A Hybrid AI-Based Approach for Early and Accurate Rice Disease Detection

Noorishta Hashmi[1,*], Mohammad Haroon[1]
[1]CSE, Integral University, Lucknow, 226022, India
Emails: noorishtahashmi@gmail.com; mharron@iul.ac.in

Abstract

Rice plant disease detection is crucial in agriculture to prevent crop loss and enhance productivity. Traditional manual inspection methods often lead to inaccuracies, delays in diagnosis, and excessive pesticide use. To address these challenges, this study proposes an Artificial Layered Fuzzy Neural Network-based African Vulture Optimization (ALFNN-AVO) algorithm for early and accurate detection of rice plant diseases. The proposed framework integrates multiple advanced techniques, including Cross Fusion former (CF former) for feature extraction, Squeeze Excitation (SE) fusion for enhancing feature representation, and Spatial Fuzzy C-Means (SPFCM) for precise segmentation of affected plant regions. Furthermore, an Artificial Layered Depth Separable Neural Network (ALDSNN) is employed for multi-class classification of rice plant diseases. The Differential Bitwise African Vultures Optimization Algorithm (DBAVOA) is introduced to optimize the hyperparameters, ensuring improved convergence and classification performance. Experimental results validate the efficiency of the proposed model, achieving an accuracy of 98.87% and an execution time of 0.09 minutes, outperforming existing methodologies. The findings demonstrate that the proposed framework offers a reliable and computationally efficient solution for real-time rice plant disease detection, contributing to sustainable agricultural practices.

Keywords: Rice plant disease detection; Cross Fusion former; Squeeze Excitation; Spatial Fuzzy C-Means; Artificial Layered Depth Separable Neural Network; African Vulture Optimization

1. Introduction

Rice (Oryza sativa) is a staple food crop globally, providing sustenance to nearly half of the world's population [1]. The cultivation and yield of rice crops are highly susceptible to various bacterial and fungal diseases, including leaf smut, brown spots, and bacterial leaf blight, which significantly reduce productivity [2]. Traditional methods of disease detection rely on manual inspection by farmers, often leading to misdiagnosis, excessive pesticide application, and yield losses [3]. The need for early and accurate disease identification has driven advancements in computational methods, particularly machine learning and deep learning techniques [4].

Rice is a primary food source in several countries, including India, China, and Pakistan, where it forms a significant part of the daily diet [5]. Different rice varieties are cultivated worldwide, valued for their rich nutritional content, minerals, and supplements [6]. The successful growth of rice plants depends on factors such as topography, seed selection, soil characteristics, biological factors, irrigation, and climate conditions. However, bacterial and fungal infections cause nearly 10-15% of rice plant damage among 70% of the crop yield [2]. Farmers often rely on pesticide applications to mitigate these diseases and protect crop productivity [7].

Traditional plant disease detection relied on farmers' experiences and standards. Manual identification is time-consuming and inaccurate, resulting in improper treatments, increased pesticide use, and poor crop quality [8].

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

Delayed diagnosis might cost a lot and reduce productivity. Early rice plant disease diagnosis and classification are needed to solve these concerns [4]. Multiple RGB image-based illness detection computer-aided systems have been developed over time. Automating rice plant disease detection and classification with machine learning and deep learning has shown great promise [9].

Conventional models are effective but often misclassify and misidentify impacted regions. MLP and SVM neural network models are used for feature extraction and classification [1]. Image quality and classification performance improve with filtering, augmentation, and normalisation. Many models misinterpret leaf picture shape, texture, pattern, edges, and colour. The vanishing gradient problem causes superfluous features, poor feature utilisation, and overfitting in deep learning models [10].

This study presents a machine learning-based method for early and accurate rice plant disease detection and classification to tackle these problems. The proposed system uses advanced neural network designs and optimisation to improve disease diagnosis and classification while reducing computational cost and overfitting.

1.1 Contribution

The principal contributions utilised in the identification and classification of rice plant diseases are delineated as follows:

Goals of the suggested research article: An effective Artificial Layered Fuzzy Neural Network based African Vulture Optimisation (ALFNN-AVO) algorithm is suggested for the precise and timely identification and categorisation of rice plant diseases.

Data acquisition and preprocessing: Images relating to rice plant diseases are obtained from several data sources, including the rice plant dataset, rice disease picture dataset, and rice leaf disease dataset. The gathered photos undergo preprocessing through image improvement, scaling, segmentation, noise reduction, and data augmentation to improve quality, diminish noise, boost contrast, and mitigate overfitting concerns.

Feature extraction and fusion: The characteristics derived from the pre-processed images are extracted using two modalities, namely self-attention and cross-attention methods, which facilitate the extraction of additional features. The Cross Fusion former (CF former) model facilitates feature extraction, while the Squeeze Excitation model integrates features from two modalities. The segmentation technique involves delineating healthy and damaged lesions in rice plant pictures based on the retrieved feature information. This segmentation job employs Spatial Fuzzy C-Means (SPFCM) to capture spatial information within the super pixels.

Detection and classification of rice plant diseases: This phase proposes an Artificial Layered Depth Separable Neural Network (ALDSNN) model for the categorisation of features in multiclass images. The hyperparameters are adjusted to identify the best parameter values of the proposed model by the application of the Differential Bitwise African Vultures Optimisation Algorithm (DBAVOA).

Results of experimental validation: The validation findings are shown during experiments, revealing that the suggested model achieved an accuracy of 98.87% and an execution time of 0.09 minutes. Comprehensive analyses are undertaken to establish the superiority of the proposed model over existing models.

The structure of this suggested article is as follows: Section 2 contains literature reviews of the various current methodologies. Section 3 demonstrates the suggested techniques and elements utilised in this work, while Section 4 provides a summary of the experimental analysis and visual representations. Finally, section 3 of this study presents the conclusion and suggests further research.

2. Literature Survey

In the context of rice plant disease detection and classification, numerous research analyses were conducted previously. Here, machine learning, deep learning and optimization related approaches were taken as a main consideration of this research article. The advantages and limitations that occurred due to the existing analyses were mentioned in this section for designing an efficient model.

2.1 Literature Related to Rice Plant Disease Detection and Classification using Deep Learning Methods

A Deep Convolutional Neural Network (DCNN) model for rice plant disease detection and classification was developed by latif et al. [11]. This updated model classified image features into six types. This CNN model was

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

optimised for memory utilisation and parameters. This article used extensive analytics to predict improved performance.

Upadhyay and Kumar [12] classified rice plant diseases with a fully connected CNN model. Leaf photos were used to identify rice plant diseases by their forms and coverage areas. The thresholding-based approach removed noise and binarized images for classification. Comparative analysis showed better detection and categorisation.

Haridasan et al. [13] developed a rice plant disease detection deep learning model using computer vision. Computer vision classified crops precisely to protect them. Photos were used to locate damaged regions. Recognition boosted agriculture applications. Jiang et al. [14] built an AB-SE-DenseNet model to diagnose rice plant diseases. Depthwise separable convolution increased training speed and parameter use, while AB lowered parameter modification time. This classification approach improved disease detection with increased recognition accuracy. For the categorisation of rice leaf diseases, Ritharson et al. [10] created the DeepRice model, an effective deep learning model. The machine learning respiratory technique was used for image acquisition, and preprocessing techniques were used to eliminate undesired areas, noise, and blur. The purpose of the confusion matrix was to show which classifications were present in the dataset.

2.2 Literature Related to Rice Plant Disease Detection and Classification using Machine Learning Models

Kishore and Kannan [15] found rice plant diseases using AdaBoostSVM. After median filtering, DWT retrieved features from pre-processed images. Diseased lesion features improved categorisation in experiments. The hybrid Inception-ResNet-SVM of Chaudhari and Malathi predicted rice plant leaf damage. Grab-Cut improved agricultural rice plant images' quality and reliability [16]. The hybrid model performed superior segmentation and classification, according to this study.

Dubey and Choubey demonstrated an Adaptive Sunflower Optimized Artificial Neural Network (ASFO-ANN) model for automatic detection [17]. The optimization algorithm was used to tune the hyperparameters of the ANN model for attaining optimal values. The level-set segmentation model was utilized to identify the affected regions in the collected images.

Joko [18] implemented the ANN model in this article for the classification of rice leaf disease. The images were gathered from rice leaf disease dataset that was pre-processed by using a labelling approach. Finally, the rice disease types were identified based on the classification model and the experiments were conducted for enhancing the ability of the model. Bharanidharan et al. illustrated a Modified Lemurs Optimization Algorithm based K-Nearest Neighbour (MLOA-KNN) model to detect multiclass diseases [19]. The healthy and affected images were collected from the data source, the classifier model was utilized for the extraction of statistical, and box-cox transformed features.

2.3 Literature Related to Rice Plant Disease Detection and Classification using Optimization Based Model

Daniya and Vigneshwari suggested a Deep Residual Network (RHGSO-DRN) model based on Rider Henry gas Solubility Optimisation to detect and classify rice plant illnesses [6]. Deep fuzzy clustering was used to extract characteristics from pre-processed photos from the rice plant dataset, the data source for image collection. Better classification performance was used to distinguish between healthy and unhealthy classes. Dubey and Choubey developed an ASFO-SVM model to predict rice plant diseases [17]. The median filter removed noise from input photos during preprocessing. To categorise it as normal or abnormal, traits were removed. The experiments used numerous performance indicators to evaluate the prediction model.

Reddy et al. created a rice plant disease diagnosis PDICNet by identifying disease relationships [21]. The ResNet-50 model extracted colour and texture features from the pictures. After that, the enhanced optimisation algorithm was shrunk to select features. Abd et al. developed an Ant Colony Optimisation-based CNN (ACO-CNN) model to identify and classify diseased rice plants [22]. Significant indicators were used to identify performance, and rigorous examinations showed successful classification.

2.4 Limitations and Research Gap

Many effective rice plant disease detection and classification studies have been done. Early analysis failed to consistently detect and categorise rice plant infections due to restrictions. Previous techniques detected and categorised rice plant diseases using deep learning, AI, and ML [23]. Farmers' laborious classification and identification errors cut crop yields. Research articles with small datasets and complexity issues are severely limited. Missing segmentation hinders classification and lesion detection. Only a few features changed categorisation performance in recent decades

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

[24]. To overcome these limits, ALFNN-AVO is suggested to accurately identify and classify rice plant diseases. Our proposed paradigm efficiently detects plant health and disease early and accurately at all stages.

3. Proposed Methodology

Figure 1 shows the proposed architecture's accurate and early detection blocks. Images from rice plant, rice leaf disease, and rice disease picture datasets. Images, contrast, and overfitting increase with preprocessing methods such image improvement, scaling, noise reduction, and data augmentation. We use CFformer model to extract efficient features that collect information that is more meaningful from different modalities. Squeezing excitation-based fusion concatenates modalities' features, while SPFCM extracts superpixel spatial information to identify afflicted regions. ALDSNN classifies pictures using extracted features. The suggested model's hyperparameters are adjusted using DBAVOA model to eliminate local optimum worries and increase optimisation algorithm searching for classification performance.

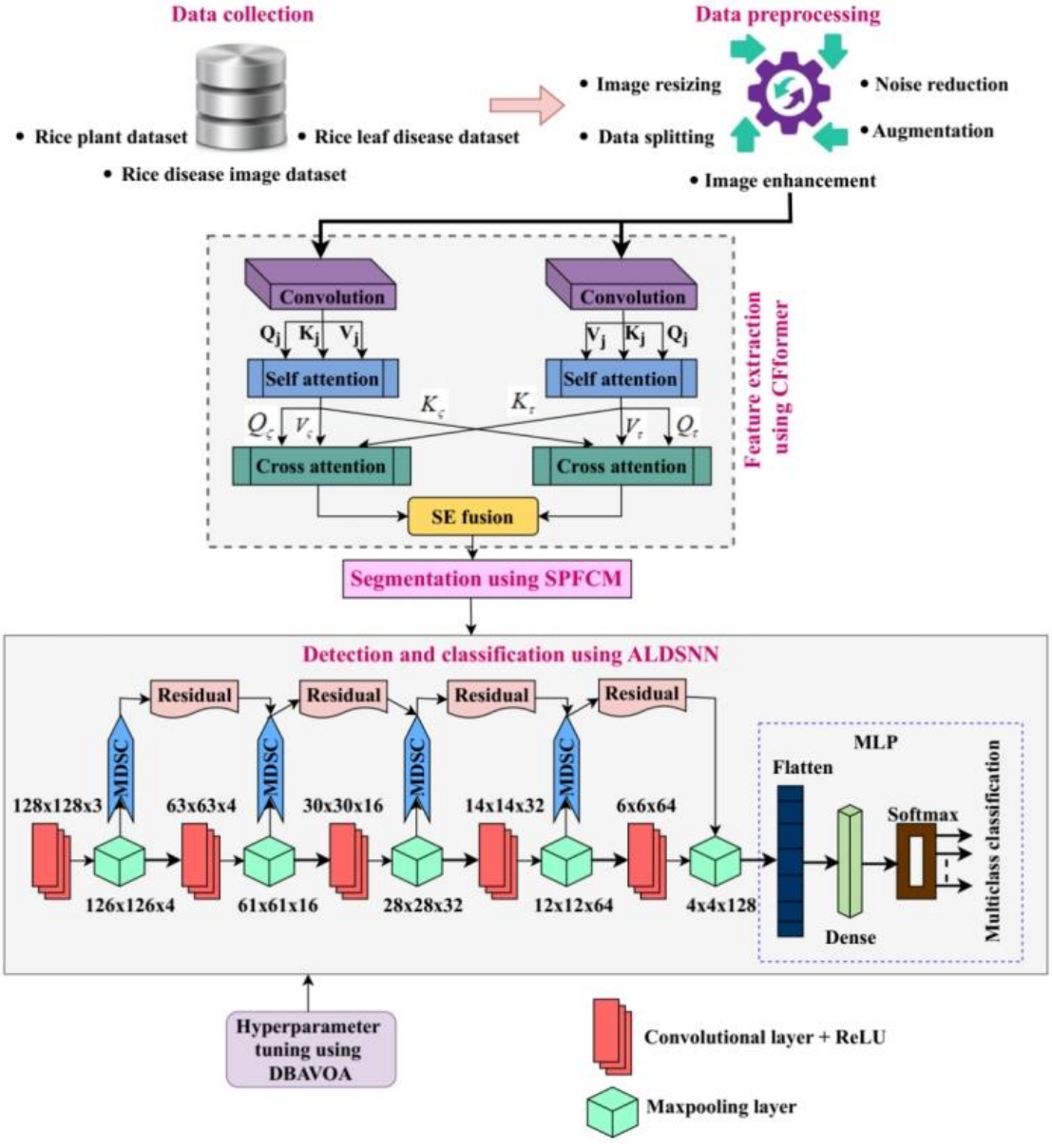


Figure 1. Proposed Model

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

3.1 Data Collection and Preprocessing

The data are gathered from a variety of rice plant leaf disease databases, featuring a variety of classes. The primary stages of the proposed model are image acquisition and preprocessing, which improve the model's suitability and image quality. To mitigate noise, enhance image contrast, resize images, and mitigate overfitting issues, the preprocessing and augmentation stages are implemented.

3.1.2 Data preprocessing

Image enhancement: Image enhancement approaches improve dataset quality, while augmentation methods increase dataset size. Images are enhanced and smoothed using edge-aware local contrast modulation. If input images include threshold values, the minimum intensity amplitude defines strong edges. Suppose enhancement and threshold are 0.5 and 0.15. The spectrum is centred by shifting the zero-frequency component with the Fourier transform and smoothed with the anisotropic diffusion filter.

Image resizing and Data splitting: For this proposed analysis, the gathered images from different sources are resized into 224×224 pixels. The pixel values in the images are scaled from 0 to 1 using a min-max scaling process, then the images are separated into training and testing sets with the proportion of 80:20 respectively.

Noise reduction: Each image presented in all datasets has noisy and error contents that are eliminated from it for the enhancement of detection accuracy and segmentation performance. The equation for noise elimination (N) from images is given as,

$$N = \frac{1}{T}\sum_{j=1}^{T}(C_j - E)(S_j - \eta) \quad (1)$$

From the above-mentioned equation, the terms such as T , C_j, E , S_j and η are represented as a combination of training and testing sets, training dataset, error of image, collected sample dataset and noise content in the images respectively.

Data augmentation: For enhancing the size of the dataset and minimizing overfitting issues, data augmentation approaches are introduced that generate minor changes to creating new images. For the augmentation purpose, the processes such as scale-in/scale-out (zooming), translation and rotation are utilized in this proposed research article. The original images are rotated into +15 to -15 degrees, the images are zoomed into 105-115% for both width and height and the images are shifted into -5 to +15.

3.2 Feature Extraction using Cross Fusion former (CF former) Model

For the feature extraction, the features are collected from different modalities such as self-attention and cross-attention transformer models. The convolutional layer is added to the transformer block as an input of the extractor that efficiently extracts features (shapes, colors, texture and edges) from the preprocessed image.

Convolutional layer: The transformer model is trained on rice photos to recognise illness patterns (filters) for image classification. For rice plant disease prediction, a neural layer identifies crucial elements in pre-processed images. The final convolution operation between filters and pre-processed pictures produces feature maps. In the feature map, the dimension of the filter ($filter$) is small compared to the preprocessed image (I). The elementwise multiplication of preprocessed image and filter are represented as a convolutional operation. The feature maps (F_{map}) is produced by the convolutional operation that is mathematically represented as,

$$F_{map} = I * filter \quad (2)$$

Self-attention block: In the self-attention block, there are three vectors such as query (Q), key (K) and value (V) are used as input [27]. The weights and values are randomly initialized to generate these three vectors and backpropagation is applied to change the weight values of the model. Backpropagation calculates and updates the loss values. The weights are multiplied with pre-processed input to get receptive vectors such as Q_j , K_j and V_j that

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

are represented in equations (3), (4) and (5) respectively. The input vector is 64×64 dimension, which helps to attain better results, and the dimension values are hyperparameterized.

$$Q_j = I_j * \omega_q \quad (3)$$

$$K_j = I_j * \omega_k \quad (4)$$

$$V_j = I_j * \omega_v \quad (5)$$

The attention score is evaluated after finding the receptive vectors that determine the importance of features. The scalar product of the key vector and query vector is used to calculate the attention score, and its mathematical formulation is given by,

$$A_{score} = Q_j * K_j \quad (6)$$

Key vector dimensions are 64 and square root is 8, so attention score is divided by key vector. Hyperparameter key = 64 determines stable gradient. The SoftMax function normalises the anticipated attention score. The normalised score outputs probabilities up to 1. Higher probability signifies rice illness class; multiply probability by value vector to get sum vector. Sum vector (weighted value vector) is calculated to find significant features and remove unnecessary ones. The self-attention layer outputs the sum vector to the feed forward network. Actual calculation is done in matrix format for rapid processing. The mathematical representation of the sum vector (S_{um}) is expressed as,

$$S_{um} = Soft\max\left(\frac{Q * K}{\sqrt{\delta k}}\right) \quad (7)$$

Here, the term δk is denoted as a dimension of the key.

Cross-attention block: The implementation of cross-attention is more suitable for the domain adaptation of rice plant images in the source and target domains. The mathematical representations of the cross-attention block in the target and source domain are computed as,

$$C\left(Q_{\varsigma}, K_{\tau}, V_{\varsigma}\right) = Soft\max\left(\frac{Q_{\varsigma} K_{\tau}^{T}}{\sqrt{\delta k}}\right) V_{\varsigma} \quad (8)$$

$$C\left(Q_{\tau}, K_{\varsigma}, V_{\tau}\right) = Soft\max\left(\frac{Q_{\tau} K_{\varsigma}^{T}}{\sqrt{\delta k}}\right) V_{\tau} \quad (9)$$

Here, the terms such as $Soft\max$, K^{T}, Q_{ς}, K_{τ}, V_{ς}, Q_{τ}, K_{ς} and V_{τ} are represented as activation function, transpose operation, query of source domain, key of target domain, value of source domain, query of target domain, key of source domain and value of target domain respectively. The Feed forward Neural Network (FNN) model is connected as an output layer of the cross-attention model that is constructed based on two fully connected layers and Gaussian Error Linear Unit (GeLU) activation function. The output of the FFN (F_{out}^{t}) model is represented as,

$$\overline{F}_{out}^{t} = Multi\left(L\left(\overline{F}_{out}^{t-1}\right)\right) + F_{out}^{t-1} \quad (10)$$

$$F_{out}^{t} = FNN\left(L\left(\overline{F}_{out}\right)\right) + \overline{F}_{out}^{t} \quad (11)$$

where L, F_{out}^{t-1} and $Multi$ are denoted as layer normalization, input and multi-head attention respectively.

Squeeze Excitation (SE) Fusion: This fusion model is designed based on the fully connected layers, activation functions and global average pooling blocks. The fusion module is basically divided into two components such as

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

squeeze and excitation operations. The size of the feature map is compressed into 1×1 features using special pooling layers; however, the channel dimension is not changed (fixed). The size of the input feature map is $c \times h \times w$ and the input set id assumed as, $X = \left[x_1, x_2, ..., x_g\right]$. The compression operation is mathematically computed as,

$$\rho_g = sqeeze\left(x_g\right) = \frac{1}{h \times w} \sum_{j=1}^{h} \sum_{i=1}^{w} x_g\left(j,i\right) \tag{12}$$

From the above equation, the squeeze operation and global information of feature maps are described as $sqeeze$ and ρ_g respectively. The excitation operation comprises a sigmoid function and fully connected layer. The input characteristic information is integrated using fully connected layer and mapping process is performed by using sigmoid activation function. The mapping relationship in the excitation operation is expressed as,

$$m = E\left(\rho, w\right) = \eta\left(f\left(\rho, w\right)\right) = \eta\left(w_2 \cdot \beta\left(w_1 \cdot \rho\right)\right) \tag{13}$$

Here, the terms such as η , β, E and w_2, w_1 are denoted as sigmoid activation function, ReLU activation function, excitation operation and weight parameters respectively. SE fusion module performs scale operation that multiples the learned weight of the channel features. Then the weights of the features are fused to form the fused feature for enhancing network performances and minimizing feature redundancy.

3.3 Segmentation Process

Segmenting rice plants uses coarse and fine feature extraction. 1) SPFCM captures superpixel spatial data. For segmentation, the super pixel spectral and texture membership function covers nearby neighbours. This stage produces coarse segmentation because it integrates local spatial information and calculates distance between cluster c. We use coarse and fine feature extraction to segment rice plants.

$$O_{function} = \sum_{i=1}^{\aleph} \sum_{l=1}^{P} \beta_l \, \vartheta_{il} \, \overline{n} - \left\| \varphi_l - \varpi_i \right\|^2 + \frac{\alpha}{A_C} \sum_{i=1}^{\aleph} \sum_{l=1}^{P} \vartheta_{il} \, \overline{m} - \left(\sum_{\tilde{f}_{C-} \in \overline{A}_{k-}} \beta_C \left\| \varphi_C - \varpi_i \right\|^2 \right) \tag{14}$$

where the terms such as $\aleph$, P , β_l , φ_l , ϑ_{il} , $\overline{A}_k$, A_C , $\|\cdot\|$, $\overline{m}$ and α are represented as number of clusters, number of predefined superpixel images, pixels in the superpixel, vector features in the superpixel, member of superpixel, adjacent set of superpixel, cardinality of adjacent set, Euclidean distance, regulates fuzziness and controlling parameter of superpixel neighbour effects respectively. The input cluster center is mathematically expressed as,

$$\varpi_i = \frac{\left(\sum_{l=1}^{P} \vartheta_{il}^{\overline{m}} \left(\beta_l \varphi_l + \frac{\alpha}{A_C} \sum_{\tilde{f}_{C-} \in \overline{A}_{k-}} \beta_C \, \varphi_C \right) \right)}{\left(\sum_{l=1}^{P} \vartheta_{il}^{\overline{m}} \left(\beta_l + \frac{\alpha}{A_C} \sum_{\tilde{f}_{C-} \in \overline{A}_{k-}} \beta_C \right) \right)} \tag{15}$$

Straight lines make membership management easy. In real-time applications, trapezoidal and triangular functions are used for their arithmetic efficiency and simple formula. Popular membership functions include s-type, p-type, generalised bell curve, Gaussian, trapezoidal, and triangular. Generalising the Gaussian distribution, fuzzy logic utilises the membership function as a degree of truth. The Gaussian membership function is,

$$\vartheta_{il}\left(y\right) = \exp\left(- \left(\frac{y - \kappa}{\lambda} \right) \right) \tag{16}$$

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

The grade of membership is quantified to the fuzzy set $\overline{F}$ and the fuzzy members are characteristics from 0 to 1. The value 0 represents y is not a member of fuzzy set and the value 1 is represents y is the member of fuzzy set. The constraints of the membership value are as follows:

$$0 \le \vartheta_{il} \le 1, for 1 \le i \le \aleph, 0 \le l \le P$$
$$0 < \sum_{l=1}^{P} \vartheta_{il} < P, for 1 \le i \le \aleph \quad (17)$$

Set the values for α, $\overline{m}$, $\aleph$ and maximum iteration as 0.2, 2, 2 and 100 respectively. The membership function is drawn in Figure 2.

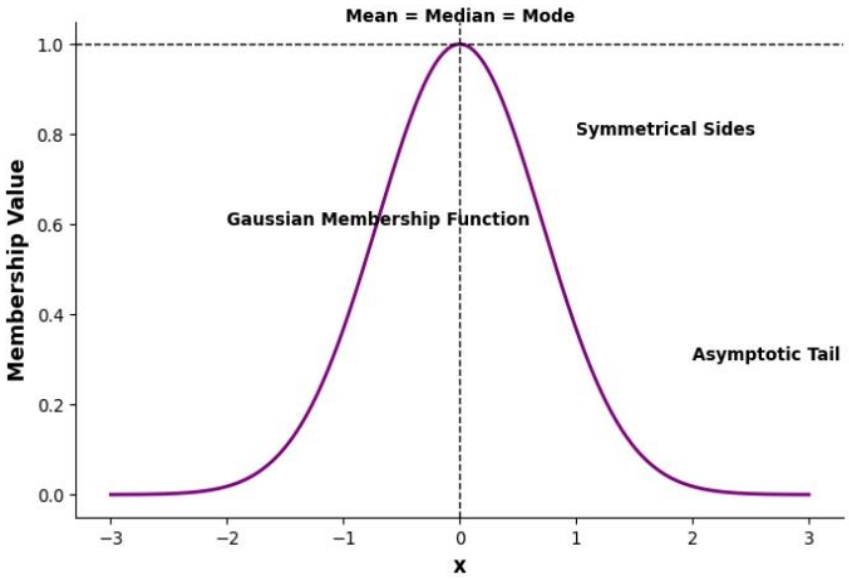


Figure 2. Gaussian membership function

3.4 Detection and Classification using Artificial Layered Depth Separable Neural Network (ALDSNN)

The detection and classification model uses MDSC, residual network, and deep neural network. Deep neural networks train better than other models, thus the detection and classification model use them. DSC and residual networks boost efficiency in this approach. Learning features and improving model performance is easy with these two computationally compatible modules.

3.4.1 Layered deep neural network

For designing layered DNN model, the different sizes and number of convolutional and pooling layers are and its performance is compared with other traditional models. Compared to other traditional models, the layered DNN model has higher training performances, and five sets of convolutional and poling layers are used for developing the proposed detection and classification model. The output of the convolutional operation is given by,

$$D(con(m,l)) = \left(\left[\frac{m_\omega - g_\omega}{\zeta} + 1 \right], \left[\frac{m_\hbar - g_\hbar}{\zeta} + 1 \right], g_\varsigma \right) \quad (18)$$

Here, the terms in the above-mentioned equation such as m_ω, g_ω, ζ, $m_\hbar$, $g_\hbar$ and g_ς are delineated as width of the filter, width of the convolutional layer, stride, height of the filter, height of the convolutional layer and channel of the convolutional layer respectively. The max pooling layer is used to reduce the dimension of the data in convolution

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

layer which has the dimension of the output is 63×63×4. The equation for calculating maxpooling output is represented as,

$$D(pool(m,l)) = \left(\left[\frac{m_\omega - g_\omega}{\zeta} + 1 \right], \left[\frac{m_\hbar - g_\hbar}{\zeta} + 1 \right], m_\varsigma \right) \quad (19)$$

The channel of the filter is denoted as m_ς and this output is taken as input to the second convolutional layer. The Rectified Linear Unit (ReLU) activation function is used at each stage along with the convolutional blocks and the mathematical representation is provided as,

$$\text{Re}\,LU(y) = Max(0, y) \quad (20)$$

3.4.2 Modified Depthwise separable convolutional layer (MDSC)

Separable convolution is also known as depthwise separable convolution that is efficient for the computational cost [26]. The depthwise separable convolution is designed by combining pointwise convolution and depthwise convolution. In this paper, the Modified depthwise detachable convolution is utilized, the 1×1 pointwise convolution is placed before applying depthwise convolutional block that makes the MDSC model more efficient. In addition, the amount of arithmetic operation is minimized when the generalization ability is persevering. The depthwise convolutional outputs are concatenated to find the resultant output. This approach helps to minimize the parameter and computation quantities, the total size of the weights is computed based on the equation as,

$$W = \varsigma \times N_\varsigma \times m_\hbar \times m_\omega \quad (21)$$

From the above-mentioned equation, the term such as ς and N_ς are delineated as number of channels and number of filters respectively. This MDSC model has the minimum number of parameters compared to the traditional convolutional layer and the parameter computation is expressed as,

$$W_{MDSC} = m_\hbar \times m_\omega \times \varsigma_{K-1} \times \varsigma_K \times 1 \times 1 \times N_\varsigma \quad (22)$$

The convolutional layer is represented as K. This MDSC makes the model more efficient and computationally easy that enhances performance. The architecture of the MDSC is illustrated in Figure 3.

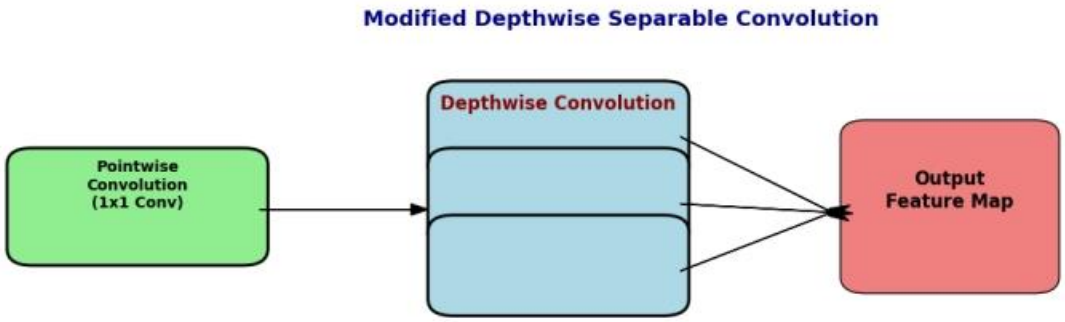


Figure 3. Architecture of MDSC

3.4.3 Residual network

The residual connection is also known as skipped connection that acts as a bypass of two networks. The residual connection integrates all features in the preceding layer with the next layer that empowers the features to enhance performance of the model. The residual connection is mathematically formulated as,

$$T(y) = R(y) + y \quad (23)$$

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

The true output value and residual learning of the layer are represented as $T(y)$ and $R(y)$ respectively. The efficient features are learned from the input images by using a residual connection that makes the model computationally efficient and enhances performance of the model. Figure 4 demonstrates the architecture of residual layers.

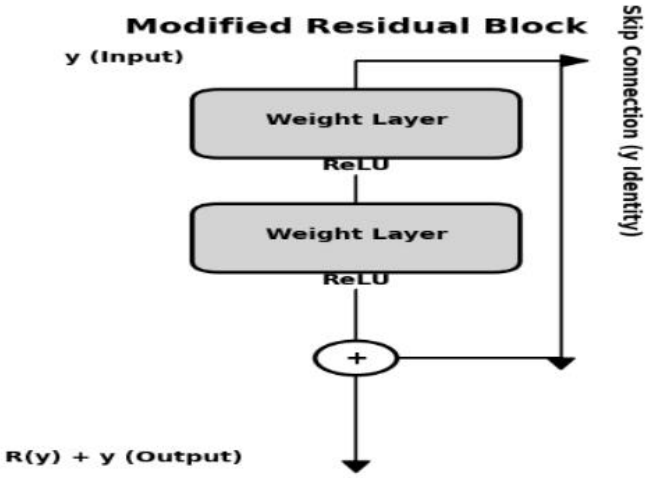


Figure 4. Architecture of residual layers

3.4.4 Multi-layer perceptron (MLP) for classification

After feature extraction, the learnt feature maps are flattened and transferred to the fully linked layer. This layer classified pre-processed images using extracted features. Flatten layer, dense layer, and SoftMax classifier comprise this classification model.

Flatten layer: Flatten layer converts pooled three-dimensional feature map to column-wise feature map. This flattens layer observes numerous pooled feature maps from series-connected pooling layers. The pooled feature map's one-by-one column arrangement helps blend all image properties into a vector.

Dense layer: generates possible neurone linkages in previous and future layers. All neurones are used for classification, and matrix vector multiplication is done between row vector neurones in the previous layer and column vector neurones in the following layer.

SoftMax classifier: The final layer of the classification model, SoftMax, reduces complexity and training time to categorise diseases. Training increases filter size in this classification layer using activation function, weight matrix, and bias vector. SoftMax layer multi-class classification classifies diseases based on probability. The equation shows the mathematical representations.

$$Final_{out} = Soft\max(dot(y, Kernel) + \beta) \quad (24)$$

The kernel and bias are represented as $Kernel$ and β respectively.

3.5 Hyperparameter Tuning Using Differential Bitwise African Vultures Optimization Algorithm (DBAVOA)

For the hyperparameter tuning, DBAVOA is proposed that has two strategies for enhancing optimization. The Bitwise Strategy (BS) is used to overcome the local optimum issues, and the Differential Evolution (DE) is applied in the exploration phase for enhancing searching ability of the proposed model.

3.5.1 African vulture's optimization algorithm (AVOA)

The nature-inspired algorithm is developed by Abdollahzadeh named as AVOA that is designed based on the orientation and feeding behaviour of the African vulture [25].

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

Phase 1: Population initialization: In the initialization stage, the best solutions are recognized as the first, best and second-best solution of vulture. The remaining vultures are assumed as third group and the equation to find the best solutions is evaluated in equation (25). Roulette wheel method is used to find ρ_2 that is calculated based on the equation (26).

$$I(j)=\begin{cases} Best\ solution_{vulture1},\ if\ \rho_j = K_1 \\ Best\ solution_{vulture2},\ if\ \rho_j = K_2 \end{cases} \quad (25)$$

$$\rho_j = \frac{\int_j}{\sum_{j=1}^{M} \int_j} \quad (26)$$

The equations provided that the terms such as $Best\ solution_{vulture1}$, $Best\ solution_{vulture2}$, K_1 and K_2, $\int_j$ and M are delineated as first best vulture, second best vulture, random values ranges from 0 to 1, fitness value of the first and second best vulture and total number of vulture respectively.

Phase 2: Starvation of vulture: If the vultures are starving, they does not contain enough energy to maintain the long-distance flight, otherwise if the vultures are not starving they have enough energy to take long jump to seek their food source. The hunger level is an indicator to shift the vultures from exploration to exploitation, and the hunger level is calculated as,

$$\rho_j = \left(2\times \Re_j + 1\right)\times \tau \times \left(1-\frac{\mathrm{I}_j}{M\mathrm{I}_j}\right)+T \quad (27)$$

$$T = d\times\left(\sin^{\gamma}\left(\frac{\pi}{2}\times\frac{\mathrm{I}_j}{M\mathrm{I}_j}\right)+\sin^{\gamma}\left(\frac{\pi}{2}\times\frac{\mathrm{I}_j}{M\mathrm{I}_j}\right)-1\right) \quad (28)$$

where $\Re_j$, τ, d, I_j, $M\mathrm{I}_j$ and γ are delineated as random value ranges between 0 and 1, random value ranges between -1 and 1, random value ranges between -2 and 2, number of current iteration, total number of iterations and parameter to change the vulture to exploitation stage respectively. The fitness value gradually decreased when the iteration increased. If the fitness value is greater than 1, the vulture moves to the exploration stage; otherwise, the vulture moves towards the exploitation stage.

Phase 3: Exploration stage: The vultures have high visual ability for seeking food sources and dead creatures, the random numbers are varied between 0 and 1 during the exploration stage. The position of the individual is updated based on the equation as,

$$\rho(j+1)=\begin{cases} I(j)-\delta(j)\times\int_j,\ if\ \Re_{\rho_1} \le \rho_1 \\ I(j)-\int_j+\Re_2\times\left((U-L)\times\Re_3+L\right),\quad if\ \Re_{\rho_1} > \rho_1 \end{cases} \quad (29)$$

The distance ($\delta(j)$) between current optimum and vulture is delineated as,

$$\delta(j)=\left|Y\times I(j)-\rho(j)\right| \quad (30)$$

From the above mentioned equations, the terms such as $I(j)$, $\Re_2$, U and L are delineated as best vulture, random number ranges between 0 and 1, upper bound variables and lower bound variables respectively.

Phase 4: Exploitation (first stage): The efficiency of the vulture is calculated in the first stage of exploitation and the position of the vulture is updated based on the below mentioned equation as,

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

$$\rho(j+1)=\begin{cases}\delta(j)-\left(\int(j)+\Re_4\right)-\upsilon(T), & if\ \rho_2 \geq \Re_{\rho_1}\\ I(j)-\left(A_1+A_2\right), & if\ \rho_2 < \Re_{\rho_1}\end{cases} \quad (31)$$

$$\upsilon(T)=I(j)-\rho(j) \quad (32)$$

$$A_1=I(j)\times\left(\frac{\Re_5\times\rho(j)}{2\pi}\right)\times\cos(\rho(j))$$

$$A_2=I(j)\times\left(\frac{\Re_6\times\rho(j)}{2\pi}\right)\times\sin(\rho(j)) \quad (33)$$

where $\upsilon(T)$, $\Re_5$ and $\Re_6$ are denoted as distance between vulture and bet vulture and random numbers between 0 and 1 respectively.

Phase 5: Exploitation (second stage): In this phase, the fitness of the vulture is smaller than 0.5 and the vultures are attacked for the food source when it satisfies the condition of $\rho_3 \geq \Re_3$. The position of the vulture is updated based on the below mentioned equation as,

$$\rho(j+1)=\frac{\alpha_1+\alpha_2}{2} \quad (34)$$

$$\alpha_1=Best\ solution_{vulture1}(j)-\frac{Best\ solution_{vulture1}(j)\times\rho(j)}{Best\ solution_{vulture1}(j)-(\rho(j))^2}\times\int_j$$

$$\alpha_2=Best\ solution_{vulture1}(j)-\frac{Best\ solution_{vulture2}(j)\times\rho(j)}{Best\ solution_{vulture2}(j)-(\rho(j))^2}\times\int_j \quad (35)$$

The vulture flocks the best vulture to clean the remaining food, and the position of the vulture is updated according to the below mentioned equation as,

$$\rho(j+1)=I(j)-|\upsilon(T)|\times\int_j\times L(\upsilon) \quad (36)$$

The levy flight (LF) strategy is implemented to enhance the effectiveness of the optimization model and is described as,

$$L(d)=0.001\times\frac{\nu\times\kappa}{|\varepsilon|^{1/\chi}}$$

$$\kappa=\left(\frac{\Gamma(1+\eta)\times\sin\left(\frac{\pi\eta}{2}\right)}{\Gamma(1+\eta 2)\times\eta\times 2\times\left(\frac{\eta-1}{2}\right)}\right)^{1/\chi} \quad (37)$$

The random values are delineated as ν and ε that is varied between 0 and 1, the constant number η is set to be 1.5.

3.5.2 Differential evolution (DE) algorithm

For enhancing search efficiency and optimization capability, the differential evolution is implemented in this paper by using DE/best/1. The mathematical representation of the DE is represented as,

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

$$\rho_j^{T+1} = \rho_{best}^T + S \times \left(\rho_{\Re 1}^T - \rho_{\Re 2}^T\right) \quad (38)$$

Here, the position of the vulture, scale factor and different integers are mathematically delineated as ρ_j^{T+1}, S and $\Re 1$ as well as $\Re 2$ respectively.

3.5.3 Bitwise strategy (BS)

Vultures hit the local optimum issue during optimisation procedure. In this study, the BS model is implemented to help the model avoid the optimisation problem and increase diversity. This BS model implements bitwise AND and OR. Bitwise AND operations between the best and freshly created solutions yield good features. OR takes AND output and the new solution as inputs. OR can transmit more useful features than AND, generating high-quality new solutions. Bitwise operations' random shuffles escape individuals from local optimum difficulties by improving solution quality and population variety. Conditions for AND and OR are as follows,

AND: If both bits are 1, then output is 1, else 0.

OR: If either bit is 1 or both are 1, then output is 1, else 0.

For example: 1001 AND 1101 = 1001, 1001 OR 1101 = 1101.

3.5.4 Proposed differential bitwise African vulture's optimization algorithm (DBAVOA) for hyperparameter tuning

For tuning the hyperparameters of the proposed rice plant disease detection and classification, the DBAVOA model is proposed that contains AVOA, BS and DE. The main advantage of combining BS is to overcome the local optimum issues and DE is to enhance searching ability of the proposed optimization model. The procedures to optimize the hyperparameters are as follows:

Initialization: The vultures are randomly assigned as, $\rho_j(T) = \{\rho_{j1}, \rho_{j2}, \ldots\ldots, \rho_{j\aleph}\}$. In the initialization phase, the number of vultures is randomly distributed within the range of [L, U] and the individuals are initialized based on the below mentioned equation as,

$$\rho_{ji} = L_i + \Re_i[0,1] \times (U_i - L_i), \quad j \in [1, M], i \in [1, D] \quad (39)$$

The terms such as D and M are delineated as number of dimensions and population size respectively. The name of the hyperparameters needs to be tuned for attaining optimal values and its ranges variation are provided as,

Total number of iterations: [10 to 1000]

Population size: [0 to 30]

Number of variables: [0 to 5]

Range of control parameters: [0.4 to 0.6]

Constant number, η : 1.5

Learning rate: [0.01 to 0.0001]

Batch size: [8, 16, 32]

Fitness value evaluation: The suggested model for rice plant disease detection and classification is evaluated using classification accuracy as the fitness function. The fitness function uses actual and anticipated probabilities to improve categorisation accuracy. Mathematics formulated as,

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad (40)$$

Here, the terms such as t_p, t_n, f_p and f_n are represented as true positive, true negative, false positive and false negative respectively.

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

*Solution updating phase:*For position updating, the exploration and exploitation phases are mainly used that takes major concentration for attaining optima solutions. These two phases come under the condition of ($I_j \le M_j$). In the exploration phase ($\int_j \ge 1$), the position of the individual is updated by using equations (41) respectively.

$$\rho(j+1)=\begin{cases} I(j)-\delta(j)\times\int_j+\rho_j^{T+1}, & if\ \Re_{\rho_1}\le\rho_1 \quad (a) \\ I(j)-\int_j+\Re_2\times\left((U-L)\times\Re_3+L\right)+\rho_j^{T+1}, & if\ \Re_{\rho_1}>\rho_1 \quad (b) \end{cases} \tag{41}$$

Otherwise, the exploitation phase is performed for updating positions of the individuals using equations (31), (34) and (36). Finally, the BS strategy is implemented for escaping individuals from the local optimum issue that categorizes the features attained from the segmented image into binary bits.

Termination criteria: Finally, the optimal parameters are attained with the highest fitness value, and the process is repeated until it satisfies the termination condition. The pseudocode to demonstrate the steps of the proposed optimization model is provided in Algorithm 1.

Algorithm 1: Pseudocode of DBAVOA

Start
Parameter initialization
Evaluate the fitness value using equation (40)
While ($I_j \le M_j$) do
  Evaluate the fitness of the vulture
  If ($j \le M$)
    Select first- and second-best vulture using equation (25)
    Evaluate τ, T and $\int_j$ using equations (27) and (28) respectively
    If ($\int_j \ge 1$) //Exploration phase
      If ($\Re_{\rho_1} \le \rho_1$)
        Update position using equations (41.a)
      Else
        Update position using equation (41.b)
    Else ($\int_j < 1$) //Exploitation phase
      If ($\Re_{\rho_2} \le \rho_2$)
        Update position using equation (31)
      End if
      If ($\Re_{\rho_3} \le \rho_3$)
        Update position using equation (34)

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

Else

Update the position using equation (36)

End if

End if

End if

Apply BS strategy

$I_j = I_j + 1$

End while

Attain best solution (optimal parameter values) with high fitness value

Stop

3.6 Training Process

To improve classification, hyperparameter adjustment and loss function evaluation are done during training. DBAVOA algorithm predicts optimal hyperparameter values for detection and classification models to improve classification performance. The loss function assessment reduces loss values and improves rice plant disease detection model classification. The optimisation problem is reduced using the categorical cross entropy loss function, computed as,

$$L = -\sum_{j=1}^{M} y_j . \log(\hat{y}_j) \quad (42)$$

Here, the terms such as y_j, $\hat{y}_j$ and M are represented as actual values, predicted values and number of samples respectively.

4. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental analyses are conducted in this section to identify the developments, trends, numerical and visual representations in the context of rice plant disease detection and classification.

4.1 Experimental Setup

Python 3.12 software and Intel® Core™ i5-5300U CPU processor, 64-bit OS, 2.30GHz processor storage, and 8GB RAM are used to anticipate the tests.

4.2 Hyperparameter Configuration

The hyperparameters of the proposed rice plant disease detection and classification model are optimised for accurate and efficient classification. Table 1 lists hyperparameters and recommended values.

Table 1: Hyperparameter tuning results

Name of the approach	**Name of the parameters**	**Optimal values**
Proposed ALFNN-AVO algorithm	Total number of iterations	1000
	Population size	30
	Number of variables	5
	Range of the control parameters	0.6

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

	η	1.5
	Learning rate	0.001
	Batch size	32

4.3 Performance Measures

Classification accuracy, precision, recall, specificity, F1-score, FPV, FNV, and MCC are used for detection and classification tasks. These subsections give accurate definitions and mathematical formulations.

Classification accuracy: The classification accuracy is defined by detecting rice diseases from all diseased and healthy samples. The mathematical formulation is described in equation (40).

Precision (P): Precision is measured by closeness values between two values and is not correct. Positive predictive value is mathematically given by,

$$P = \frac{t_p}{t_p + f_p} \tag{43}$$

Recall (R): Recall identifies the diseased instances from all diseased and healthy instances that demonstrate the efficiency of the machine-learning model. The recall is mathematically described as,

$$R = \frac{t_p}{t_p + f_n} \tag{44}$$

Specificity (S): This metric is used to identify the healthy instances of the dataset and also known as false positive value that helps to find the ability of the proposed model. The specificity is expressed as,

$$S = \frac{t_n}{t_n + f_n} \tag{45}$$

F1-score: The harmonic meaning of recall and precision is evaluated for the F1-score analysis and this analysis is used to identify the correct predictions of the model. The F1-score analysis is mathematically formulated as,

$$F = 2 \times \frac{P \times R}{P + R} \tag{46}$$

Mathew's Correlation coefficient (MCC): This is the statistical model for the evaluation purpose, and it is defined as the difference between predicted and actual values. The equation for the MCC is defined as,

$$MCC = \frac{(t_p \times t_n) - (f_p \times f_n)}{\sqrt{(t_p + f_p)(t_p + f_n)(t_n + f_p)(t_n + f_n)}} \tag{47}$$

False positive Value (FPV) is used to find the probability of false alarm rate and False Negative Value (FNV) determines the probability of diseased instances that are missed by the test.

4.4 Performance Analysis

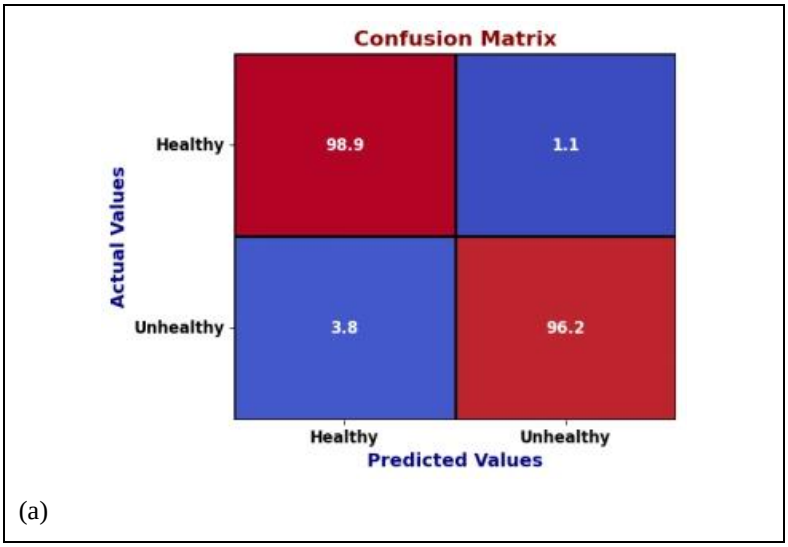
Performance analysis evaluates model strengths and shortcomings to improve efficiency and quality. The tabular performance analysis includes systematic observations to improve decision-making and performance. Table 2 shows the suggested model's performance rates by measure.

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

Table 2: Performance analysis results

Name of the performance measures	Performance values
Classification accuracy	98.87%
Precision	98.9%
Recall	98.2%
Specificity	97.9%
F1-score	98.3%
MCC	95.8%
FPV	1.7%
FNV	2.9%
Execution time	0.09 minutes

The confusion matrix is involved in this paper or the identification of classification performances attained from each dataset for each class. The confusion matrix is drawn in the tabular format that helps to differentiate the correctly and incorrectly predicted samples. The confusion matrices for the datasets are portrayed in Figure 5.



(a)

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

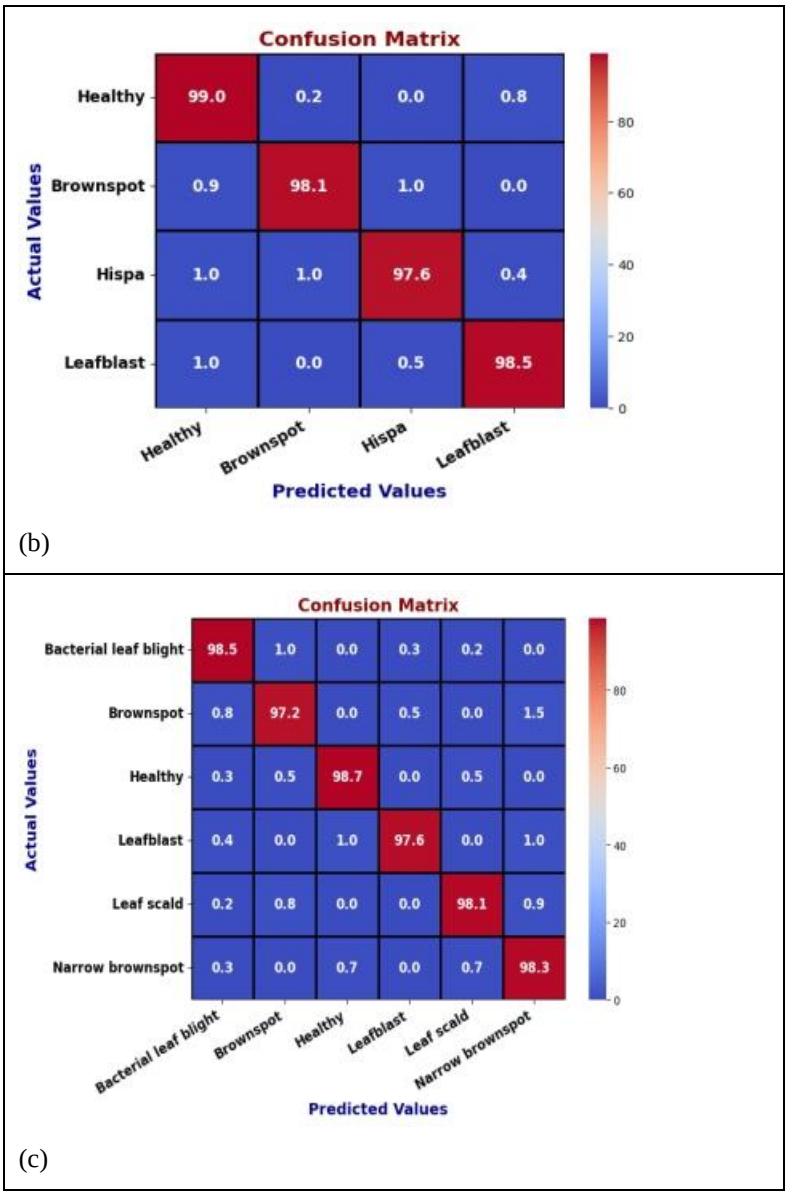


Figure 5. Confusion matrix for classification performances using (a) Rice plant dataset (b) Rice disease image dataset (c) Rice leaf disease dataset

Figure 6(a-b) represents the accuracy and loss analyses of the training and testing datasets of the proposed model. Here, three types of datasets are used for capturing rice plant images and the training set got better performance compared to testing set. In the accuracy analyses, the training accuracy has 0.988 and testing accuracy has 0.964. The loss analyses provided that the training and testing were set as 0.12 and 0.36 respectively.

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

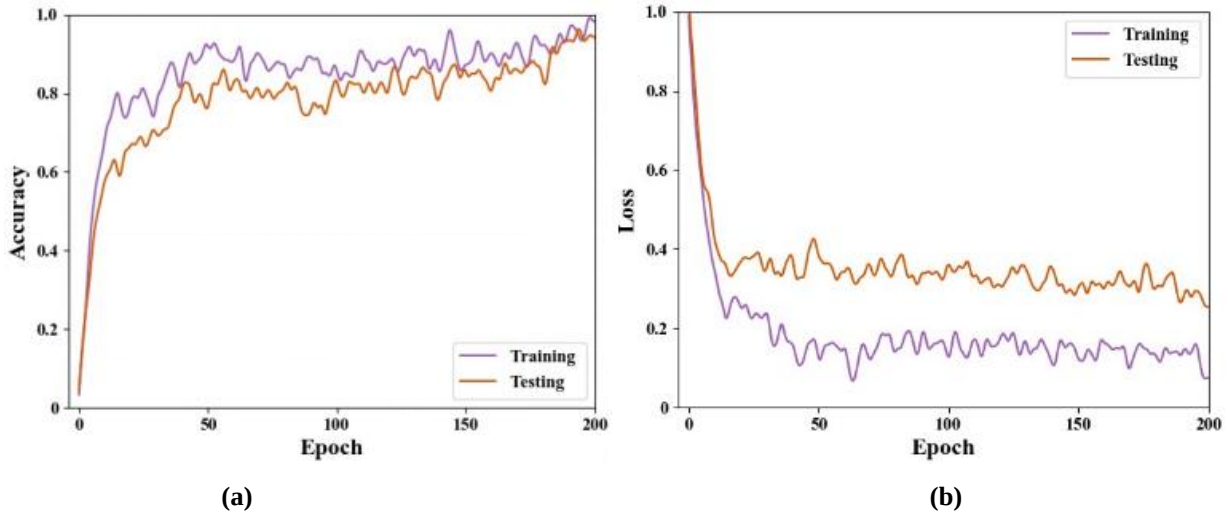


Figure 6. Performance of training and testing sets for (a) Accuracy (b) Loss

Area Under Receiver Operating Curve (AUROC) is used for analysing the classification performances that helps to correctly rank the model's ability. This analysis is performed for the performances of each dataset. The AUCROC values of 0.989, 0.99, and 0.985 are predicted from the rice plant dataset, the rice disease image dataset and rice leaf disease dataset, respectively. This analysis is conducted in Figure 7.

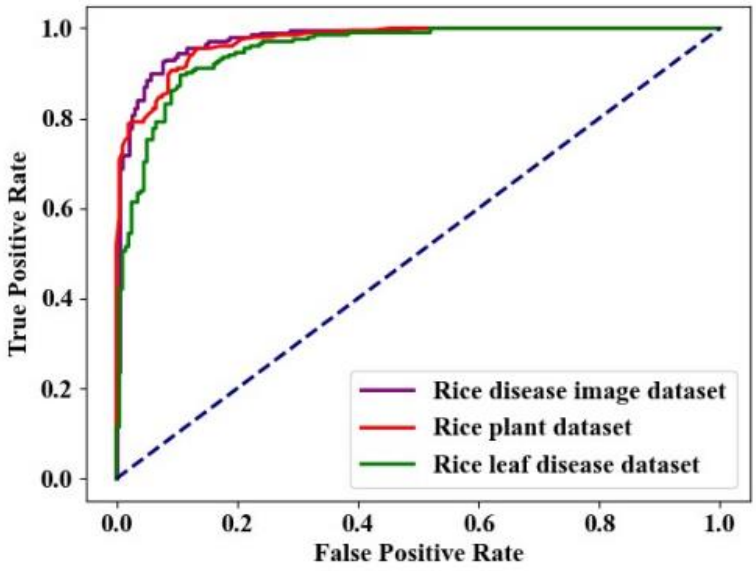


Figure 7. AUROC analysis for the classification analysis

5. Conclusion

In this paper, the proposed ALFNN-AVO model is introduced for the detection and classification of rice plant diseases. The data are collected from different data sources such as rice plant dataset, rice disease image dataset and rice leaf disease dataset that are pre-processed by using image enhancement, image resizing, noise reduction and data augmentation for enhancing suitability and image quality of the model. The CF former model is applied for efficiently extracting features of the input images and the fusion model is used to concatenate features attained from transformer

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

model. In this fusion model, the features from the different modalities are fused by using SE model. The diseased regions of the pre-processed images are identified by using the segmentation process named SPFCM. Finally, the ALDSNN model is applied for the detection and classification of rice plant diseases because this model is efficient and computationally compatible. The hyperparameters of the proposed model are tuned by using an efficient optimization model that overcomes the local optimum issues and enhances searching ability of the proposed optimization model. Here, comprehensive analyses are conducted for attaining superior performance of the proposed model that uses different performance measures as well as different approaches. This paper concludes that the proposed model has better performances in all analyses compared to other approaches. This research article will be extended to detect other types of diseases in the rice plant and other plant varieties. In future, the data can be collected from real-agricultural field by using advanced types of cameras for detecting wide range of crop diseases used in real time applications. This proposed model will be integrated with another advanced approach for enhancing more reliability and resilience.

Acknowledgement

We would like to extend our sincere appreciation to Integral University for their invaluable support in facilitating our research endeavours. The Manuscript Control Number Manuscript Communication Number: IU/R&D/2025-MCN0003670 issued by Integral University serves as a testament to their commitment to advancing academic research. This recognition underscores the significance of our work, and we are grateful for the university's dedication to fostering scholarly contributions in our field.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] P. Kulkarni and S. Shastri, "Rice leaf diseases detection using machine learning," *J. Sci. Res. Technol.*, pp. 17–22, 2024.

[2] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances," *J. Big Data*, vol. 11, no. 1, p. 5, 2024.

[3] K. S. Archana, S. Srinivasan, S. P. Bharathi, R. Balamurugan, T. N. Prabakar, and A. S. F. Britto, "A novel method to improve computational and classification performance of rice plant disease identification," *J. Supercomput.*, pp. 1–21, 2022.

[4] T. S. Poornappriya and R. Gopinath, "Rice plant disease identification using artificial intelligence approaches," *Int. J. Electr. Eng. Technol.*, vol. 11, no. 10, pp. 392–402, 2022.

[5] N. Mandal et al., "Spectral characterization and severity assessment of rice blast disease using univariate and multivariate models," *Front. Plant Sci.*, vol. 14, p. 1067189, 2023.

[6] T. Daniya and S. Vigneshwari, "Rice Plant Leaf Disease Detection and Classification Using Optimization Enabled Deep Learning," *J. Environ. Informatics*, vol. 42, no. 1, 2023.

[7] S. P. Sankareshwaran et al., "Optimizing rice plant disease detection with crossover boosted artificial hummingbird algorithm-based AX-RetinaNet," *Environ. Monit. Assess*, vol. 195, no. 9, p. 1070, 2023.

[8] E. Moupojou et al., "FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning," *IEEE Access*, vol. 11, pp. 35398–35410, 2023.

[9] C. K. Sunil, C. D. Jaidhar, and N. Patil, "Systematic study on deep learning-based plant disease detection or classification," *Artif. Intell. Rev.*, vol. 56, no. 12, pp. 14955–15052, 2023.

[10] P. I. Ritharson et al., "DeepRice: A deep learning and deep feature-based classification of rice leaf disease subtypes," *Artif. Intell. Agric.*, vol. 11, pp. 34–49, 2024.

[11] G. Latif, S. E. Abdelhamid, R. E. Mallouhy, J. Alghazo, and Z. A. Kazimi, "Deep learning utilization in agriculture: Detection of rice plant diseases using an improved CNN model," *Plants*, vol. 11, no. 17, p. 2230, 2022.

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025

[12] S. K. Upadhyay and A. Kumar, "A novel approach for rice plant diseases classification with deep convolutional neural network," *Int. J. Inf. Technol.*, vol. 14, no. 1, pp. 185–199, 2022.

[13] A. Haridasan, J. Thomas, and E. D. Raj, "Deep learning system for paddy plant disease detection and classification," *Environ. Monit. Assess*, vol. 195, no. 1, p. 120, 2023.

[14] M. Jiang et al., "Rice disease identification method based on attention mechanism and deep dense network," *Electronics*, vol. 12, no. 3, p. 508, 2023.

[15] K. K. Kishore and E. Kannan, "Detection of rice plant disease using AdaBoostSVM classifier," *Agronomy J.*, vol. 114, no. 4, pp. 2213–2229, 2022.

[16] D. J. Chaudhari and K. Malathi, "Detection and prediction of rice leaf disease using a hybrid CNN-SVM model," *Opt. Mem. Neural Netw.*, vol. 32, no. 1, pp. 39–57, 2023.

[17] R. K. Dubey and D. K. Choubey, "Efficient prediction of blast disease in paddy plant using optimized support vector machine," *IETE J. Res.*, vol. 70, no. 4, pp. 3679–3689, 2024.

[18] S. Joko, "Utilization of Artificial Neural Network in Rice Plant Disease Classification Using Leaf Image," *Int. J. Res. Sci. Eng.*, vol. 4, no. 2, pp. 1–10, 2024.

[19] N. Bharanidharan et al., "Multiclass Paddy Disease Detection Using Filter Based Feature Transformation Technique," *IEEE Access*, 2023.

[20] S. R. Reddy, G. S. Varma, and R. L. Davuluri, "Resnet-based modified red deer optimization with DLCNN classifier for plant disease identification and classification," *Computers Electr. Eng.*, vol. 105, p. 108492, 2023.

[21] Y. M. Abd Algani et al., "Leaf disease identification and classification using optimized deep learning," *Measurement: Sensors*, vol. 25, p. 100643, 2023.

[22] N. A. Farooqui et al., "Precision agriculture and predictive analytics: Enhancing agricultural efficiency and yield," *Intelligent Techniques Predictive Data Anal.*, pp. 171–188, 2024.

[23] S. N. Ahmed and P. Prakasam, "A systematic review on intracranial aneurysm and hemorrhage detection using machine learning and deep learning techniques," *Progress Biophys. Mol. Biol.*, 2023.

[24] R. Liu et al., "Improved African vulture optimization algorithm based on quasi-oppositional differential evolution operator," *IEEE Access*, vol. 10, pp. 95197–95218, 2022.

[25] Y. Zhang et al., "High-Precision Seedling Detection Model Based on Multi-Activation Layer and Depth-Separable Convolution Using Images Acquired by Drones," *Drones*, vol. 6, no. 6, p. 152, 2022.

[26] W. Zheng et al., "Lightweight transformer image feature extraction network," *PeerJ Comput. Sci.*, vol. 10, p. e1755, 2024.

DOI: https://doi.org/10.54216/FPA.210107
Received: February 13, 2025 Revised: May 31, 2025 Accepted: July 04, 2025