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,*], Saeed Ali Bahaj [3]

[1]Department of Computer Science, Ahmadu Bello University, Zaria 810107, Nigeria
[2]Artifical Intelligence & Data Analytics Lab. College of Computer and Information Sciences Prince Sultan University Riyadh, 11586 Saudi Arabia
[3]Department of Management Information System, College of Business Administration, Prince Sattam Bin Abdulaziz University, 11942, AlKharj, Saudi Arabia
**Corresponding author: tsaba@psu.edu.sa; Tel.: +966 11 494 8000*

Abstract: Timely and accurate identification of rice leaf diseases is critical for optimizing crop productivity and safeguarding global food security. This study 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. The proposed model demonstrates superior performance by effectively tuning critical hyperparameters, including batch size, learning rate, dropout rate, and the number of neurons. Evaluations conducted on the RLD 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 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 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 laboratory tests, are

https://doi.org/10.14716/ijtech.v16i5.7766
Received April 2025; Revised May 2025; Accepted June 2025

time-consuming, error-prone, and costly, especially across large-scale farms (Naqi et al., 2025a; Shah et al., 2023).

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 (Chakrabarty et al., 2024; Pattnaik et al., 2021). Despite their effectiveness, training CNNs from scratch is resource-intensive and requires large annotated datasets that are often unavailable in the field of agriculture. Transfer learning mitigates this by fine-tuning pre-trained CNNs, enabling high performance even with limited data (Naqi et al., 2025b; Yuan et al., 2022; Ayesha et al., 2021).

Nevertheless, the performance of CNN depends heavily on optimal hyperparameter settings, such as learning rate, batch size, and number of filters, making hyperparameter tuning a critical step. Traditional methods, such as manual tuning, grid search, and random search, are often inefficient, computationally expensive, and time-consuming (Rehman et al., 2023; 2021). These challenges have driven researchers to reframe hyperparameter tuning as a complex optimization problem, which has been increasingly addressed through metaheuristic algorithms (Mohammed et al., 2025).

Inspired by natural phenomena such as evolution, swarm behavior, and physics, metaheuristic algorithms offer a strategic exploration of high-dimensional and nonlinear search spaces. Their adaptability and effectiveness have been widely demonstrated in deep learning model optimization for various applications. For example, Artificial Bee Colony (Farea et al., 2024), Artificial Namib Beetle Optimization (Rao and Vasumathi, 2024), and Jaya Artificial Ecosystem-Based Optimization (Babu and Philip, 2024) have been employed to fine-tune CNNs. However, the No Free Lunch (NFL) theorem (Wolpert and 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. Although effective, the original POA struggles with premature convergence and limited exploration. 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 to overcome these limitations.

The proposed approach combines CNNs with mPOA to optimize rice leaf disease detection using transfer learning. The method demonstrates superior accuracy, precision, recall, and F1-score across disease classes by fine-tuning a comprehensive set of CNN hyperparameters. This integration of deep learning and MHO 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 global food security efforts.

This study demonstrates that models developed for the detection of rice leaf diseases exhibit limited generalizability to novel environments and cultivars, primarily due to variations in leaf morphology and background conditions. Future work will incorporate multilocation datasets and domain adaptation. The potential of the model for real-time, mobile-based diagnostics offers practical benefits for timely agricultural interventions.

The key contributions of this study are as follows:

1. Various preprocessing methods, including contrast enhancement, normalization, noise reduction, and data augmentation, were applied to improve the leaf image quality for more effective detection of rice leaf diseases.

2. Proposed a modified POA incorporating mutated random opposition-based learning and Brownian motion strategies to enhance the performance of the traditional POA for superior optimization.

3. The modified POA was applied for efficient hyperparameter optimization of DenseNet121, thereby improving disease detection.
4. Thorough evaluation of the presented model using rice disease was performed to confirm its effectiveness.
5. A comparative analysis of the presented model in contrast to other advanced rice disease detection methods available in the literature was performed.

The manuscript is organized as follows: Section 2 provides a review of recent deep learning approaches for rice leaf disease detection. Section 3 describes the CNN architecture and the TL approach. Section 4 introduces POA and its mathematical formulations. Section 5 presents the modified POA. Section 6 outlines the proposed RLD detection model and its performance evaluation. Finally, 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 ML to advanced deep learning models. Goluguri et al. (2021) combined deep CNN with LSTM and artificial fish swarm optimization, achieving 97.5% accuracy but faced scalability issues due to computational demands. Daniya and Vigneshwari (2023) proposed a Rider Water Wave-based neural network that achieves 90.8% accuracy and is limited by single-dataset testing. Hossain et al. (2024) introduced a deep learning-based CP 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 that achieved 97.29% accuracy, which was 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 (Manjupriya and Leema, 2025; Mohammed et al., 2025; Barakat et al., 2023). 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 (Emam et al., 2024; Mahmmod et al., 2023; Gaspar et al., 2021). Transfer learning addresses data scarcity by leveraging pretrained models, such as DenseNet121, which uses dense connectivity to enhance feature reuse and mitigate vanishing gradients (Mofrad and Valizadeh, 2023). This study employs DenseNet121 with transfer learning, fine-tuning the classifier while optimizing hyperparameters using mPOA.

3.1. DenseNet pretrained models

DenseNet is a powerful convolutional neural network (CNN) architecture with 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, thereby 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 the 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 convolution, pooling, and normalization transition layers. These models are typically pre-trained on large datasets, such as ImageNet, to capture generalized visual features useful for

transfer learning. Figure 1 shows the architecture of DenseNet-121, as initially applied to the ImageNet dataset.

Transfer learning with DenseNet involves fine-tuning a pretrained 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, 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.

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

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		3×3 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		$1 \times 1\ conv$ 2×2 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		$1 \times 1\ conv$ 2×2 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		$1 \times 1\ conv$ 2×2 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 $pool$ $1000 - D$ fc, softmax		$1 \times 1 \times 1024$	$1 \times 1 \times 1664$ $1 \times 1 \times 100$	$1 \times 1 \times 1920$

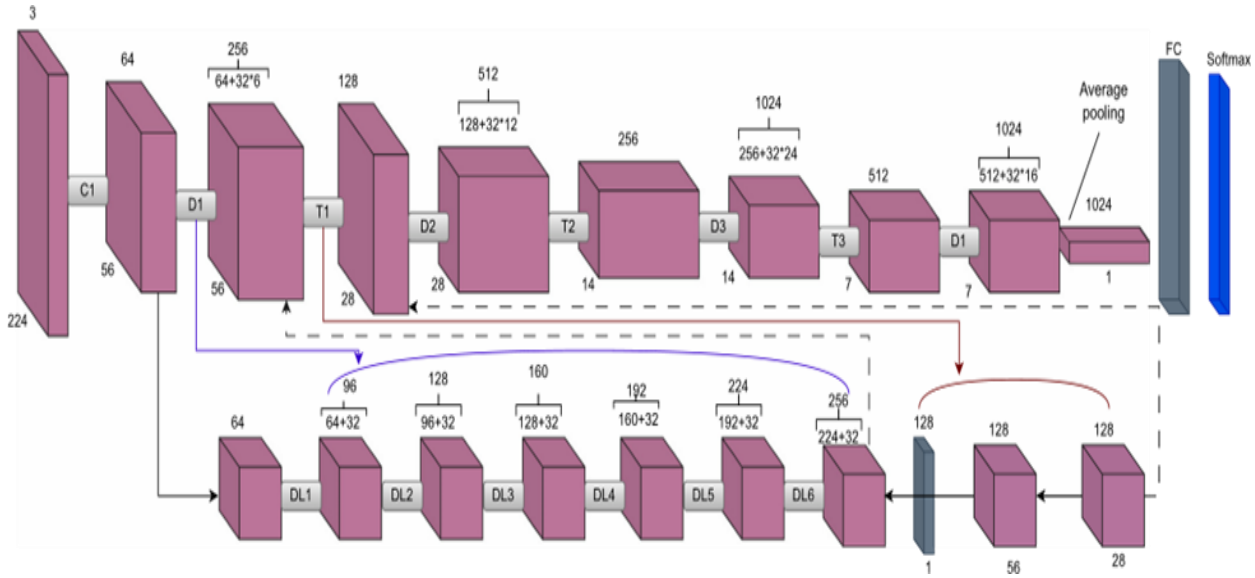

Figure 1 Main DenseNet-121 CNN architecture as applied to ImageNet data

4. Parrot Optimization Algorithm

The POA developed by Lian et al. (2024) is a novel and effective metaheuristic algorithm inspired by the behavioral traits of domesticated Pyrrhura Molinae parrots, such as foraging, remaining stationary, vocalizing, and exhibiting caution toward unfamiliar entities. These behavioral patterns

serve as the foundational principles for the POA development. This section presents an overview of the POA and its foundational mathematical framework.

4.1. Initial POA stage

The POA, 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 parameters' values, thereby defining a possible solution. Parameters such as the population size (N_pop), the highest number of iterations (MaxIter), and the search region boundaries denoted by the lower bound (lwb) and the upper bound (upb) are considered for POA initialization. This start process is mathematically expressed in Equation 1.

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

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

4.2. Conduct of POA hunting

During the hunting phase in POA, 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. Consequently, Equation 2 governs the variation in their position.

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

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

$$P_{mean}^{curIt} = \frac{1}{N_{pop}} \sum_k^{N_{pop}} P_k^{curIt} \tag{3}$$

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

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 expressed by Equation 5.

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

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

4.4. Conduct of PO communication

Parrots, belonging to the Pyrrhura Molinae family, are inherently social creatures with 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 corresponds to the mean position of the current population. Equation 6 provides a mathematical expression for this phenomenon.

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

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. This behavior in POA is mathematically expressed through Equation 7.

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

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

In POA, the procedure continues until the specified circumstances for termination are fulfilled. The pseudocode is provided in Algorithm 1.

Algorithm 1: PO algorithm pseudo − code

1: PO parameter initialization
2: Randomly initialized the position of the solution agents
3: ***For*** $i = MaxIter$ *do*
4:
Compute the fitness function and fine the best position
5: ***For*** $j = 1: N_{pop}$ ***do***
6: $st = randi([1,4]$
7: ***If*** $st == 1$ ***Then***
8: *foraging conduct and update posing using Eq.* (2)
9: ***Elseif*** $st == 2$ ***Then***
10: *Staying conduct and update posing using Eq.* (5)
11: ***Elseif*** $st == 3$ ***Then***
12:
C*ommunicating conduct and update posing using Eq.* (6)
13: ***Elseif*** $st == 4$ ***Then***
14:
fear of strangers conduct and update posing using Eq. (7)
15: ***End For***
16: ***End For***
17: Return the 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 local search capabilities of the

POA and speed up the global search procedure to overcome its drawbacks. The main goal of mPOA is to reduce the issue of stagnation at local optima while achieving faster convergence. To provide a comprehensive overview, we begin by examining the difficulties with the traditional PO algorithm.

5.1. Issues with the original POA

Although effective, the original POA struggles with high-dimensional problems due to premature convergence and limited exploration. To overcome this, we propose a modified POA (mPOA) that integrates mOBL and BR. mOBL enhances start and accelerates convergence, whereas BR improves exploration. The effectiveness of mPOA was validated by optimizing a pretrained 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 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 the incorporation of the OBL.

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

$$y_O = lwb + upb - y \tag{8}$$

where lwb and upb denote lower and upper bounds, respectively

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

$$y_O = lwb_q + upb_q - y_q \tag{9}$$

In OBL, the solution Y is substituted with its complementary counterpart Y_O, determined by an activation function. If the fitness of Y represented as $f(Y)$ is greater than that of Y_O denoted as $f(Y_O)$, Y is preserved; otherwise, Y is replaced with Y_O. This updating process facilitates the evolution of the solution population by selecting the best solution between Y and Y_O. This 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.

$$y_{mOBL} = lw_q + up_q - \varepsilon \times r \tag{10}$$

In this context, r denotes a value within the range [0,1], and ε represents the mutation scale, a small constant that regulates the mutation intensity. 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 ability of the algorithm to efficiently escape from local optima.

5.3. Brownian motion

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

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

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

5.4. Start of the modified POA

Produce a starting population of potential solutions within the search space's stated limits. The initial population is enhanced 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 transition of the population 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).

$$Fit_i = \alpha \times Err_i + \beta \times \frac{d_i}{D} \quad (12)$$

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

5.6. Modified POA fitness evaluation

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

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

5.7. modified POA update phase

It is essential to evaluate the solutions at each iteration to find the best candidates and improve the freshly created solutions for subsequent phases. After computing each individual's fitness, 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 St ϵ rand [1, 4] is then examined. If $St = 1$, the position is updated using the Brownian motion (BR) strategy as per Equation 11 instead of the Lévy strategy. Similarly, if $St = 2$, the position is updated using BR based on Equation (11) instead of employing the staying behavior. Algorithm 2 presents the mPOA pseudo-code.

Algorithm 2: Modified POA pseudo − code

1: Initilize the mPOA parametres
2: Initilize the position of all agents using Eqn. (1)and evaluate their fitness
3: ***for*** i = N_{pop} ***do***
4: Perform mOBL on the initial population using Eqn. 10 and save results in Y_{mOBL}
5: Evaluete the fitness $of\ Y_{mOBL}$ and save result in $mOBL_{fit}$
6: $if\ mOBL_{fit} < F_i\ then$
7: $Y_i = Y_{mOBL}$
8: **end if**
8: **end for**
3: ***For*** $i = MaxIter\ do$
4: $Compute\ the\ fitness\ function\ and\ fine\ the\ best\ position$ (12)
5: ***For*** $j = 1: N_{pop}$ ***do***
6: $st = randi([1,4]$
7: ***If*** $st == 1$ ***Then***
8: $update\ positiong\ using\ Eqn.$ (11)
9: ***Elseif*** $st == 2$ ***Then***
10: $Update\ position\ using\ BR\ based\ on\ Eqn.$ (11)
11: ***Elseif*** $st == 3$ ***Then***
12: $Perform\ communicating\ conduct\ and\ update\ posing\ using\ Eq.$ (6)
13: ***Elseif*** $st == 4$ ***Then***
14:
$Perform\ fear\ of\ strangers\ conduct\ and\ update\ posing\ using\ Eq.$ (7)
15: ***End For***
16: ***End For***
17: Return the best solution obtain

6. Proposed rice leaf disease model

This section presents the methodology underlying the rice leaf disease detection model, which is designed to significantly enhance the performance of CNN architectures. The proposed approach integrates the mPOA with the DenseNet121 framework to optimize its performance for improved disease detection and classification. First, the mPOA is employed to determine the optimal DenseNet121 hyperparameter configurations. Transfer learning methods are then applied to train the DenseNet121. Upon completion of the training, the performance of the model 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 the validation data being used for subsequent assessment. The methodology outlines a detailed pipeline for building the DL model, starting with the dataset acquisition and ending with the classification results generation. The presented model consists of several stages, as shown in Figure 2. These stages include data collection, preprocessing, hyperparameter tuning, learning, testing, and evaluation.

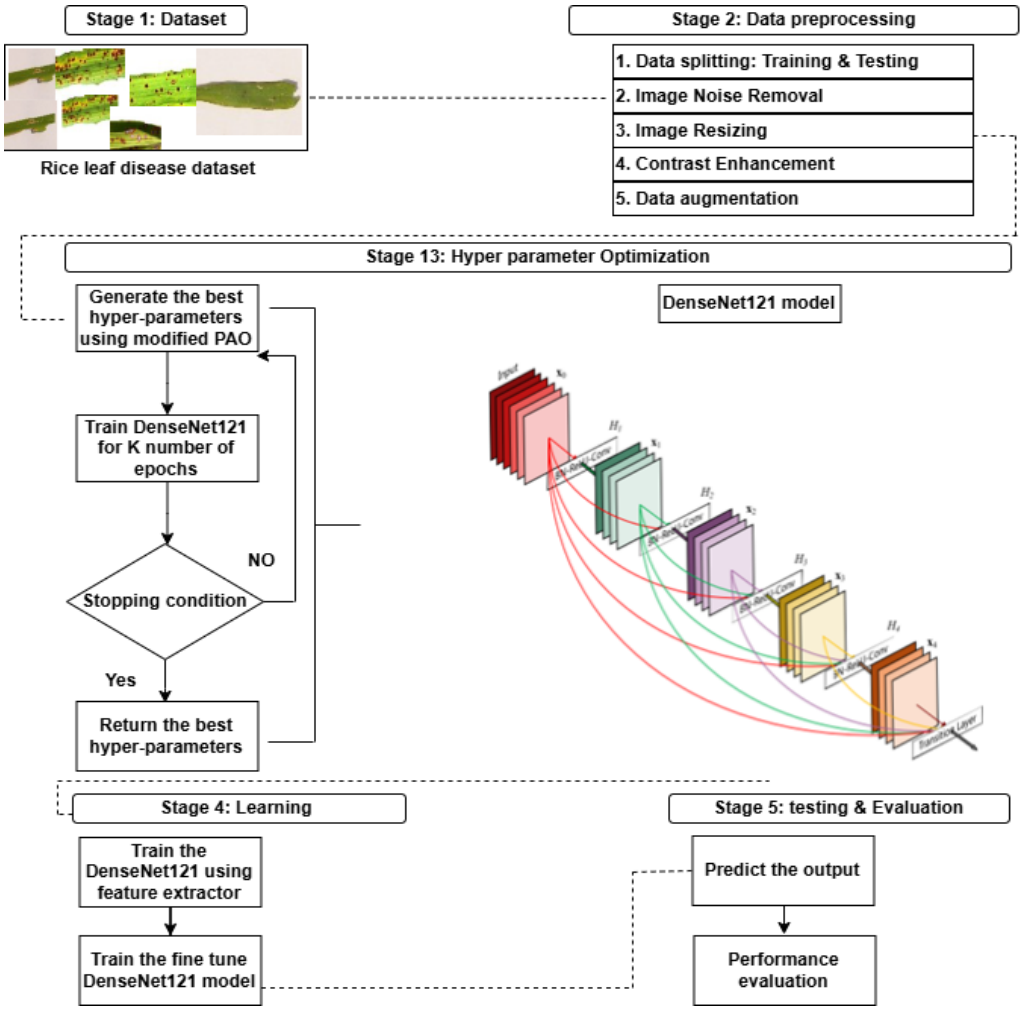

Figure 2 Structural diagram of the proposed rice disease detection model

6.1. Acquisition of datasets

The data used in this study were obtained from the publicly accessible Kaggle cloud data repositor. It comprises 1,600 images distributed across four distinct classes: Hispa, Brown Spot, Leaf Blast, and Healthy leaves. Figure 3 depicts the class spread of the data.

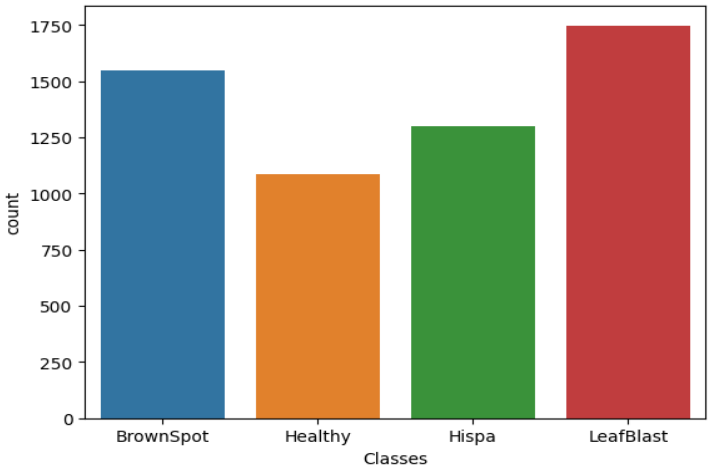

Figure 3 Class Distribution of Rice Leaf Images in the Kaggle Dataset

6.2. Data Preprocessing

Images of crop leaf diseases are frequently affected by substantial noise and poor contrast, thereby limiting the accuracy of disease detection. Since the clarity and sharpness of these images are critical for effective diagnosis, a useful strategy to enhance detection accuracy is to improve image clarity by eliminating noise frequencies. Accordingly, this study investigates various image preprocessing techniques aimed at addressing these challenges and improving the image quality of rice disease for more precise and reliable detection.

6.2.1. Noise reduction

Digital imaging plays a crucial role in image processing; however, noise from acquisition devices can hinder analysis, especially in rice disease classification. Noise reduction enhances signal quality and classification accuracy. It is essential to identify noise sources and apply effective reduction techniques (Huang et al., 2024). Nonlinear filters, such as 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 expressed in Equation 14.

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

where N defines the surrounding neighborhood at location (a, b).

6.2.2. Image Normalization (INN)

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 enhances model convergence during training. In the initial stage of normalization, the brightness levels of the input images are adjusted to lie within the range of 0 to 1, as outlined by Razmjooy et al. (2020). The mathematical formulation employed to achieve brightness normalization is given in Eq. (15).

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

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

6.2.3. Contrast enrichment

The contrast is vital for image quality, representing brightness variability. Low contrast compresses the tonal range, causing blurriness. Enhancing contrast increases tonal variation and sharpness. The methodologies and algorithms discussed in this study leverage techniques that apply histogram correction to address the challenges of low image contrast. Specifically, the

histogram equalization method has been identified as an effective approach to rectifying these issues (Razmjooy et al., 2020).

6.2.4. Dataset augmentation

Data augmentation is crucial in deep learning to overcome limited data and class imbalance, particularly in binary classification, where underrepresented (minority) classes can critically impact model performance. Scarcity of samples of key disease classes exacerbate these challenges in crop leaf disease detection. 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 ability of the model to learn from minority classes. This preprocessing step effectively mitigates the risks of poor generalization and high misclassification costs associated with imbalanced datasets. A number of augmentation approaches are implemented in this study, as shown in Table 2.

Table 2 Augmentation approaches and 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 the Hyperparameters

Several models were evaluated to identify the most suitable pretrained CNN for rice leaf disease classification, 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 RLD 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. However, the added classifier is trained on the enlarged dataset using the features extracted from the frozen layers. This step aligns with the third phase described in Algorithm 3.

The same data augmentation techniques as in earlier stages are used to enhance sample diversity and introduce new feature variations to maintain consistency. Fine-tuning is initiated for further optimization after the feature extraction process reaches a performance plateau with no significant improvement over several training epochs. The fine-tuned classifier comprises 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 mPOA determining the size of neurons and the dropout factor. The output layer, designed for multiclass classification, includes four neurons and employs the softmax activation function. This process corresponds to the second stage of Algorithm 3. All the last four layers of the DenseNet121 convolutional backbone are frozen at the optimization stage, 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 stable performance is achieved, marking the point where further improvements cease to occur.

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 $RLD_{D_{test}}$ to form $RLD_{Dn_{test}}$
Reduce noise in $RLD_{D_{agtrain}}$
Enhance contrast of images in $RLD_{D_{agtrain}}$
// **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 $RLD_{D_{agtrain}}$ using $RLD_{Dn_{test}}$
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 $RLD_{D_{agtrain}}$ using $RLD_{D_{ntest}}$
end for

Several regularization techniques were employed to mitigate the risk of overfitting during model training. These include data augmentation (e.g., flipping, rotation, and scaling) as described 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. Close alignment of these curves indicates reduced overfitting and strong generalization performance.

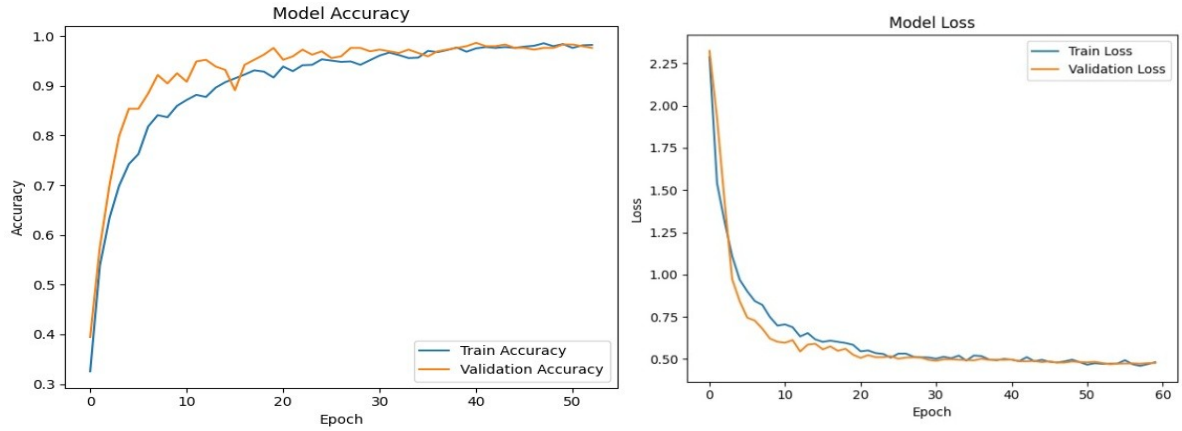

Figure 4 (a) Proposed model training and accuracy validation (b)

6.5. Testing and evaluation of the results

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

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

$$\frac{trP+trN}{trP+trN+faP+faN} \tag{16}$$

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

$$\frac{trP}{trP+faN} \tag{17}$$

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

$$\frac{trP}{trP+faP} \tag{18}$$

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

$$\frac{2\times recall\times precision}{recall+precision} \tag{19}$$

where trP denotes the number of true positive predictions, trN denotes true negative predictions, faP represents false positive predictions, and faN denotes false negative predictions.

7. Performance evaluation of the rice disease detection model

This section provides an analysis of the results achieved using the proposed deep learning framework for rice leaf disease detection. 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. Hyperparameter optimization 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 importance of each hyperparameter in enhancing the model's effectiveness. The maximum number of iterations of the optimization algorithm 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, which represents the number of samples processed per iteration, had a search range of 1–64. 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 50–550, allowing for flexible adjustment of model capacity.

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

To determine the optimal DenseNet121 training epoch count, several values were evaluated. We 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 reduce the validation loss. The test set loss degree after 16 training epochs served as the benchmark for evaluating the effectiveness of the proposed method. After training, the optimal values for the hyperparameters: dropout rate, batch size, and number of neurons in the initial dense layer) are determined. The following were the final optimized hyperparameter values determined by the POA: 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 efficiency of DenseNet121. Table 3 presents the parameter settings for the mPOA integrated with DenseNet121.

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 optimal values

Here, the enhanced DenseNet121 is trained using the mPOA-determined optimal hyperparameters, serving as a feature extractor for the augmented training dataset. The effectiveness of the model was subsequently assessed on the validation dataset across a maximum of 16 epochs. The training process was halted early if no progress was detected over 10 consecutive epochs to mitigate overfitting. This approach utilizes the early stopping technique, as described by Prechelt (2002) and Bai et al. (2021). Given that the RLD dataset involves multi-class classification, the categorical cross-entropy loss was employed to optimize the DenseNet121 model. The Adam optimizer with a 2×10-7 learning used at the feature extraction stage. A decay learning rate schedule 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 10 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 knowledge extracted from the initial phase is retained, enhancing the overall model performance.

7.3. Evaluation of the efficacy of the presented model

This section provides a comprehensive assessment of the efficacy of the presented model. 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 with those obtained from a DenseNet121 configured using manual hyperparameter tuning. The hyperparameter settings optimized by the mPOA are detailed in Table 4. The manually tuned DenseNet121 model employed a batch size of 8, a dropout factor of 0.72, 120 neurons, and a 0.001 starting learning rate of 0.001. The efficacy of the mPOA-enhanced DenseNet121 was assessed using standard metrics such as accuracy, precision, recall, and f-measure. Table 5 presents these metrics, offering a detailed assessment of the performance of the model. For the manually tuned DenseNet121 model, the accuracy, precision, recall, and F-measure were approximately 94.8%, 94.8%, and 94.23%, respectively. Conversely, the DenseNet121 optimized with the mPOA demonstrated noticeably superior efficacy across all metrics. Specifically, the model achieved an accuracy of approximately 98.5%, precision of 98.6%, recall of 98.4%, and F-measure of 98.5%. These findings demonstrate 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 mPOA over manual hyperparameter tuning. Figures 4(a) and 4(b) illustrate the training and validation accuracy and loss curves, respectively, of the presented model, 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

7.4. Performance assessment of the mPOA DenseNet121 and original POA DenseNet121 models

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 RLD classification. Both models were assessed based on their respective hyperparameter configurations and classification efficacy (Table 6). Both approaches share a similar learning factor of 0.0001.

The DenseNet121 enhanced with mPOA showcases notable improvements in hyperparameter configurations compared to the original POA-optimized version. For example, the mPOA uses a smaller batch size of 8, as opposed to 13 in the original model, signifying a more streamlined training procedure with fewer examples per repetition. Furthermore, the dropout factor is significantly lowered to 0.1 in the modified model compared to 0.84 in the original model, 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 significantly enhance 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 RLD classification, leading to superior performance outcomes. Figure5 provides a graphical comparison of the performance metrics for the mPOA-enhanced DenseNet121, original POA-

enhanced DenseNet121, and standard DenseNet121, further illustrating the proposed approach's efficacy. However, While the mPOA introduces additional computation during training, the final DenseNet121-based model remains efficient at inference time, making it suitable for deployment in constrained environments. The proposed model achieves a balance between accuracy and computational cost with fewer parameters (7.98 million) 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

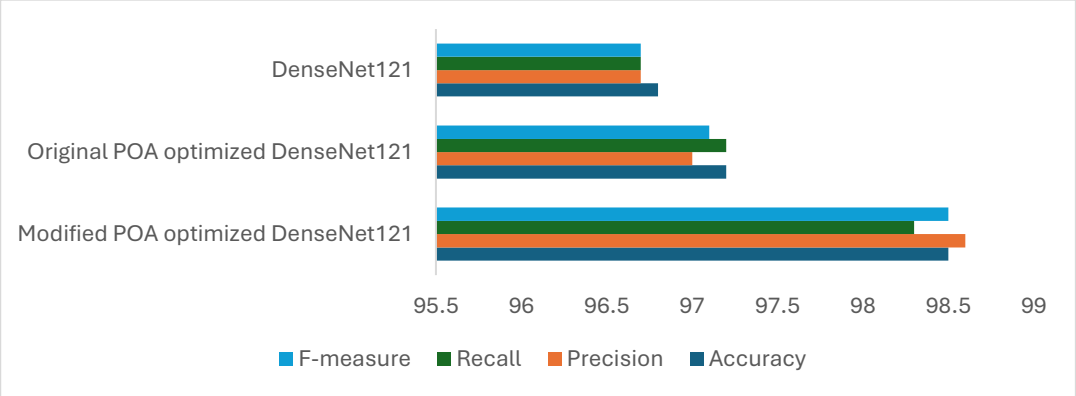

Figure 5 Comparison of mPOA DensNet121 with POA-DenseNet121 and original DenseNet121

7.5. Performance assessment of the mPOA DenseNet121 model with other pretrained models

This section presents a comparative evaluation of the proposed mPOA-DenseNet121 model against several widely recognized pretrained deep learning models, including VGG19, EfficientNetB0, InceptionV3, DenseNet201, and ResNet50, for rice leaf disease detection. These models were selected based on their proven performance in earlier research addressing disease detection challenges. The objective of this study is to judge the robustness and efficacy of the proposed approach by contrasting its performance with those of benchmark models. This analysis provides a comprehensive evaluation of the capability of the proposed model in rice leaf disease detection relative to established methods in the field. Table 7 provides a detailed comparison of mPOA-DenseNet121 with the aforementioned pretrained deep learning models. The results reveal that the presented model outperforms all others across all evaluated metrics, demonstrating its exceptional performance in accurately detecting RLDs. Notably, EfficientNetB0, InceptionV3, and ResNet50 emerged 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 mPOA-DenseNet121 model's capability in achieving precise classification of rice leaf disease images. The graphical representation of this comparative analysis is illustrated in Figure6, further emphasizing the effectiveness of the presented method over existing methods.

Although an exhaustive hyperparameter search could yield marginal improvements in theory, such approaches are computationally intensive and often impractical for real-world applications. In this study, the use of the modified POA achieves a balance between performance and resource efficiency. We define a model as "good enough" when it consistently achieves over 98% 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 scope of optimization.

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

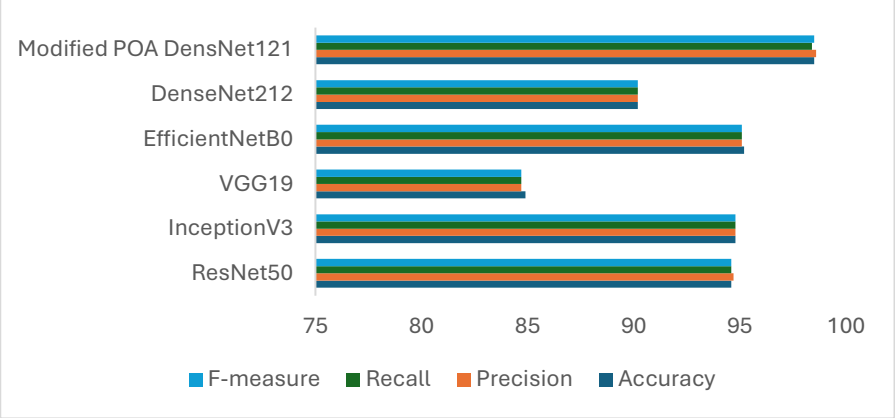

Figure 6 Comparison of mPOA DensNet121 with other pretrained models

8. Conclusions and Future Work

This study introduces an mPOA-DenseNet121 model tailored for rice leaf disease detection, leveraging deep learning and optimization developments. The proposed model demonstrated outstanding performance in all evaluation metrics compared with popular pretrained models, such as EfficientNetB0, DenseNet201, VGG19, InceptionV3, and ResNet50. This outstanding performance highlights the mPOA's efficacy in optimizing critical hyperparameters to enhance DenseNet121's classification accuracy. 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 the model is 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.

Acknowledgements

The authors would like to acknowledge the support of Prince Sultan University, Riyadh Saudi Arabia for APC support of this publication.

Author Contributions

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

Conflict of Interest

The authors declare no conflicts of interest.

References

Adamu, A, Abdullahi, M, Junaidu, SB & Hassan, IH 2021, 'An hybrid particle swarm optimization with crow search algorithm for feature selection', *Machine Learning with Applications*, vol. 6, article 100108, https://doi.org/10.1016/j.mlwa.2021.100108

Ayesha, H, Iqbal, S, Tariq, M, Abrar, M, Sanaullah, M, Abbas, I, Rehman, A, Niazi, NFK & Hussain, S 2021, 'Automatic medical image interpretation: State of the art and future directions', *Pattern Recognition*, vol. 114, article 107856, https://doi.org/10.1016/j.patcog.2020.107856

Babu, RR & Philip, FM 2024, 'Optimized deep learning for skin lesion segmentation and skin cancer detection', *Biomedical Signal Processing and Control*, vol. 95, article 106292, https://doi.org/10.1016/j.bspc.2024.106292

Bai, Y, Yang, E, Han, B, Yang, Y, Li, J, Mao, Y, Niu, G & Liu, T 2021, 'Understanding and improving early stopping for learning with noisy labels', *Advances in Neural Information Processing Systems*, vol. 34, pp. 24392-24403

Barakat, M, Chung, GC, Lee, IE, Pang, WL & Chan, KY 2023, 'Detection and sizing of durian using zero-shot deep learning models', *International Journal of Technology*, vol. 14, no. 6, pp. 1206-1215, https://doi.org/10.14716/ijtech.v14i6.6640

Chakrabarty, A, Ahmeda, ST, Islam, MF, Aziz, SM & Maidin, SS 2024, 'An interpretable fusion model integrating light weight CNN and transformer architectures for rice leaf disease identification', *Ecological Informatics*, vol. 82, article 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', *Journal of the Science of Food and Agriculture*, vol. 100, no. 7, pp. 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*, vol. 182, article 103472, https://doi.org/10.1016/j.advengsoft.2023.103472

Emam, MM, Houssein, EH, Samee, NA, Alohali, MA & Hosney, ME 2024, 'Breast cancer diagnosis using optimized deep convolutional neural network based on transfer learning technique and improved Coati optimization algorithm', *Expert Systems with Applications*, vol. 255, article 124581, https://doi.org/10.1016/j.eswa.2024.124581

Faramarzi, A, Heidarinejad, M, Mirjalili, S & Gandomi, AH 2020, 'Marine predators' algorithm: A nature-inspired metaheuristic', *Expert Systems with Applications*, vol. 152, article 113377, https://doi.org/10.1016/j.eswa.2020.113377

Farea, E, Saleh, RA, AbuAlkebash, H, Farea, AA & Al-antari, MA 2024, 'A hybrid deep learning skin cancer prediction framework', *Engineering Science and Technology, an International Journal*, vol. 57, article 101818, https://doi.org/10.1016/j.jestch.2024.101818

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

Goluguri, NV, Devi, KS & 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*, vol. 33, pp. 5869–5884, https://doi.org/10.1007/s00521-020-05364-x

Hassan, IH, Abdullahi, M, Aliyu, MM, Yusuf, SA & 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*, vol. 16, article 200114, https://doi.org/10.1016/j.iswa.2022.200114

Hossain, S, Seyam, TA, Chowdhury, A, Ghose, R, Rahaman, A, Hadika, Z & Pathak, A 2025, 'Enhancing agricultural diagnostics: Advanced training of pre-trained CNN models for paddy leaf disease detection', *Machine Learning*, vol. 10, no. 1, pp. 1–13, https://doi.org/10.11648/j.mlr.20251001.11

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*, vol. 57, article 105546, https://doi.org/10.1016/j.bspc.2023.105546

Ibrahim, AT, Abdullahi, M, Kana, AFD, Mohammed, MT & Hassan, IH 2024, 'Categorical classification of skin cancer using a weighted ensemble of transfer learning with test time augmentation', *Data Science and Management*, 2024, pp. 174-184, 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*, vol. 52, no. 12, pp. 13250-13261, https://doi.org/10.1109/TCYB.2021.3107415

Lian, J, Hui, G, Ma, L, Zhu, T, Wu, X, Heidari, AA, Chen, Y & Chen, H 2024, 'Parrot optimizer: Algorithm and applications to medical problems', *Computers in Biology and Medicine*, vol. 172, article 108064, https://doi.org/10.1016/j.compbiomed.2024.108064

Maijeddah, UI, Yusuf, SA, Abdullahi, M & Hassan, IH 2024, 'A hybrid transfer learning model with optimized SVM using honey badger optimization algorithm for multi-class lung cancer classification', *Science World Journal*, vol. 19, no. 4, pp. 977-986, https://doi.org/10.4314/swj.v19i4.10

Manjupriya, R & Leema, AA 2025, 'Efficient epileptic seizure detection with optimal channel selection and FIXUPPACTBI-LSTM deep learning model', *International Journal of Technology*, vol. 16, no. 2, pp. 706-721, https://doi.org/10.14716/ijtech.v16i2.7333

Mofrad, FB & 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*, vol. 213, article 119261, https://doi.org/10.1016/j.eswa.2022.119261

Mohammed, H, Majeed, H, Al-mafrachi, BAR, Al-Khaffaf, MS, & Saad, A. 2025, 'A novel deep learning approach for classification of abnormal teeth in panoramic x-rays', *International Journal of Theoretical & Applied Computational Intelligence,* vol. 2025, pp. 22–34, https://ijtaci.com/index.php/ojs/article/view/7

Naqi, SAE, Iqbal, K, Khan, AA, Khan, R, Jamil, S & Ishtiaq, U 2025a, 'Diseases detection from apple leaf using deep transfer learning approach', *International Journal of Theoretical & Applied Computational Intelligence*, 2025, pp. 57-70, https://ijtaci.com/index.php/ojs/article/view/13

Naqi, SAE, Jamil, S, Khan, MAA, Naveed, F, Arshad, A & Ishtiaq, U 2025b, 'Automated pathological assessment of potato leaf diseases through convolutional neural networks', *International Journal of Theoretical & Applied Computational Intelligence,* vol. 2025, pp. 106–124. https://ijtaci.com/index.php/ojs/article/view/28

Nugroho, YN, Harwahyu, R, Sari, RF, Nikaein, N & Cheng, RG 2023, 'Performance evaluation of anomaly detection system on portable LTE telecommunication networks using Open Air Interface and ELK', *International Journal of Technology*, vol. 14, no. 3, pp. 549–560, https://doi.org/10.14716/ijtech.v14i3.4237

Pattnaik, G, Shrivastava, VK & Parvathi, K 2021, 'Tomato pest classification using deep convolutional neural network with transfer learning, fine tuning and scratch learning', *Intelligent Decision Technologies*, vol. 15, no. 3, pp. 433-442, https://doi.org/10.3233/IDT-200192

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

Preethi, P, Swathika, R, Kaliraj, S, Premkumar, R & Yogapriya, J 2024, 'Deep learning–based enhanced optimization for automated rice plant disease detection and classification', *Food and Energy Security*, vol. 13, no. 5, article e70001, https://doi.org/10.1002/fes3.70001

Rahayu, DS, Husodo, ZA, Pidanic, J, Li, X & Suhartanto, H 2025, 'A technique to predict bankruptcy using ultimate ownership network as key indicators', *International Journal of Technology*, vol. 16, no. 1, pp. 275-288, https://doi.org/10.14716/ijtech.v16i1.7516

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

Razmjooy, N, Ashourian, M, Karimifard, M, Estrela, VV, Loschi, HJ, Do Nascimento, D, França, RP & Vishnevski, M 2020, 'Computer-aided diagnosis of skin cancer: A review', *Current Medical Imaging Reviews*, vol. 16, no. 7, pp. 781-793, https://doi.org/10.2174/1573405616666200129095242

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

Rehman, A, Kashif, M, Abunadi, I & Ayesha, N 2021, '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)*, IEEE, pp. 101–104

Ritharson, PI, Raimond, K, Mary, XA & Robert, JE 2024, 'DeepRice: A deep learning and deep feature-based classification of rice leaf disease subtypes', *Artificial Intelligence in Agriculture*, vol. 11, pp. 34-49, https://doi.org/10.1016/j.aiia.2023.11.001

Shah, SR, Qadri, S, Bibi, H, Shah, SMW, Sharif, MI & 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*, vol. 13, no. 6, article 1633, https://doi.org/10.3390/agronomy13061633

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*, vol. 14, no. 8, pp. 1717-1727, https://doi.org/10.14716/ijtech.v14i8.6859

Wolpert, DH & Macready, WG 1997, 'No free lunch theorems for optimization', *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, https://doi.org/10.1109/4235.585893

Yuan, Y, Chen, L, Wu, H & Li, L 2022, 'Advanced agricultural disease image recognition technologies: A review', *Information Processing in Agriculture*, vol. 9, no. 1, pp. 48-59, https://doi.org/10.1016/j.inpa.2021.01.003

Yusuf, HM, Ali, YS, Abubakar, AH, Abdullahi, M & Hassan, IH 2024, 'A systematic review of deep learning techniques for rice disease recognition: Current trends and future directions', *Franklin Open*, article 100154, https://doi.org/10.1016/j.fraope.2024.100154