International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I11P218

Volume 71 Issue 11, 171-180, November 2023

Original Article

An Intelligent Framework for Rice Plant Disease Recognition and Classification using Deep Learning and Reptile Search Algorithm

D. Felicia Rose Anandhi[1], S. Sathiamoorthy[2]

[1]*Department of Computer and Information Science, Faculty of Science, Annamalai University, Villupuram, Tamilnadu, India.*
[2]*Annamalai University PG Extension Centre, Villupuram, Tamilnadu, India.*

[1]*Corresponding Author : feliciarosephd@gmail.com*

Received: 22 July 2023 Revised: 21 September 2023 Accepted: 27 September 2023 Published: 04 November 2023

Abstract *- Rice is one of the majorly utilized primary crops globally, providing sustenance for the worldwide population's crucial segment. However, if left untreated, rice plants are susceptible to various diseases that can cause substantial yield losses. Hence, precise and timely recognition of rice plant diseases is substantial for effectively managing the disease and ensuring optimal crop production. Therefore, this study concentrates on constructing an Automated Rice Plant Disease Detection utilizing a Reptile Search Algorithm with Deep Learning (ARPDD-RSADL) technique. The purpose of the ARPDD-RSADL technique lies in the accurate and proficient classification of rice plant ailments using a parameter-tuned DL model. To accomplish this, the ARPDD-RSADL technique performs image preprocessing in two stages: Unsharp Masking (UM) based filtering and contrast enhancement. In addition, the ARPDD-RSADL technique designs the Squeeze and Excitation ResNet (SE-ResNet) model for feature extraction purposes. The Stacked Sparse Auto-Encoder (SSAE) model is used with RSA based hyperparameter optimizer for plant disease detection. The design of RSA helps to adjust the parameters relevant to the SSAE model optimally, and it assists in attaining advanced accuracy in detection. An extensive experimentation analysis was accomplished to validate the effectual rice plant disease classification results of the ARPDD-RSADL method. The simulation values portrayed the ARPDD-RSADL method's excellence in diverse evaluation measures. Investigational outcomes illustrate the effectiveness of the proposed model in accurately classifying and recognizing several rice plant diseases.*

Keywords *- Rice plant disease, Computer vision, Agriculture, Deep learning, Reptile search algorithm.*

1. Introduction

The majority of the world population relies upon rice food, and it becomes a great difficulty for the agriculture sector to ensure food security [1]. The primary reasons for diseases are microorganisms and fungi. Such unhealthy rice crops give rise to a reduction in rice productivity that affects the economy with immense loss to the agriculturalists every year [2]. Therefore, detecting diseases in agricultural goods in their previous stage is of utmost importance for quality improvement and to avoid production loss [3]. Commonly, the identification of rice plant disease can be performed either based on experimental outcomes by culturing microbes in the laboratories or based on the visual valuation of the indications [4]. Visual evaluation is a subjective approach and is predisposed to error. In addition to these limitations, the orthodox methods need specialists to find the disease, and it is hard for agronomists to gain admittance to professionals due to the interior area of the farming field [5]. Such issues have impelled researchers to inspect different methods to devise automatic categorization and identification methods for rice plant diseases.

Computer Vision (CV) methods that can identify plant diseases would be useful tools for resistance breeding and disease management [6]. The latest pattern recognition and image processing methods have found some solutions for disease diagnosis to help agriculture experts and farmers [7]. Images of the various parts of the plant were captured for developing plant disease detection mechanisms. A precise and potent real-time recognition of disease mechanisms may assist in devising reduction methods to guarantee financial food safety on a large scale and rice crop protection on a small scale [8]. Further, precise disease detection of the rice plant using Deep Learning (DL) and CV can offer the basis for the site-precise agrochemical application.

Conversely, the outline of image analysis tools is a potential algorithm for identifying plant ailments and

continually detecting plant health conditions [9]. The application of CV and Artificial Intelligence (AI) to the automated diagnosis and detection of rice plant ailments is broadly studied, as manual plant disease monitoring is labour-intensive, tedious, and time-consuming. Currently, DL methods are abundantly used in detecting diseases [10].

This study concentrates on constructing an Automated Rice Plant Disease Detection utilizing a Reptile Search Algorithm with Deep Learning (ARPDD-RSADL) technique. The purpose of the ARPDD-RSADL technique lies in the accurate and proficient classification of rice plant ailments using a parameter-tuned DL model. To accomplish this, the ARPDD-RSADL technique performs image preprocessing in two stages: Unsharp Masking (UM) based filtering and contrast enhancement. In addition, the ARPDD-RSADL technique designs the Squeeze and Excitation ResNet (SE-ResNet) model for feature extraction purposes. The Stacked Sparse Auto-Encoder (SSAE) model is used with RSA based hyperparameter optimizer for plant disease detection. The design of RSA helps to adjust the parameters relevant to the SSAE model optimally, and it assists in attaining advanced accuracy in detection. An extensive experimentation analysis was accomplished to validate the effectual rice plant disease classification results of the ARPDD-RSADL method.

2. Related Works

Zhang et al. [11] introduced a High-Quality Image Augmentation (HQIA) approach for building higher-quality rice leaf ailment imageries related to dual Generative Adversarial Networks (GAN). Firstly, the initial instances are exploited for training Wasserstein GAN (WGAN-GP) to make pseudo-data instances. For generating higher-quality pseudo-data instances, the above-mentioned instances are given into Optimized-Real-ESRGAN. Eventually, the higher quality pseudo-data instances are given into Convolutional Neural Networks (CNN) based classification of the disease, and pointers verify the efficacy of the approach. Lamba et al. [12] presented a new NN-related hybrid method (GCL). GCL was a dataset-augmenting combination of CNN with GAN and Long Short-Term Memory (LSTM). GAN was employed for dataset augmenting; CNN extracted the attributes, and LSTM categorized the different rice ailments.

Zahro et al. [13] target to find ailments that happen in the rice plant leaves with the CNN approach. CNN is one of the effective approaches utilized for classifying an image. In [14], an Attention-relevant Depthwise Separable NN was designed, including Bayesian Optimization (ADSNN-BO) to categorize and find rice diseases from paddy leaf imagery. Still, rice ailment detection is mainly performed manually. The author devised the ADSNN-BO approach relevant to the augmented attention mechanism and MobileNet structure. To attain AI-supported precise and rapid disease detection, in addition to tuning hyper-parameters of the method, the Bayesian optimization method was adopted. Jiang et al. [15] present a rice disease detection approach related to DenseNet. This approach utilizes DenseNet as a benchmark method and uses channel attention system squeeze-and-excitation to strengthen the favourable structures while overpowering unfavourable attributes.

In [16] examined existing methods to find plant leaf disease utilizing DL and high-end imaging approaches, along with their difficulties. The conventional CNN has the problem of utilizing more computing power. To solve this problem, this study devised a Dilated CNN (DCNN) approach with Global Average Pooling (GAP) that can be built by varying usual CNN convolutional kernel having enlarged convolution kernel and Fully Connected (FC) in classical CNN substituted by GAP. In [17], a new Deep Neural Network (DNN) classifier algorithm was advanced for finding rice disease with the help of plant images. Classification errors are reduced by maximizing preconceptions and weightage in the DNN method through a Crow Search Algorithm (CSA) in the typical finetuning and pre-training.

Tunio et al. [18] employ fusion deep CNN transfer learning with rice plant imaging or the recognition and categorization of several rice ailments; Transfer Learning (TL) is employed for generating the DL approach by implementing Rice_Leaf_Dataset from a subordinate source. In [19], the presented ECA-ConvNeXt approach integrated the Efficient Channel Attention (ECA) component that augmented the extracting process by implementing just a scarce parameter. TL was enforced for loading finetuning, and pre-training weights were utilized. Ramesh and Vydeki [20] presented a model utilizing Optimized DNN with the Jaya Algorithm. The elimination of the RGB imageries under the preprocessing is transformed into HSV, and depending on the saturation and hue parts, binary imageries are put under extraction for splitting the attacked and non-attacked regions.

3. The Proposed Model

The present study designed a novel ARPDD-RSADL approach for precisely classifying and recognizing plant disease. This approach exploits the CV and DL concepts with a tuning process. The method encompasses UM-based noise removal, contrast enhancement, SE-ResNet feature extraction, SSAE classification, and RSA-based tuning process to reach this. Fig. 1 exhibits the complete workflow.

3.1. Image Preprocessing

In the preliminary phase, the ARPDD-RSADL technique performs image preprocessing in two stages: UM-based filtering and contrast enhancement. UM is an image enhancement model used for sharpening the captured images [21]. The image with Gaussian blur used the original imagery, and imagery sharp details are calculated as a difference between the original images. In the beginning, the input imagery was blurred with a Gaussian blur filter. The blur amount and radius are two important variables for Gaussian

blurry imagery. The radius has a consequence on the edge size that should be extended. The amount of contrast, darkness, and lightness added to the edges of the images is based on the following expression.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-((x^2+y^2)/(2\sigma^2))} \quad (1)$$

In Eq. (1), x and y denote the horizontal and vertical distance from the source, and σ refers to the Gaussian distribution. $I_{enhanced}$ represent attained image after improvement as follows:

$$I_{enhanced} = \left(I_{original} + contrast_{value} * (I_{blur})\right) \quad (2)$$

In Eq. (2), I_{blur} denotes the unsharp image and $I_{original}$ represents the original image. Next, the Contrast Limited Adaptive Histogram Equalization (CLAHE) approach was leveraged to improve the contrast level of the noise-removed images.

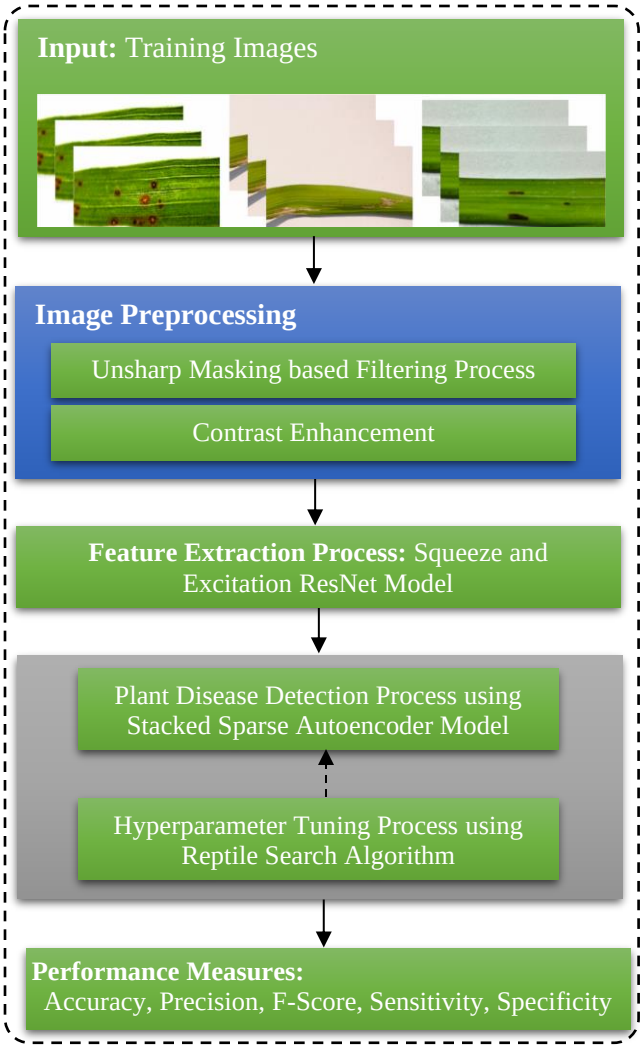

Fig. 1 Working flow of ARPDD-RSADL approach

3.2. Feature Extraction

Once the input images are preprocessed, the subsequent phase is to produce feature vectors using the SE-ResNet approach. ResNet added a shortcut connecting the branch exterior to the convolution layer to execute an easy constant mapping and for processing fundamental Residual Learning (RL) units. The next resolves the problem of network degradation, which can be complex to train once CNN is too deep by consecutively stacking RL units, making it feasible for training DCNNs [22]. The fundamental RL unit neither establishes novel variables nor enhances the computing complexity.

The principle of ResNet is:

$$x_l = h(x_l) + F(x_l, W_L) \quad (3)$$

$$x_{l+1} = f(y_1) \quad (4)$$

Whereas, $h()$ refers to direct mapping and $f()$ denotes activation function.

The residual block is formulated as:

$$x_{l+1} = x_l + F(x_l, W_1) \quad (5)$$

The connection between the deeper L, and l layer:

$$x_L = x_l + \sum_{i=1}^{L-1} F(x_i, W_i) \quad (6)$$

Based on the chain rule for derivative utilized in BP, a gradient of loss functions ε in terms of x_l is

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L}\left(1 + \frac{\partial}{\partial x l}\sum_{i=1}^{L-1} F(x_i, W_i)\right)$$
$$= \frac{\partial \varepsilon}{\partial x_L} + \frac{\partial \varepsilon}{\partial x_L}\frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, W_i) \quad (7)$$

During the trained model, $\frac{\partial}{\partial x_l}\sum_{i=1}^{L-1} F(x_i, W_i)$ could not be -1 constantly; therefore, there was not an issue of gradient disappearance from the remaining networks, and $\frac{\partial \varepsilon}{\partial x_l}$ refers that the gradient of $the\ L$ layer was directly taken from some l layer that is lower than it.

The feature extraction by CNN over convolution layer stacking is a high-dimensional feature. A few cases are lost, whereas the remaining ResNet block was to skip feature extraction by any convolution layer and merge attributes before n layers having convolution features after n layers to either high- or low-dimensional features can be maintained, and the network efficiency was enhanced. The GAP is also utilized for replacing the FC layer from the typical CNN. The GAP supports the correspondence among mapping features and classifications on the FC layer that is more appropriate for

convolution structures. Besides, there are no parameters optimized from the GAP that avoid overfitting. Additionally, GAP combines spatial data and is further robust regarding the spatial transformation of input.

SE-ResNet concentrations on the interdependency among convolved feature channels utilizing 1-D convolutional. The SE block has been accomplished by a squeeze operation that provides a summary of the entire data of all the feature maps and an excitation operation that scales the significance of all the map features. During this method, the squeeze function extracts the important data of all the channels, and the excitation function calculates the dependency among channels utilizing an FC layer with a non-linear function.

3.3. Image Classification

For plant disease detection, the SSAE model is used. SSAE was introduced for higher-level feature learning from lapping imaging patches in the training process. SSAE refers to an unsupervised model of DL, which has a fundamental layer for feature learning [23]. First, the feature-learning model is discussed by SAE and later presents the stack SAE; lastly, sigmoid layers with unsupervised SSAE are applied for finetuning the SVseg method. Fig. 2 depicts the construction of the SSAE method.

The basic element for SSAE, AE, works for feedforward non-linear NN training. It encompasses three major layers: hidden, input, and output layers. There exist many nodes that make every layer of AE; this node establishes a full connection between the nodes of neighbouring layers. The AE comprises the encoder-decoder processing stage.

The input vector presentation can be encrypted at the encoding phase to connect the AE Hidden Layer (HL) and the input layer. At the same time, during the decoding process, the AE suggests an input vector reconstructed from encoded feature learning in the HL. The AE aims to define the input data representation that creates a better reconstruction. Input image patches x_j are fed into AE during training, and reduces error factor for all network connection weights has been adopted by the following equation:

$$Arg\text{Min}_{W,b,\widehat{W},\widehat{b}} \sum_{i=1}^{N} \left| x_i - \left(\widehat{W}\left(\sigma(W x_i + b)\right) + \widehat{b} \right) \right|_2^2 \qquad (8)$$

Where b and σ denote weights, biases, and activation functions of AE parameters. Assume an input vector x_j, an AE initially encrypts input into representing $h_i = \sigma(W x_i + b)$, here h_i denotes the x_i response of HL neurons, and h indicates the dimension associated with neurons in the HL. The AE needs the HL dimension to be lesser than the dimension of the input layer for feature extraction from the patches of an input image; otherwise, error minimization can result in a trivial solution.

An alternative method termed SAE imposed sparsity regularization on AE-HLs rather than a limitation of HL dimension. SAE performs the regularization of HL response to evade trivial solutions that fundamental AEs tend towards. The AEs require the dimension of HL to be lesser than the input layer. Specifically, the sparsity regulation was executed on AE to make it infinitesimal. To construct a balance between reconstruction power and the HLs sparsity, for each input node, just the best-hidden nodes response which drives SAE for representing training sets in sparse feature:

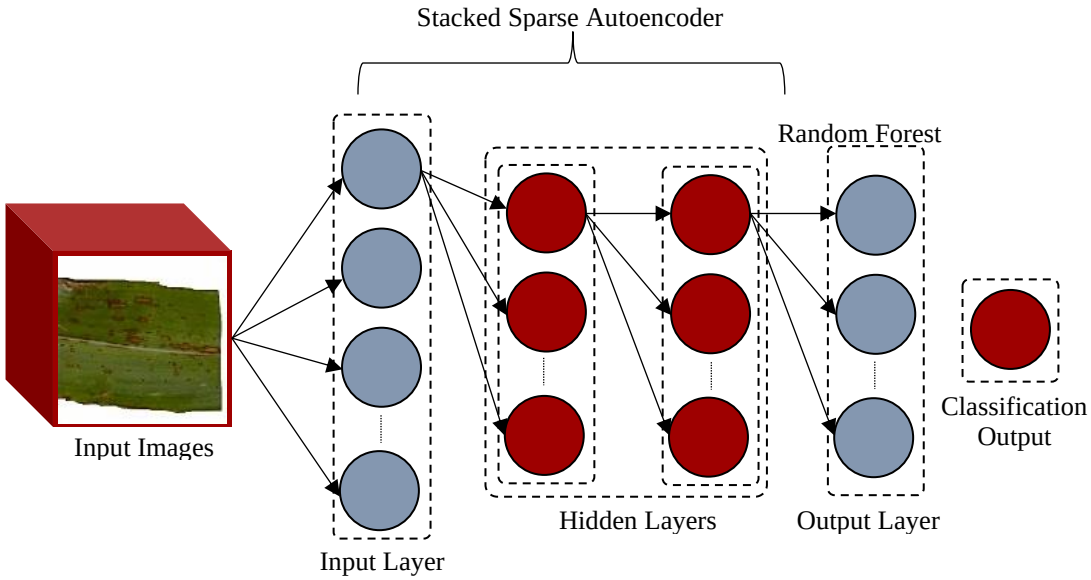

Fig. 2 Architecture of SSAE

$$Arg\text{Min}_{W,b,\widehat{W},\hat{b}} \sum_{i=1}^{N} \left| x_i - \left(\widehat{W}\left(\sigma(Wx_i - b)\right) + \hat{b}\right)\right|_2^2 + \delta \sum_{j=1}^{M} KL(\rho|\rho^j) \quad (9)$$

$$KL(\rho|\rho^j) = \rho log\frac{\rho}{\rho^j} + (1-\rho) log\frac{1-\rho}{1-\rho^j} \quad (10)$$

From the expression, M represents the dimension of the HL, $and\ \delta$ indicates the balancing parameter between reconstruction and sparsity. The term $(\rho|\rho^j)$, called the Kullback-Leibler equation, represents deviation in two Bernoulli distributions, which have ρ and ρ^j probabilities. The sparsity was minimalized once ρ^j is closer to ρ for all the j^{th} hidden neurons. SAE learned the low-level feature from image patches of vertebrae. However, lower-level feature learning is insufficient because of differences in the appearance of vertebrae. On the other hand, based on human perception, abstract high-feature was more powerful to CT images' inhomogeneity. Depending on low-level feature representation, a study used SSAE for higher-level feature learning. The stacking of SAE, called SSAE, contracts deep hierarchy. The SAE is stacked to present the lower-level output layer of SAE as the input layer for higher-level SAEs to study abstract higher-level features from the input image patch.

3.4. Hyperparameter Tuning using RSA

In the last stage, the TSA adjusts the hyperparameter values of the SSAE method. RSA is simulated by the predatory approach of alligators, further precise by seeking for prey [24]. The heuristic is separated into 2 crocodile-hunting approaches: exploitation and exploration. The crocodile behaviour is fragmented into two classes of exploring strategy: (a) belly walking and (b) high walking. Hunting collaboration and coordination are considered. Between the exploitation and exploration foraging phases, the RSA method was modified. Based on these methodologies, the overall amount of iterations was given in the following.

The exploration stage met dual conditions: belly-walking $\left(t > \frac{T}{4} \text{ and } t \le \frac{2T}{4}\right)$ and high walking $\left(t \le \frac{T}{4}\right)$. Furthermore, the exploitation phase was conditioned by hunting cooperation $\left(t > \frac{3T}{4} \text{ and } t \le T\right)$ and hunting coordination $\left(t > \frac{2T}{4} \text{ and } t \le \frac{3T}{4}\right)$. On real-time optimization problems, this new metaheuristic has previously accomplished positive prospects.

The stage-by-stage method of RSA is given below:

The RSA method can be performed by starting with a solution selected randomly and producing it as

$$y_{(i,j)} = rand \times (UB - LB) + LB, \quad j = 1,2,\cdots,n \quad (11)$$

In the encircling stage, the location update formula can be depicted as:

$$y_{(i,j)}(t+1) = \begin{cases} Best_{j_1}(t) \times -\eta_1(t) \times \beta_1 - R_{1(i,j)} \times rand, & t \le \frac{T}{4} \\ Best_{j_1}(t) \times x_{(r_1,j)} \times ES_1(t) \times rand, & t > \frac{T}{4} \text{ and } t \le \frac{2T}{4} \end{cases} \quad (12)$$

The prior optimal solution accomplished is $st_{j_1}(t)$, whereas $rand$ indicates a random integer within [0,1]. Moreover, β_1 is a critical parameter which affected exploratory performance, while t and T reflect the prevailing and overall iteration numbers.

$ES\ (t)$ is the stochastic value amongst $[-2, 2]$ across entire iterations estimated with Eq. (13), $x_{(r_1,j)}$ denotes the arbitrary location of $i\text{-}th$ solutions and $R_{1(i,j)}$ is the reduced search region that is calculated by Eq. (15), r_1 shows the random selection within $[1N]$, N represents the overall amount of solutions, and Eq. (14) is used for calculating $\eta_{1(i,j)}$. It determines the hunting operator to the j^{th} location of the i^{th} solution.

$$ES_1(t) = 2 \times r_3 \times \left(1 - \frac{1}{T}\right) \quad (13)$$

$$\eta_{1(i,j)} = Best_{j_1}(t) \times P_{1(i,j)} \quad (14)$$

$$R_{1(i,j)} = \frac{Best_{j_1}(t) - x_{(r_2,j)}}{Best_{j_1}(t) + \epsilon} \quad (15)$$

Now e signifies small values, r_2 belongs to $[1, N]$, and r_3 indicates the arbitrary number within $[-1, 1]$. Eq. (16) was utilized to calculate the new solution In the RSA algorithm's exploitation stage.

$$y_{(i,j)}(t+1) = \begin{cases} Best_{j_1}(t) \times P_{1(i,j)} \times rand, & t > \frac{2T}{4} \text{ and } t \le \frac{3T}{4} \\ Best_{j_1}(t) - \eta_{1(i,j)}(t) \times \epsilon - R_{1(i,j)} \times rand, & t > \frac{3T}{4} \text{ and } t \le T \end{cases} \quad (16)$$

Where $P_{1(i,j)}(t)$ represents the discrepancy of percentage defined using Eq. (17) amongst $the\ j^{th}$ location of the better one and the j^{th} location of the existing one.

$$P_{1(i,j)}(t) = \alpha + \frac{x_{(i,j)} - M_1(\text{x}_\text{i})}{Best_{j_1}(t) \times (UB_j - LB_j) + \epsilon} \quad (17)$$

In Eq. (17), α, another variable with a fixed value of 0.1, was utilized for restraining the exploration precision, and $M(x)$ is computed as:

$$M_1(\text{x}_\text{i}) = \frac{1}{n}\sum_{j=1}^{n} x_{(i,j)} \quad (18)$$

The pseudocode of RSA is given below.

Algorithm 1: Pseudocode of RSA

Initializes the parameter randomly$\alpha, \beta_1, \epsilon$, and so on.
Setup the candidate solution: $X = x_{(i,j)}, \quad i = 1,2,\cdots,N, j = 1,2,\cdots,N$
While $(t < T)$ do
Define the objective function for potential solutions (X).
choose the better solution yet
Upgrade the ES by Eq. (13),
RSA's beginning
For $(i = 1\ to\ N)do$ for $(j = 1\ to\ n)do$
Update the value of η, R, P utilizing Eqs. (14), (15), and (16), correspondingly
If $\left(t \le \frac{T}{4}\right)$ then
$y_{(i,j)}(t+1) = Best_{j_1} \times -\eta_{1(i,j)}(t) \times \beta_1 - R_{1(i,j)} \times rand,$
Else if $\left(t > \frac{T}{4} \text{ and } t \le \frac{2T}{4}\right)$ then
$y_{(i,j)}(t+1) = Best_{j_1} \times \chi_{(r_1,j)} \times ES_1(t) \times \text{rand},$
Else if $\left(t > \frac{2T}{4} \text{ and } t \le \frac{3T}{4}\right)$ then
$y_{(i,j)}(t+1) = Best_{j_1}(t) \times P_{1(i,j)} \times rand,$
Else
$y_{(i,j)}(t+1) = Best_{j_1}(t) - \eta_{1(i,j)}(t) \times e - R_{1(i,j)} \times rand,$
End if
End for
End for
$t = t + 1$
End while
Return the better solution (Best(X))

The RSA approach grows a fitness function (FF) to make superior results. It controls a positive value to exemplify the best solution for the candidate's performance. Here, the lessened classifier error rate is supposed to be FF, as written in Eq. (19).

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{no.of\ misclassified\ instances}{Total\ no.of\ instances} \times 100 \qquad (19)$$

4. Results and Discussion

The rice plant leaf disease recognition outputs of the ARPDD-RSADL method are tested on the dataset [25], encompassing 115 samples and 3 classes, as illustrated in Table 1. Fig. 3 denotes the instance imageries.

Table 1. Dataset description

Class	No. of Samples
Bacterial Leaf Blight	40
Brown Spot	37
Leaf Smut	38
Total Samples	**115**

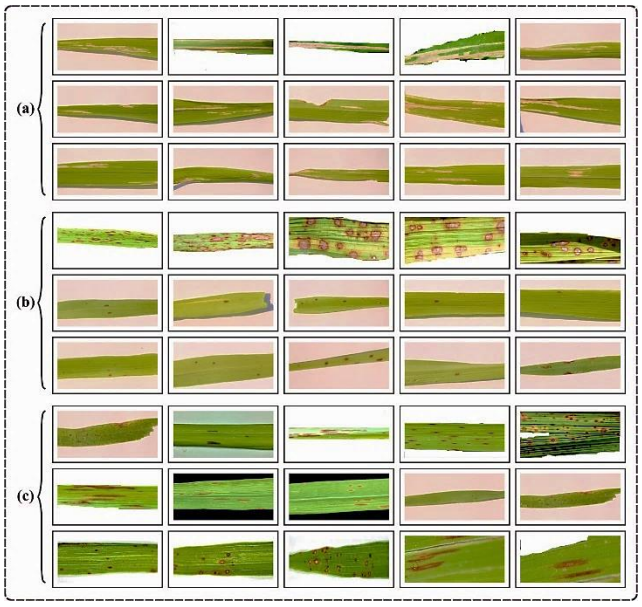

Fig. 3 Sample Images a) Bacterial Leaf Blight b) Brown Spot c) Leaf Smut

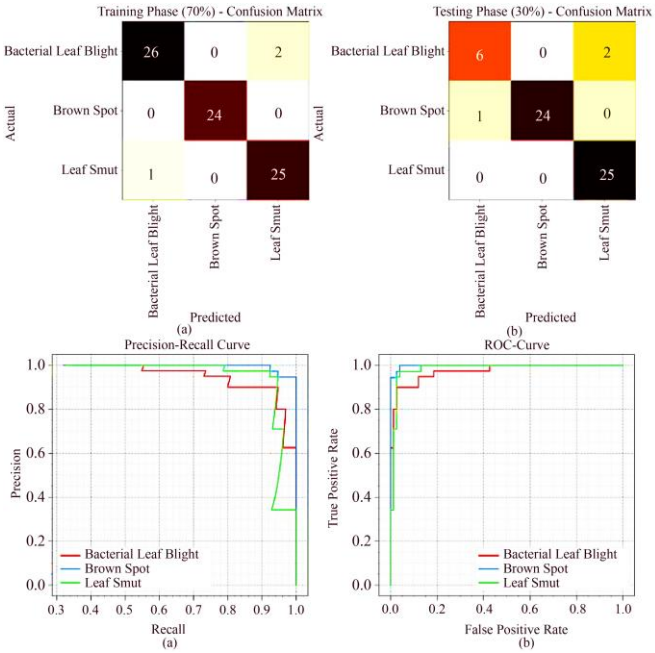

Fig. 4 Classifier output of (a-b) Confusion matrix, Curves of (c) PR, and (d) ROC

Fig. 4 specifies the classifier outputs of the ARPDD-RSADL method on the test dataset. Fig. 4a specifies the confusion matrix portrayed by the ARPDD-RSADL method on 70% of TRP.

The figure portrayed that the ARPDD-RSADL approach detected 28 samples on BLB, 24 samples on BS, and 25 samples on LS. As well, Fig. 4b portrays the confusion matrix presented by the ARPDD-RSADL approach on 30% of TSP.

Table 2. Rice plant disease detection outcome of ARPDD-RSADL methodology on 70:30 of TRP/TSP

Class	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{score}
Training Phase (70%)					
Bacterial Leaf Blight	96.25	96.55	93.33	98.00	94.92
Brown Spot	100.00	100.00	100.00	100.00	100.00
Leaf Smut	96.25	92.59	96.15	96.30	94.34
Average	**97.50**	**96.38**	**96.50**	**98.10**	**96.42**
Testing Phase (30%)					
Bacterial Leaf Blight	85.71	85.71	60.00	96.00	70.59
Brown Spot	94.29	100.00	84.62	100.00	91.67
Leaf Smut	85.71	70.59	100.00	78.26	82.76
Average	**88.57**	**85.43**	**81.54**	**91.42**	**81.67**

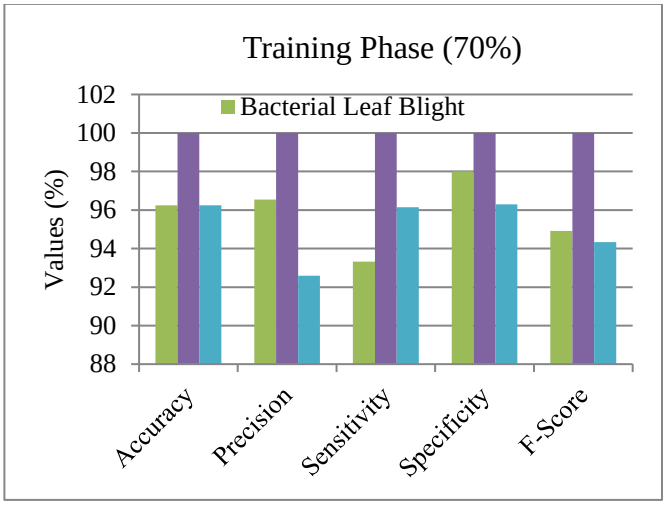

Fig. 5 Rice plant disease detection output of ARPDD-RSADL method on 70% of TRP

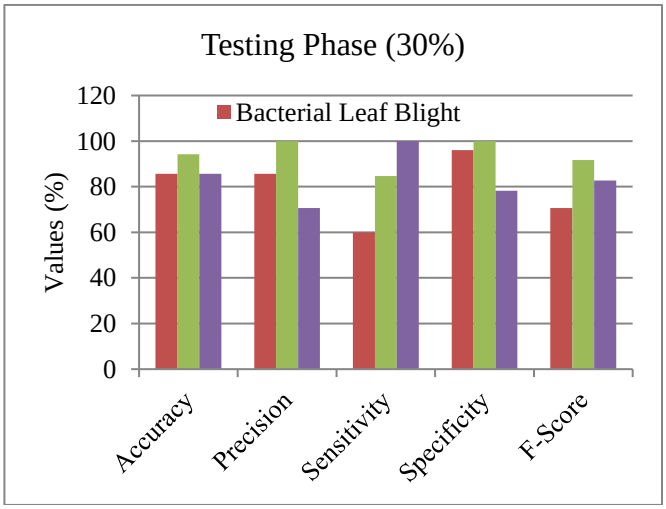

Fig. 6 Rice plant disease detection output of ARPDD-RSADL method on 30% of TSP

The outputs pointed out that the ARPDD-RSADL technique has detected 6 samples on BLB, 11 samples on BS, and 12 samples on LS. Similarly, Fig. 4c illustrates the PR examination of the ARPDD-RSADL technique. The figure specifies that the ARPDD-RSADL method has gained maximal PR performance in the overall class. Lastly, Fig. 4d shows the ROC examination of the ARPDD-RSADL method. The figure portrayed that the ARPDD-RSADL method has gained maximum ROC values on different class labels.

In Table 2, the comprehensive rice plant disease detection results are given. Fig. 5 offers detailed disease detection and classification outputs of the ARPDD-RSADL method provided under 70% of TRP. The figure indicates that the ARPDD-RSADL method reaches effectual detection under distinct classes. On bacterial leaf blight, the ARPDD-RSADL technique obtains average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 96.25%, 96.55%, 93.33%, 98%, and 94.92% respectively. Also, on the brown spot, the ARPDD-RSADL approach gains average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 100%, 100%, 100%, 100%, and 100% respectively. In addition, on leaf smut, the ARPDD-RSADL method obtains an average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 96.25%, 92.59%, 96.15%, 96.30%, and 94.34% correspondingly.

Fig. 6 offers detailed disease detection and classification results of the ARPDD-RSADL method, which is provided under 30% of TSP. The figure indicates that the ARPDD-RSADL approach reaches effectual detection under distinct classes. On bacterial leaf blight, the ARPDD-RSADL technique obtains average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 85.71%, 85.71%, 60%, 96%, and 70.59% correspondingly. Similarly, on the brown spot, the ARPDD-RSADL technique obtains an average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 94.29%, 100%, 84.62%, 100%, and 91.67% correspondingly. Moreover, on leaf smut, the ARPDD-RSADL method obtains an average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 85.71%, 70.59%, 100%, 78.26%, and 82.76% correspondingly.

The average detection results of the ARPDD-RSADL model are given in Fig. 7. The result specifies that the ARPDD-RSADL model recognizes the diseases proficiently. As a sample, on 70% of TRP, the ARPDD-RSADL approach offers average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 97.50%, 96.38%, 96.50%, 98.10%, and 96.42% respectively. Additionally, on 30% of TSP, the ARPDD-RSADL approach offers average $accu_y$, $prec_n$, $sens_y$, $spec_y$, and F_{score} of 88.57%, 85.43%, 81.54%, 91.42%, and 81.67% correspondingly.

Fig. 8 examines the ARPDD-RSADL method's accuracy in the process of training and validation of the test data. The

output portrayed the ARPDD-RSADL method as having greater accuracy values over greater epochs. Additionally, the greater validation accuracy over training portrayed that the ARPDD-RSADL approach learns productively on the test data.

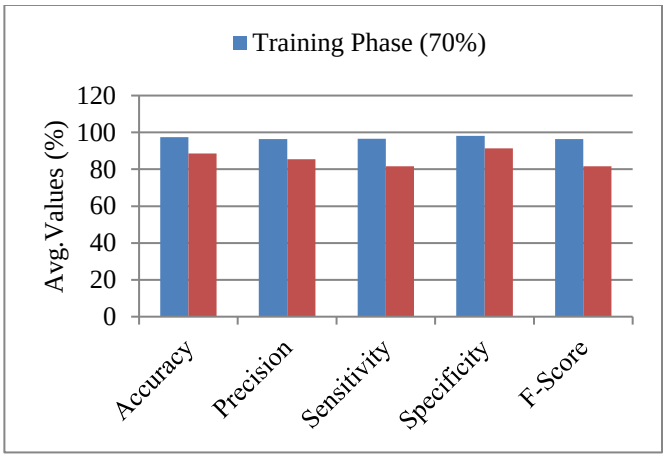

Fig. 7 Average output of ARPDD-RSADL methodology on 70:30 of TRP/TSP

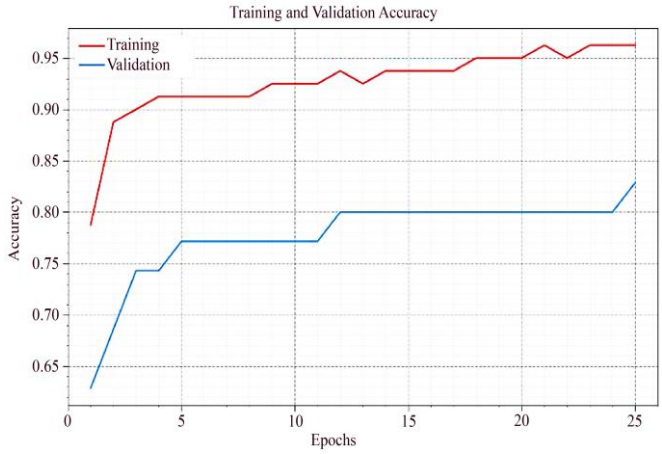

Fig. 8 Accuracy curve of the ARPDD-RSADL methodology

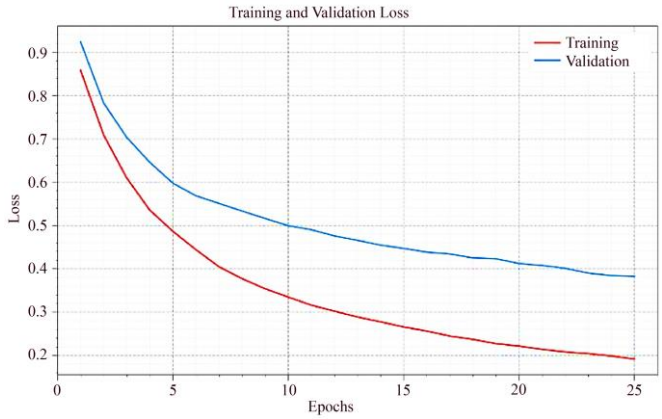

Fig. 9 Loss curve of the ARPDD-RSADL method

Table 3. Relative output of ARPDD-RSADL method with other approaches

Approaches	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	F_{score}
DNN-JOA	94.25	81.24	83.70	94.04	88.74
DNN	90.00	74.91	73.46	89.42	81.51
DAE	86.04	67.58	68.02	87.18	77.03
ANN	80.01	60.87	63.30	81.58	68.31
CNN	94.00	94.00	94.00	94.00	94.00
SIFT-SVM	91.10	86.66	86.66	90.03	86.66
SIFT-KNN	93.33	90.60	90.00	92.34	90.14
ARPDD-RSADL	97.50	98.10	96.38	96.50	96.42

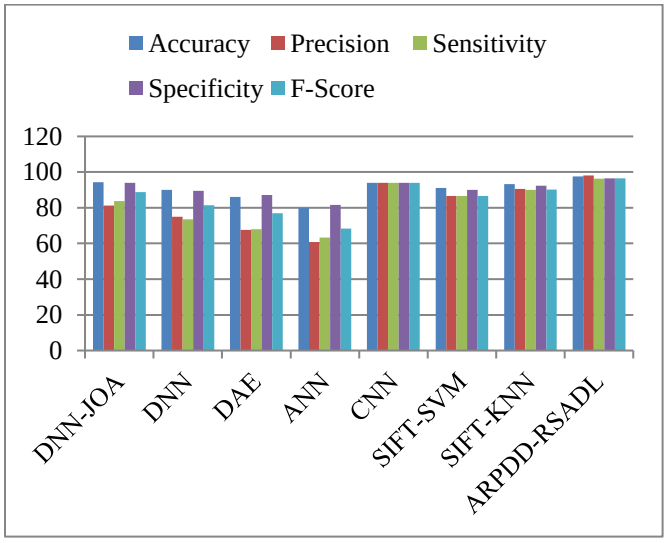

Fig. 10 Relative output of ARPDD-RSADL method with other approaches

Fig. 9 examines the ARPDD-RSADL methodology's loss in the training and validation depicted on the test data in Fig. 9. The output highlights that the ARPDD-RSADL methodology attains greater training and validation loss values. The ARPDD-RSADL model learned effectually on test data.

Table 3 and Fig. 10 offer a broad output of the ARPDD-RSADL method [26, 27]. The figure demonstrates the improvements of the ARPDD-RSADL method over other ones. Based on $accu_y$, the ARPDD-RSADL technique reaches an increased value of 97.50% while the DNN-JOA, DNN, DAE, ANN, CNN, SIFT-SVM, and SIFT-KNN models accomplish decreased $accu_y$ values of 94.25%, 90%, 86.04%, 80.01%, 94%, 91.10%, and 93.33% respectively.

Table 4. CT output of ARPDD-RSADL method with other approaches

Approaches	Computational time (sec)
DNN-JOA	0.29
DNN	0.26
DAE	0.27
ANN	0.25
CNN	0.25
SIFT-SVM	0.29
SIFT-KNN	0.27
ARPDD-RSADL	0.19

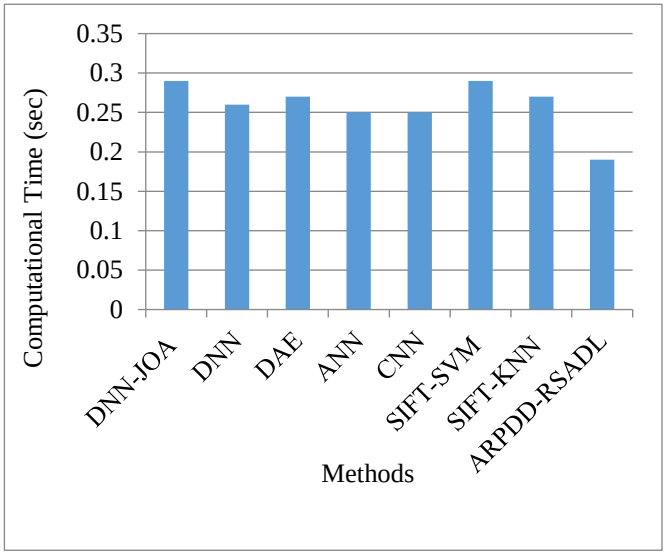

Fig. 11 CT outcome of ARPDD-RSADL method with other methods

Concurrently, based on $prec_n$, the ARPDD-RSADL approach reaches an increased value of 98.10% while the DNN-JOA, DNN, DAE, ANN, CNN, SIFT-SVM, and SIFT-KNN methods accomplish decreased $prec_n$ values of 81.24%, 74.91%, 67.58%, 60.87%, 94%, 86.66%, and 90.60% respectively. Eventually, based on $sens_y$, the ARPDD-RSADL technique reaches an increased value of 96.38% while the DNN-JOA, DNN, DAE, ANN, CNN, SIFT-SVM, and SIFT-KNN methods accomplish decreased $sens_y$ values of 83.70%, 73.46%, 68.02%, 63.30%, 94%, 86.66%, and 90%. Eventually, based on $spec_y$, the ARPDD-RSADL method reaches an increased value of 96.50% while the DNN-JOA, DNN, DAE, ANN, CNN, SIFT-SVM, and SIFT-KNN methods accomplish decreased $spec_y$ values of 94.04%, 89.42%, 87.18%, 81.58%, 94%, 90.03%, and 92.34% correspondingly.

At the final stage, the Computation Time (CT) investigation of the ARPDD-RSADL model is related with current approaches made in Table 4 and Fig. 11.

The result highlighted that the ARPDD-RSADL technique reaches a minimal CT of 0.19s, whereas the DNN-JOA, DNN, DAE, ANN, CNN, SIFT-SVM, and SIFT-KNN models obtain increased CT of 0.29s, 0.26s, 0.27s, 0.25s, 0.25s, 0.29s, and 0.27s subsequently. Hence, the ARPDD-RSADL technique can be implemented for robust rice plant ailment detection.

5. Conclusion

In this study, a new ARPDD-RSADL methodology for precisely classifying and recognizing the leaf diseases of the rice plant is designed. The ARPDD-RSADL methodology exploits the concept of CV and DL models with a hyperparameter tuning strategy. To achieve this, the ARPDD-RSADL method encompasses several processes: UM-based noise removal, contrast enhancement, SE-ResNet feature extraction, SSAE classification, and RSA-based hyperparameter tuning.

The construction of RSA helps to adjust the parameters associated with the SSAE method optimally, and it assists in attaining advanced accuracy in detection. A widespread experimentation assessment was achieved to validate the effectual rice plant ailment classification results of the ARPDD-RSADL model. The simulation values portrayed the authority of the ARPDD-RSADL model in terms of different evaluation measures.

References

[1] Santosh Kumar Upadhyay, and Avadhesh Kumar, "A Novel Approach for Rice Plant Diseases Classification with Deep Convolutional Neural Network," *International Journal of Information Technology*, vol. 14, no. 1, pp. 185-199, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] K. Kishore Kumar, and E. Kannan, "Detection of Rice Plant Disease Using AdaBoostSVM Classifier," *Agronomy Journal*, vol. 114, no. 4, pp. 2213-2229, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Wan-Jie Liang et al., "Rice Blast Disease Recognition Using a Deep Convolutional Neural Network," *Scientific Reports*, vol. 9, pp. 1-10, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Norhalina Senan et al., "An Efficient Convolutional Neural Network for Paddy Leaf Disease and Pest Classification," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, pp. 116-122, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ghulam Gilanie et al., "Ricenet: Convolutional Neural Networks-based Model to Classify Pakistani Grown Rice Seed Types," *Multimedia Systems*, vol. 27, no. 5, pp. 867-875, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Shuai Feng et al., "A Deep Convolutional Neural Network-Based Wavelength Selection Method for Spectral Characteristics of Rice Blast Disease," *Computers and Electronics in Agriculture*, vol. 199, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] N.V. Raja Reddy Goluguri, K. Suganya Devi, and P. Srinivasan, "Rice-net: An Efficient Artificial Fish Swarm Optimization Applied Deep Convolutional Neural Network Model for Identifying the Oryza Sativa Diseases," *Neural Computing and Applications*, vol. 33, no. 11, pp. 5869-5884, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Narendra Pal Singh Rathore, and Lalji Prasad, "Automatic Rice Plant Disease Recognition and Identification Using Convolutional Neural Network," *Journal of Critical Reviews*, vol. 7, no. 15, pp. 6076-6086, 2020. [Google Scholar] [Publisher Link]

[9] Murat Koklu, Ilkay Cinar, and Yavuz Selim Taspinar, "Classification of Rice Varieties with Deep Learning Methods," *Computers and Electronics in Agriculture*, vol. 187, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Manoj Agrawal, and Shweta Agrawal, "Rice Plant Diseases Detection and Classification Using Deep Learning Models: A Systematic Review," *Journal of Critical Reviews*, vol. 7, no. 11, pp. 4376-4390, 2020. [Google Scholar] [Publisher Link]

[11] Zhao Zhang et al., "A High-Quality Rice Leaf Disease Image Data Augmentation Method based on a Dual GAN," *IEEE Access*, vol. 11, pp. 21176-21191, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Shweta Lamba, Anupam Baliyan, and Vinay Kukreja, "A Novel GCL Hybrid Classification Model for Paddy Diseases," *International Journal of Information Technology*, vol. 15, pp. 1127-1136, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] L.B Zahro, Ika Hesti Agustin, and Zainur Rasyid Ridlo, *Classification of Disease in Rice Plant Leaves Using the Method Convolutional Neural Networks*, Proceedings of the 1st International Conference on Neural Networks and Machine Learning 2022 (ICONNSMAL 2022), Springer Nature, vol. 177, pp. 195-216, 2023. [Google Scholar] [Publisher Link]

[14] Yibin Wang, Haifeng Wang, and Zhaohua Peng, "Rice Diseases Detection and Classification Using Attention based Neural Network and Bayesian Optimization," *Expert Systems with Applications*, vol. 178, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Minlan Jiang et al., "Rice Disease Identification Method based on Attention Mechanism and Deep Dense Network," *Electronics*, vol. 12, no. 3, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] S. Senthil Pandi et al., "Rice Plant Disease Classification Using Dilated Convolutional Neural Network with Global Average Pooling," *Ecological Modelling*, vol. 474, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Shankarnarayanan Nalini et al., "Paddy Leaf Disease Detection Using an Optimized Deep Neural Network," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 1117-1128, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Muhammad Hanif Tunio et al., "Identification and Classification of Rice Plant Disease Using Hybrid Transfer Learning," In *2021 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 525-529, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Xiaoqi Wang et al., "ECA-ConvNeXt: A Rice Leaf Disease Identification Model Based on ConvNeXt," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6234-6242, 2023. [Google Scholar] [Publisher Link]

[20] S. Ramesh, and D. Vydeki "Recognition and Classification of Paddy Leaf Diseases Using Optimized Deep Neural Network with Jaya Algorithm," *Information Processing in Agriculture*, vol. 7, no. 2, pp.249-260, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Vipul Narayan et al., "Enhance-Net: An Approach to Boost the Performance of Deep Learning Model Based on Real-Time Medical Images," *Journal of Sensors*, vol. 2023, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Tie Chen et al., "A Non-Intrusive Load Monitoring Method based on Feature Fusion and SE-ResNet," *Electronics*, vol. 12, no. 8, pp. 1-14, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Syed Furqan Qadri et al., "SVseg: Stacked Sparse Autoencoder-Based Patch Classification Modeling for Vertebrae Segmentation," *Mathematics*, vol. 10, no. 5, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Jithendra Thandra, and Shaik Sharief Basha, "A Hybridized Machine Learning Approach for Predicting COVID-19 Using Adaptive Neuro-Fuzzy Inference System and Reptile Search Algorithm," *Diagnostics*, vol. 13, no. 9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Harshadkumar B. Prajapati, Jitesh Shah, and Vipul K. Dabhi, "Detection and Classification of Rice Plant Diseases," *Intelligent Decision Technologies*, vol. 11, no. 3, pp. 357-373, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[26] T. Nagarathinam, and K. Rameshkumar, "A Survey on Cluster Analysis Techniques for Plant Disease Diagnosis," *SSRG International Journal of Computer Science and Engineering*, vol. 3, no. 6, pp. 11-17, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[27] R.P. Narmadha, N. Sengottaiyan, and R.J. Kavitha, "Deep Transfer Learning based Rice Plant Disease Detection Model," *Intelligent Automation and Soft Computing*, vol. 31, no. 2, pp. 1257-1271, 2022. [CrossRef] [Google Scholar] [Publisher Link]