelliah, "Mango leaf disease identification and classification using a CNN architecture optimized by crossover-based levy flight distribution algorithm", *Neural Computing and Applications*, Vol. 34, No. 9, pp. 7311-7324, 2022.

[2] A. K. Dewangan, S. Kumar and T. B. Chandra, "Leaf-Rust and Nitrogen Deficient Wheat Plant Disease Classification using Combined Features and Optimized Ensemble Learning", *Research Journal of Pharmacy and Technology*, Vol. 15, No. 6, pp. 2531-2538, 2022.

[3] T. N. Pham, L. V. Tran and S. V. T. Dao, "Early disease classification of mango leaves using feed-forward neural network and hybrid metaheuristic feature selection", *IEEE Access*, Vol. 8, pp. 189960-189973, 2020.

[4] P. Bedi, P. Gole and S. K. Agarwal, "18 Using deep learning for image-based plant disease detection", *Internet of Things and Machine Learning in Agriculture*, pp. 369-402, 2021, doi:

[5] S. J. Wei, D. F. A. Riza and H. Nugroho, "Comparative study on the performance of deep learning implementation in the edge computing: Case study on the plant leaf disease identification", *Journal of Agriculture and Food Research*, p. 100389, 2022.

[6] P. D. Kusuma and A. L. Prasati "Guided Pelican Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 179–190, 2022, doi: 10.22266/ijies2022.1231.18.

[7] P. D. Kusuma and M. Kallista "Stochastic Komodo Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 156-166, 2022, doi: 10.22266/ijies2022.0831.15.

[8] P. D. Kusuma et al. "Extended Stochastic Coati Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 482–494, 2023, doi: 10.22266/ijies2023.0630.38.

[9] P. D. Kusuma and A. Dinimaharawati, "Fixed Step Average and Subtraction Based Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 339-351, 2022, doi: 10.22266/ijies2022.0831.31.

[10] F. A. Zeidabadi and M. Dehghani, "POA: Puzzle Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, pp. 273-281, Feb. 2022, doi: 10.22266/ijies2022.0228.25.

[11] J. Annrose, N. Rufus, C. R. Rex and D. G. Immanuel, "A cloud-based platform for soybean plant disease classification using archimedes optimization based hybrid deep learning model", *Wireless Personal Communications*, Vol. 122, No. 4, pp. 2995-3017, 2022.

[12] Y. Guo, J. Zhang, C. Yin, X. Hu, and Y. Zou *et al.*, "Plant disease identification based on deep learning algorithm in smart farming", *Discrete Dynamics in Nature and Society*, 2020, doi:

[13] S. Nandhini and K. Ashokkumar, "An automatic plant leaf disease identification using DenseNet-

121 architecture with a mutation-based henry gas solubility optimization algorithm", *Neural Computing and Applications*, Vol. 34, No. 7, pp. 5513-5534, 2022.

[14] D. Aqel, S. A. Zubi, A. Mughaid, and Y. Jararweh, "Extreme learning machine for plant diseases classification: a sustainable approach for smart agriculture", *Cluster Computing*, Vol. 25, No. 3, pp. 2007-2020, 2022.

[15] T. R. Gadekallu, D. S. Rajput, M. Reddy, K. Lakshmanna, S. Bhattacharya *et al.*, "A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU", *Journal of Real-Time Image Processing*, Vol. 18, No. 4, pp. 1383-1396, 2021.

[16] R. Cristin, B. S. Kumar, C. Priya, and K. Karthick, "Deep neural network based Rider-Cuckoo Search Algorithm for plant disease detection", *Artificial Intelligence Review*, Vol. 53, No. 7, pp. 4993-5018, 2020.

[17] A. Pavithra and G. Kalpana, "An Efficient Plant Disease Identification Using Intense Transfer Learning Empowered Efficient Net with Kernel Extreme Learning Machine", In: *Proc. of 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India: IEEE, pp. 909–914, 2022. doi: 10.1109/ICSSIT53264.2022.9716330.

[18] X. Yu, J. Fan, J. Chen, P. Zhang, Y. Zhou *et al.*, "NestNet: A multiscale convolutional neural network for remote sensing image change detection", *International Journal of Remote Sensing*, Vol. 42, No. 13, pp. 4898-4921, 2021.

[19] E. H. Houssein, E. D. Helmy, D. Oliva, A. A. Elngar and H. Shaban, "Multi-level thresholding image segmentation based on nature-inspired optimization algorithms: a comprehensive review", *Metaheuristics in Machine Learning: Theory and Applications*, pp. 239-265, 2021.

[20] A. lnuaim, M. Zakariah, W. A. Hatamleh, H. Tarazi, and V. Tripathi *et al.*, "Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet", *Computational Intelligence and Neuroscience*, 2022.

[21] Y. Bektaş and H. Karaca, "Red deer algorithm based selective harmonic elimination for renewable energy application with unequal DC sources", *Energy Reports*, Vol. 8, pp. 588-596, 2022.

[22] X. Zhu, J. Chu, K. Wang, S. Wu, and W. Yan *et al.*, "Prediction of rockhead using a hybrid N-XGBoost machine learning framework", *Journal of Rock Mechanics and Geotechnical Engineering*, Vol. 13, No. 6, pp. 1231-1245, 2021.

[23] X. Zhang, L. Jiao, A. Paul, Y. Yuan, and Z. Wei *et al.*, "Semisupervised particle swarm optimization for classification", *Mathematical Problems in Engineering*, 2014, doi:

[24] https://www.kaggle.com/vipoooool/new-plant-diseases-dataset/data#

[25] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks", *Symmetry*, Vol. 10, No. 1, p. 11, 2017.

[26] C. Bi, J. Wang, Y. Duan, B. Fu, and J. R. Kang *et al.*, "MobileNet based apple leaf diseases identification", *Mobile Networks and Applications*, pp. 1-9, 2020.

[27] B. Liu, Z. Ding, Y. Zhang, D. He, and J. He, "Kiwifruit Leaf Disease Identification Using Improved Deep Convolutional Neural Networks", In: *Proc. of 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, Madrid, Spain, pp. 1267–1272, 2020, doi: 10.1109/COMPSAC48688.2020.00-82.