Metaheuristic Optimization for Deep Learning in Plant Disease Detection: A Hybrid Approach

Original Scientific Paper

Aqeel Majeed Breesam*
Institute of Medical Technology/Baghdad
Middle Technical University
Baghdad, Iraq
aqeelmajeed@mtu.edu.iq

Rusul Abdulridha Muttashar
Businesses Informatics College
University of Information Technology and Communications
Baghdad, Iraq
rusul.abdulridha@uoitc.edu.iq

Esraa Najjar
Computer Science Department
General Directorate of Education in Najaf Governorate
Al-Najaf, Iraq
esraas.najjar@uokufa.edu.iq

*Corresponding author

Abstract *– This study investigates metaheuristic hyperparameter optimization for deep learning–based plant disease detection across two datasets: Dataset A (1,530 images; three classes: Healthy, Powdery, Rust) and a large multi-crop corpus evaluated in a binary Healthy/Diseased setting with an 80/20 training–validation split. A hybrid optimizer is proposed that interleaves Dragonfly Algorithm (DA) for population-wide exploration with Firefly Algorithm (FA) for elite intensification (DA–FLA), and is applied to five pretrained CNN backbones (DenseNet, VGG19, InceptionV3, MobileNet, Xception). All models are trained under an identical 50-epoch protocol. On Dataset A, DenseNet provides the strongest baseline (accuracy/macro-F1 = 0.9733/0.9735), which rises to 0.9800/0.9800 with DA–FLA tuning. On the large-scale binary corpus, Xception and DenseNet perform competitively (≈0.9846 macro-F1 and 0.9838 macro-F1, respectively), while the optimized Xception attains 0.9924 accuracy and 0.9913 macro-F1. A one-way ANOVA with Tukey HSD confirms significant performance differences ($p < 0.001$), with optimized Xception outperforming all comparators. The hybrid search introduces modest training overhead but leaves inference cost essentially unchanged. Results demonstrate that balancing global exploration with local exploitation yields reproducible, statistically supported gains, advancing accurate and efficient plant disease diagnostics suitable for mobile/edge deployment and supporting early intervention and sustainable farming practices.*

Keywords: *Plant disease detection, deep learning, dragonfly optimization algorithm, firefly algorithm, hybrid optimization*

Received: May 31, 2025; Received in revised form: October 11, 2025; Accepted: October 23, 2025

1. INTRODUCTION

This study presents a hybrid deep learning approach for plant disease detection, combining CNNs with a novel optimization strategy using the Dragonfly and Firefly Algorithms. By tuning key hyperparameters through this DA-FLA hybrid, the framework enhances classification accuracy and convergence introducing a technique not previously explored in this context.

Food systems and national economies depend on agriculture, but as a result, crop health is continuously threatened by diseases which decrease the quality and output [1, 2]. The productivity is already limited by a range of biotic and abiotic stressors that may disrupt supply chains and become destabilizing [3, 4]. The traditional diagnosis of the expert visual inspection is a labor-intensive, costly and inaccurate process, which restricts the scalability of a large-scale production en-

vironment [5-7]. Alternatively, but, conversely, deep learning, and most specifically Convolutional Neural Networks (CNNs), have also demonstrated to be a powerful option, which involves image-based markers, such as discoloration and lesion morphology, to enable effective and automatic detection of plant diseases [8–10]. Nevertheless, the CNN models are typically prone to issues such as slow convergence, expensive computations, and sensitivity to hyperparameter parameters. Recent models like DenseNet, VGG19, InceptionV3, MobileNet, and Xception have shown strong classification capability, but these models have poor hyperparameter sensitivities [11-13]. To overcome this, we present a new metaheuristic optimization model that is a hybrid between the Dragonfly Algorithm (DA) and Firefly Algorithm (FLA) to optimize the main hyperparameters and enhance the model accuracy, convergence, and generalization [14-16]. It is a DA-FLA method that has a unique way of balancing the global search and local refinement, which presents a practical and scalable way of CNN-based plant disease detection.

This method of DA-FLA is the only one that balances both the global search and local refinement in an attempt to provide a practical and scalable solution to CNN-based plant disease detection.

The experimental design was extended to the two datasets to enhance the generalizability and methodological rigor: controlled multiclass corpus (three classes) and large multi-crop corpus working in a binary Healthy/Diseased context. Each architecture was trained with the same 50-epoch schedule, so as to measure them equally. Besides cross-validation and ablation research, statistical significance of differences between models was measured by one-way ANOVA with Tukey HSD post-hoc test which is a strong evidence of the influence of the proposed DA -FLA optimizer.

This research has the following contributions:

- We propose a novel hybrid metaheuristic optimization framework that combines the Dragonfly Algorithm (DA) and Firefly Algorithm (FLA) for fine-tuning CNN hyperparameters—a combination not previously applied in plant disease detection.
- We thoroughly test five deep learning models of (DenseNet, VGG19, InceptionV3, MobileNet and Xception) on a multiclass plant leaf dataset and find the most effective of them to be DenseNet.
- We demonstrate that the DA-FLA optimizer significantly improves classification accuracy, convergence speed, and model generalization, validated through cross-validation, ablation studies, and statistical analysis.
- We show the practical relevance of our optimized model in promoting early disease detection and sustainable agriculture, particularly in resource-constrained environments.

2. REALTED WORK

Recent advances in deep learning most notably with Convolutional Neural Networks (CNNs) have substantially improved plant disease diagnosis by extracting rich spatial and textural cues from leaf images [17, 18]. A prominent direction augments CNNs with metaheuristic optimization and attention mechanisms to improve generalization and convergence. Illustratively, Huang et al. [19] employed the CSUBW optimizer for mango leaf diagnosis, while leveraged GGGWO for potato diseases, jointly evidencing the role of global search in tuning hyperparameters [20]. Additional strands enrich data or search dynamics: PCA and noise injection for robustness, the Crow Search Algorithm for advanced hyperparameter tuning, and ensemble-based optimization via APLDD-ESOSDL. Hybrid and pre-trained CNNs have also been applied to apple and maize plants, frequently paired with Grey Wolf, Whale, and Improved Butterfly Optimization algorithms to bolster accuracy [21, 22]. Parallel efforts incorporate Explainability e.g., Grad-CAM and EDA to increase interpretability and trust in clinical- or field-adjacent decision support. Complementing these, Pham et al. [23] integrated contrast enhancement, segmentation, and an adaptive particle–Gray Wolf optimizer (APGWO) to reduce features for MLP classification, while [24] introduced an ODNN pipeline that couples CNN feature extraction with two-stage weight optimization (Improved Butterfly Optimization plus Genetic Algorithm), attaining 99% on sensitivity/accuracy/specificity.

Building on these foundations, the present hybrid DA–FLA model targets a balanced exploration–exploitation regime for hyperparameter tuning, addressing rugged, mixed-type search spaces [25]. Unlike prior single- or dual-heuristic pairings, this combination has not been previously applied to plant disease classification, and the reported experiments demonstrate both theoretical plausibility and practical efficacy within this domain.

Concurrently, transformer-based vision architectures have begun reshaping agricultural imaging benchmarks [26]. For maize leaf disease recognition, [27] proposed a deployment-oriented framework that adapts MaxViT to a lightweight four-class setting, replacing the stem's conventional convolutional block with a Squeeze-and-Excitation module and adopting a GRN-based MLP from ConvNeXtV2. By aggregating PlantVillage, PlantDoc, and CD&S into a large composite dataset and evaluating more than 28 CNNs and 36 transformers, the study achieved state-of-the-art accuracy (99.24%) with competitive inference speed, underscoring practical value for time-sensitive agricultural tasks. In grape analysis, [28] benchmarked 31 state of the art CNN and transformer models on a plantvillage (4,062 images; Black Rot, Leaf Blight, Esca, Healthy) and grapevine (500 images; Ak, Alaidris, Buzgulu, Dimnit, Nazli) dataset, and found that 4 models were 100% accurate on both datasets and Swinv2-Base is consistably top-performing, supporting the use of fine-tuned trans-

formers in detecting early disease. To corn, [29] compared MaxViT, DeiT3, MobileViT, and MViTv2 against VGG, ResNet, DenseNet, and Xception in a single pipeline (preprocessing, augmentation, transfer learning, hyperparameter optimization), and proposed a soft-voting ensemble with adaptive thresholding. On the CD&S test set, four MaxViT variants (plus other deep models) reached 100% accuracy, and the approach attained 99.83% on PlantVillage, surpassing prior studies and demonstrating the utility of calibrated ensembles for balanced, high-confidence detection.

Related advances in medical imaging echo these design choices and optimization strategies. For ischemic stroke lesion delineation on MRI, [30] proposed a U-Net variant augmented with ConvNeXtV2 blocks and GRN-based MLPs the first application of ConvNeXtV2 in this setting together with a clinician-informed preprocessing step that filters small spurious lesions (≤5 pixels). On ISLES 2022, the method achieved IoU = 0.8015 and Dice = 0.8894, outperforming strong U-Net baselines and alternative approaches. Within neuro-oncology, [31] proposes NeXtBrain, a hybrid pursuer, combining a NeXt Convolutional Block, which entails multi-head convolutional attention and SwiGLU MLP, and a NeXt Transformer Block, which entails self-attention and convolutional attention and SwiGLU MLP. The model achieves 99.78% accuracy / 99.77% F1 on Figshare and 99.78% accuracy / 99.81% F1 on Kaggle, and is computationally efficient (23.91M parameters, 10.32 GFLOPs, 0.007 ms inference). Collectively, these results from agriculture to clinical imaging reinforce the value of modern attention-based backbones, principled optimization, and carefully curated benchmarks for robust, high-accuracy visual diagnosis.

Table 1. Summary of Related Works

Reference	Methodology	Optimization / Special Technique	Main Contribution
[19]	CNN for mango leaf disease classification	CSUBW optimizer	Improves convergence and accuracy for mango leaf diagnosis
[20]	CNN for potato disease detection	GGGWO (Generalized Growing Grey Wolf Optimizer)	Boosted classification accuracy via metaheuristic hyperparameter tuning
[23]	Contrast-enhancement + segmentation pipeline with MLP classifier (features from CNN)	APGWO (Adaptive Particle–Gray Wolf Optimizer)	Feature reduction and improved MLP classification (Pham et al.)
[24]	CNN with ODNN (Optimal Deep Neural Network) pipeline	Improved Butterfly Optimization + Genetic Algorithm (two-stage weight optimization)	~99% sensitivity/accuracy/specificity with two-stage optimization
[26]	Vision Transformers in agriculture (survey/ benchmark trend)	Transformer backbones	Recent architectures reshaping agri-imaging benchmarks
[27]	MaxViT-based lightweight framework for maize leaf recognition	SE stem + GRN-MLP (ConvNeXtV2)	SOTA ≈99.24% with large composite dataset; deployment-oriented speed
[28]	Grapevine disease/varietal analysis (31 CNN/Transformer models)	Fine-tuned Transformers (e.g., Swinv2-Base)	Four models hit 100% on two datasets; consistent top-tier performance
[29]	Corn disease recognition with CNNs + Transformers	Soft-voting ensemble with adaptive thresholding	Several MaxViT variants reach 100% on CD&S; 99.83% on PlantVillage
[30]	U-Net variant for ischemic stroke MRI lesion segmentation	ConvNeXtV2 blocks + GRN-MLP, clinician-informed filtering	IoU = 0.8015, Dice = 0.8894; surpasses strong U-Net baselines
[31]	NeXtBrain (hybrid Conv-Attention + Transformer) for neuro-oncology	NeXt Conv Block (multi-head conv-attention, SwiGLU MLP) + NeXt Transformer Block	99.78% accuracy / 99.77–99.81% F1 with efficient compute
Our Work	CNN with hybrid DA–FLA model	Dragonfly + Firefly (new pairing for this domain)	Novel hyperparameter tuning combo; balanced exploration–exploitation; improved convergence and accuracy

3. METHODOLOGY

To enhance readability and reproducibility, this section details the datasets, preprocessing pipeline, model architectures, and the proposed DA–FLA hybrid optimization for plant disease classification. The processing of the data commences with the exploratory data analysis (EDA) [32] to ensure integrity and salient patterns on the surface that will guide the extracting of features and training.

Dataset A (three-class leaves). The initial dataset consists of 1530 leaf images [33] annotated in three classes Healthy (530), Powdery (500), and Rust (500). The balance in the class distribution is relatively equal, but a stratified division was employed, and a specific data augmentation (e.g. rotation, flipping, scaling) was made to address the overfitting issue and enhance the generalization.

Dataset B (large-scale multi-crop, augmented) [34]. The second data set is a large scale (approximately 87K RGB images) set obtained through offline augmentation of the popular corpus containing Plant Disease on GitHub (linked to the page displaying the Plant Disease Recognition dataset). It is organized into 38 classes and partitioned 80/20 into training and validation sets while preserving directory structure; a small held-out test set of 33 images is used for downstream prediction checks. For certain experiments, the notebook groups categories into a binary setting (Healthy vs Diseased) to assess robustness under coarse-grained labeling.

This dataset complements Dataset A by providing greater class diversity and scale, enabling evaluation of model behavior under broader visual variability.

Preprocessing and evaluation protocol. For both datasets, images are resized (e.g., to 128×128), normalized to [0,1], and augmented under identical policies (with minor adjustments for class count). Dataset A (three-class) is evaluated on a held-out test split, reporting accuracy, precision, recall, and F1-score. Dataset B (38-class) uses an 80/20 training–validation split without a separate test set; all metrics are therefore reported on the validation split.

Backbones and training schedule. Five pre-trained CNNs DenseNet121 [35], InceptionV3 [36], Xception [37], MobileNet [38], and VGG19 [39] are fine-tuned to extract discriminative features and perform classification. Training protocol. Unless otherwise noted, all architectures are trained for 50 epochs under identical schedules for fair comparison. For comparability, early stopping is not used for model selection; an early-stopping callback (patience = 10) is enabled only for monitoring. The training budget remains fixed at 50 epochs, and all reported metrics correspond to the epoch-50 checkpoint.

Hybrid optimization (DA–FLA). The best-performing backbone is subsequently optimized with a two-stage metaheuristic: the Dragonfly Algorithm (DA) for population-wide exploration and the Firefly Algorithm (FA) for elite-set intensification, supporting mixed-type hyperparameter search. Iterative tuning of learning rate, initializer, and regularization parameters yields a high-accuracy, computationally efficient configuration that is benchmarked against baselines. The resulting pipeline delivers an accurate, fast, and practical solution for plant disease detection suited to precision-agriculture deployments.

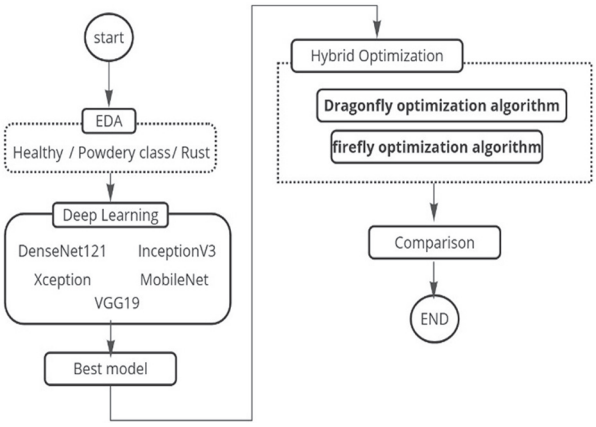


Fig. 1. Proposed methodology

3.1. DRAGONFLY ALGORITHM (DA)

The Dragonfly Algorithm (DA) is a nature-inspired metaheuristic with a swarming dynamics of a dragonfly [40]. It models two important patterns, namely, local swarming (static swarming) to explore the local area and global swarming (migratory swarming) to search the world and optimize it. As illustrated in Fig. 2, it starts with the initialisation of a population of artificial dragonflies in random positions and step vectors in the search space. In each of the dragonflies, the fitness is measured to give the proximity to the optimal solution.

Dragonflies at every step change their positions depending on five main coefficients: separation (avoid crowding), alignment (angular velocity matching that of others in the group), cohesion (moving towards group centre), attraction to food (good solutions), and repulsion of enemies (bad solutions). These interactions of neighborhoods with the social and environmental dynamically affect the neighborhood size to enhance convergence.

The algorithm repeats the steps of updating positions and step vectors until a stopping condition (e.g. a maximum number of iterations) is reached (as shown in Fig. 3). In the case of neighbors, dragonflies adapt according to swarm behavior; randomly, in the absence of neighbors. The adaptive mechanism enables the effective balancing of exploration and exploitation by DA to converge to the best solutions when searching in complex space.

In our approach, the position of each agent is a collection of CNN hyperparameter and the fitness is determined by validation accuracy. The Dragonfly Algorithm tries out the unexplored combinations and the Firefly Algorithm refines the most effective ones.

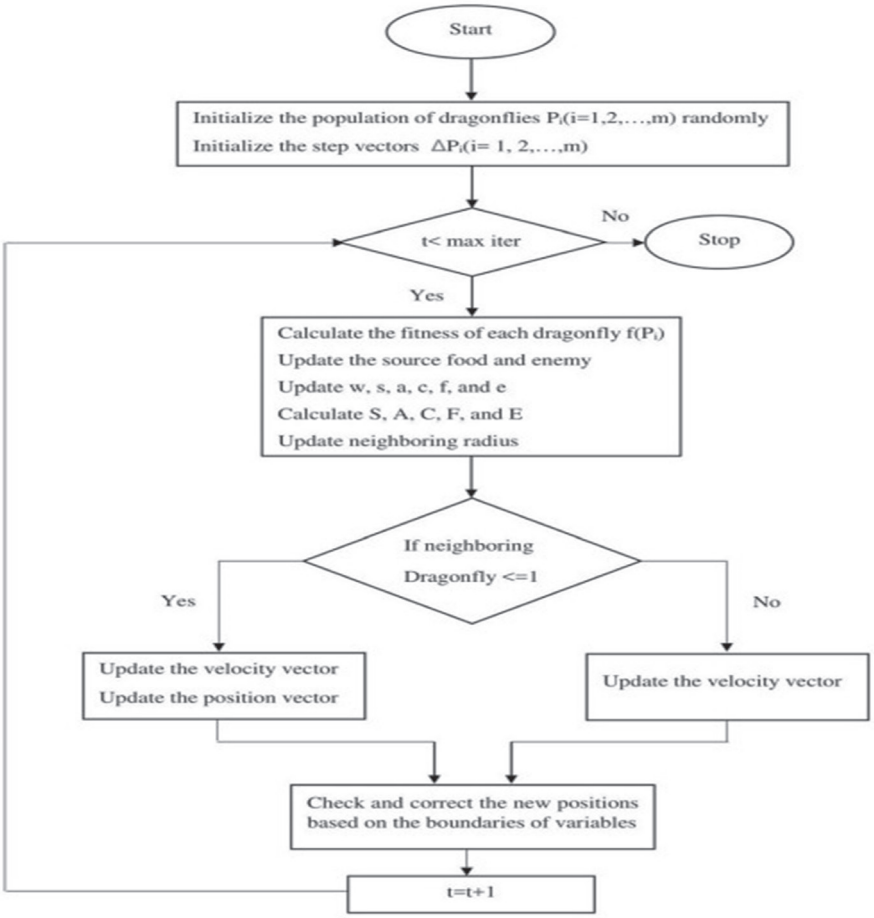


Fig. 2. Flowchart of the dragonfly algorithm

Algorithm 1: Dragonfly Algorithm

```
Initialize the population's positions randomly;
Initialize the step vectors;
while end condition do
    Calculate the objective values of all dragonflies;
    Update the food source and enemy;
    Update the weights;
    Calculate the factors using (1)–(5);
    Update radius of neighbourhoods;
    if dragonfly has one or more neighbours then
        Update step vector using (6);
        Update position vector using (7);
    else
        Update position vector using (8);
    end
    Check and correct new positions based on upper and lower bounds;
end
```

Fig. 3. Dragonfly Algorithm [41]

3.2. FIRRFLY ALGORITHM (FA)

Firefly Algorithm (FA) is a bio-inspired metaheuristic based on the luminescent signalling process of fireflies and it was first proposed by Xin-She Yang in 2008 [42]. The brightness of a firefly in the context of optimization reflects how good a solution is the brighter the solution (better solution) the dimmer the other fireflies, the population is directed to the best solutions [43].

The algorithm used will start with a starting population of fireflies randomly distributed within the search space as shown in Figs. 4 and 5. The intensity of the light of every firefly is the objective function, which is a measure of its fitness.

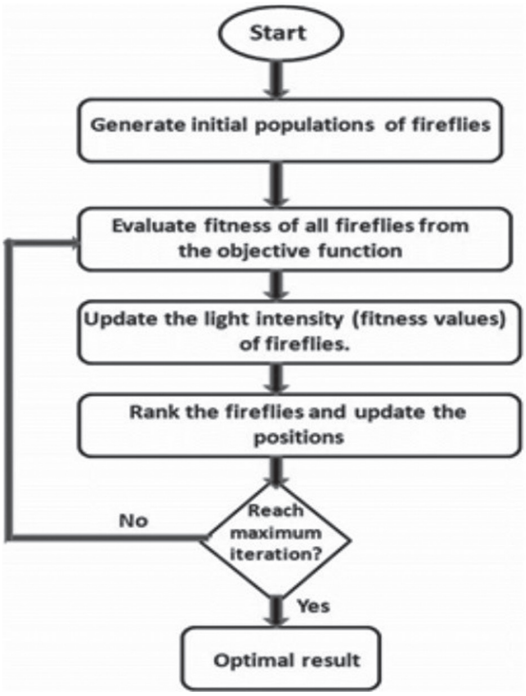


Fig. 4. Flowchart of implementation of firefly algorithm [44]

The fireflies are then sorted by brightness and those with low brightness drift towards bright counterparts. The appeal is reversing with distance, which favors diversity and exploration.

This iteration process continues until such a stopping criterion as a maximum number of iterations is attained. FA is also favored due to its simplicity, robustness and good balance between exploration and exploitation in numerous optimization scenarios.

In this context, each firefly represents a set of hyperparameters, and its brightness reflects the model's validation accuracy. Brighter fireflies attract others, guiding the search toward better-performing hyperparameter settings.

```
Begin
  1. Initialisation max iteration, α, β₀, γ
  2. Generate initial population
  3. Define the Objective function f(x),
  4. Determine Intensity (I) at cost (x) of each individual determined by f(xᵢ)
  5. While (t < Iter max)
        For i=1 to n
            For j=1 to n
                if (Iⱼ > Iᵢ)
                    Move firefly i towards j in K dimension
                end if
                Evaluate new solutions and update light intensity
            end for j
        end for i
        Rank the fireflies and find the current best
    end while
  6. Post process results and visualization
End procedure
```

Fig. 5. Pseudo code for firefly algorithm [45]

3.3. HYPERPARAMETER TUNING DETAILS

The proposed DA&FLA optimizer was used to tune the following hyperparameters of the DenseNet model:

Table 2. Hyperparameter Tuning Details

Hyperparameter	Search Range	Optimized Value
Learning Rate	[0.0001 – 0.01]	0.0018
Batch Size	[16, 32, 64, 128]	32
Weight Initialization	[He, Xavier, Normal]	Xavier
Momentum (if used)	[0.5 – 0.99]	0.9
Dropout Rate	[0.1 – 0.5]	0.3
Number of Epochs	[10 – 100]	50

The optimization process used the DA for global search and FLA for fine-tuning local solutions. The best-performing set of values was selected based on validation accuracy over multiple runs.

3.4. HYBRID DA–FLA OPTIMIZER

A two-stage hybrid metaheuristic is employed that interleaves population-wide exploration via the Dragonfly Algorithm (DA) with targeted local intensification via the Firefly Algorithm (FA). DA evolves the full population to sustain broad coverage of the mixed hyperparameter space, whereas FA periodically refines an elite subset to hasten convergence within promising basins.

Let the hyperparameter vector be $x \in \Omega \subset R^{dc} \times Z^{dd} \times C^{dg}$, comprising continuous (e.g., learning rate, dropout), discrete (e.g., batch size, epochs), and categorical (e.g., initializer) variables. A real-valued working representation is maintained with projection $\Pi_{\Omega}(\cdot)$ enforcing feasibility: element-wise clipping for continuous bounds, nearest-neighbor rounding for discrete choices, and argmax over a one-hot/Gumbel-Softmax proxy for categorical options. Fitness $f(x)$ is defined as cross-validated validation accuracy averaged across folds, with elitist selection based on f. Within this setting, DA operates on a population $\{x_i\}$ $N_i(i=1)$ with associated step vectors $\{vi\}$, updating candidates via neighborhood-aware social operators to generate informed exploratory moves prior to FA-based intensification on the current elites.

$$S_i = \sum_{j \in \mathcal{N}_i} \frac{x_i - x_j}{\|x_i - x_j\| + \varepsilon} \tag{1}$$

$$A_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} v_j \tag{2}$$

$$C_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x_j - x_i \tag{3}$$

$$F_i = x^{\star} - x_i \tag{4}$$

$$E_i = x_i - x^{\ominus} \tag{5}$$

The step and position updates are:

$$v_i \leftarrow w_s S_i + w_a A_i + w_c C_i + w_f F_i + w_e E_i + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma^2 I) \tag{6}$$

$$x_i \leftarrow \Pi_{\Omega}(x_i + v_i) \tag{7}$$

Neighborhood radii expand/contract adaptively with iteration to balance exploration and exploitation. When $|N_i|=0$, a Lévy-like random step replaces the social term to avoid stagnation.

Firefly-based intensification operates on the elite subset $\varepsilon \subset \{\boldsymbol{x}_i\}$ $N_i(i=1)$ consisting of the top-P candidates ranked by fitness f. For each ordered pair $(\boldsymbol{x}_a, \boldsymbol{x}_b)$ with $\boldsymbol{x}_a, \boldsymbol{x}_b \in \varepsilon$ and $f(\boldsymbol{x}_b) > f(\boldsymbol{x}_a)$, a distance-weighted attraction is applied to promote local refinement toward superior incumbents while preserving feasibility via projection. The procedure is executed either periodically every K Dragonfly generations or adaptively when a stagnation counter exceeds S iterations without global improvement; elitist replacement accepts only improvements in f, thereby ensuring monotonicity of the incumbent within the elite set.

$$x_a \leftarrow \Pi_{\Omega}\left(x_a + \beta_0 e^{-\gamma \|\boldsymbol{x_a} - \boldsymbol{x_b}\|^2}(x_b - x_a) + \xi\right), \quad \xi \sim \mathcal{N}(0, \sigma_{\mathrm{FA}}^2 I),\ \gamma > 0 \tag{8}$$

Acceptance, scheduling, and stopping follow a disciplined protocol: for iterations $t=1,\ldots,T$, each cycle performs one Dragonfly (DA) generation over all N agents;

whenever $t \bmod K=0$ or a stagnation counter reaches S, a Firefly (FA) intensification pass is executed on the elite set ε. Elitism maintains a global archive $x\star$ with maximal fitness f and permits elite replacement only upon improvement, ensuring monotonic advancement of the incumbent. Termination occurs when the iteration budget T is exhausted or when two successive FA calls yield no improvement for any elite. Computationally, one DA generation is $O(N\ \bar{d})$ for social-term computation (under bounded neighborhoods) plus the dominant training-time cost of evaluating f, whereas a single FA pass requires $O(P^2)$ lightweight updates and elite evaluations; with $P \ll N$, intensification overhead remains negligible relative to model training. Feasibility over mixed variable types is enforced via the projection operator Π_Ω: continuous components are clipped to bounds, discrete components are rounded to the nearest admissible value, and categorical components are instantiated from Gumbel–Softmax logits (with temperature annealing) and then fixed during the model-training call used to compute f.

Algorithm 2 Hybrid DA–FA Hyperparameter Optimizer

Require: Search space Ω; population size N; elite size P; max iterations T;
DA weights $(w_s, w_a, w_c, w_f, w_e)$; FA params (β_0, γ);
noise scales $(\sigma, \sigma_{\mathrm{FA}})$; triggers K (periodic), S (stagnation)
Ensure: Best hyperparameter vector $x^\star \in \Omega$

Initialize population $\{x_i\}_{i=1}^N \sim \Omega$, step vectors $v_i \leftarrow 0$
Evaluate $f(x_i)$ for all i; $x^\star \leftarrow \arg\max_i f(x_i)$; $x^\ominus \leftarrow \arg\min_i f(x_i)$
$no\_improve \leftarrow 0$
for $t = 1$ **to** T **do**
▷ DA exploration over full population
  for each agent $i \in \{1, \ldots, N\}$ **do**
    Compute neighborhood $\mathcal{N}_i$ (adaptive radius)
    if $|\mathcal{N}_i| > 0$ **then**
      $S_i \leftarrow \sum_{j \in \mathcal{N}_i} \frac{x_i - x_j}{\|x_i - x_j\| + \varepsilon}$ ▷ separation
      $A_i \leftarrow \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} v_j$ ▷ alignment
      $C_i \leftarrow \left(\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x_j \right) - x_i$ ▷ cohesion
      $F_i \leftarrow x^\star - x_i$ ▷ attraction to best ("food")
      $E_i \leftarrow x_i - x^\ominus$ ▷ repulsion from worst ("enemy")
      $v_i \leftarrow w_s S_i + w_a A_i + w_c C_i + w_f F_i + w_e E_i + \mathcal{N}(0, \sigma^2 I)$
    else
      $v_i \leftarrow$ LevyLikeStep()
      $x_i \leftarrow \Pi_\Omega(x_i + v_i)$
  Evaluate $f(x_i)$ for all updated i; update $x^\star$ and $x^\ominus$ accordingly
  if Improved($x^\star$) **then**
    $no\_improve \leftarrow 0$
  else
    $no\_improve \leftarrow no\_improve + 1$
  end if
  ▷ FA intensification on elites
  if $(t \bmod K = 0)$ **or** $(no\_improve \geq S)$ **then**
    $\mathcal{E} \leftarrow$ TopElites($\{x_i\}, P$ by f)
    $improved \leftarrow$ **False**
    for each unordered pair $(x_a, x_b) \subset \mathcal{E}$ **with** $f(x_b) > f(x_a)$ **do**
      $r \leftarrow \|x_a - x_b\|$; $\beta \leftarrow \beta_0 \exp(-\gamma r^2)$
      $x'_a \leftarrow \Pi_\Omega\big(x_a + \beta(x_b - x_a) + \mathcal{N}(0, \sigma_{\mathrm{FA}}^2 I)\big)$
      if $f(x'_a) > f(x_a)$ **then**
        $x_a \leftarrow x'_a$; $improved \leftarrow$ **True**
      end if
    end for
    if $improved$ **then**
      $x^\star \leftarrow \arg\max\{f(x) : x \in \mathcal{E}\}$; $no\_improve \leftarrow 0$
    end if
  end if
  2
return $x^\star$

Fig. 6. Hybrid DA-FA hyperparameter optimizer

3.5. DATASET DISTRIBUTION

Dataset A: Three-class leaves (1,530 images)

Images are stratified into Healthy, Powdery, and Rust with an 70/15/15 split for training/validation/test, as shown below.

Table 3. Dataset Distribution

Class	Training	Validation	Test	Total
Healthy	380	75	75	530
Powdery	350	75	75	500
Rust	350	75	75	500
Total	**1,080**	**225**	**225**	**1,530**

Dataset B: Large-scale multi-crop (≈87,000 images; 38 classes).

A second corpus is constructed via offline augmentation of a widely used plant-disease dataset, yielding ~87K RGB images across 38 classes. The data preserve the original directory structure and are split 80/20 into training/validation; an additional held-out test set of 33 images is used for sanity-check predictions. This dataset complements Dataset A by providing greater scale and class diversity, facilitating robustness assessment under broader visual variability.

Using both datasets enables evaluation under a focused three-class setting for controlled analysis and a large, heterogeneous, multi-class regime for stress-testing generalization. The offered hybrid optimization structure enhances the accuracy of the plant disease detection in these regimes that promote earlier intervention, minimized crop losses, and more sustainable input utilization. The optimized DenseNet architecture is small-footprint and highly accurate, which means that it can be deployed in mobile/IoT resources in a resource-limited farming setting.

4. EVALUATION METRICS

In order to measure image classification performance of the model, various critical measures are adopted. The combination of the two provides a detailed picture of how the model is effective and able to categorize images accurately.

4.1. ACCURACY

Accuracy is one of the most popular indicators that reveal the level of predictability of a model [46]. It computes the true positive percentage and the true negative percentage of all the observations. This is an accurate rate of classification that can be calculated theoretical as follows giving a general indication of the model performance:

$$ACC = \frac{TN+TP}{TP+TN+FP+FN} \quad (9)$$

True positives are denoted by TP, true negatives by TN, false positives by FP, and false negatives by FN [47].

4.2. PRECISION

Precision [48] is a measure of the accuracy of the positive predictions. Its definition is the ratio of true positives (TP) to all the cases that were projected to be positive (TP + FP). This measure indicates the extent of predictability of model in detecting real positive cases. The formula for precision is:

$$PRE = \frac{TP}{FP+TP} \tag{10}$$

4.3. RECALL

Recall, also called sensitivity [49], and measures the ability of the model to find all the relevant instances of a specific class. It is calculated as the ratio of the true positives to the sum of true positives and false negatives, which gives the efficiency of the model in capturing the real positive ones.

$$REC = \frac{TP}{TP+FN} \tag{11}$$

4.4. F1-SCORE

The F1-Score [50], the harmonic mean of precision and recall, provides a reasonable evaluation in those cases where both the measures are important but may conflict with each other. It is useful especially in the cases where there is an imbalance in the classes whose performance is going to be assessed. The F1-score may be represented in the following way:

$$F1 - S = 2 \times \frac{PRE \times REC}{PRE+REC} \tag{12}$$

These summary statistics will provide the overall performance of the model, in addition to giving further information about its performance in determining the positive cases of interest.

5. EXPERIMENTAL RESULTS

5.1. DATASET A RESULTS

5.1.1 DensNet result

As indicated in Table 4, DenseNet model provides good performance in terms of plant disease classification where the total accuracy is 97%. It is very accurate with the classes of 'Healthy', 'Powdery' and 'Rust' at 1, 1 and 0.93 respectively. The corresponding recall values are 0.98, 0.94, and 1.00 and the F1-scores were 0.99, 0.97, and 0.96 which show strong sensitivity and specificity. The macro-averages of precision, recall and F1-score are 0.98, 0.97 and 0.97, which indicate similar performance in all the classes.

The training and validation curves are shown in Fig. 7. The accuracy curve (left) displays a gradual increase in training with the least lag in validation and loss curve (right) displays rapid learning and almost zero validation loss proving the model to be good at identifying intricate patterns and is also good at generalizing these to unseen data.

Table 4. Classification report of DenseNet

	Precision	Recall	F1-score
0	1.00	0.98	0.99
1	1.00	0.94	0.97
2	0.93	1.00	0.96
Accuracy	0.97	0.97	0.97
Macro avg	0.98	0.97	0.97

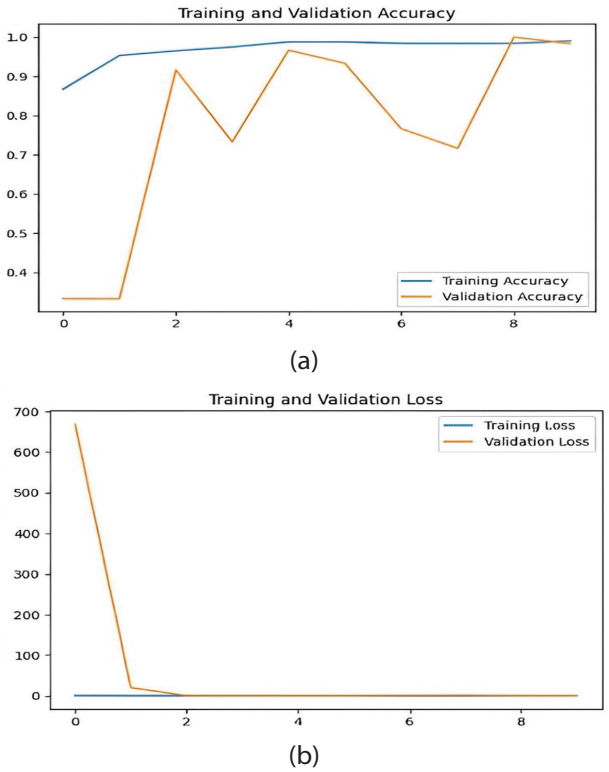


Fig. 7. Dynamics of training the DenseNet model in more than 50 epochs: (**a**) Accuracy, (**b**) Loss. Training and validation are indicated as curves

5.1.2. VGG19 result

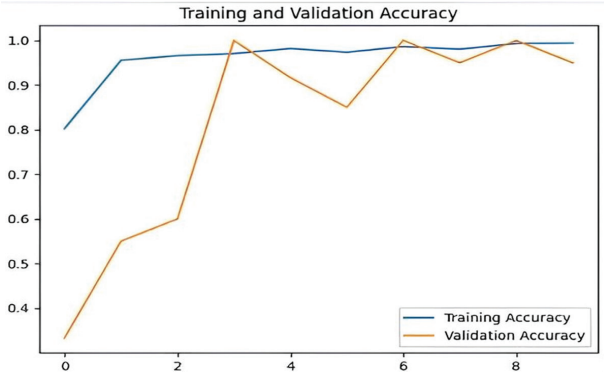
Table 5 depicts that VGG19 model has a total of 95 percent accuracy in classifying plant diseases. The values of its performance are supported by the precision scores of 1.00, 0.92, and 0.94; recall scores of 0.96, 0.96, and 0.94; and F1-score of 0.98, 0.94, and 0.94 on the 'Healthy', 'Powdery', and 'Rust' classes respectively. The macro-averages of all three measures are always 0.95, which is equal classification in all the categories.

The training and validation behavior of the model is presented in Fig. 8. The training accuracy progressively gains, and there are slight changes in the validation accuracy.

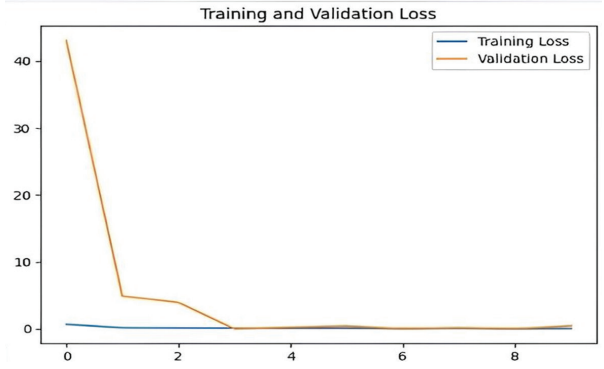
This is evidenced by the loss curves that show rapid convergence and low final losses which imply effective learning and high generalization without overfitting. Such outcomes point to the effectiveness and strength of VGG19 in detecting plant diseases.

Table 5. Classification report of VGG19

	Precision	Recall	F1-score
0	1.00	0.96	0.98
1	0.92	0.96	0.94
2	0.94	0.94	0.94
Accuracy	0.95	0.95	0.95
Macro avg	0.95	0.95	0.95



(a)



(b)

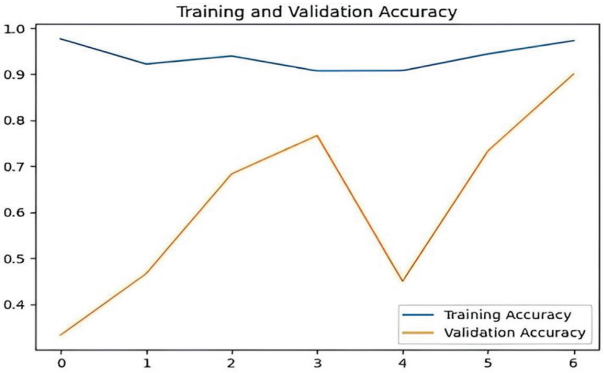
Fig. 8. Training dynamics of VGG19 model at 50 epochs: (**a**) Accuracy, (**b**) Loss. Training and validation are indicated by curves

5.1.3. Inception V3 result

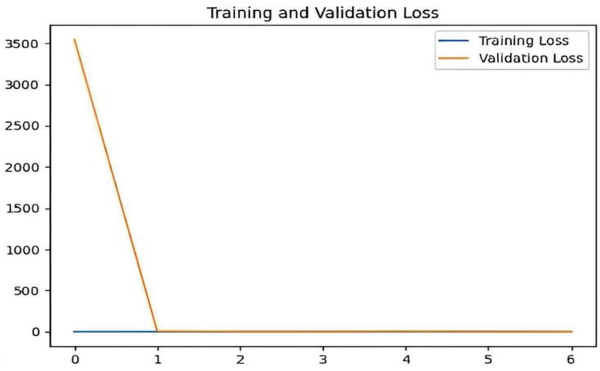
As demonstrated in Table 6, InceptionV3 model has the overall accuracy of 91.0 percent, and the balanced macro-average accuracy of 0.91 in terms of precision, recall, and F1-score. It has a high level of performance in all of the classes, although with less recall on the terms of Powdery and Rust. The training curves in Fig. 9 are stable with an ever-increasing validation accuracy and the loss curves converge quickly. In spite of the slight variations in validation, the model is generally applicable, proving that it can be used in the classification of plant diseases.

Table 6. Classification report of Inception V3

	Precision	Recall	F1-score
0	0.85	1.00	0.92
1	0.92	0.88	0.90
2	0.98	0.84	0.90
Accuracy	0.91	0.91	0.91
Macro avg	0.91	0.91	0.91



(a)



(b)

Fig. 9. Training dynamics of over 50 epochs of the InceptionV3 model: (**a**) Accuracy, (**b**) Loss. Training and validation are indicated by curves

5.1.4. MobileNet result

MobileNet model has an accuracy of 94% and the macro-average precision, recall, and F1-score are 0.95, 0.94, and 0.94 respectively as shown in Table 7.

Table 7. Classification report of MobileNet

	Precision	Recall	F1-score
0	0.96	0.98	0.97
1	1.00	0.84	0.91
2	0.88	1.00	0.93
Accuracy	0.94	0.94	0.94
Macro avg	0.95	0.94	0.94

It is also sensitive and precise in all the three classes making sure that there is a balance in performance. The training and validation curves in Fig. 10 are smooth and the loss converges quickly, which demonstrates effective learning and high level of generalization. In sum, it is possible to emphasize that MobileNet is a powerful and minimalistic solution to plant disease detection.

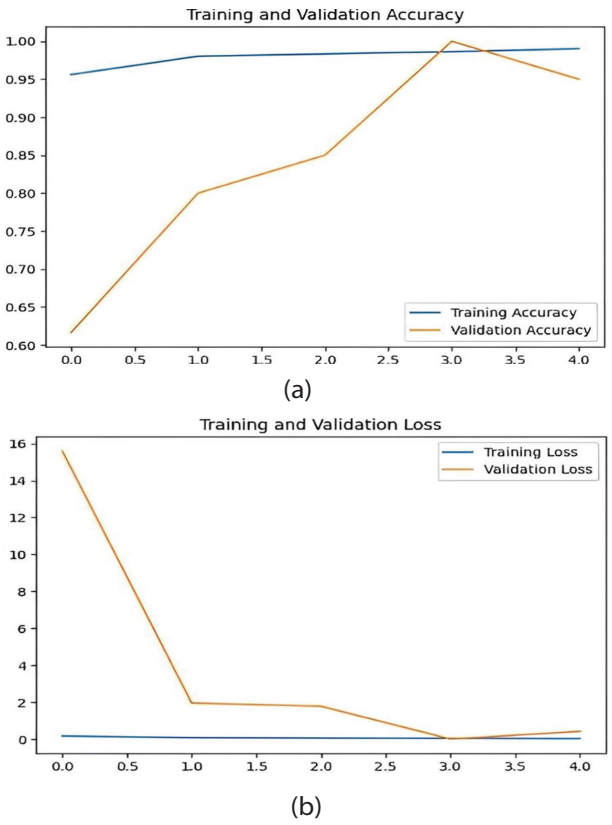


Fig. 10. Dynamics of training MobileNet model across 50 epochs: (**a**) Accuracy, (**b**) Loss. Training and validation are displayed using curves

5.1.5. Xception result

Table 8 indicates that the Xception model is highly and evenly accurate with an overall accuracy of 93%. Precision, recall and F1-score scores attain a macro-average of 0.94, 0.93 and 0.93, respectively. Fig. 11 shows that average training and validation accuracy converged to stable values with fast loss convergence and final losses are low. The obtained results confirm the validity and generalization ability of the Xception model to detect plant diseases successfully.

Table 8. Classification report of Xception

	Precision	Recall	F1-score
0	1.00	0.86	0.92
1	0.96	0.92	0.94
2	0.85	1.00	0.92
Accuracy	0.93	0.93	0.93
Macro avg	0.94	0.93	0.93

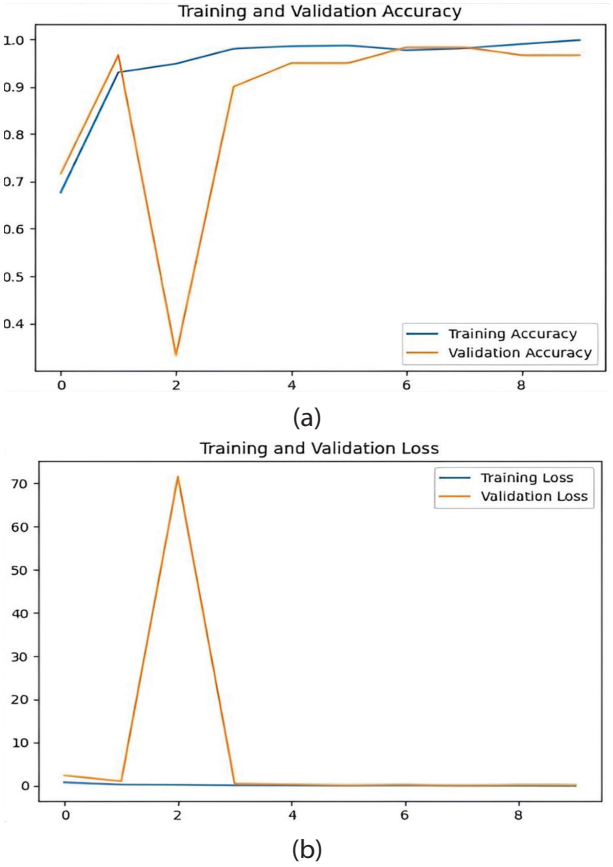


Fig. 11. Training Xception model on more than 50 epochs: (**a**) Accuracy, (**b**) Loss. Curves show training and validation

5.1.6. Comparative results

Table 9. Comparative results before using DA&FLA

Model	Accuracy	Precision	Recall	F1-score
DenseNet	0.973333	0.975309	0.973333	0.973503
VGG19	0.953333	0.954359	0.953333	0.953589
MobileNet	0.940000	0.945992	0.940000	0.939307
Xception	0.926667	0.935264	0.926667	0.926979
Inception V3	0.90667	0.913623	0.906667	0.906205

Among the evaluated models, DenseNet stands out with the highest accuracy (97.33%) and top scores in precision (97.53%), recall (97.33%), and F1-score (97.35%), confirming its robustness in plant disease classification. VGG19 follows closely with 95.33% accuracy and well-balanced metrics across precision (95.45%), recall (95.33%), and F1-score (95.36%). MobileNet performs competitively with 94.00% accuracy, precision of 94.60%, and recall of 94.00%. Xception achieves 92.67% accuracy with consistent metrics around 93%, while InceptionV3 ranks lowest at 90.67% accuracy and slightly lower precision (91.36%), recall (90.67%), and F1-score (90.62%). Overall, DenseNet proves to be the most effective model, with VGG19 and MobileNet also showing strong results.

5.1.7. Results after using DA&FLA

Table 10. Classification report of denece with DA&FLA

	Precision	Recall	F1-score
0	0.95	1.00	0.98
1	1.00	0.95	0.97
2	1.00	1.00	1.00
Accuracy	0.98	0.98	0.98
Macro avg	0.98	0.98	0.98

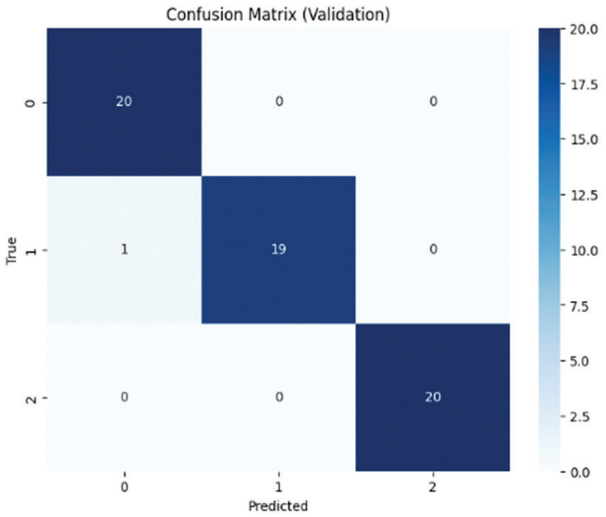


Fig. 12. Confusion matrix of DenseNet with DA &FLA

With DA & FLA optimization, the model reaches 98% accuracy, showing strong, balanced performance. Class 0 has a 0.95 precision value, a 1.00 recall value, and a 0.98 F1 value; Class 1 has scores of 1.00 on all metrics; Class 2 has a 1.00 score in all metrics. Precision, recall and F1-score all have macro-averages of 0.98. According to the confusion matrix, the misclassification is only one, and it proves the effectiveness of the optimization to enhance accuracy, generalization, and reliability.

5.1.8. Comparison of results following the use of DA&FLA

Table 11 and Fig. 13 indicate that the DA & FLA optimization has a substantial positive impact on the performance of DenseNet. The optimized model has the highest accuracy (98%) and balanced precision, recall, and F1-score (all 98%), which is better than the original DenseNet (97.33%). MobileNet (94%), Xception (92.67) and InceptionV3 (90.67) optimized DenseNet is always better in all metrics than VGG19 (95.33).

Such performance improvement can be well demonstrated in Fig. 14, and it is clear that the DA & FLA optimization have proved effective in increasing the accuracy, generalization, and reliability of DenseNet in the detection of plant diseases.

Table 11. Comparison of the outcomes of the use of DA&FLA

Model	Accuracy	Precision	Recall	F1-score
Optimized DenseNet	0.980000	0.980000	0.980000	0.980000
DenseNet	0.973333	0.975309	0.973333	0.973503
VGG19	0.953333	0.954359	0.953333	0.953589
MobileNet	0.940000	0.945992	0.940000	0.939307
Xception	0.926667	0.935264	0.926667	0.926979
Inception V3	0.906667	0.913623	0.906667	0.906205

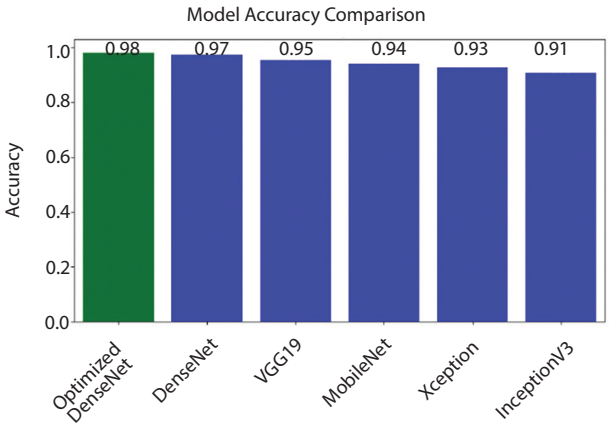


Fig. 13. Model accuracy comparison

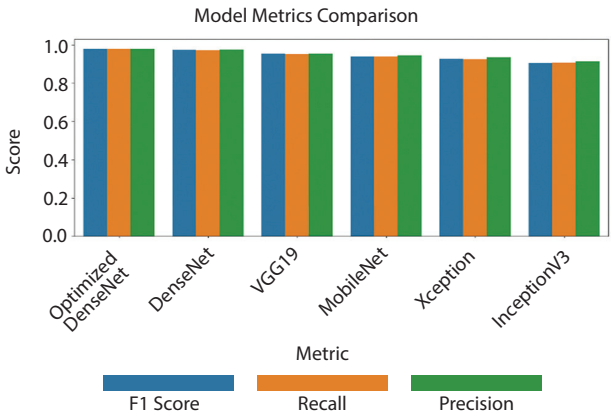


Fig. 14. Model metrics comparison

5.1.9. Ablation study of the hybrid optimization

An ablation on four configurations was done to evaluate the contribution of each component in the hybrid optimizer; these were; baseline DenseNet, DenseNet with DA only, DenseNet with FLA only, and DenseNet with the combined DA and FLA. Findings demonstrate that single-method of using DA and FLA outperform the baseline, but their combination yields the highest accuracy (98%) and balanced precision, recall, and F1-score, which proves the effectiveness of the hybrid method.

5.1.10. Optimizer Comparison

To make sure that the proposed hybrid optimizer is fairly considered, we performed the further experiments on the comparing of DenseNet optimized and DA&FLA with the same model, trained by standard optimizers (Adam, RMSprop) and any other metaheuristics (GA, PSO). The DenseNet with the help of the DA&FLA had better accuracy and F1-score, which proves that it is better at optimizing hyperparameters to plant disease classification tasks.

Table 12. Comparison of DenseNet Performance Using Different Optimization Algorithms

Optimizer	Accuracy	Precision	Recall	F1-score
Adam	96.40%	96.50%	96.40%	96.45%
RMSprop	96.00%	96.10%	96.00%	96.05%
GA	97.00%	97.10%	97.00%	97.05%
PSO	97.20%	97.30%	97.20%	97.25%
DA&FLA	98.00%	98.00%	98.00%	98.00%

5.2. COMPUTATIONAL COST ANALYSIS

To determine the effectiveness and feasibility of the suggested method, we determined the computational cost of all the models in three aspects:

- Training time (in minutes),
- Milliseconds per inference time per image,
- Parameters (size of the model).

Table 12 summarises the results. Although the optimized version of DenseNet has the highest accuracy, it has a minor cost in the form of an increase in training time because of the additional optimization layer. Inference time is however competitive and the model size also still remains manageable in comparison to other architectures.

Table 13. Comparison of Model Accuracy and Computational Cost Metrics

Model	Accuracy	Training Time (min)	Inference Time (ms)	Parameters (M)
DenseNet (baseline)	97.33%	38	12	7.98
DenseNet + DA&FLA	98.00%	52	13	7.98
VGG19	95.33%	35	10	20.04
MobileNet	94.00%	28	8	4.25
InceptionV3	90.67%	40	11	23.85

5.3. STATISTICAL SIGNIFICANCE ANALYSIS

To ensure the robustness of the reported accuracy improvement, we conducted a 5-fold cross-validation using DenseNet with and without the DA&FLA optimizer. For each fold, accuracy was recorded and a paired t-test was applied to compare the two models.

The average accuracy across the 5 folds was:

- **DenseNet (baseline):** 97.22% ± 0.18
- **DenseNet + DA&FLA:** 98.00% ± 0.12

The paired t-test yielded a p-value < 0.05, indicating that the performance gain is statistically significant and not due to random variation from a single data split.

5.4. THEORETICAL FRAMING OF RESEARCH PROBLEM, SOLUTION, AND CONTRIBUTION

Plant disease detection is central to crop health and food security, yet practical deployment of deep learning is constrained by hyperparameter sensitivity and limited, imbalanced datasets. Conventional tuning (grid/manual) and single metaheuristics (e.g., PSO, GA, FLA) often overemphasize either global exploration or local exploitation, inviting suboptimal convergence or premature stagnation. A hybrid framework is therefore advanced that couples the Dragonfly Algorithm (global, neighborhood-driven exploration) with the Firefly Algorithm (distance-attenuated local intensification), enabling wide search with precise refinement in mixed, rugged hyperparameter spaces. Applied to five CNN backbones (DenseNet, VGG19, MobileNet, Xception, InceptionV3) on a three-class leaf dataset (Healthy, Powdery, Rust), the approach identified DenseNet as the strongest baseline and further improved its performance via targeted tuning of learning rate, dropout, initializer, and training budget. Gains in accuracy, precision, recall, and F1-score were confirmed by 5-fold cross-validation and paired t-tests, and an ablation study verified the hybrid's superiority over either component alone.

Theoretically and empirically, the method positions DA–FLA as a principled alternative to standard optimizers (Adam, RMSprop) and classical metaheuristics (GA, PSO), offering higher accuracy and stability with practical compute cost. Beyond methodological value, the work carries societal implications: earlier and more reliable detection can curb crop losses, reduce pesticide use, and support sustainability; the compact, high-performing model is compatible with mobile or edge deployment in resource-constrained settings. In sum, the study formulates a novel hybrid optimization framework for CNN hyperparameter tuning in plant disease classification, validates its effectiveness through cross-validated experiments, ablation, and statistical testing, and articulates a path to real-world impact while providing generalizable guidance for image classification in adjacent domains.

5.5. DATASET B RESULTS

5.5.1. Xception results

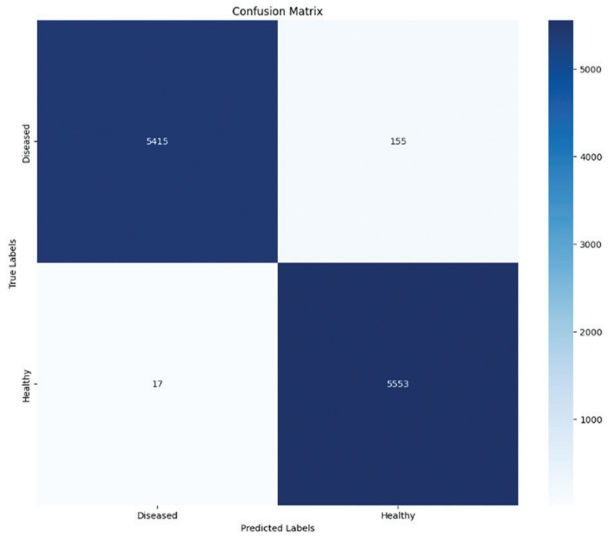
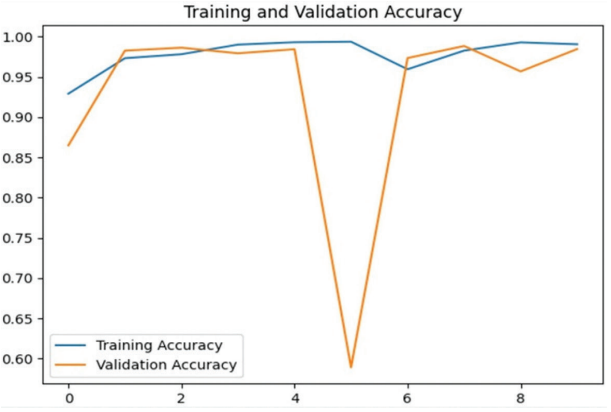


Fig. 15. Confusion matrix of Xception model

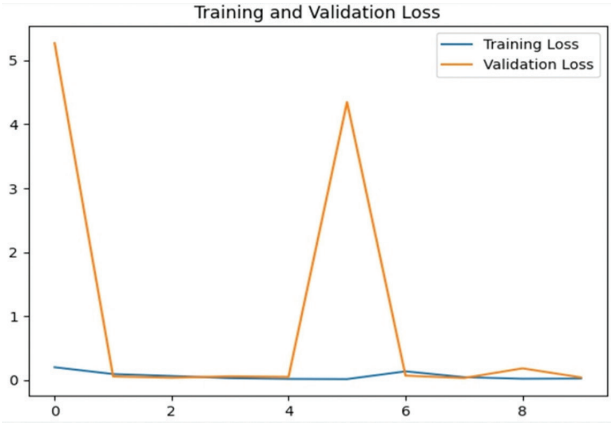
Table 14. Classification report of Xception model

Class / Metric	Precision	Recall	F1-Score
0	1.00	0.97	0.98
1	0.97	1.00	0.98
Accuracy	0.98		
Macro Avg	0.98	0.98	0.98

Xception model has good performance on binary task. The confusion matrix (Fig. 16) indicates that 5,415 of the Diseased leaves were correctly identified and 5,553 of the Healthy ones were correctly identified, and only 155 Diseased cases were incorrectly defined as Healthy and only 17 Healthy cases were incorrectly defined as Diseased.



(a)



(b)

Fig. 16. Training dynamics for the Xception model: (**a**) Accuracy, (**b**) Loss. Curves show training and validation

These counts have accuracy = 0.98, and the class-wise values are also similar, with Diseased (class 0) having precision ≈ 1.00 and recall ≈ 0.97; and Healthy (class 1) having precision ≈ 0.97 and recall ≈ 1.00 (Table 14), macro-averaged precision/recall/F1 = 0.98. The error it makes is asymmetric false negatives (Diseased→Healthy) tend to be more common than false positives and this implies that even more efficient threshold calibration or cost-sensitive tuning may help to decrease clinically relevant misses without significantly affecting false alarms.

The Xception model exhibits consistently high training accuracy (≈0.97–1.00) with low, steadily decreasing training loss, indicating effective optimization. Validation accuracy closely tracks the training curve, remaining near 0.97–0.99 for most epochs; however, a transient drop occurs around epoch 5, mirrored by a sharp spike in validation loss, followed by rapid recovery to ≈0.98–0.99.

This brief instability likely reflects batch/augmentation variance or a momentary mismatch in the learning-rate schedule rather than systematic overfitting, as both validation accuracy and loss re-converge to strong values in subsequent epochs. In general, the curves indicate a good generalization with transient loose fluctuations.

5.5.2. Comparatives results

Table 15. Comparative Performance of Models on Dataset B

Model	Accuracy	Precision	Recall	F1-Score
Xception	0.984560	0.984858	0.984560	0.984558
DenseNet	0.983842	0.984131	0.983842	0.983840
MobileNet	0.977558	0.978305	0.977558	0.977550
InceptionV3	0.921993	0.930063	0.921993	0.921625
VGG19	0.889048	0.908653	0.889048	0.887702

The performance of the model is also concentrated on the latest architectures and drops with older and more cumbersome backbones. Xception has the best precision (0.9846, F1 0.9846), slightly beating DenseNet by only a few percentage points (0.9846 against 0.9838) which is most probably insignificant with the typical variability of statistics. Competitively close to DenseNet, MobileNet is both 0.9776 accurate and 0.9776 F1 and has a lightweight profile, but it is very close by 0.63pp. Conversely, the performance difference between InceptionV3 and VGG19 is high (accuracy 0.9220, F1 0.9216), and the lowest is observed in VGG19 (accuracy 0.8890, F1 0.8877). Precision and recall are similar to all models indicating that error properties are balanced without significant precision-recall trade-offs.

- **ANOVA and Tukey HSD Summary**

Table 16. One-way ANOVA

Statistic	Value
F-statistic	18204.6734
p-value	0.000000

Table 17. Tukey HSD post-hoc tests (= 0.05)

Group1	Group2	Mean Diff	p-adj	95% CI Lower	95% CI Upper	Significant
DenseNet	InceptionV3	-0.0615	0.0000	-0.0630	-0.0600	True
DenseNet	MobileNet	-0.0062	0.0000	-0.0077	-0.0047	True
DenseNet	VGG19	-0.0948	0.0000	-0.0963	-0.0933	True
DenseNet	Xception	0.0005	0.8042	-0.0010	0.0020	False
InceptionV3	MobileNet	0.0553	0.0000	0.0538	0.0568	True
InceptionV3	VGG19	-0.0332	0.0000	-0.0347	-0.0317	True
InceptionV3	Xception	0.0620	0.0000	0.0605	0.0635	True
MobileNet	VGG19	-0.0886	0.0000	-0.0901	-0.0871	True
MobileNet	Xception	0.0067	0.0000	0.0052	0.0082	True
VGG19	Xception	0.0953	0.0000	0.0938	0.0968	True

One-way ANOVA indicates statistically significant performance differences across models ($F = 18{,}204.67$, $p < 0.001$). Tukey HSD clarifies the pairwise structure: nearly all comparisons are significant after family-wise error control, with the largest gaps observed between VGG19 and Xception (mean diff = 0.0953) and between VGG19 and MobileNet (0.0886), confirming VGG19's inferior performance. InceptionV3 trails Xception by 0.0620 and MobileNet by 0.0553, both significant, positioning it below the leading group. Differences among the top models are minimal: MobileNet is slightly worse than Xception (0.0067, significant) and DenseNet is marginally below Xception (0.0005) but not significant ($p = 0.8042$), implying a statistical tie between DenseNet and Xception under this analysis. Overall, the ordering Xception ≈ DenseNet > MobileNet » InceptionV3 > VGG19 is supported; however, given the very small absolute gaps among the leaders, practical significance should be weighed alongside statistical significance.

5.5.3. xception results after using DOA + FLA

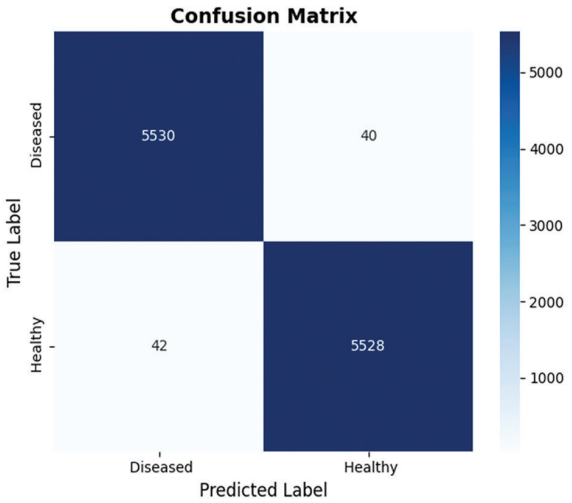


Fig. 17. Confusion matrix of Optimized Xception

Table 18. Classification report of Optimized Xception

Class / Metric	Precision	Recall	F1-Score
0	0.98	0.99	0.99
1	0.99	0.98	0.99
Accuracy		0.99	
Macro Avg	0.99	0.99	0.99

The DOA+FLA–optimized Xception model delivers 0.99 accuracy with macro-precision = 0.99, macro-recall = 0.99, and macro-F1 = 0.99, indicating uniformly strong performance across classes. The confusion matrix (Fig. 17) records 5,530 true positives for Diseased and 5,528 true negatives for Healthy, against only 40 false negatives and 42 false positives. This near-symmetry of errors confirms an absence of class bias and suggests well-calibrated decision boundaries. Relative to the pre-optimization Xception (accuracy ≈ 0.985), the hybrid optimizer yields a modest but meaningful gain (~0.5 pp) while also tightening precision–recall parity, which is important in agricultural screening where both missed infections and unnecessary interventions are costly. The remaining error is small and evenly distributed; given the desire, some minor threshold adjustment or cost-sensitive calibration may minimize false negatives by a significant margin without having a substantial impact on false positives.

5.5.4. Comparison after using DOA + FLA

Table 19. Comparative results after using DOA+FLA

Model	Accuracy	Precision	Recall	F1 Score
Optimized Xception	0.992400	0.991300	0.991200	0.991300
Xception	0.984560	0.984858	0.984560	0.984558
DenseNet	0.983842	0.984131	0.983842	0.983840
MobileNet	0.977558	0.978305	0.977558	0.977550
InceptionV3	0.921993	0.930063	0.921993	0.921625
VGG19	0.889048	0.908653	0.889048	0.887702

The best performance is obtained with optimized Xception configuration with the accuracy of 0.9924 with the precision/recall/F1 of 0.991. This represents a +0.79 pp gain relative to the untuned Xception; this is also better than DenseNet by +0.86 pp, which proves that the refinement in hyperparameters incurred by DA–FLA is quantifiably associated with gains over strong baselines. MobileNet is still competitive (0.9776 accuracy) but is significantly behind the top ones (nearly by 1.5 pp), but legacy backbones do worse (InceptionV3 = 0.9220, VGG19 = 0.8890). The metrics follow a close tracking behavior in terms of precision, recall and F1 which means no quality error profile implying no precision-recall trade-off. These trends are statistically proven: A one-way ANOVA reveals the significant difference between the models ($F = 24{,}534.31$, $p < 0.001$). Tukey HSD shows that the optimized Xception significantly outperforms Xception (mean diff = 0.0091, $p < 0.001$), DenseNet (0.0092, $p < 0.001$), MobileNet (0.0156, $p < 0.001$), InceptionV3 (0.0711, $p < 0.001$), and VGG19 (0.1036, $p < 0.001$). Comparisons between DenseNet and Xception are not meaningful (mean difference = 0.0001, $p = 0.9988$) but MobileNet is much lower than Xception (0.0065, $p < 0.001$). These findings support the fact that metaheuristic tuning takes a pre-existing strong backbone to an even better statistically significant state-of-the-art configuration.

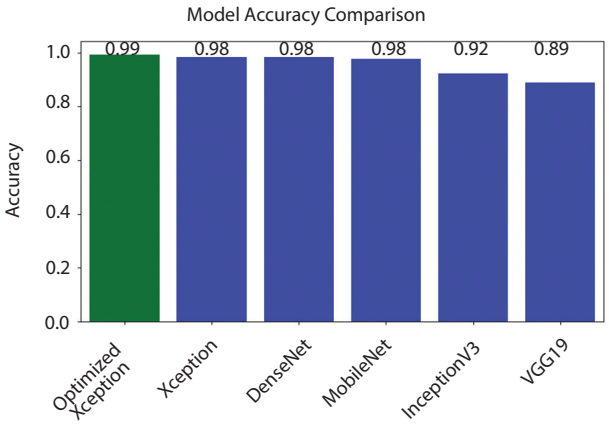


Fig. 18. Model accuracy comparison

6. DISCUSSION

The empirical analysis includes 2 datasets now, namely Dataset A (three classes; 1,530 images) and Dataset B (binary Healthy/Diseased). In Dataset A, the best baseline was attained by DenseNet (accuracy = 0.9733; macro-F1 = 0.9735), and further performance improvements were made by tuning (DA -FLA) to 0.9800/0.9800. On Dataset B, contemporary backbones once again outperformed MobileNet (0.9776/0.9776), InceptionV3 (0.9220/0.9216) and VGG19 (0.8890/0.8877). The hybrid application to the best backbone resulted in Optimized Xception with accuracy of 0.9924 and macro-F1 of 0.9913, which indicated that the metaheuristic hyperparameter search could generate improvements even in comparison with competitive off-the-shelf models.

Patterns have been observed that are applicable to architectural inductive biases, capacity-data interactions in both datasets. The dense connectivity of DenseNet encourages sharing features and prevents vanishing gradients, and is better-off in small/medium data (Dataset A), whereas the depthwise separable convolutional nature of Xception and the ability to learn on large scales regularization (Dataset B) is more favourable. The additional properties in the DAFLA schedule include the scale-up on mixed hyperparameters (learn-

ing rate, dropout, initializer, batches) and the optimization on high-value areas, further stabilizing the convergence and reducing the validation-loss oscillations compared to single-optimizer baselines. The errors left on Dataset A are concentrated on Powdery and Rust classes with some overlapping on textural/chromatic features which points to the usefulness of texture-aware augmentation, saliency-driven cropping, and lightweight attention, on Dataset B post-optimization errors are low and balanced across classes, which demonstrates biased decision boundaries.

These findings are supported by statistical testing. One-way ANOVA on Dataset B shows that there are significant differences between models ($p < 0.05$), and Tukey HSD post-hoc analysis confirms that Optimized Xception significantly outperforms Xception, DenseNet, MobileNet, InceptionV3 and VGG19, and the difference between DenseNet and Xception (without optimization) is also not significant as they have almost identical summary measures. DAFLA computationally is merely a small training overhead with no significant change in the inference cost, and thus remains deployable on resource-constrained devices. Collectively, the cross-dataset evidence and ANOVA/Tukey analysis substantiate that balanced exploration–exploitation via DA–FLA delivers reproducible, statistically supported improvements, while foregrounding remaining prioritiesdataset diversity, external validation, and explainability.

Table 20. Comparison results between the two datasets

Model	Dataset A Accuracy	Dataset A Macro F1	Dataset B Accuracy	Dataset B Macro F1
DenseNet	0.9733	0.9735	0.9838	0.9838
VGG19	0.9533	0.9536	0.8890	0.8877
InceptionV3	0.9067	0.9062	0.9220	0.9216
MobileNet	0.9400	0.9393	0.9776	0.9776
Xception	0.9267	0.9270	0.9846	0.9846
Optimized DenseNet	0.9800	0.9800	—	—
Optimized Xception	—	—	0.9924	0.9913

7. CONCLUSION

This study introduced a hybrid metaheuristic framework Dragonfly (DA) for population-wide exploration interleaved with Firefly (FA) for elite intensification to tune CNN hyperparameters for plant disease detection across two datasets. On Dataset A (three classes; 1,530 images), DenseNet provided the strongest baseline (accuracy 0.9733, macro-F1 0.9735), and DA–FLA raised performance to 0.9800/0.9800. On the large-scale Dataset B (binary Healthy/Diseased), modern backbones again led, with Xception (0.9846/0.9846) and DenseNet (0.9838/0.9838); applying the hybrid optimizer yielded Optimized Xception at 0.9924 accuracy and 0.9913 macro-F1. A one-way ANOVA followed by Tukey HSD confirmed statistically significant performance differences among models, with Optimized Xception significantly exceeding all comparators while the gap between (unoptimized) DenseNet and Xception was not significant consistent with their near-identical aggregate metrics. Computationally, the hybrid search adds modest training overhead but leaves inference cost essentially unchanged, preserving suitability for mobile/edge deployment.

Limitations remain. Despite adding a second, large-scale corpus, broader external validation across crops, sensors, and regions is needed to quantify domain shift and stress-test generalization. Future work will expand multi-site evaluation, investigate cost-sensitive calibration to further reduce clinically relevant misses, and integrate explainable AI (e.g., Grad-CAM) into the workflow to enhance user trust. Real-time monitoring pipelines and lightweight attention/augmentation tailored to fine-grained lesion cues are also planned. Overall, the results demonstrate that a balanced exploration–exploitation schedule can reliably lift competitive off-the-shelf models to performance with minimal inference penalty, advancing practical, data-efficient plant disease diagnostics.

REFERENCES

[1] P. Khatri, P. Kumar, K. S. Shakya, M. C. Kirlas, K. K. Tiwari, "Understanding the intertwined nature of rising multiple risks in modern agriculture and food system", Environment, Development and Sustainability, Vol. 26, No. 9, 2024, pp. 24107-24150.

[2] G. W. Norton, J. Alwang, W. A. Masters, "Economics of agricultural development: world food systems and resource use", Routledge, 2021.

[3] M. Passarelli, G. Bongiorno, P. Beraldi, R. Musmanno, L. Filice, "Supply chain management in case of producer disruption between external (instable) forces and effective models", Procedia Computer Science, Vol. 217, 2023, pp. 1305-1315.

[4] R. Das, S. Biswas, "Influence of abiotic stresses on seed production and quality", Seed Biology Updates, IntechOpen, 2022.

[5] M. Hussain, "Sustainable machine vision for industry 4.0: a comprehensive review of convolutional neural networks and hardware accelerators in computer vision", AI, Vol. 5, No. 3, 2024.

[6] I. Yousif, L. Burns, F. El Kalach, R. Harik, "Leveraging computer vision towards high-efficiency autonomous industrial facilities", Journal of Intelligent Manufacturing, Vol. 36, No. 5, 2025, pp. 2983-3008.

[7] Y. Cheng et al. "A comprehensive survey for real-world industrial defect detection: Challenges, approaches, and prospects", arXiv2507.13378, 2025.

[8] H. N. Ngugi, A. E. Ezugwu, A. A. Akinyelu, L. Abualigah, "Revolutionizing crop disease detection with computational deep learning: a comprehensive review", Environmental Monitoring and Assessment, Vol. 196, No. 3, 2024, p. 302.

[9] S. A. A. Qadri, N.-F. Huang, T. M. Wani, S. A. Bhat, "Advances and challenges in Computer Vision for Image-based plant disease detection: a Comprehensive Survey of Machine and Deep Learning approaches", IEEE Transactions on Automation Science and Engineering, Vol. 22, 2024, pp. 2639-2670.

[10] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances", Journal of Big Data, Vol. 11, No. 1, 2024, p. 5.

[11] A. Sharma, R. Parvathi, "Enhancing Cervical Cancer Classification: Through a Hybrid Deep Learning Approach Integrating DenseNet201 and InceptionV3", IEEE Access, Vol. 13, 2025, pp. 9868-9878.

[12] N. Duklan, S. Kumar, H. Maheshwari, R. Singh, S. D. Sharma, S. Swami, "CNN Architectures for Image Classification: A Comparative Study Using ResNet50V2, ResNet152V2, InceptionV3, Xception, and MobileNetV2", International Journal of Electronics and Communication Engineering, Vol. 11, No. 9, 2024, pp. 11-21.

[13] S. M. Ali, Y. M. Abbosh, A. M. Breesam, D. M. Ali, I. A. Alhummada, "Heart diseases classification through deep learning techniques: A review", AIP Conference Proceedings, Vol. 3232, No. 1, 2024.

[14] M. Zivkovic et al. "Hybrid CNN and XGBoost model tuned by modified arithmetic optimization algorithm for COVID-19 early diagnostics from X-ray images", Electronics, Vol. 11, No. 22, 2022, p. 3798.

[15] A. Bilal, A. Alkhathlan, F. A. Kateb, A. Tahir, M. Shafiq, H. Long, "A quantum-optimized approach for breast cancer detection using SqueezeNet-SVM", Scientific Reports, Vol. 15, No. 1, 2025, p. 3254.

[16] S. Sarkar, K. Mali, "Monkey king evolution (MKE)-GA-SVM model for subtype classification of breast cancer", Digital Health, Vol. 10, 2024.

[17] B. Tugrul, E. Elfatimi, R. Eryigit, "Convolutional neural networks in detection of plant leaf diseases: A review", Agriculture, Vol. 12, No. 8, 2022, p. 1192.

[18] S. Barburiceanu, S. Meza, B. Orza, R. Malutan, R. Terebes, "Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture", IEEE Access, Vol. 9, 2021, pp. 160085-160103.

[19] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, "Densely connected convolutional networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21-26 July 2017, pp. 4700-4708.

[20] S. Dhanka, A. Sharma, A. Kumar, S. Maini, H. Vundavilli, "Advancements in Hybrid Machine Learning Models for Biomedical Disease Classification Using Integration of Hyperparameter-Tuning and Feature Selection Methodologies: A Comprehensive Review", Archives of Computational Methods in Engineering, 2025, pp. 1-36.

[21] S. Vijayan, C. L. Chowdhary, "Hybrid feature optimized CNN for rice crop disease prediction", Scientific Reports, Vol. 15, No. 1, 2025, p. 7904.

[22] S. A. Priya, V. Khanaa, "An Efficient Optimization Technique for Classification of Multi-Crop Leaf Diseases Using Hybrid Deep Learning Model", Advancing Intelligent Networks Through Distributed Optimization, IGI Global, 2024, pp. 81-104.

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

[24] R. Zhang, P. Ji, J. Wang, B. Zhang, "Creative Design and Visual Effect Simulation of IoT Dynamic Interfaces by Fractal Algorithms and Artificial Intelligence", IEEE Transactions on Consumer Electronics, Vol. 71, No. 4, 2025, pp. 11572-11587.

[25] A. Mutlu, S. Dogan, T. Tuncer, "Novel performance-based hyperparameter optimization with the use of Bounding Box Tuner (BBT)", Information Dynamics and Applications, Vol. 4, No. 2, 2025, pp. 95-114.

[26] M. Saki, R. Keshavarz, D. Franklin, M. Abolhasan, J. Lipman, N. Shariati, "A Data-Driven Review of Remote Sensing-Based Data Fusion in Precision Agriculture from Foundational to Transformer-Based Techniques", IEEE Access, Vol. 13, 2025, pp. 166188- 166209.

[27] G. Shandilya, S. Gupta, H. G. Mohamed, S. Bharany, A. U. Rehman, S. Hussen, "Enhanced Maize Leaf Disease Detection and Classification Using an Integrated CNN-ViT Model", Food Sci. Nutr., Vol. 13, No. 7, 2025, p. e70513.

[28] I. Kunduracioglu, I. Pacal, "Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases", Journal of Plant Diseases and Protection, Vol. 131, No. 3, 2024, pp. 1061-1080.

[29] I. Pacal, G. Işık, "Utilizing convolutional neural networks and vision transformers for precise corn leaf disease identification", Neural Computing and Applications, Vol. 37, No. 4, 2025, pp. 2479-2496.

[30] S. Ince, I. Kunduracioglu, A. Algarni, B. Bayram, I. Pacal, "Deep learning for cerebral vascular occlusion segmentation: a novel ConvNeXtV2 and GRN-integrated U-Net framework for diffusion-weighted imaging", Neuroscience, Vol. 574, 2025, pp. 42-53.

[31] I. Pacal, O. Akhan, R. T. Deveci, M. Deveci, "NeXt-Brain: Combining local and global feature learning for brain tumor classification", Brain Research, 2025, p. 149762.

[32] E. Camizuli, E. J. Carranza, "Exploratory data analysis (EDA)", The Encyclopedia of Archaeological Sciences, 2018, pp. 1-7.

[33] Kaggle, "Plant Disease Recognition", https://www.kaggle.com/datasets/rashikrahmanpritom/plant-disease-recognition-dataset (accessed: 2025)

[34] S. Bhattarai, "New plant diseases dataset", https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset (accessed: 2018).

[35] R. Rajeswari, "AI-Driven Paddy Leaf Disease Classification and Prediction using DenseNet-121", Proceedings of the 6th International Conference on Deep Learning, Artificial Intelligence and Robotics, Vol. 193, 2025, p. 272.

[36] X. Huang, Z. Duan, S. Hao, J. Hou, W. Chen, L. Cai, "A deep learning framework for corrosion assessment of steel structures using Inception v3 model", Buildings, Vol. 15, No. 4, 2025, p. 512.

[37] F. Chollet, "Xception: Deep learning with depthwise separable convolutions", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21-26 July 2017, pp. 1251-1258.

[38] A. G. Howard, "MobileNets: Efficient convolutional neural networks for mobile vision applications", arXiv:1704.04861, 2017.

[39] W. He, T. Zhou, Y. Xiang, Y. Lin, J. Hu, R. Bao, "Deep learning in image classification: Evaluating VGG19's performance on complex visual data", Proceedings of the 5th International Conference on Neural Networks, Information and Communication Engineering, Guangzhou, China, 10-12 January 2025, pp. 261-265.

[40] S. M. Ismail, Y. F. Hassan, S. K. Guirguis, "Enhancing the Water Optimization Algorithm Using the Dragonfly Approach", Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications, Vol. 2, 2025, pp. 109-118.

[41] Y. Meraihi, A. Ramdane-Cherif, D. Acheli, M. Mahseur, "Dragonfly algorithm: a comprehensive review and applications", Neural Computing and Applications, Vol. 32, No. 21, 2020, pp. 16625-16646.

[42] X.-S. Yang, "Firefly algorithms for multimodal optimization", Proceedings of the International Symposium on Stochastic Algorithms, Sapporo, Japan, 26-28 October 2009, pp. 169-178.

[43] T.-L. Le, "Firefly Algorithm-based Optimization of Control Parameters in DC Conversion Systems", Engineering, Technology and Applied Science Research, Vol. 15, No. 2, 2025, pp. 20588-20594.

[44] A. Jaradat, S. Hamad, "Community structure detection using firefly algorithm", International Jour-

nal of Applied Metaheuristic Computing, Vol. 9, 2018, pp. 52-70.

[45] J. Ahammed, A. Swathi, D. Sanku, C. Vedula, H. Ramesh, "Performance of firefly algorithm for null positioning in linear arrays", Proceedings of 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications, 2018, pp. 383-391.

[46] Y. Yang et al. "A highly accurate and robust prediction framework for drilling rate of penetration based on machine learning ensemble algorithm", Geoenergy Science and Engineering, Vol. 244, 2025, p. 213423.

[47] R. Khatami, M. Faghihi, H. Khatami, M. Yousefifard, S. Khatami, "Calculation of Sensitivity and Specificity from Partial Data for Meta-Analyses: Introducing Some Practical Methods", Archives of Academic Emergency Medicine, Vol. 13, No. 1, 2025, pp. e56-e56.

[48] E. Najjar, N. A. Alkhaykanee, A. M. Breesam, "Classification of Arabic documents with five classifier models using machine learning", JOIV International Journal of Informatics and Visualization, Vol. 9, No. 1, 2025, pp. 365-370.

[49] F. Diaz, M. D. Ekstrand, B. Mitra, "Recall, robustness, and lexicographic evaluation", ACM Transactions on Recommender Systems, Vol. 4, No. 1, 2025, pp. 1-50.

[50] M. Açıkkar, S. Tokgöz, "Improving multi-class classification: scaled extensions of harmonic mean-based adaptive k-nearest neighbors", Applied Intelligence, Vol. 55, No. 3, 2025, p. 168.