International Journal of Technology

http://ijtech.eng.ui.ac.id

Research Article

Exploiting the strength of modified parrot optimization algorithm for enhancing rice leaf disease detection using convolutional neural network and transfer learning

Ibrahim Hayatu Hassan [1], Anees Ara[2], Salma Idris [2], Tanzila Saba[2*] and Saeed Ali Bahaj[3]

[1]Department of Computer Science, Ahmadu Bello University, Zaria 810107, Nigeria;
[2] AIDA Lab. CCIS Prince Sultan University Riyadh 11586 Saudi Arabia;
[3] MIS Department College of Business Administration, Prince Sattam Bin Abdulaziz University, 11942, AlKharj, Saudi Arabia

[*]Corresponding author: drstanzila@gmail.com

Abstract: Timely and accurate identification of rice leaf diseases is paramount for optimizing crop productivity and safeguarding global food security. This research developed an innovative deep learning framework that incorporates the DenseNet121 architecture, optimized through a modified Parrot Optimization Algorithm (POA), to achieve precise classification of rice leaf diseases. The modified POA, an enhanced variant of the original algorithm, integrates Mutation random opposition-based learning (mROB) and Brownian motion mechanisms to improve optimization efficiency. By effectively tuning critical hyperparameters, including batch size, learning rate, dropout rate, and the number of neurons, the proposed model demonstrates superior performance. Evaluations conducted on rice leaf disease dataset revealed that the modified POA-DenseNet121 model outperformed established pretrained models such as VGG19, DenseNet201, InceptionV3, EfficientNetB0, and ResNet50. The proposed model achieved remarkable performance metrics, including a 98.5% accuracy, 98.6% precision, 98.4% recall, and 98.5% F-measure. Furthermore, the application of optimization strategies, including step decay learning schedules and early stopping, enhanced the model's robustness and minimized the risk of overfitting. This study underscores the potential of the modified POA-DenseNet121 framework as a scalable and efficient tool for advancing agricultural diagnostics and addressing challenges in rice disease management.

Keywords: Disease detection; Parrot optimization algorithm; Rice leaf disease; Transfer learning; Technological development

1. Introduction

Rice is a fundamental food source for nearly half of the global population, making its stable production crucial for global food security (Chen et al., 2020). However, rice-based agriculture faces persistent economic, ecological, and social challenges, with diseases posing one of the most

https://doi.org/xx/ijtech.xx
Received date; Revised date; Accepted date

significant threats. The International Rice Research Institute (IRRI) estimates that rice diseases can cause yield losses of up to 80%, further aggravating food insecurity (Ritharson et al., 2024; Yusuf et al., 2024). Traditional disease detection methods, such as visual inspection and lab tests, are time consuming, error-prone, and costly, especially across large-scale farms (Shah et al., 2023; Barman et al., 2024).

Advancements in artificial intelligence (AI), particularly deep learning, have shown promise in revolutionizing agricultural diagnostics. Convolutional Neural Networks (CNNs) have emerged as effective tools for automatically learning hierarchical features from disease images, thus facilitating accurate rice disease classification (Pattnaik et al., 2021; Chakrabarty et al., 2024). Despite their effectiveness, training CNNs from scratch is resource-intensive and requires large annotated datasets often unavailable in the field of agriculture. Transfer learning mitigates this by fine-tuning pre-trained CNNs, enabling high performance even with limited data (Ayesha et al., 2021; Yuan et al., 2022).

Nevertheless, CNN performance heavily depends on optimal hyperparameter settings such as learning rate, batch size, and number of filters, making hyperparameter tuning a critical step. Traditional methods like manual tuning, grid search, and random search are often inefficient, computationally expensive, and time-consuming (Rehman et al., 2021; Barman et al., 2024; Rehman et al., 2023). These challenges have driven researchers to reframe hyperparameter tuning as a complex optimization problem, increasingly addressed through metaheuristic algorithms.

Metaheuristic algorithms, inspired by natural phenomena such as evolution, swarm behavior, and physics, offer strategic exploration of high-dimensional and non-linear search spaces. Their adaptability and effectiveness have been widely demonstrated in optimizing deep learning models for various applications. For example, Artificial Bee Colony (Ebraheem et al., 2024), Artificial Namib Beetle Optimization (Rao & Vasumathi, 2024), and Jaya Artificial Ecosystem-Based Optimization (Babu & Philip, 2024) have all been employed to fine-tune CNNs. However, the No Free Lunch (NFL) theorem (Wolpert & Macready, 1997) asserts that no single algorithm performs best across all problems, prompting the need for tailored or hybrid approaches.

In this context, the Parrot Optimizer Algorithm (POA) (Lian et al., 2024), inspired by the behaviors of Pyrrhura molinae parrots offers a compelling optimization framework. Though effective, the original POA struggles with premature convergence and limited exploration. To overcome these limitations, this study introduces a modified POA (mPOA) that integrates mutated random opposition-based learning (mOBL) and Brownian motion to enhance population diversity and convergence speed.

The proposed approach combines CNNs with mPOA for optimized rice leaf disease detection using transfer learning. By fine-tuning a comprehensive set of CNN hyperparameters, the method demonstrates superior accuracy, precision, recall, and F1-score across disease classes. This integration of deep learning and metaheuristic optimization highlights the growing potential of AI in precision agriculture, particularly for scalable and cost-effective disease diagnostics.

Overall, this study contributes a novel, hybridized optimization framework that addresses critical challenges in CNN training and demonstrates its effectiveness in improving rice crop health monitoring, thereby supporting food security efforts on a global scale.

This study highlights that rice leaf disease detection models may struggle with unseen environments and cultivars due to diverse leaf morphology and backgrounds. Future work will incorporate multi-location datasets and domain adaptation. The model's potential for real-time, mobile-based diagnostics offers practical benefits for timely agricultural interventions.

The key contributions of this study are outlined as follows:

1. Applied various preprocessing methods, including contrast enhancement, normalization, noise reduction and data augmentation, to improve the leaf image quality for more effective detection of rice leaf diseases.
2. Proposed a modified Parrot Optimizer algorithm (POA) incorporating mutated random opposition-based learning and Brownian motion strategies to enhance the performance of the traditional POA for superior optimization.
3. Applied the modified POA for efficient hyperparameter optimization of DenseNet121, thereby improving the disease detection.
4. Performed thorough evaluations of the presented model using rice disease to confirm its effectiveness.
5. Performed a comparative analysis of the presented model in contrast to other advanced rice disease detection methods available in the literature.

This manuscript is delineated in the following sequence: Section 2 offers a review of recent deep learning-based rice leaf disease detection models. Section 3 describes the CNN architecture and the transfer learning approach. Section 4 introduces the POA and its mathematical formulations. Section 5 presents the modified POA. Section 6 outlines the proposed rice leaf disease detection model and its performance evaluation. Lastly, Section 7 concludes the paper and explores potential avenues for future research.

2. Literature Review

Recent studies on rice leaf disease detection have evolved from traditional machine learning to advanced deep learning models. Goluguri et al. (2021) combined deep CNN with LSTM and artificial fish swarm optimization, achieving 97.5% accuracy but facing scalability issues due to computational demands. Daniya and Vigneshwari (2023) proposed a Rider Water Wave-based Neural Network, achieving 90.8% accuracy, limited by single-dataset testing. Hossain et al. (2024) introduced a Deep Learning-based Crested Porcupine Optimizer model with ConvNeXt-L and CVAE, but its computational complexity hinders real-time use. Preethi et al. (2024) developed a hybrid DNN with Enhanced Artificial Shuffled Shepherd Optimization, achieving 97.29% accuracy, constrained by reliance on high-resolution images. These studies highlight the need for efficient, scalable models, which this work addresses through mPOA and DenseNet121.

3. CNN and Transfer learning

CNNs excel in image classification by extracting complex features through convolutional, activation, pooling, and batch normalization layers (Barakat et al., 2024; Manjupriya et al., 2025). The classification section uses dense and dropout layers to categorize features, with softmax activation for multi-class tasks (Maijeddah et al., 2024; Ibrahim et al., 2024). Hyperparameter tuning (e.g., learning rate, dropout factor) is critical but challenging due to the vast search space (Gaspar t al., 2021; Emam et al., 2024; Mahmmod et al., 2023). Transfer learning addresses data scarcity by leveraging pretrained models like DenseNet121, which uses dense connectivity to enhance feature reuse and mitigate vanishing gradients (Mofrad & Valizadeh, 2023). This study employs DenseNet121 with transfer learning, fine-tuning the classifier while optimizing hyperparameters using mPOA.

3.1 DenseNet pre-trained models

DenseNet is a powerful convolutional neural network (CNN) architecture that has significantly enhanced transfer learning performance across various vision tasks. It features a distinctive dense connectivity pattern, where each layer receives inputs from all preceding layers, encouraging feature reuse and mitigating vanishing gradient issues. DenseNet models come in various depths, including DenseNet-121, DenseNet-169, and DenseNet-201, with 121, 169, and 201 layers respectively. These variants balance computational cost and representational power, allowing practitioners to select a model based on task complexity and available resources. Table 1 provides a summary of these pre-trained models, particularly those trained on the ImageNet dataset

All DenseNet versions include four dense blocks composed of 1×1 and 3×3 convolutional layers, separated by transition layers with convolution, pooling, and normalization. These models are typically pre-trained on large datasets like ImageNet, capturing generalized visual features useful for transfer learning. The architecture of DenseNet-121, as initially applied to the ImageNet dataset, is shown in Figure 1.

Transfer learning with DenseNet involves fine-tuning a pre-trained model rather than training from scratch. Freezing early convolutional layers, which detect common low-level features such as edges, accelerates training and reduces overfitting especially with small datasets. However, freezing too many layers may limit the model's ability to learn high-level, task-specific features. Thus, selecting an appropriate fine-tuning strategy is essential for achieving optimal performance.

Table 1: Architectural settings for different DenseNet CNN architectures trained using ImageNet.

Layers	DenseNet-121	DenseNet-169	DenseNet-201	Output size		
				DenseNet-121	DenseNet-169	DenseNet-201
Input				$224 \times 224 \times 3$		
Conv.	$7 \times 7\ conv, stride = 2$			$112 \times 112 \times 64$		
Pooling	$3 \times 3 \max pool, stride = 2$			$56 \times 56 \times 64$		
Block-1	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 6$			$56 \times 56 \times 256$		
TL-1	$1 \times 1\ conv$ $2 \times 2\ \text{avg}\ pool, stride = 2$			$56 \times 56 \times 128$ $28 \times 28 \times 128$		
Block-2	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 12$			$28 \times 28 \times 512$		
TL-2	$1 \times 1\ conv$ $2 \times 2\ \text{avg}\ pool, stride = 2$			$28 \times 28 \times 256$ $14 \times 14 \times 256$		
Block-3	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 48$	$14 \times 14 \times 1024$	$14 \times 14 \times 1280$	$14 \times 14 \times 1792$
TL-3	$1 \times 1\ conv$ $2 \times 2\ \text{avg}\ pool, stride = 2$			$14 \times 14 \times 512$ $7 \times 7 \times 512$	$14 \times 14 \times 600$ $7 \times 7 \times 600$	$14 \times 14 \times 896$ $7 \times 7 \times 896$
Block-4	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1\ conv \\ 3 \times 3\ conv \end{bmatrix} \times 32$	$7 \times 7 \times 1024$	$7 \times 7 \times 1664$	$7 \times 7 \times 1920$
CL	7×7 global avg $pool$ $1000 - \text{D fc, softmax}$			$1 \times 1 \times 1024$	$1 \times 1 \times 1664$ $1 \times 1 \times 100$	$1 \times 1 \times 1920$

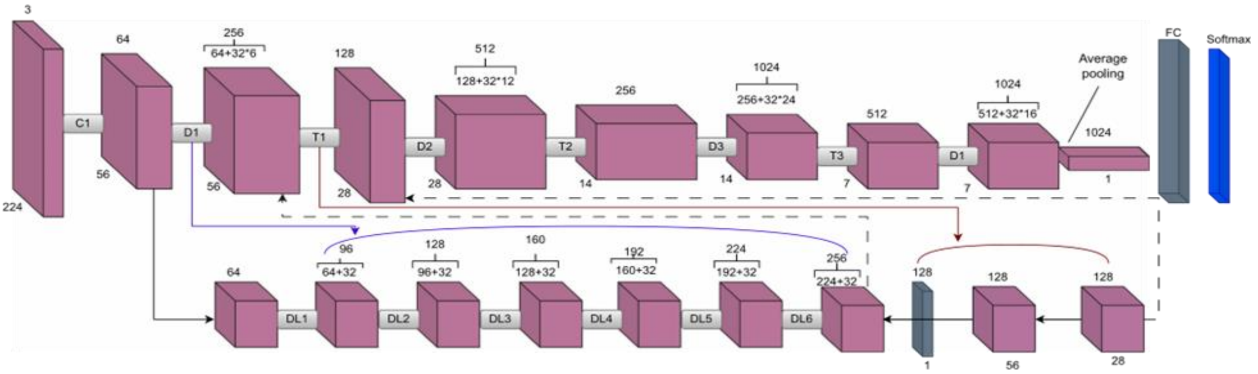

Figure 1 The main architecture of DenseNet-121 CNN architecture, as applied to the ImageNet data.

The diagram includes the following components: D represents the dense blocks, T denotes the transition layers, FC stands for the fully connected layers, DL refers to the dense layers, and C represents the initial convolution and pooling layers.

4 Parrot Optimization Algorithm (POA)

The POA by Lian et al. (2024) is a novel and effective metaheuristic algorithm inspired by the behavioural traits of domesticated Pyrrhura Molinae parrots, such as foraging, remaining stationary, vocalizing, and exhibiting caution towards unfamiliar entities. These behavioural patterns serve as the foundational principles for the development of the POA. This section presents an overview of the POA along with its foundational mathematical framework.

4.1 POA Initialization stage

The POA, as introduced by Lian et al. (2024), is an innovative population-based metaheuristic approach, where each parrot in the population symbolizes a potential solution to the optimization issue. The position of each Pyrrhura Molinae within the search space is mapped to the values of the parameters, thereby defining a possible solution. For POA initialization, parameters such as the size of the population size (N_pop), the highest number of iterations (MaxIter), and the boundaries of the search region denoted by the lower bound (lwb) and the upper bound (upb) are considered. This initialization process is mathematically formulated in Equation 1.

$$\boldsymbol{P_i^0 = lwb + r * (upb - lwb)} \qquad (1)$$

Where r signifies a number randomly produced in range of [0, 1] and $\boldsymbol{P_i^0}$ denotes the $\boldsymbol{i_{th}}$ Pyrrhura Molinae position in the starting stage.

4.2 POA hunting conduct

In POA, during the hunting phase, the parrots assess the possible location of food by observing its surroundings or by referring to the position of the leader. Subsequently, they move toward the identified region. As a result, the variation in their position is controlled by the equation presented in Equation 2.

$$\boldsymbol{P_i^{curIt+1} = \left(P_i^{curIt} - P_{best}\right) * Lv(D) + r(0,1) * \left(1 - \frac{curIt}{MaxIter}\right)^{\frac{2curIt}{MaxIter}} * P_{mean}^{curIt}} \qquad (2)$$

Where, $\boldsymbol{P_i^{curIt}}$ defines the current region and $\boldsymbol{P_i^{curIt+1}}$ indicates the location after the next update. $\boldsymbol{P_{mean}^{curIt}}$ signifies the average location within the existing population, and Lv(D) refers to a Levy distribution, which characterizes the parrots' flight pattern. $\boldsymbol{P_{best}}$ represents the optimal position achieved up to this point, from the initialization stage to the current phase, and also indicates the current position of the leader. $\boldsymbol{curIt}$ represents the current iteration. $\boldsymbol{\left(P_i^{curIt} - P_{best}\right) * Lv(D)}$ represent movement in relation to one's position based on the owner and $\boldsymbol{r(0,1)}$. $\boldsymbol{\left(1 - \frac{curIt}{MaxIter}\right)^{\frac{2curIt}{MaxIter}} * P_{mean}^{curIt}}$ refers to the monitoring of the overall position of the population in order to better direct the search for the food's location. The average position of current swarm, $\boldsymbol{P_{mean}^{curIt}}$ is computed using the mathematical expression depicted in Equation 3 and the Lv(D) can be determine using rule define in Equation 4. $\boldsymbol{\gamma}$ is given a magnitude of 1.5.

$$\boldsymbol{P_{mean}^{curIt} = \frac{1}{N_{pop}} \sum_{k}^{N_{pop}} P_k^{curIt}} \qquad (3)$$

$$\begin{cases} \boldsymbol{Lv(D)} = \frac{\boldsymbol{\mu-\sigma}}{|\boldsymbol{v}|^{\frac{1}{7}}} \\ \boldsymbol{\mu \sim N_{pop}(0, D)} \\ \boldsymbol{v \sim N_{pop}(0, D)} \\ \boldsymbol{\sigma} = \left(\frac{\boldsymbol{\tau(1+\gamma) * \sin\left(\frac{\pi\gamma}{2}\right)}}{\boldsymbol{\tau\left(\frac{1+\gamma}{2}\right) * \gamma . 2^{\frac{1+\gamma}{2}}}} \right)^{\boldsymbol{\gamma+1}} \end{cases} \tag{4}$$

4.3 POA staying conduct

The highly social Pyrrhura molinae primarily demonstrates a characteristic behavior of swiftly flying to a specific area on its owner's body, where it stays motionless for a particular period. This behavior is mathematically represented by Equation 5.

$$\boldsymbol{P_i^{curIt+1} = P_i^{curIt} + P_{best} * Lv(D) + r(0,1) * \ ones(1, d)} \tag{5}$$

$\boldsymbol{ones(1, D)}$ signifies all-1 vector of D dimension, $\boldsymbol{P_i^{curIt} + P_{best}}$ represents the flight to the host, and the procedure of randomly halting at a portion of the host's body is defined by $\boldsymbol{r(0,1) * ones(1, d)}$.

4.4 PO communication conduct

Parrots, belonging to the Pyrrhura Molinae family, are inherently social creatures that display a strong tendency for group communication. Their communication behavior includes both hovering to join the flock and interacting without flying. The POA assumes that these actions have an equal chance of occurring. The center of the flock is represented by the mean position of the current population. Equation 6 provides a mathematical expression for this phenomenon.

$$\boldsymbol{P_i^{curIt+1}} = \begin{cases} \boldsymbol{0.2 * r(0,1) * (1 - \frac{curIt}{MaxIter}) * (P_i^{curIt} - P_{mean}^{curIt}), pr \leq 0.5} \\ \boldsymbol{0.2 * r(0,1) * exp\left(-\frac{curIt}{r(0,1) * MatIter}\right), pr > 0.5} \end{cases} \tag{6}$$

4.5 POA Fear of strangers' conduct

Parrots of the Pyrrhura Molinae species, like other birds, exhibit an instinctual fear of unfamiliar individuals. In response to this fear, they tend to seek safety by distancing themselves from strangers and finding refuge with their owners. In POA, this behavior can be mathematically represented by Equation 7.

$$\boldsymbol{P_i^{curIt+1} = P_i^{curIt} + r\ (0,1) * \cos\left(0.5\pi * \frac{curIt}{MaxIter}\right) * (P_{best} - P_i^{curIt}) - \cos(r\ (0,1) * \pi) *} \\ \boldsymbol{\left(\frac{curIt}{MaxIter}\right)^{\frac{2}{MaxIter}} * (P_i^{curIt} - P_{best})} \tag{7}$$

where $\boldsymbol{r\ (0,1) * \cos\left(0.5\pi * \frac{curIt}{MaxIter}\right) * (P_{best} - P_i^{curIt})}$ represents the procedure of reorienting to fly in the direction of the owner and $\boldsymbol{\cos(r\ (0,1) * \pi) * \left(\frac{curIt}{MaxIter}\right)^{\frac{2}{MaxIter}} * (P_i^{curIt} - P_{best})}$ indicates the procedure of going away from the strangers.

In POA, the procedure will go on until the specified circumstances for termination are fulfilled. The pseudocode is indicated by Algorithm 1.

```
Algorithm 1: PO algorithm pseudo − code
PO parameter initialization
Randomly initialized the position of the solution agents
For $i = MaxIter$ do
    Compute the fitness function and fine the best position
    For $j = 1: N_{pop}$ do
    $st = randi([1, 4]$
    $\boldsymbol{If}\ st == 1\ \boldsymbol{Then}$
        $foraging\ conduct\ and\ update\ posing\ using\ Eq.\ (2)$
    $\boldsymbol{Elseif}\ st == 2\ \boldsymbol{Then}$
        $Staying\ conduct\ and\ update\ posing\ using\ Eq.\ (5)$
    $\boldsymbol{Elseif}\ st == 3\ \boldsymbol{Then}$
        $Communicating\ conduct\ and\ update\ posing\ using\ Eq.\ (6)$
    $\boldsymbol{Elseif}\ st == 4\ \boldsymbol{Then}$
        $fear\ of\ strangers\ conduct\ and\ update\ posing\ using\ Eq.\ (7)$
  $\boldsymbol{End\ For}$
$\boldsymbol{End\ For}$
Return best solution obtain
```

5 Modified Parrot optimizer (mPOA)

The following section presents an enhanced variant of the POA, referred to as the modified parrot optimization algorithm (mPOA), which seeks to improve the POA's local search capabilities and speed up the global search procedure in order to get over its drawbacks. The main goal of mPOA is to lessen the issue of stagnation at local optima while achieving faster convergence. In order to give a comprehensive overview, we start by examining the difficulties with the traditional PO algorithm.

5.1 Issues with original POA

The original Parrot Optimization Algorithm (POA), though effective, struggles with high-dimensional problems due to premature convergence and limited exploration. To overcome this, we propose a modified POA (mPOA) integrating opposition-based mutation learning (mOBL) and Brownian motion (BR). mOBL enhances initialization and accelerates convergence, while BR improves exploration. The mPOA's effectiveness was validated by optimizing a pre-trained DenseNet-121 model for rice leaf disease detection, demonstrating improved performance in navigating complex search spaces and identifying optimal hyperparameter configurations.

5.2 Opposition based mutation learning approach (OBL)

OBL enhances convergence in metaheuristic algorithms by simultaneously exploring original and opposite solutions, increasing the chance of locating global optima. It is especially effective when initial solutions are suboptimal, accelerating convergence and improving performance by expanding the search space and enabling the selection of superior solutions from a broader solution pool (Adamu et al., 2022). The following subsection describes how the OBL is incorporated.

Opposite values: In OBL, Equation 8 is used to determine the opposite of a real integer $\boldsymbol{y}$ inside the interval $[\boldsymbol{lwb}, \boldsymbol{upb}]$.

$$\boldsymbol{y_O = lwb + upb - y} \quad (8)$$

where $\boldsymbol{lwb}$ and $\boldsymbol{upb}$ denote lower and upper bounds respectively

Opposite vectors: If $\boldsymbol{Y = [y_1, y_2, \ldots, y_n]}$ is a vector, where $\boldsymbol{y_1, y_2, \ldots, y_n} \in \boldsymbol{R}$ and $\boldsymbol{y_j} \in [\boldsymbol{lw_q}, \boldsymbol{up_q}]$. The opposite vector $\boldsymbol{Y_O = [y_{O1}, y_{O2}, \ldots, y_{On}]}$ is computed based on Equation 9.

$$\boldsymbol{y_O = lwb_q + upb_q - y_q} \quad (9)$$

In OBL, the solution $\boldsymbol{Y}$ is substituted with its complementary counterpart $\boldsymbol{Y_O}$, determined by an activation function. If the fitness of $\boldsymbol{Y}$ represented as $\boldsymbol{f(Y)}$ is greater than that of $\boldsymbol{Y_O}$ denoted as $\boldsymbol{f(Y_O)}$, $\boldsymbol{Y}$ is preserved; otherwise, Y is replaced with $\boldsymbol{Y_O}$. This updating process facilitates the evolution of the solution population by selecting the better solution between $\boldsymbol{Y}$ and $\boldsymbol{Y_O}$. The paper presents a modified version of this approach, referred to as the mutated random opposition-based learning (mROBL) strategy, which is outlined in Equation 10.

$$\boldsymbol{y_{mOBL} = lw_q + up_q - \varepsilon \times r} \quad (10)$$

In this context, $\boldsymbol{r}$ denotes a value within the range [0,1], and $\boldsymbol{\varepsilon}$ represents the mutation scale, a small constant that regulates the intensity of the mutation. Unlike Equation 9, the mutated opposite solution, as outlined in Equation 10, introduces a higher degree of randomness. This increased randomness promotes greater diversity within the population, thereby improving the algorithm's capability to efficiently escape from local optima.

5.3 Brownian motion

The length of each phase in the Brownian motion method is governed by a normal gaussian distribution function, with a mean of zero (μ = 0) and a variance of one (σ^2 = 1). The function that describes this motion at a given point y is specified in Equation 11 (Faramarzi et al., 2020).

$$\boldsymbol{F_{BR}(y,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right)} \tag{11}$$

here, $\boldsymbol{F_{BR}(y,\mu,\sigma)}$ denotes the Brownian motion probability density function, the mean is represented by μ and the standard deviation by $\boldsymbol{\sigma}$.

5.4 Initialization stage of modified POA

Produce a starting population of potential solutions in the stated limits of the search space. Enhance the initial population by employing the mutated random (mROBL) strategy, which evaluates the fitness of each solution's opposite counterpart and updates the solution if the opposing counterpart demonstrates superior fitness, as described in Equation 10. Consequently, the application of Equation 10 effectively enhances population diversity and aids in overcoming local optima by facilitating the population's transition to unexplored regions of the search space.

5.5 Fitness function

The fitness function assesses how well a particular solution approximates the optimal solution for the given problem. The fitness value for each search agent is calculated using Equation 12.

$$\boldsymbol{Fit_i = \alpha \times Err_i + \beta \times \frac{d_i}{D}} \tag{12}$$

The $\boldsymbol{\alpha}$ is assign a value of 0.7 and $\boldsymbol{\beta = 1 - \alpha}$. The α parameter strikes an equilibrium between number of feature subsets $\boldsymbol{FS}$ $(\boldsymbol{d_i})$ and the error rate $(\boldsymbol{Err})_{\boldsymbol{i}}$ of classification.

5.6 Modified POA fitness evaluation

Each parrot fitness value is chosen using Eqn. (13)

$$\boldsymbol{if\ mOBL_{fit} < F_i\ then \begin{cases} Y(i,:) = Y_{mOBL} \\ F_i = mOBL_{fit} \end{cases}} \tag{13}$$

5.7 modified POA update phase

It is essential to evaluate the solutions at each iteration in order to find the best candidates and improve the freshly created solutions for subsequent phases. After computing the fitness of each individual, their positions are updated by applying the initial phases of the POA. The foraging behavior phases are implemented as described in Equations (2–4). The parameter the St $\boldsymbol{\epsilon}$ rand [1, 4] is then examined. If $\boldsymbol{St = 1}$, the position is updated using the Brownian motion (BR) strategy as per Equation 11 instead of the Lévy strategy. Similarly, if $\boldsymbol{St = 2}$, the position is updated using BR based on Equation (11) instead of employing the staying behavior. Algorithm 2 presents the pseudo-code for the mPOA.

```
**Algorithm 2**: Modified POA pseudo − code
Initilize the mPOA parametres
Initilize the position of all agents using Eqn. (1)and evaluate their fitness
$\boldsymbol{for}$ i = $\mathrm{N_{pop}}$ $\boldsymbol{do}$
    Perform mOBL on the initial population using Eqn. 10 and save results in $Y_{mOBL}$
    Evaluete the fitness $of\ Y_{mOBL}$ and save result in $mOBL_{fit}$
    $if\ mOBL_{fit} < F_i\ then$
        $Y_i = Y_{mOBL}$
    **end if**
**end for**
$\boldsymbol{For}\ i = MaxIter\ do$
    $Compute\ the\ fitness\ function\ and\ fine\ the\ best\ position$ (12)
    $\boldsymbol{For}\ j = 1{:}\ N_{pop}\ \boldsymbol{do}$
    $st = randi([1, 4])$
      If $st == 1$ Then
          update positiong using Eqn. (11)
      Elseif $st == 2$ Then
          Update position using BR based on Eqn. (11)
      Elseif $st == 3$ Then
          Perform communicating conduct and update posing using Eq. (6)
      Elseif $st == 4$ Then
          Perform fear of strangers conduct and update posing using Eq. (7)
    End For
End For
Return best solution obtain
```

6 Proposed rice leaf disease detection model

This part presents the methodology underlying the presented rice leaf disease detection model, designed to significantly enhance the performance of CNN architectures. The approach integrates the mPOA with DenseNet121 framework to optimize it performance for improved disease detection and classification. Firstly, the mPOA is employed to determine the optimal hyperparameter configurations of DenseNet121. Afterward, transfer learning methods are applied to train the DenseNet121. Upon completion of the training, the model's performance is assessed on a separate validation set. More precisely, the training and validation datasets are used to optimize the hyper-parameters and train the DenseNet121, with subsequent assessment carried out using the validation data. The methodology outlines a detailed pipeline for building the deep learning model, starting with the acquisition of the dataset and ending with the generation of classification results. The presented model consists of several stages, as illustrated in Figure 2. These stages include data collection, preprocessing, hyperparameter tuning, the learning phase, and the testing and evaluation phase.

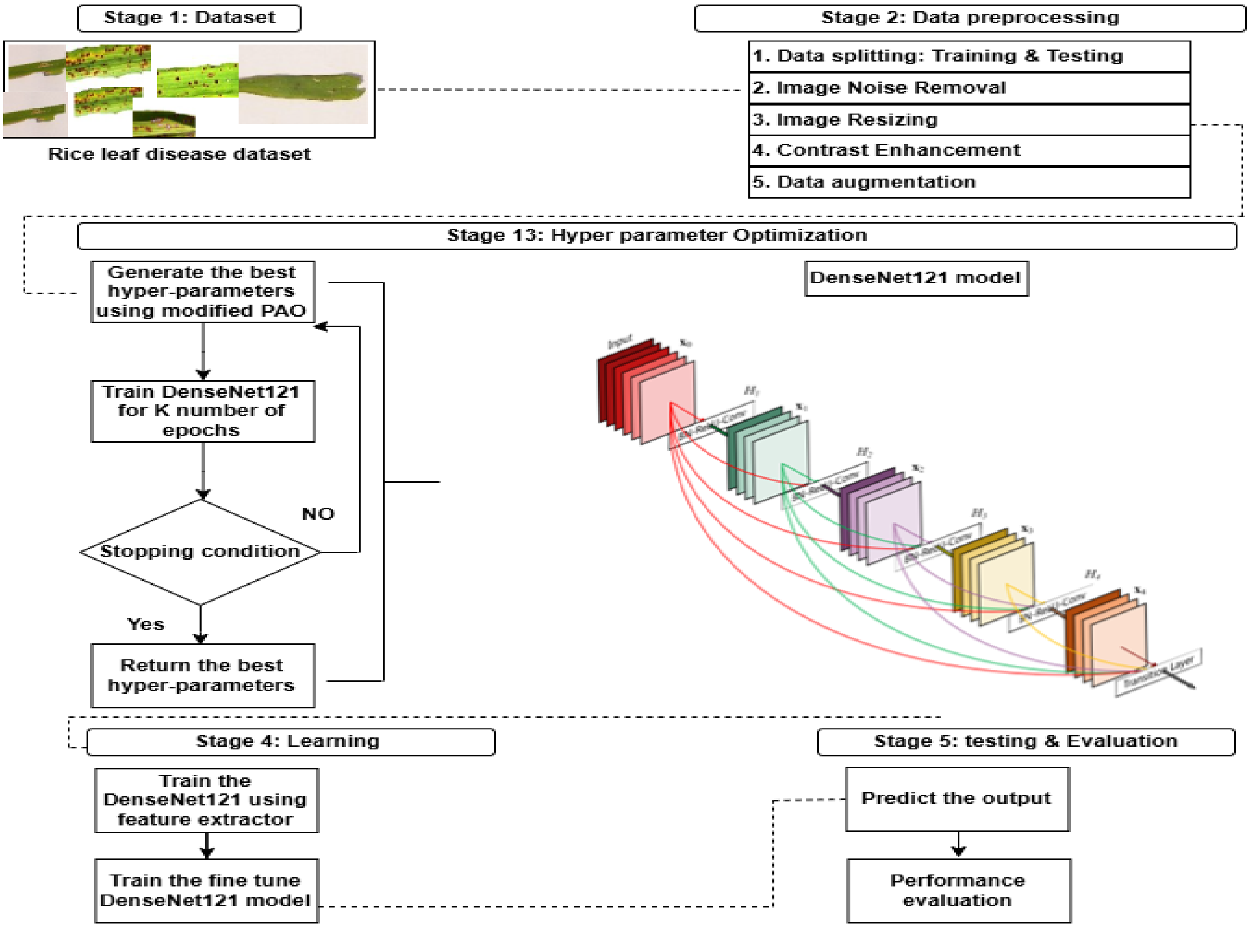
Stage 1: Dataset
Rice leaf disease dataset
Stage 2: Data preprocessing
1. Data splitting: Training & Testing
2. Image Noise Removal
3. Image Resizing
4. Contrast Enhancement
5. Data augmentation
Stage 13: Hyper parameter Optimization
Generate the best hyper-parameters using modified PAO
Train DenseNet121 for K number of epochs
NO
Stopping condition
Yes
Return the best hyper-parameters
DenseNet121 model
Input
Stage 4: Learning
Train the DenseNet121 using feature extractor
Train the fine tune DenseNet121 model
Stage 5: testing & Evaluation
Predict the output
Performance evaluation

Figure 2 The structural diagram of the proposed rice disease detection model.

6.1 Dataset acquisition

The data utilized in this work was from the publicly accessible Kaggle cloud data repository (Tejaswini et al., 2022). It comprises a total of 1,600 images distributed across four distinct classes: Hispa, Brown Spot, Leaf Blast, and Healthy leaves. The class spread of the data is depicted in Figure 3.

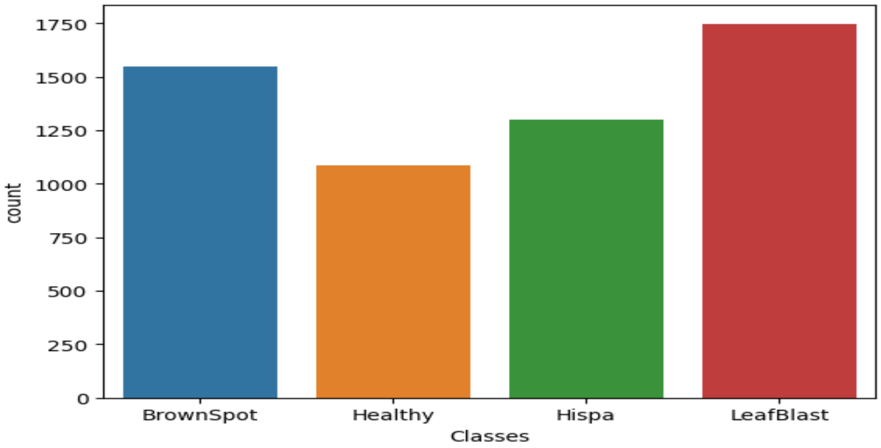

Figure 3 The Data distribution

6.2 Data Preprocessing

Images of crop leaf diseases are often characterized by significant noise and low contrast, which can hinder accurate disease detection. Since the clarity and sharpness of these images are critical for effective diagnosis, improving image clarity by eliminating noise frequencies is considered a useful strategy to enhance detection accuracy. Accordingly, this study investigates various image preprocessing techniques aimed at addressing these challenges and improving the quality of rice disease images for more precise and reliable detection.

6.2.1 Noise reduction

Digital imaging plays a crucial role in image processing, but noise from acquisition devices can hinder analysis, especially in rice disease classification. Reducing noise enhances signal quality and classification accuracy. Identifying noise sources and applying effective reduction techniques is essential (Huang et al., 2024). Nonlinear filters like the median filter are particularly effective, preserving edges while suppressing noise by replacing each pixel with the median of its neighbors. This technique involves replacing each pixel median rate calculated from its surrounding pixels, as mathematically expressed in Equation 14.

$$\boldsymbol{X_{(a,b)} = median(y_{i,j} : i, j \in N)} \quad (14)$$

where $\boldsymbol{N}$ defines the surrounding neighborhood at location $(\boldsymbol{a}, \boldsymbol{b})$.

6.2.2 Image Normalization

Normalization is a fundamental aspect of preprocessing, mostly in tasks involving image resizing and brightness normalization. These processes play a critical role in standardizing pixel values, which significantly contributes to enhancing model convergence during training. In the initial stage of normalization, the brightness levels of input images are adjusted to lie within the range of zero to one, as outlined by Razmjooy et al. (2020). The mathematical formulation employed to achieve this brightness normalization is provided in Eqn. (15).

$$\boldsymbol{IC_K = (IC - IC_{min}) \times \frac{IC_{K\,max} - IC_{K\,min}}{IC_{max} - IC_{min}} + IC_{K\,min}} \quad (15)$$

where $\boldsymbol{IC}$ is the input image limited to the range of $\boldsymbol{IC_{max}}$ and $\boldsymbol{IC_{min}}$, and $\boldsymbol{IC_K}$ contains the new adjusted image, having its limits described by $\boldsymbol{IC_{K\,max}}$ and $\boldsymbol{IC_{K\,min}}$. The images in the dataset are scaled down to 224 × 224 each.

6.2.3 Contrast enrichment

Contrast is vital for image quality, representing brightness variability. Low contrast compresses tonal range, causing blurriness. Enhancing contrast increases tonal variation and sharpness. The methodologies and algorithms discussed in this study leverage histogram correction techniques to address challenges associated with low image contrast. Specifically, the histogram equalization method is identified as an effective approach to rectify these issues (Razmjooy et al., 2020).

6.2.4 Dataset augmentation

Data augmentation is crucial in deep learning for overcoming limited data and class imbalance, particularly in binary classification where underrepresented (minority) classes can critically impact model performance. In crop leaf disease detection, scarce samples of key disease classes exacerbate these challenges. By artificially generating new images through techniques such as rotation, flipping, scaling, and color jitter data augmentation increases dataset size, balances class distributions, and enhances the model's ability to learn from minority classes. This preprocessing step effectively mitigates the risks of poor generalization and high misclassification costs associated with imbalanced datasets. In this study, a number of augmentation approaches are implemented as shown in Table 2.

Table 2 Augmentation approaches and their parameter values

Approaches	Values
Shearing	0.2

Zooming	0.25
Width shift	0.2
Height shift	0.2
Rotation	10
Feature-wise Centering	True
Fill mode	Reflect
Vertical flip	True
Horizontal flip	True

6.3 Optimization of Hyper-parameters

To identify the most suitable pre-trained CNN for rice leaf disease classification, several models were evaluated, including VGG16, DenseNet121, Xception, InceptionV3, and MobileNet, with DenseNet121 achieving the best performance. Consequently, DenseNet121 was selected and fine-tuned using transfer learning by replacing its classifier and optimizing four key hyperparameters, including learning rate, batch size, dropout rate, and dense layer size. Together, these parameters define a 4-dimensional search space, with each point representing a unique mixture of hyperparameter values optimized to improve model performance.

6.4 Training Stage

In this stage, DenseNet121 is applied to the rice leaf disease dataset using a blend of feature extraction and fine-tuning methods. Initially, the network serves as a feature extractor, keeping its convolutional layers frozen to prevent updates during training. The added classifier, however, is trained on the enlarged dataset using the features extracted from frozen layers. This step aligns with the third phase described in Algorithm 3. To maintain consistency, the same data augmentation techniques as in earlier stages are used to enhance sample diversity and introduce new feature variations. After the feature extraction process reaches a performance plateau with no significant improvement over several training epochs, fine-tuning is initiated for further optimization. The fine-tuned classifier is composed of four key layers: a flattening layer, a dense layer, a dropout layer, and a final dense layer. The initial dense layer uses the ReLU activation function, with the size of neurons and the dropout factor determined by the mPOA. The output layer, designed for multi-class classification, includes four neurons and employs the softmax activation function. This process corresponds to the second stage outlined in Algorithm 3. At the optimization stage, all the last four layers of the DenseNet121 convolutional backbone are frozen, allowing only these final layers and the added classifier to be trainable. These components are trained simultaneously, as detailed in the fourth phase of Algorithm 3. The enhanced model undergoes additional training across multiple epochs until it achieves stable performance, marking the point where further improvements cease to occur.

To mitigate the risk of overfitting during model training, several regularization techniques were employed. These include data augmentation (such as flipping, rotation, and scaling) as describe in table 2, dropout layers (with optimized rates determined by mPOA), and early stopping based on validation loss trends. The model’s robustness was further validated by analyzing the convergence behavior between training and validation curves, as illustrated in Figure 4. The close alignment of these curves indicates reduced overfitting and strong generalization performance.

```
Algorithm 3  Learning Stage of Proposed Rice Leaf Diseases (RLD) Detection Model
Input:  Training set ($RLD\_D_{train}$), test set ($RLD_{D_{test}}$), hyperparameter values
Output:  RLD Trained model
    // Phase 1:  Preprocessing Stage
    Perform data augmentation on $RLD\_D_{traain}$ to generate  $RLD\_D_{agtraain}$
    Resize images in $RLD\_D_{agtraain}$ to form $RLD\_D_{agtraain}$
    Resize images in $\mathrm{RLD}_{\mathrm{D}_{\mathrm{test}}}$ to form $\mathrm{RLD}_{\mathrm{Dn}_{\mathrm{test}}}$
    Reduce noise in $\mathrm{RLD}_{\mathrm{D}_{\mathrm{agtrain}}}$
    Enhance contrast of images in $\mathrm{RLD}_{\mathrm{D}_{\mathrm{agtrain}}}$
    // Phase 2: Building DenseNet121 Model
    Load DenseNet121 without top layers as $Conv\_part$
    for each layer in $Conv\_part$ do
        Set $Layer.trainable = False$
    end for
    Add $Conv\_part$ to the model
    Add Flatten layer to the model
    Add a Dense layer to the model
    Add Dropout layer to the model
    Add a Dense layer to the model
    //Phase 3: Training DenseNet121 Model using Feature Extraction Method
    for each epoch in range(1, n) do
        Train the fine − tuned DenseNet121 on $\mathrm{RLD}_{\mathrm{D}_{\mathrm{agtrain}}}$ using $\mathrm{RLD}_{\mathrm{Dn}_{\mathrm{test}}}$
    end for
    //Phase 4: Training DenseNet121 Model using Fine − Tuning Method
    for each layer in $Conv\_part.layers[: -4]$ do
        Set $Layer.trainable = False$
    end for
    for each layer in $Conv\_part.layers[-4: ]$ do
        Set $Layer.trainable = True$
    end for
    for each epoch in range(1, n) do
        Train the fine − tuned DenseNet121 on $\mathrm{RLD}_{\mathrm{D}_{\mathrm{agtrain}}}$ using $\mathrm{RLD}_{\mathrm{D}_{\mathrm{ntest}}}$
    end for
```

6.5 Testing and Evaluation

Five evaluation measures are used in this stage: F-score, Accuracy, Sensitivity, Specificity, and Precision. The effectiveness of the suggested rice leaf disease classification model is evaluated using these metrics, which are frequently employed in classification issues (Skhvediani et al., 2023; Nugroho et al., 2023; Rahayu et al., 2025)

Accuracy: shows the percentage of samples that have been appropriately identified (Hassan et al. 2022) and is calculated using Equation 16

$$\frac{\boldsymbol{trP+trN}}{\boldsymbol{trP+trN+faP+faN}} \qquad (16)$$

Recall: measures how well a model can find all relevant instances in a dataset and is calculated using Equation 17.

$$\frac{\boldsymbol{trP}}{\boldsymbol{trP+faN}} \qquad (17)$$

Precision: measures the precision of positive classification and is determined using Equation 18

$$\frac{\boldsymbol{trP}}{\boldsymbol{trP+faP}} \qquad (18)$$

F-score: F-score is a measure of test accuracy and is calculated by Equation 19.

$$\frac{\boldsymbol{2 \times recall \times precision}}{\boldsymbol{recall+precision}} \qquad (19)$$

where *trP* donate number of true positive predictions, *trN* define true negative predictions, *faP* represent false positive predictions, and *faN* depict false negative predictions.

7 Performance evaluation of rice disease detection model

This segment provides an analysis of the results achieved using the proposed deep learning framework for disease detection rice leaf. The entire methodology was executed using the Kaggle notebook environment and implemented in Python, leveraging the TensorFlow library for model development and evaluation.

7.1 Optimizing hyperparameters using the modified POA

This section details the optimization ranges for various hyperparameters tuned using the mPOA algorithm. Table 3 outlines the parameter arrangements for integrating the DenseNe121 with the mPOA, emphasizing the critical important of each hyperparameter in enhancing the effectiveness of model. The optimization algorithm's maximum number of iterations was set to 50, with a population size of 30, representing the count of candidate solutions. The search space dimension, defined by the four hyperparameters, enables their simultaneous optimization. The learning rate, which governs weight adjustments during training, was constrained to a range of $1e{-}7$ to $1e{-}3$, ensuring controlled updates that preserve previously learned features. The batch size, representing the number of samples processed per iteration, had a search range between 1 and 64. The dropout rate, used for regularization, was varied between 0.1 and 0.9 to prevent overfitting. The number of neurons in the initial dense layer was optimized within a range of 50 to 550, allowing for flexible model capacity adjustments. To determine the optimal count of DenseNet121 training epochs, several values were evaluated. It was found that fewer than 16 epochs led to suboptimal accuracy; thus, the training epochs were set to 16. The key goal of using the mPOA was to lessen the validation loss. The test set loss degree after 16 training epochs served as the benchmark for evaluating the proposed method's effectiveness. After training, the optimal values for the hyperparameters: dropout rate, batch size, and the number of neurons in the initial dense layer are determined. The final optimized hyperparameter values determined by the POA were as follows: a learning rate of 0.0001, a dropout rate of 0.1, a batch size of 0.9, and 125 neurons in the first dense layer. The findings depict the efficacy of the modified POA in enhancing the DenseNet121 for improved efficiency. Table 3 presents the parameter settings for the mPOA integrated with DenseNet121.

Table 3 mPOA-DenseNet121 parameter settings

Parameter	Value
Maximum iteration count	50
Size of the Population	30
Dimension	4
st	[1,4]
β	1.5
Learning factor (α)	
Batch Size	[1,64]
Dropout factor	[0.1,0.9]
Count of Neurons	[50,550]
Maximum Training epochs of DenseNet121	16

Table 4 Optimum values of DenseNet121 hyperparameters found using mPOA

Hyperparameter	Optimum value
Learning rate (α)	0.0001
Batch size	8

Dropout rate	0.1
Count of neurons	120

7.2 DenseNet121 Training using Optimum Values

Here, the enhanced DenseNet121 is trained using the optimal hyper-parameters determined using the mPOA, serving as a feature extractor for the augmented training dataset. The model's effectiveness was subsequently assessed on the validation dataset across a maximum of 16 epochs. To mitigate overfitting, the training process was halted early if no progress was detected over 10 consecutive epochs. This approach utilized the early stopping technique, as described by Prechelt (2002) and Bai et al. (2021). Given that the rice leaf disease dataset involves multi-class classification, the categorical cross-entropy loss was employed to optimize the DenseNet121 model. Adam optimizer with a 2×10-7 learning used at the feature extraction stage. A decay learning rate schedular was implemented at the fine-tuning stage, using the methodologies outlined by Iiduka (2021). This plan began with an initial learning rate that decreased by a factor of 0.2 after every ten training epochs. The decision to adopt a smaller learning factor during fine-tuning aimed to maintain the essential features learned during the feature extraction stage while minimizing substantial alterations. This ensures that the extracted knowledge from the initial phase is retained, enhancing the overall model performance.

7.3 Assessing the efficacy of the presented model

This section provides a comprehensive assessment of the presented model efficacy. To demonstrate the effectiveness of the mPOA in determining the optimum value of DenseNet121 hyperparameters for the model and achieving superior accuracy, its results are compared to those obtained from a DenseNet121 configured using manual hyper-parameter tuning. The hyperparameter settings optimized by the mPOA are detailed in Table 4, while the manually tuned DenseNet121 model employed a batch size of 8, a 0.72 dropout factor, 120 neurons, and a 0.001 starting learning rate. The efficacy of the DenseNet121 enhanced with the mPOA was assessed using standard metrics such as accuracy, precision, recall and f-measure. Table 5 presents these metrics, offering a detailed assessment of model performance. For the manually tuned DenseNet121 model, the results include an accuracy of approximately 94.8%, precision of 94.8%, 94.8% recall and 94.23% F-measure. Conversely, the DenseNet121 optimized with the mPOA demonstrated noticeably superior efficacy across all metrics. Specifically, the model achieved an accuracy of approximately 98.5%, 98.6% precision, 98.4%, recall and an 98.5% F-measure. These findings depict the efficacy of the mPOA in enhancing the DenseNet121, enabling improved accuracy and reliability for rice leaf disease classification tasks. The considerable enhancement in performance metrics demonstrates the advantage of employing the mPOA over manual hyperparameter tuning. Figure 4(a) and 4(b) illustrate the training and validation accuracy and loss curves of the presented model respectively, providing further evidence of its robust and reliable performance.

Table 5 Comparison of classification results between mPOA optimized DenseNet121 and original POA enhanced DenseNet121

Measures	mPOA optimized DenseNet121	DenseNet121
Accuracy	98.5	96.8
Precision	98.6	96.7
Recall	98.4	96.7
F-measure	98.5	96.7

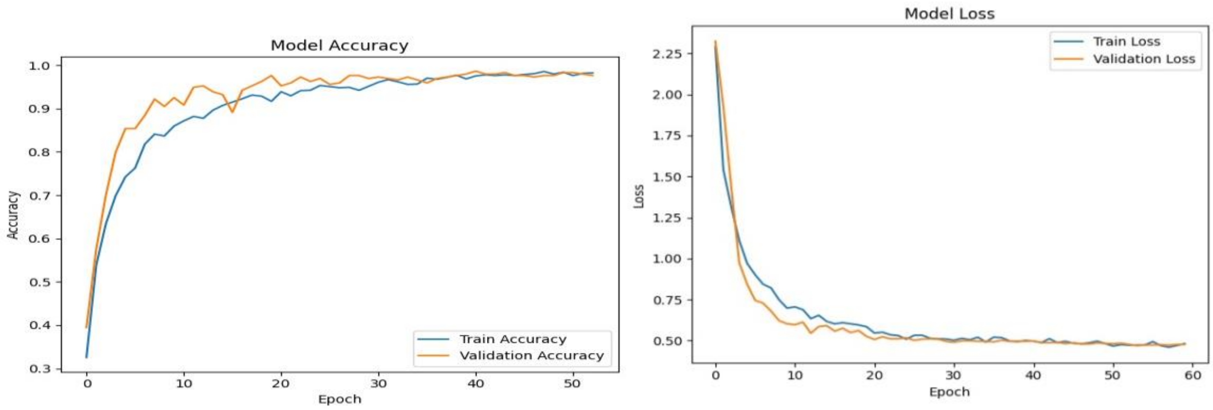

Figure 4 (a) Proposed model training and validation accuracy (b)

7.4 Performance assessment of mPOA DenseNet121 model and original POA DenseNet121 model

This section presents a comparative analysis of the optimized DenseNet121 using the modified parrot optimization algorithm (POA) and the original POA in the context of rice leaf disease classification. Both models were assessed based on their respective hyperparameter configurations and the classification efficacy, as outlined in Table 6. Notably, both approaches share similar learning factor of 0.0001.

The DenseNet121 enhanced with mPOA showcases notable improvements in hyperparameter configurations compared to the version optimized by the original POA. For example, the mPOA uses a lesser batch size of 8, as opposed to 13 in the original model, signifying a more streamlined training procedure with less examples per repetition. Furthermore, the dropout factor is significantly lowered to 0.1 in the modified model, compared to 0.84 in the original, demonstrating improved regularization and a reduced risk of overfitting. Additionally, the size of neurons in the first dense layer is decrease to 120 in the mPOA model, down from 180 in the original model, reflecting a more compact and efficient feature representation. These refined hyperparameter configurations contribute to significant enhancements in the classification performance of the mPOA-optimized DenseNet121 model. Evaluation measures such as accuracy, precision, recall, and F-measure show notable improvements over those achieved by the original POA-optimized model.

In summary, the findings underscore the effectiveness of the mPOA in optimizing DenseNet121 for rice leaf disease classification, leading to superior performance outcomes. Fig. 5 provides a graphical comparison of the performance metrics for the mPOA enhanced DenseNet121, the original POA enhanced DenseNet121, and the standard DenseNet121, further illustrating the efficacy of the proposed approach. However, While the mPOA introduces additional computation during training, the final DenseNet121-based model remains efficient at inference time, suitable for deployment in constrained environments. The proposed model achieves a balance between accuracy and computational cost, with fewer parameters (7.98 million) parameters than VGG16 and faster inference (22 ms) than ResNet50, making it practical for real-world applications.

Table 6 Comparison of classification results between mPOA optimized DenseNet121 and POA

Measures	MPOA optimized DenseNet121	POA optimized DenseNet121
Accuracy	98.5	97.2
Precision	98.6	97.0
Recall	98.4	97.2
F-measure	98.5	97.1

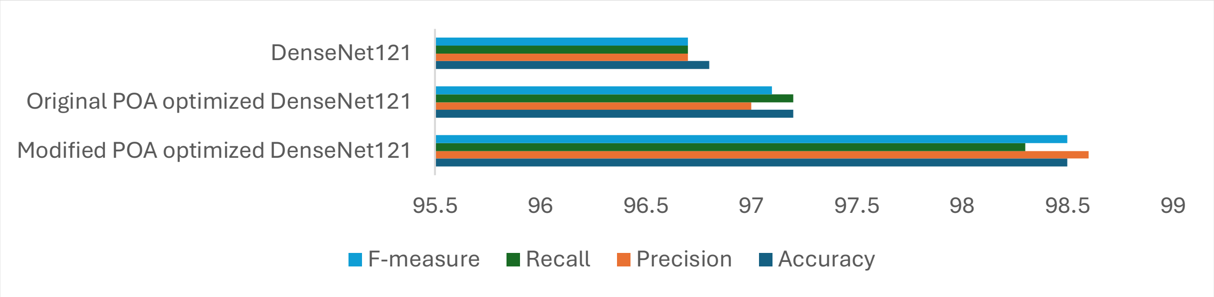

Figure 5 Comparison of mPOA DensNet121 with POA-DenseNet121 and original DenseNet121

7.5 Performance assessment of mPOA DenseNet121 model with other pre-trained models

This section presents a comparative evaluation of the proposed mPOA-DenseNet121 model for rice leaf disease detection against several widely recognized pretrained deep learning models, including VGG19, EfficientNetB0, InceptionV3, DenseNet201, and ResNet50. These models were chosen regarding their proven performance in earlier research addressing disease detection challenges. The objective is to judge the robustness and efficacy of the presented approach by contrasting its performance with those of benchmark models. This analysis provides a comprehensive evaluation of the proposed model's capability in rice leaf disease detection relative to established methods in the field. Table 7 provides a detailed comparison of the mPOA-DenseNet121 with the aforementioned pretrained deep learning models. The results reveal that the presented model surpasses all others across all evaluated metrics, showcasing its exceptional performance in accurately detecting rice leaf diseases. Notably, EfficientNetB0, InceptionV3, and ResNet50 emerge as the next best-performing models, achieving accuracies of 95.2%, 94.8%, and 94.6%, respectively. However, these results are still outperformed by the proposed mPOA-DenseNet121 model, underscoring its superiority. This significant improvement across all performance metrics highlights the capability of the mPOA-DenseNet121 model in achieving precise classification of rice leaf disease images. The graphical representation of this comparative analysis is illustrated in Fig. 6, further emphasizing the effectiveness of the presented method over existing methods.

Although an exhaustive hyperparameter search could, in theory, yield marginal improvements, such approaches are computationally intensive and often impractical for real-world applications. The use of the modified POA in this study achieves a balance between performance and resource efficiency. We define a model as "good enough" when it consistently achieves over 98% in accuracy, precision, recall, and F-measure levels suitable for field deployment. The lower bounds of model usefulness emerge in scenarios involving poor image quality, occlusions, or mixed infections, which may impact classification accuracy. Meanwhile, the performance observed in this study reflects an upper bound based on the current dataset and optimization scope.

Table 7 Comparison of classification results between mPOA enhance DenseNet121 with other pre-trained models

Model	Accuracy	Precision	Recall	F-measure
ResNet50	94.6	94.7	94.6	94.6
InceptionV3	94.8	94.8	94.8	94.8
VGG19	84.9	84.7	84.7	84.7
EfficientNetB0	95.2	95.1	95.1	95.1
DenseNet212	90.2	90.2	90.2	90.2

mPOA DensNet121	98.5	98.6	98.4	98.5

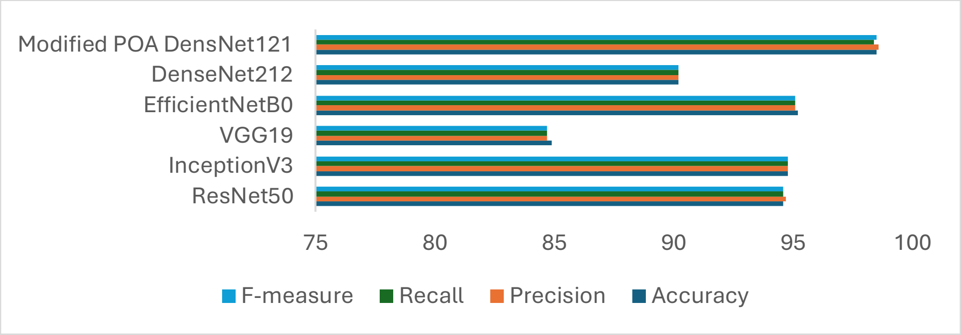

Figure 6 Comparison of mPOA DensNet121 with other pre-trained models

8 Conclusion and future work

This work introduces a mPOA-DenseNet121 model tailored for rice leaf disease detection, leveraging developments in deep learning and optimization. The proposed model demonstrated outstanding performance in all the evaluation metrics considered in contrast to popular pre-trained models such as EfficientNetB0, DenseNet201, VGG19, InceptionV3, and ResNet50. This outstanding performance highlights the efficacy of the mPOA in optimizing critical hyperparameters to enhance the classification accuracy of DenseNet121. The findings underscore the potential of the proposed approach in addressing challenges associated with rice leaf disease detection, offering a reliable and accurate tool for agricultural diagnostics. Furthermore, the practical use of this model extends to the development of lightweight mobile or embedded devices that can assist farmers with disease detection and suggest targeted treatment strategies, such as fungicide application, irrigation changes, or crop rotation, thereby improving agricultural decision-making.

Despite the high accuracy, one limitation is the potential decline in model performance when exposed to entirely different rice varieties or novel environmental conditions.

Building on the current findings, several avenues for future research include, (1) evaluating the model's performance on diverse crop disease datasets to generalize its applicability across agricultural domains, (2) exploring hybrid optimization algorithms that combine POA with other metaheuristic approaches to further enhance model performance, (3) developing lightweight versions of the model suitable for deployment on edge devices with limited computational resources and (4) extending the framework to include disease severity estimation, providing farmers with actionable insights for targeted interventions.

Acknowledgement

The authors would like to acknowledge the support of AIDA Lab CCIS Prince Sultan University, Riyadh Saudi Arabia for APC support.

Author Contributions

Ibrahim Hayatu Hassan: Concept, Design, Methodology, software Writing-original draft. Tanzila Saba: Concept, Review. Salma Idris Implementation, Review. Anees Ara: Design, Review, Funding sourcing. Salma Idris: Methodology, Review. Anees Ara: Concept, review, founding sourcing.

Conflict of Interest

The authors declare no conflicts of interest

References

Adamu, A., Abdullahi, M., Junaidu, S. B., & Hassan, I. H. (2021). An hybrid particle swarm optimization with crow search algorithm for feature selection. Machine Learning with Applications, 6, 100108, https://doi.org/10.1016/j.mlwa.2021.100108

Ayesha, H., Iqbal, S., Tariq, M., Abrar, M., Sanaullah, M., Abbas, I., ... & Hussain, S. (2021). Automatic medical image interpretation: State of the art and future directions. Pattern Recognition, 114, 107856.

Babu, R. R., & Philip, F. M. (2024). Optimized deep learning for skin lesion segmentation and skin cancer detection. Biomedical Signal Processing and Control, 95, 106292, https://doi.org/10.1016/j.bspc.2024.106292

Bai, Y., Yang, E., Han, B., Yang, Y., Li, J., Mao, Y., Niu, G. and Liu, T., 2021. Understanding and improving early stopping for learning with noisy labels. Advances in Neural Information Processing Systems, 34, pp.24392-24403.

Barman, U., Das, D., Sonowal, G., & Dutta, M. (2024). Innovative approaches to rice (Oryza sativa) crop health: a comprehensive analysis of deep transfer learning for early disease detection. Yuzuncu Yil Univ.J.Agricult.Sci., 34(2), 314–322, https://doi.org/10.29133/yyutbd.1402821

Barakat, M., Chung, G. C., Lee, I. E., Pang, W. L., & Chan, K. Y. (2023). Detection and sizing of durian using zero-shot deep learning models. International Journal of Technology, 14(6), 1206-1215, https://doi.org/10.14716/ijtech.v14i6.6640

Chakrabarty, A., Ahmeda, S. T., Islam, M. F., Aziz, S. M., & Maidin, S. S. (2024). An interpretable fusion model integrating light weight CNN and transformer architectures for rice leaf disease identification. EcologicalInformatics, 82(2024), 102718, https://doi.org/10.1016/j.ecoinf.2024.102718

Chen, J., Zhang, D., Nanehkaran, Y., & Li, D. (2020). Detection of rice plant diseases based on deep transfer learning. J. Sci. Food Agric., 100(7), 3246-3256, https://doi.org/10.1002/jsfa.10365

Daniya, T., & Vigneshwari, S. (2023). Rider Water Wave-enabled deep learning for disease detection in rice plant. Advances in Engineering Software, 182(2023), 103472, https://doi.org/10.1016/j.advengsoft.2023.103472

Ebraheem, F., Radhwan, A. S., Humam, A., Abdulgbar, A. F., & Mugahed, A. A.-a. (2024). A hybrid deep learning skin cancer prediction framework. Engineering Science and Technology, an International Journal, 57, 101818, https://doi.org/10.1016/j.jestch.2024.101818

Emam, M. M., Houssein, E. H., Samee, N. A., Alohali, M. A., & Hosney, M. E. (2024). Breast cancer diagnosis using optimized deep convolutional neural network based on transfer learning technique and improved Coati optimization algorithm. Expert Systems with Applications, 255(2024), 124581, https://doi.org/10.1016/j.eswa.2024.124581

Faramarzi, A., Heidarinejad, M., Mirjalili, S., & Gandomi, A. H. (2020). Marine predators' algorithm: A nature-inspired metaheuristic. Expert Systems with Applications, 113377, https://doi.org/10.1016/j.eswa.2020.113377

Gaspar, A., Oliva, D., Cuevas, E., Zaldívar, D., Pérez, M., & Pajares, G. (2021). Hyperparameter optimization in a convolutional neural network using metaheuristic algorithms. Metaheuristics in machine learning: theory and applications (pp. 37–59). Springer, https://doi.org/10.1007/978-3-030-70542-8_2

Goluguri, N. V., Devi, K. S., & Srinivasan, P. (2021). Rice-net: an efficient artificial fish swarm optimization applied deep convolutional neural network model for identifying the Oryza sativa diseases. Neural Computing and Applications, 33(2021), 5869–5884, https://doi.org/10.1007/s00521-020-05364-x

Hassan, I. H., Abdullahi, M., Aliyu, M. M., Yusuf, S. A., & Abdulrahim, A. (2022). An improved binary manta ray foraging optimization algorithm based feature selection and random forest classifier for network intrusion detection. Intelligent Systems with Applications, 16, 200114, https://doi.org/10.1016/j.iswa.2022.200114

Huang, Q., Ding, H., & Razmjooy, N. (2024). Oral cancer detection using convolutional neural network optimized by combined seagull optimization algorithm. Biomedical Signal Processing and Control, 57, 105546, https://doi.org/10.1016/j.bspc.2023.105546

Hossain, S., Seyam, T.A., Chowdhury, A., Ghose, R., Rahaman, A., Hadika, Z. and Pathak, A., 2025. Enhancing Agricultural Diagnostics: Advanced Training of Pre-Trained CNN Models for Paddy Leaf Disease Detection. Machine Learning, 10(1), pp.1-13, https://doi.org/10.11648/j.mlr.20251001.11

Ibrahim, A. T., Abdullahi, M., Kana, A. F. D., Mohammed, M. T., & Hassan, I. H. (2024). Categorical classification of skin cancer using a weighted ensemble of transfer learning with test time augmentation. Data Science and Management, 2024, 1-32, https://doi.org/10.1016/j.dsm.2024.10.002

Iiduka, H. (2021). Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. IEEE Transactions on Cybernetics, 52(12), 13250-13261, https://doi.org/10.1109/TCYB.2021.3107415

Javed, R., Saba, T., Alahmadi, T. J., Al-Otaibi, S., AlGhofaily, B., & Rehman, A. (2024). EfficientNetB1 Deep Learning Model for Microscopic Lung Cancer Lesion Detection and Classification Using Histopathological Images. Computers, Materials & Continua, 81(1), 810-825. doi:10.32604/cmc.2024.052755

Lian, J., Hui, G., Ma, L., Zhu, T., Wu, X., Heidari, A.A., Chen, Y. and Chen, H., 2024. Parrot optimizer: Algorithm and applications to medical problems. Computers in Biology and Medicine, 172, p.108064, doi: 10.1016/j.compbiomed.2024.108064

Maijeddah, U. I., Yusuf, S. A., Abdullahi, M., & Hassan, I. H. (2024). A hybrid transfer learning model with optimized SVM using honey badger optimization algorithm for multi-class lung cancer classification. Science World Journal, 19(4), 977-986. doi: 10.4314/swj.v19i4.10

Manjupriya, R., & Leema, A. A. (2025). Efficient Epileptic Seizure Detection with Optimal Channel Selection and FIXUPPACTBI-LSTM Deep Learning Model. International Journal of Technology, 16(2), 706-721, https://doi.org/10.14716/ijtech.v16i2.7333

Mofrad, F. B., & Valizadeh, G. (2023). DensNet-based transfer learning for lv shape classification: Introducing a novel information fusion and data augmentation using statistical shape/color modelling. Expert Systems with Applications, 213, 119261. doi:10.1016/j.eswa.2022.119261

Mahmmod, T., Ayesha, N., Mujahid, M., & Rehman, A. (2024, March). Customized Deep Learning Framework with Advanced Sampling Techniques for Lung Cancer Detection using CT Scans. In 2024 Seventh International Women in Data Science Conference at Prince Sultan University (WiDS PSU) (pp. 110-115). IEEE.Nugroho, Y. N., Harwahyu, R., Sari, R. F., Nikaein, N., & Cheng, R. G. (2023). Performance Evaluation of Anomaly Detection System on Portable LTE Telecommunication Networks Using Open Air Interface and ELK. International journal of technology, 14(3), 549-560. doi:10.14716/ijtech.v14i3.4237

Pattnaik, G., Shrivastava, V.K. and Parvathi, K., 2021. Tomato pest classification using deep convolutional neural network with transfer learning, fine tuning and scratch learning. Intelligent Decision Technologies, 15(3), pp.433-442. doi: 0.3233/IDT-200192

Prechelt, L., 2002. Early stopping-but when? In Neural Networks: Tricks of the trade (pp. 55-69). Berlin, Heidelberg: Springer Berlin Heidelberg, https://doi.org/10.1007/3-540-49430-8_3

Preethi, P., Swathika, R., Kaliraj, S., Premkumar, R. and Yogapriya, J., 2024. Deep Learning–Based Enhanced Optimization for Automated Rice Plant Disease Detection and Classification. Food and Energy Security, 13(5), p.e70001, https://doi.org/10.1002/fes3.70001

Rao, N.R. and Vasumathi, D., 2024. Segmentation and detection of skin cancer using deep learning-enabled artificial Namib beetle optimization. Biomedical Signal Processing and Control, 96, p.106605, https://doi.org/10.1016/j.bspc.2024.106605

Rahayu, D. S., Husodo, Z. A., Pidanic, J., Li, X., & Suhartanto, H. (2025). A Technique to Predict Bankruptcy Using Ultimate Ownership Network as Key Indicators. International Journal of Technology, 16(1), 275-288. doi: https://doi.org/10.14716/ijtech.v16i1.7516

Razmjooy, N., Ashourian, M., Karimifard, M., Estrela, V.V., Loschi, H.J., Do Nascimento, D., França, R.P. and Vishnevski, M., 2020. Computer-aided diagnosis of skin cancer: a review. Current Medical Imaging Reviews, 16(7), pp.781-793, https://doi.org/10.2174/1573405616666200129095242

Ritharson, P.I., Raimond, K., Mary, X.A. and Robert, J.E., 2024. DeepRice: A deep learning and deep feature-based classification of Rice leaf disease subtypes. Artificial Intelligence in Agriculture, 11, pp.34-49, https://doi.org/10.1016/j.aiia.2023.11.001

Rehman, A., Kashif, M., Abunadi, I., & Ayesha, N. (2021, April). Lung cancer detection and classification from chest CT scans using machine learning techniques. In 2021 1st international conference on artificial intelligence and data analytics (CAIDA) (pp. 101-104). IEEE.

Khan, A. R., Abunadi, I., Alghofaily, B., Ali, H., & Saba, T. (2023). Automatic diagnosis of rice leaves diseases using hybrid deep learning model. Journal of Advances in Information Technology, 14(3), 418-425. doi:10.12720/jait.14.3.418-425

Nugroho, Y. N., Harwahyu, R., Sari, R. F., Nikaein, N., & Cheng, R. G. (2023). Performance Evaluation of Anomaly Detection System on Portable LTE Telecommunication Networks Using Open Air Interface and ELK. International journal of technology, 14(3), 549-560. doi:10.14716/ijtech.v14i3.4237

Rahayu, D. S., Husodo, Z. A., Pidanic, J., Li, X., & Suhartanto, H. (2025). A Technique to Predict Bankruptcy Using Ultimate Ownership Network as Key Indicators. International Journal of Technology, 16(1), 275-288. doi: https://doi.org/10.14716/ijtech.v16i1.7516

Rehman, A. (2023, December). Brain stroke prediction through deep learning techniques with ADASYN strategy. In 2023 16th International Conference on Developments in eSystems Engineering (DeSE) (pp. 679-684). IEEE.

Shah, S.R., Qadri, S., Bibi, H., Shah, S.M.W., Sharif, M.I. and Marinello, F., 2023. Comparing inception V3, VGG 16, VGG 19, CNN, and ResNet 50: A case study on early detection of a rice disease. Agronomy, 13(6), p.1633, https://doi.org/10.3390/agronomy1306163

Skhvediani, A., Rodionova, M., Savchenko, N., & Kudryavtseva, T. (2023). Prediction of the Road Accidents Severity Level: Case of Saint-Petersburg and Leningrad Oblast. International Journal of Technology, 14(8), 1717-1727. doi: https://doi.org/10.14716/ijtech.v14i8.6859

Wolpert, D.H. and Macready, W.G., 1997. No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), pp.67-82, https://doi.org/10.1109/4235.585893

Yuan, Y., Chen, L., Wu, H. and Li, L., 2022. Advanced agricultural disease image recognition technologies: A review. Information Processing in Agriculture, 9(1), pp.48-59, https://doi.org/10.1016/j.inpa.2021.01.003

Yusuf, H.M., Ali, Y.S., Abubakar, A.H., Abdullahi, M. and Hassan, I.H., 2024. A systematic review of deep learning techniques for rice disease recognition: Current trends and future directions. Franklin Open, p.100154, https://doi.org/10.1016/j.fraope.2024.100154