Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 8 (2026) 3470

Journal of the Nigerian Society of Physical Sciences

HybOptic-CNN: A hybrid WOA-GWO-optimized convolutional neural network model for enhanced plant disease detection in the Nigerian environment

Lois Onyejere Nwobodo[a], Chinonso J. Okonkwo[b], Edith Angela Ugwu[c], Udeh Chukwuma Callistus[c], Okorie Kingsley Maduabuchi[c], Ngene John Ndubisi[c], Agbo Kenechukwu Martin[c,*]

[a]Department of Computer Engineering, Enugu State University of Science and Technology
[b]Department of Computer Science, Chukwuemeka Odumegwu Ojukwu University, Anambra State, Nigeria
[c]Department of Computer Science, Enugu State University of Science and Technology

Abstract

Plant diseases threaten agricultural productivity, and automated image analysis can support early identification of visible disease symptoms. This study introduces HybOptic-CNN, a convolutional neural network (CNN) whose learning rate and batch size are selected using a hybrid Whale Optimization Algorithm–Grey Wolf Optimizer (WOA-GWO). Nine disease classes were selected from the 22-class CCMT field-image dataset, and 152 local farm leaf images were collected in Enugu State, Nigeria. Of the local images, 122 (80.3%) were added to the model-development data for training and validation, whereas 30 (19.7%) formed an independent Nigerian hold-out set excluded from augmentation, class balancing, early stopping, validation, and hyperparameter selection. Across 10 model-development runs, the optimized model achieved 96.8 ± 0.4% mean validation accuracy, 95.2 ± 0.5% macro-precision, 94.9 ± 0.6% macro-recall, and 95.0 ± 0.5% macro-F1, compared with 90.3 ± 0.9% validation accuracy and 86.3 ± 1.2% macro-F1 for the baseline. The optimized model improved mean validation accuracy by 6.5 percentage points and converged 14.6 epochs earlier. On the independent 30-image Nigerian hold-out, HybOptic-CNN achieved 93.3% accuracy and 93.1% macro-F1 across four represented disease classes. A web application integrating the trained classifier was also demonstrated. These results support improved model-development performance through hybrid hyperparameter selection and motivate broader multi-location field evaluation.

DOI: 10.46481/jnsps.2026.3470

Keywords: Plant disease detection, Convolutional neural network, Hybrid optimization, Whale Optimization Algorithm, Grey Wolf Optimizer

Article History:
Received: 28 April 2026
Received in revised form: 19 June 2026
Accepted for publication: 20 August 2026
Available online: 11 September 2026

Communicated by: B. J. Falaye

1. Introduction

Agriculture is one of the core foundations of Nigeria's economy, as it provides employment and contributes substantially to food security and rural livelihoods. Cassava, maize, rice, yam, and tomato are some of the major crops prone to different kinds of plant diseases associated with fungi, bacteria, viruses, and pests [1]. Nigeria's tropical climate can exacerbate these diseases; high humidity and temperature variations create favourable conditions for pathogen proliferation. Early and accurate plant disease detection is therefore critical for preventing large crop-yield losses and sustaining agricultural productivity

*Corresponding Author Tel. No.: +234-706-602-2745.
Email address: kennygeemartin@gmail.com (Agbo Kenechukwu Martin)

[2].

Manual visual observation by farmers or agricultural extension officers remains a conventional method of plant disease detection in Nigeria. Although useful in some instances, subjectivity, slow diagnosis, and the limited availability of experts in rural communities restrict its application, particularly across vast farmlands [3]. Moreover, disease identification can be difficult because of variations in lighting conditions, leaf morphology, and disease presentation [4]. These constraints highlight the need for automated and intelligent systems capable of providing timely and dependable disease detection across varied agricultural settings [5].

Recent developments in deep learning, particularly convolutional neural networks (CNNs), have been highly successful in image classification, including plant disease detection [6]. CNNs can automatically learn hierarchical features from leaf images, including edges, textures, and lesion patterns, and can classify healthy and diseased crops [7]. Nonetheless, CNN models depend strongly on effective hyperparameter settings, such as learning rate, number of layers, kernel size, and dropout rate [8]. Traditional optimization methods, such as grid search or manual tuning, are computationally costly and may fail to identify optimal configurations within a complex search space [9].

To address these limitations, this study combines a CNN with a hybrid Whale Optimization Algorithm and Grey Wolf Optimizer (WOA-GWO) [10–12]. WOA contributes global exploration of the hyperparameter search space, while GWO strengthens exploitation around promising candidate solutions. The hybrid procedure is used to improve the model-development configuration with respect to validation loss, classification performance, and convergence behaviour. The inclusion of locally collected Nigerian field images further enables assessment under environmental variations that differ from the principal CCMT data source.

This work makes five main contributions. It develops and evaluates the HybOptic-CNN framework for nine-class plant disease classification using field-acquired CCMT images supplemented by locally collected Nigerian images; applies a hybrid WOA-GWO procedure to select key training hyperparameters using validation loss as the optimization objective, thereby reducing reliance on manual trial-and-error tuning; incorporates locally collected Nigerian field images into model development while retaining an independent local hold-out subset to assess transfer under variations in illumination, background, occlusion, and disease presentation; demonstrates functional software integration through a web-based plant disease classification application; and provides a repeated-run comparison between the baseline CNN and the WOA-GWO-optimized configuration, complemented by an independent Nigerian hold-out assessment.

2. Related works

Recent research has shown substantial progress in deep-learning-based systems for plant disease detection. Reddy *et al.* [13] designed a portable mobile tool to identify plant disease and its severity through a custom PyTorch CNN and classical image processing. Trained on the PlantVillage dataset, the model achieved 92.06% test accuracy, with per-class precision, recall, and F1-scores exceeding 85% across 39 categories. In addition to classification, the paper evaluated disease severity using an OpenCV and NumPy pipeline to measure the ratio of diseased area to total leaf area and provide interpretable percentage results. The system was implemented as a React Native application with a Flask backend, providing real-time predictions and Grad-CAM images in approximately 1–2 s. Similarly, Kennedy and Chelladurai [14] implemented a MobileNet-based web diagnostic system achieving approximately 95% accuracy and integrating confidence thresholds and interactive feedback to enhance reliability. Jothilakshmi and Sharanesh [6] further extended disease detection to autonomous rover-based monitoring, where InceptionResNetV2 achieved 95.24% accuracy, demonstrating the applicability of CNN models in large-scale agricultural settings.

Several studies have focused on efficient and accurate model architectures. Ashurov *et al.* [2] enhanced MobileNetV2 using depth-wise separable convolutions, Squeeze-and-Excitation (SE) blocks, and residual connections, achieving 98% accuracy and an F1-score of 98.2% with a lightweight 8.5 MB model suitable for real-time deployment. Karuna *et al.* [15] compared VGG16, VGG19, GoogleNet, and DenseNet201, with DenseNet201 achieving 99.17% validation accuracy and 98.82% test accuracy. Chandrasekaran *et al.* [16] applied transfer learning with ResNet on the New Plant Diseases Dataset containing more than 87,000 images, achieving 99.3% accuracy and outperforming CNN, MobileNet, and EfficientNet architectures. Sujatha *et al.* [17] combined CNN feature extractors with SVM and kNN classifiers, achieving up to 99.1% accuracy on certain datasets, although performance varied with crop complexity. These studies indicate that deeper architectural developments and hybrid CNN–machine-learning models can improve classification robustness.

Extensive reviews also indicate broader trends and issues in the field. Wang *et al.* [18] pointed out that deep-learning approaches, including classification, detection, segmentation, and transfer learning, are more effective than conventional diagnostic methods, although generalisation and computational cost remain challenges. In a review of 278 papers, Upadhyay *et al.* [19] indicated that multimodal imaging methods, such as RGB, multispectral, and hyperspectral sensing, may be important for future applications because deep-learning models consistently outperform conventional machine-learning models. Having analysed 160 studies, Pacal *et al.* [20] found that classification is well developed, but further research on detection, severity estimation, and segmentation is needed. Shoaib *et al.* [21] also reported that classification accuracies frequently exceed 95%, yet challenges such as environmental variability, limited datasets, and computational constraints persist.

Architectures beyond traditional CNNs have also been proposed. Borhani *et al.* [22] explored the use of a Vision Transformer (ViT) and reported an F1-score of 0.9877 on the PlantVillage dataset using fewer parameters than CNN-based

models, although inference was slower. Rodríguez-Lira *et al.* [23] observed that residual architectures remain prominent, while MobileNet-type networks are attractive for edge-oriented applications. Isinkaye *et al.* [24] stressed the importance of combining deep learning and content-based filtering systems to offer treatment suggestions in addition to disease detection. Islam *et al.* [25] introduced PlantCareNet, achieving up to 97% accuracy across multiple datasets and demonstrating adaptability to region-specific agricultural conditions, although visually similar diseases remained challenging. Important gaps therefore remain in hyperparameter selection, field-image transfer, and locally contextualized evaluation. These gaps motivate the present HybOptic-CNN study and its combination of repeated validation analysis with an independent Nigerian field-image hold-out.

3. Research methodology

This study adopted a quantitative experimental design to develop and evaluate the proposed HybOptic-CNN for a nine-class plant disease classification problem. The model-development data comprised the selected CCMT field images supplemented by 122 of the 152 locally collected Nigerian images. The Nigerian images were divided 122:30 (80.3%:19.7%) into model-development and independent hold-out subsets, respectively. The 30-image Nigerian hold-out was excluded from all model-development activities, including fitting, validation, augmentation, class balancing, early stopping, and WOA-GWO optimization, until final evaluation of the selected model. Partitioning was performed before study-specific data augmentation to prevent variants of the same original image from appearing in different partitions. The CNN was trained using the Adam optimizer and cross-entropy loss, while WOA-GWO tuned the learning rate and batch size according to the lowest validation loss. Model-development performance was evaluated using accuracy, macro-precision, macro-recall, macro-F1, convergence, and class-wise metrics across repeated runs. The selected model was then evaluated separately on the Nigerian hold-out before software deployment.

3.1. Data collection

In this study, the Crop Pest and Disease Detection (CCMT) dataset was used as the principal image source. The published dataset contains 24,881 raw images and 102,976 augmented images of cashew, cassava, maize, and tomato collected under varied field conditions in Ghana, with 22 original classes [26]. The original class distribution and study inclusion are shown in Table 1.

Of the 22 original CCMT categories, nine disease categories comprising 11,712 original images were selected for the classification experiment. The 13 excluded categories comprised healthy classes, pest classes, and disease categories outside the defined nine-class classification scope.

The original classes are Cashew (Anthracnose, Gummosis, Healthy, Leaf Miner, and Red Rust); Cassava (Bacterial Blight, Brown Spot, Green Mite, Healthy, and Mosaic); Maize (Fall Armyworm, Grasshopper, Healthy, Leaf Beetle, Leaf Blight, Leaf Spot, and Streak Virus); and Tomato (Healthy, Leaf Blight, Leaf Curl, Septoria Leaf Spot, and Verticillium Wilt). Nine disease classes were retained for the present classification task: Cashew Anthracnose, Cashew Gummosis, Cashew Red Rust, Cassava Bacterial Blight, Cassava Mosaic, Maize Leaf Blight, Maize Streak Virus, Tomato Leaf Blight, and Tomato Verticillium Wilt. The remaining 13 classes, comprising healthy categories, pest categories, and disease categories outside the defined nine-class scope, were excluded. The CCMT source reports raw crop totals of 6,549 cashew, 7,508 cassava, 5,389 maize, and 5,435 tomato images.

To increase local contextual relevance, 152 additional leaf images were collected from farms in Ngwo, Udi LGA, Enugu State, Nigeria, covering Cassava Mosaic, Cassava Bacterial Blight, Maize Leaf Blight, and Tomato Leaf Blight. The Nigerian collection was divided at the original-image level using the approximately 80:20 allocation reported in the study: 122 images (80.3%) formed the local model-development subset and were incorporated into the training/validation workflow, while 30 images (19.7%) were reserved as an independent Nigerian hold-out set. The hold-out images were excluded from model fitting, validation-based early stopping, WOA-GWO hyperparameter selection, oversampling, and augmentation. The local-image distribution is summarized in Table 2.

The Nigerian field-image collection was partitioned at the original-image level before augmentation. The 122-image development subset was incorporated into the training/validation workflow, while the 30-image hold-out was kept completely separate until completion of model selection. All images were resized so that the shorter side equalled 224 pixels while preserving aspect ratio, centre-cropped to 224 × 224 pixels, and normalized to [0, 1]. Figure 1 presents examples of the Nigerian disease images used in the study.

3.2. Data preprocessing

All CCMT and Nigerian images were processed using a consistent pipeline. Partition assignment was performed at the original-image level before any study-specific augmentation so that augmented variants of a source image remained within the same partition. For the Nigerian collection, 122 images belonged to the model-development pool and followed the same training/validation procedure as the CCMT development data, while the 30-image Nigerian hold-out remained completely separate.

Random rotations, horizontal and vertical flips, zoom, brightness adjustment, and shearing were applied only to training images. Validation images and Nigerian hold-out images received only deterministic resizing, centre cropping, and normalization. Pixel values were normalized to [0, 1]. Class imbalance was addressed within the training data using oversampling and class-weighted loss; neither procedure was applied to validation or hold-out images. The resulting data were organized into the nine retained disease classes and loaded into PyTorch DataLoaders for model development, repeated validation analysis, and final hold-out evaluation. Accordingly, augmentation, oversampling, class-weight adjustment, model

Table 1: Original CCMT class distribution and study inclusion.

Crop	Original CCMT class	Raw images	Study status
Cashew	Anthracnose	1,729	Retained
Cashew	Gummosis	392	Retained
Cashew	Healthy	1,368	Excluded
Cashew	Leaf Miner	1,378	Excluded
Cashew	Red Rust	1,682	Retained
Cassava	Bacterial Blight	2,614	Retained
Cassava	Brown Spot	1,481	Excluded
Cassava	Green Mite	1,015	Excluded
Cassava	Healthy	1,193	Excluded
Cassava	Mosaic	1,205	Retained
Maize	Fall Armyworm	285	Excluded
Maize	Grasshopper	673	Excluded
Maize	Healthy	208	Excluded
Maize	Leaf Beetle	948	Excluded
Maize	Leaf Blight	1,006	Retained
Maize	Leaf Spot	1,259	Excluded
Maize	Streak Virus	1,010	Retained
Tomato	Healthy	500	Excluded
Tomato	Leaf Blight	1,301	Retained
Tomato	Leaf Curl	518	Excluded
Tomato	Septoria Leaf Spot	2,343	Excluded
Tomato	Verticillium Wilt	773	Retained
Total	22 classes	24,881	9 retained; 13 excluded

Table 2: Distribution of locally collected Nigerian images.

Dataset component	Number of images	Percentage	Experimental role
Model-development subset	122	80.3%	Training/validation workflow
Independent Nigerian hold-out	30	19.7%	Final post-selection field evaluation
Total	152	100.0%	—

(a) Cassava Mosaic (b) Maize Leaf Blight (c) Tomato Leaf Blight

Figure 1: Samples from the collected dataset.

fitting, early stopping, and hyperparameter selection were restricted to the model-development data and were not applied to the 30 Nigerian hold-out images.

3.3. The CNN architecture

The CNN was designed to learn discriminative visual features from RGB leaf images resized to $224 \times 224 \times 3$. The feature extractor comprises four convolutional blocks with 32, 64,

128, and 256 filters, respectively, using 3×3 kernels. Each convolution is followed by ReLU activation, batch normalization, 2×2 max pooling, and dropout. The fixed convolutional-block dropout schedule is 0.30, 0.30, 0.40, and 0.50 for Blocks 1–4, respectively. These architectural settings were held constant in the baseline and optimized models so that the comparison isolates the effect of the reported WOA-GWO-selected training hyperparameters.

After the final convolutional block, the feature maps are flattened and passed through fully connected layers of 128 and 64 neurons. Both dense layers use ReLU activation and a fixed dropout rate of 0.40. The output layer contains nine neurons corresponding to the nine retained disease classes. During training, PyTorch CrossEntropyLoss is computed directly from the output logits; softmax is applied only during inference to obtain class probabilities. In the revised specification, filter counts, kernel size, dropout schedule, and the maximum epoch cap are fixed architectural/training settings. The WOA-GWO search is therefore described only for the training hyperparameters that differ between the baseline and optimized configurations, principally learning rate and batch size.

Filter counts, kernel size, dropout schedule, fully connected dimensions, and the maximum epoch cap were fixed for both configurations. WOA-GWO selected only the learning rate and batch size.

3.4. Hybrid WOA-GWO hyperparameter optimization

The Hybrid Whale Optimization Algorithm and Grey Wolf Optimizer (WOA-GWO) was used to select training hyperparameters for the fixed CNN architecture. In the revised specification, the search concerns the learning rate and batch size, while the filter configuration, 3×3 kernel size, layer-specific dropout schedule, and maximum epoch cap are treated as fixed design choices. Each candidate solution therefore represents a candidate learning-rate/batch-size configuration. Validation cross-entropy loss is used consistently as the fitness function, with lower validation loss indicating a better candidate. The WOA component promotes global exploration of the search space, whereas the GWO component promotes exploitation around high-performing candidate solutions [10–12].

$$X = [\eta, B]. \tag{1}$$

In Eq. (1), η denotes the learning rate and B denotes the batch size. Each candidate solution in the population therefore represents one learning-rate/batch-size configuration. The objective (fitness) function is presented in Eq. (2).

$$f(X) = \text{Validation loss}. \tag{2}$$

Lower $f(X)$ indicates better performance.

The GWO simulates grey wolves' hunting strategy and social hierarchy: alpha (α), beta (β), delta (δ), and omega (ω). Equation (3) presents the position-update model as

$$\begin{aligned} D_\alpha &= |C_1 \cdot X_\alpha - X_i|, \\ D_\beta &= \left|C_2 \cdot X_\beta - X_i\right|, \\ D_\delta &= |C_3 \cdot X_\delta - X_i|. \end{aligned} \tag{3}$$

Then, the new position of agent X_i is obtained by averaging the influences of the three leaders using Eq. (4):

$$\begin{aligned} X_i(t+1) = \frac{1}{3}(X_\alpha - A_1 \cdot D_\alpha + X_\beta - A_2 \cdot D_\beta \\ + X_\delta - A_3 \cdot D_\delta). \end{aligned} \tag{4}$$

The coefficients A and C are defined in Eq. (5):

$$A = 2a \cdot r_1 - a, \qquad C = 2 \cdot r_2. \tag{5}$$

Here, X_α, X_β, and X_δ are the position vectors of the best three solutions (the α, β, and δ wolves) at the current iteration; X_i is the position vector of the ith candidate solution (the ω wolf); A_1, A_2, and A_3 are coefficient vectors calculated using Eq. (5) for each leader; and C_1, C_2, and C_3 are random coefficient vectors calculated using Eq. (5) for each leader. The control parameter a decreases linearly from 2 to 0 over the iterations and is computed as $a = 2 - 2t/t_{\max}$, where t is the current iteration and $t_{\max}$ is the maximum number of iterations. The vectors r_1 and r_2 are drawn from a uniform distribution on $[0, 1]$, that is, $r_1, r_2 \sim U(0, 1)$.

The purpose of the GWO component is to exploit the best-known solutions found so far while guiding the population toward promising regions of the search space. As the parameter a decreases, the algorithm transitions from global exploration to local exploitation.

WOA simulates humpback whales' bubble-net feeding using encircling-prey and spiral-updating models, as shown in Eqs. (6) and (7), respectively.

$$D = |X^* - X_i|, \qquad X_i(t+1) = X^* - A \cdot D. \tag{6}$$

$$X_i(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*. \tag{7}$$

Here, X^* is the best solution found so far, b is the logarithmic spiral constant, $l \sim U(-1, 1)$, and A decreases linearly from 2 to 0. This component encourages global exploration and helps avoid premature convergence.

The hybrid WOA-GWO optimization procedure is summarized in Algorithm 1.

The WOA-GWO optimization used a population of 10 candidate solutions and 15 iterative update cycles. The initial population required 10 fitness evaluations, followed by 150 evaluations across the 15 optimization iterations. Thus, the complete search required 160 candidate-model evaluations, calculated as $10 + (10 \times 15)$. Each candidate evaluation used a fixed 10-epoch training budget, and validation cross-entropy loss was used consistently as the fitness criterion.

3.5. The HybOptic-CNN model

The HybOptic-CNN is the fixed CNN architecture trained using the WOA-GWO-selected training configuration. The optimization objective is validation loss; therefore, the optimization should be interpreted as targeting predictive model-development performance rather than computational efficiency. The architecture contains four convolutional feature-extraction

Algorithm 1: Hybrid WOA-GWO hyperparameter optimization pseudocode.

Component	Specification
Input	CNN model architecture
Search variables	Learning rate $\eta \in [0.0001, 0.0010]$; batch size $B \in \{16, 32, 64, 128\}$
Population size, N	10
Maximum iterations, MaxIter	15
WOA/GWO switching probability, p	0.50
Fitness-evaluation training budget	10 epochs per candidate solution
Random-seed procedure	Seed 42 was used for WOA-GWO population initialization and candidate fitness evaluation. The subsequent repeated baseline and optimized training runs were independently initialized and were not deliberately paired.
Fitness function	Validation cross-entropy loss; lower values indicate better fitness
Output	Optimized hyperparameter vector $X^* = [\eta^*, B^*]$
Procedure	1. Initialize population $\{X_i \mid i = 1, \ldots, N\}$, where $N = 10$, within the learning-rate range $[0.0001, 0.0010]$ and batch-size set $\{16, 32, 64, 128\}$. 2. Evaluate fitness $f(X_i)$ for each candidate by training the CNN for 10 epochs and computing validation cross-entropy loss. 3. Identify X^* as the candidate with the lowest validation loss. 4. For $t = 1, \ldots, 15$, update each candidate. Draw r uniformly from $[0, 1]$. If $r < 0.50$, update X_i using the WOA encircling/spiral rule; otherwise, update X_i using the GWO alpha, beta, and delta leaders. Project the learning rate to $[0.0001, 0.0010]$ and map the batch size to the nearest valid value in $\{16, 32, 64, 128\}$. Re-evaluate the candidate by training for 10 epochs, compute the validation cross-entropy loss, and update X^* if the candidate improves the fitness. Decrease a linearly from 2 to 0 over the 15 iterations.
Return	$X^* = [0.0007, 64]$ as the selected training-hyperparameter configuration.

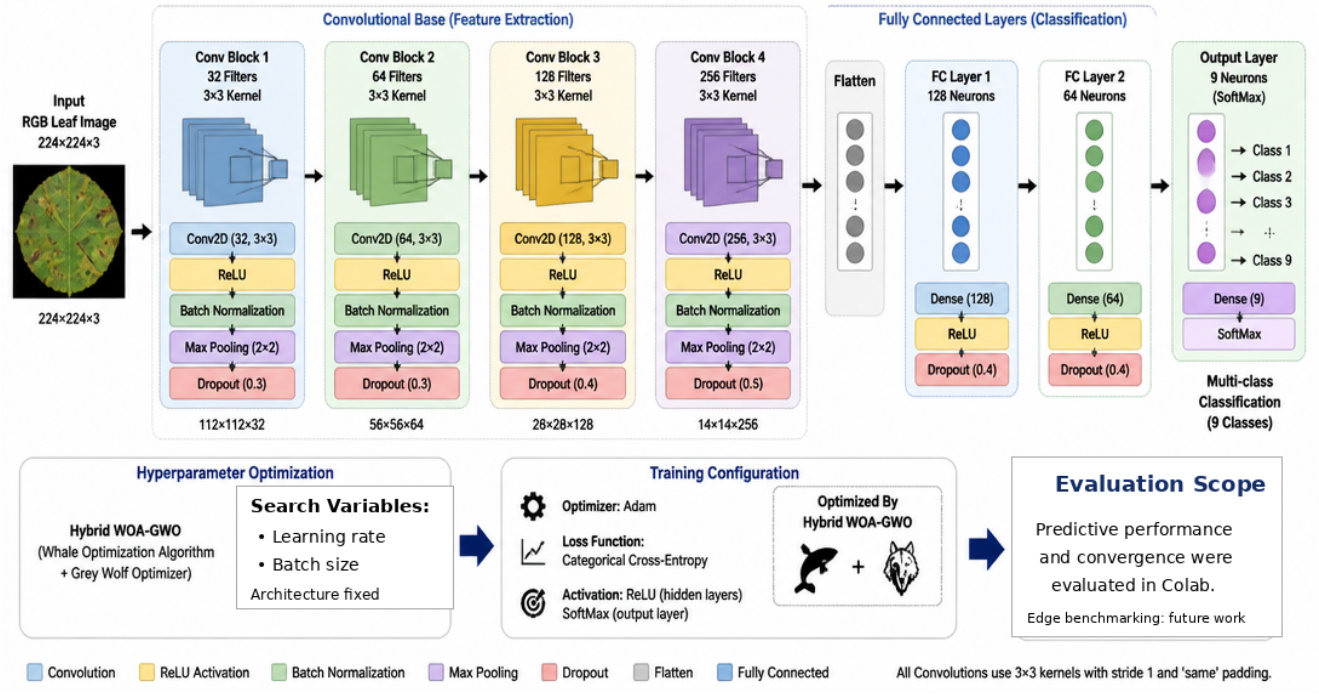

Figure 2: Architecture of the HybOptic-CNN model.

blocks, max-pooling operations, two fully connected layers, and a nine-class output layer, as summarized in Figure 2.

The HybOptic-CNN in Figure 2 accepts RGB leaf images of 224 × 224 × 3 pixels and uses four convolutional blocks with 32, 64, 128, and 256 filters and 3 × 3 kernels. The fixed dropout values are 0.30, 0.30, 0.40, and 0.50 across convolutional Blocks 1–4, followed by fully connected layers of 128 and 64 neurons with dropout 0.40 after each dense layer. The final layer outputs nine logits, with softmax used at inference. In the reported comparison, the architecture is held constant while WOA-GWO selects the learning rate and batch size.

This configuration combines CNN-based feature learning with metaheuristic selection of training hyperparameters. Its performance is evaluated against the manually configured baseline in Section 4 using repeated validation runs, followed by a supplementary assessment on the independent 30-image Nigerian hold-out. The web implementation demonstrates software-level applicability, while broader multi-location field testing and hardware-specific efficiency benchmarking are left for subsequent deployment studies.

3.6. System implementation

The HybOptic-CNN was implemented in Python using PyTorch and trained in Google Colab with GPU acceleration. Image preprocessing included resizing, centre cropping, normalization, and training-only augmentation. Adam optimization and cross-entropy loss were used, with early stopping based on validation loss. Accuracy, macro-precision, macro-recall, macro-F1, and per-class metrics were used for the repeated validation comparison of the baseline and optimized configurations.

The 30 Nigerian hold-out images were not used during model fitting, validation, early stopping, augmentation, or WOA-GWO selection. After the optimized configuration had been selected, the final model was fixed and evaluated separately on the Nigerian hold-out using only resizing, centre cropping, and normalization. The trained classifier was subsequently integrated into a web-based application to demonstrate functional software integration. Hardware-specific quantities such as optimized-model training time, WOA-GWO search cost, parameter count, model size, FLOPs, inference latency, memory consumption, and smartphone/edge-device performance were outside the present evaluation.

4. Experimental results and discussion

The proposed HybOptic-CNN was compared with a non-optimized baseline CNN on the same nine-class classification task. The two configurations use the same underlying architecture and differ in the reported training hyperparameters. The analysis distinguishes single-run descriptive results from the 10-run mean ± standard deviation validation results. Validation loss was used for WOA-GWO selection, so Tables 4–7 quantify model-development performance. A separate 30-image Nigerian hold-out, excluded from all model-development procedures, was retained for supplementary local-field assessment after the final configuration had been fixed.

4.1. Training protocols

For the baseline non-optimized CNN protocol, the baseline model used the fixed CNN architecture described in Section 3.3 and the same model-development dataset and data partitions used for the optimized configuration. It was trained using Adam with learning rate = 0.001, batch size = 32, convolutional dropout values of 0.30, 0.30, 0.40, and 0.50, fully connected dropout values of 0.40 and 0.40, and a maximum epoch cap of 75. Early stopping was based on validation loss. No WOA-GWO hyperparameter selection was applied.

For the optimized HybOptic-CNN protocol, the optimized model retained the identical architecture, development dataset, preprocessing procedure, and data partitions but used the WOA-GWO-selected learning rate of 0.0007 and batch size of 64. Validation cross-entropy loss was the optimization objective. Filters, kernel size, layer-specific dropout values, fully connected dimensions, and the 75-epoch maximum were fixed and were not optimized. The baseline and optimized settings are summarized in Table 3.

4.2. Performance of the non-optimized CNN

The non-optimized CNN was trained using the same model-development dataset as the optimized HybOptic-CNN, comprising the nine retained CCMT disease classes supplemented with the 122 Nigerian images assigned to the model-development pool. Identical training/validation partitions and preprocessing procedures were maintained for both configurations so that the comparison reflected differences in training hyperparameters rather than differences in data composition. Figure 3 presents the training and validation behaviour of the baseline model.

Figure 3 shows that the non-optimized CNN learned the nine-class task, with the single-run validation accuracy reaching 90.5% by epoch 75. Training and validation losses declined during optimization, although the widening separation between training and validation curves after approximately epoch 20 suggests mild overfitting. Table 4 summarizes this single-run baseline validation performance. These values provide the model-development reference against which the WOA-GWO-optimized configuration is compared; the independent Nigerian hold-out is treated separately in Section 4.4.

Table 4 results indicate that the non-optimized CNN achieved moderate predictive performance, with macro-averaged precision, recall, and F1-score computed directly from the nine per-class values reported in Table 5. The macro-average F1-score of 86.5% reflects the simple arithmetic mean across all nine disease classes, giving each class equal weight regardless of sample size. Lower performance was observed in visually challenging and under-represented classes such as Cashew Red Rust (F1 = 82.5%) and Maize Leaf Blight (F1 = 84.8%), indicating sensitivity to class imbalance and subtle symptom patterns. These findings establish the baseline against which the optimized model is compared.

The F1-scores obtained from the non-optimized CNN ranged from 82.5% to 89.1% in the displayed single run, reflecting uneven validation performance across the nine disease

Table 3: Selected hyperparameters for the optimized HybOptic-CNN.

Hyperparameter	Baseline	Optimized (WOA-GWO)
Learning rate	0.001	0.0007 (selected)
Batch size	32	64 (selected)
Filters (Blocks 1–4)	32, 64, 128, 256	32, 64, 128, 256 (fixed)
Kernel size	3×3	3×3 (fixed)
Conv-block dropout	0.30, 0.30, 0.40, 0.50	Same (fixed)
FC-layer dropout	0.40, 0.40	Same (fixed)
Maximum epochs	75	75 (fixed cap)

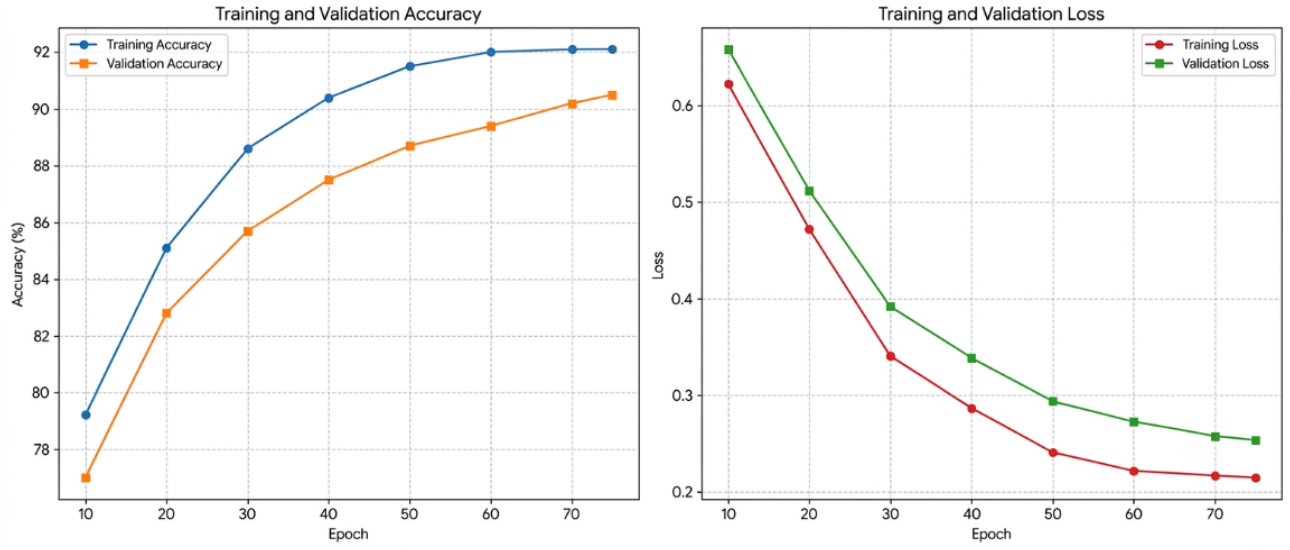

Figure 3: Training and validation accuracy and loss for the non-optimized CNN.

Table 4: Overall performance metrics for the non-optimized CNN.

Metric	Non-optimized CNN
Training accuracy (%)	92.1
Validation accuracy (%)	90.5
Macro-average precision (%)	86.9
Macro-average recall (%)	86.1
Macro-average F1-score (%)	86.5
Convergence epoch	75
Training time (GPU h)	3.2

classes. Cashew Red Rust and Maize Leaf Blight recorded the lowest F1-scores, suggesting that these classes were more difficult for the baseline configuration under the available data. These baseline validation results provide the reference point for assessing whether WOA-GWO selection improves accuracy, macro-level class balance, and convergence behaviour. The independent Nigerian field-image assessment is reported separately in Section 4.4 so that model-development validation and local hold-out evaluation remain clearly distinguished.

4.3. *Performance of the HybOptic-CNN*

The HybOptic-CNN was evaluated on the same nine-class validation task as the baseline. The optimized configuration reached higher validation performance and earlier convergence in the reported runs. Figure 4 presents the single-run learning curves, whereas the repeated-run comparison is summarized separately in Table 6.

The optimized CNN demonstrated improved validation learning dynamics, reaching near-maximum validation accuracy earlier than the baseline in the displayed single-run curves. For the repeated-run analysis, 10 baseline runs and 10 optimized runs are summarized as mean ± standard deviation. The two sets of repeated runs were analysed as independent groups using Welch's two-sample t-test ($n = 10$ per group). The optimized model achieved 96.8 ± 0.4% validation accuracy and 95.0 ± 0.5% macro-F1, compared with 90.3 ± 0.9% and 86.3 ± 1.2% for the baseline. The corresponding Welch tests indicate very large between-group differences (validation accuracy: $t = 20.87$, df = 12.42, $p = 4.79 \times 10^{-11}$; macro-F1: $t = 21.16$, df = 12.03, $p = 6.88 \times 10^{-11}$). The standardized effect sizes are interpreted together with the absolute percentage-point changes reported in Table 6.

For the repeated-run comparison, 10 baseline runs and 10 optimized runs were summarized as mean ± standard deviation.

Table 5: Single-run per-class validation metrics for the non-optimized CNN.

Crop	Disease class	Precision (%)	Recall (%)	F1-score (%)
Cashew	Anthracnose	88.5	87.9	88.2
Cashew	Gummosis	86.2	85.6	85.9
Cashew	Red Rust	83.1	81.9	82.5
Cassava	Bacterial Blight	89.5	88.7	89.1
Cassava	Mosaic	88.0	87.2	87.6
Maize	Leaf Blight	85.3	84.3	84.8
Maize	Streak Virus	86.7	86.0	86.3
Tomato	Leaf Blight	88.8	88.2	88.5
Tomato	Verticillium Wilt	85.8	85.0	85.4

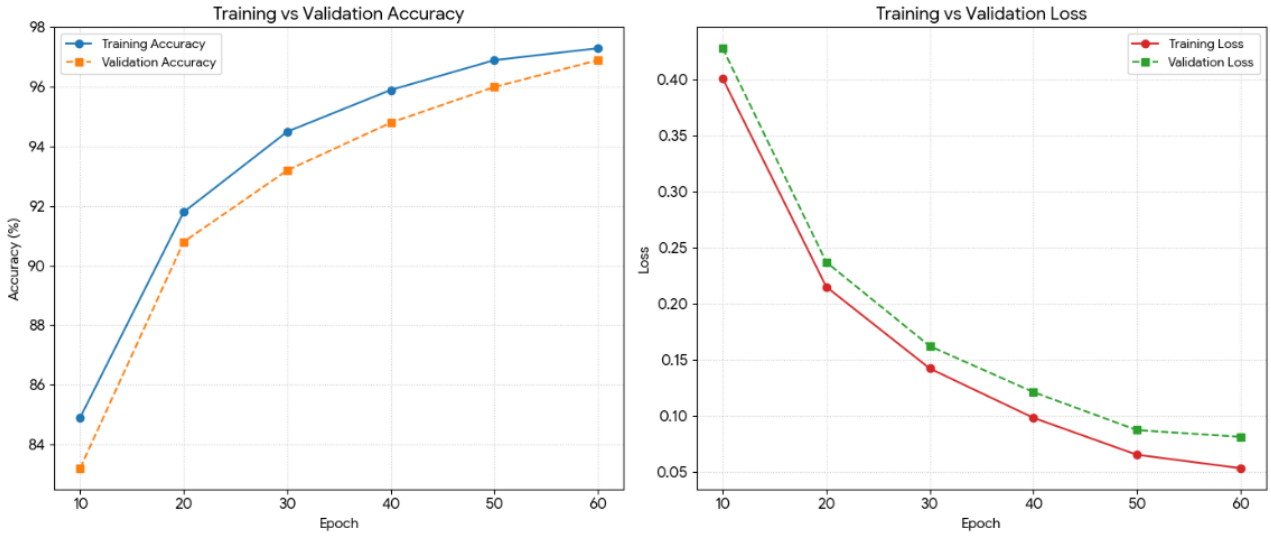

Figure 4: Training and validation accuracy and loss for the optimized CNN model.

Table 6: Repeated-run overall performance metrics.

Metric	Baseline (mean ± SD)	HybOptic-CNN (mean ± SD)	Welch t (df)	p-value	Cohen's d
Validation accuracy (%)	90.3 ± 0.9	96.8 ± 0.4	20.87 (12.42)	4.79×10^{-11}	9.33
Macro-avg precision (%)	86.7 ± 1.1	95.2 ± 0.5	22.25 (12.57)	1.80×10^{-11}	9.95
Macro-avg recall (%)	85.9 ± 1.3	94.9 ± 0.6	19.88 (12.67)	6.28×10^{-11}	8.89
Macro-avg F1-score (%)	86.3 ± 1.2	95.0 ± 0.5	21.16 (12.03)	6.88×10^{-11}	9.46
Convergence epoch	74.2 ± 1.5	59.6 ± 1.8	-19.70 (17.43)	2.34×10^{-13}	-8.81

The runs were treated as independent groups rather than deliberately matched pairs; consequently, Welch's two-sample t-test was used ($n = 10$ per group). The optimized model achieved 96.8 ± 0.4% validation accuracy and 95.0 ± 0.5% macro-F1, compared with 90.3 ± 0.9% and 86.3 ± 1.2%, respectively, for the baseline. Welch's test indicated a substantial difference in validation accuracy ($t = 20.87$, df = 12.42, $p = 4.79 \times 10^{-11}$, Cohen's $d = 9.33$) and macro-F1 ($t = 21.16$, df = 12.03, $p = 6.88 \times 10^{-11}$, Cohen's $d = 9.46$). Full statistical results are presented in Table 6.

Table 7 presents class-wise metrics from a representative optimized run, whereas Table 6 presents the primary 10-run mean ± standard deviation comparison. Accordingly, the arithmetic macro averages obtained from the class-wise values in Table 7 should not be interpreted as the repeated-run performance values reported in Table 6.

The single-run per-class results in Table 7 show higher values than the baseline table across all nine evaluated disease categories, including Cashew Red Rust, whose F1-score increased from 82.5% in the baseline single run to 92.3% in the optimized single run. All nine optimized single-run F1-scores exceeded 92%. These findings indicate improved class-wise validation performance under the experimental conditions considered. The independent Nigerian hold-out provides an additional local-field assessment for the four disease classes represented in the Nigerian collection, while broader geographical and nine-class field generalization would require a larger multi-location hold-out dataset.

4.4. Independent Nigerian hold-out assessment

The Nigerian hold-out consisted of 30 field images representing Cassava Mosaic, Cassava Bacterial Blight, Maize Leaf Blight, and Tomato Leaf Blight. These images were excluded from training, validation, augmentation, early stopping, class balancing, and WOA-GWO hyperparameter selection. After model selection, the fixed HybOptic-CNN was evaluated on the hold-out using only resizing, centre cropping, and normalization. As shown in Table 8, the model correctly classified 28 of the 30 images, achieving 93.3% accuracy, 93.3% macro-precision, 93.3% macro-recall, and 93.1% macro-F1. The result was slightly below the 96.8 ± 0.4% mean validation accuracy, indicating good transfer to locally collected Nigerian field images.

The final HybOptic-CNN was evaluated on the independent Nigerian hold-out comprising 30 field images across Cassava Bacterial Blight, Cassava Mosaic, Maize Leaf Blight, and Tomato Leaf Blight. The model correctly classified 28 of the 30 images, corresponding to an overall accuracy of 93.3%. Macro-precision and macro-recall were both 93.3%, while the macro-F1 score was 93.1%. The confusion matrix showed perfect classification of the eight Cassava Bacterial Blight images and the seven Tomato Leaf Blight images, while one Cassava Mosaic image was classified as Maize Leaf Blight and one Maize Leaf Blight image was classified as Tomato Leaf Blight. These results indicate that the optimized model maintained strong classification performance on locally acquired Nigerian field images that were excluded from model development.

4.5. Software integration testing results

To demonstrate functional software integration of the proposed HybOptic-CNN, the trained classifier was incorporated into a web-based plant disease classification application. The interface allows a user to upload a plant-leaf image, after which the backend applies the model preprocessing pipeline and returns the predicted disease class together with the model's softmax output probability. Figures 5–8 illustrate example outputs

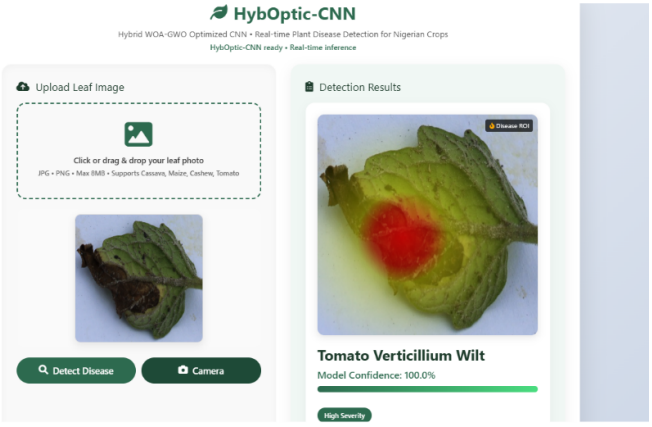

Figure 5: Tomato Verticillium Wilt classification result.

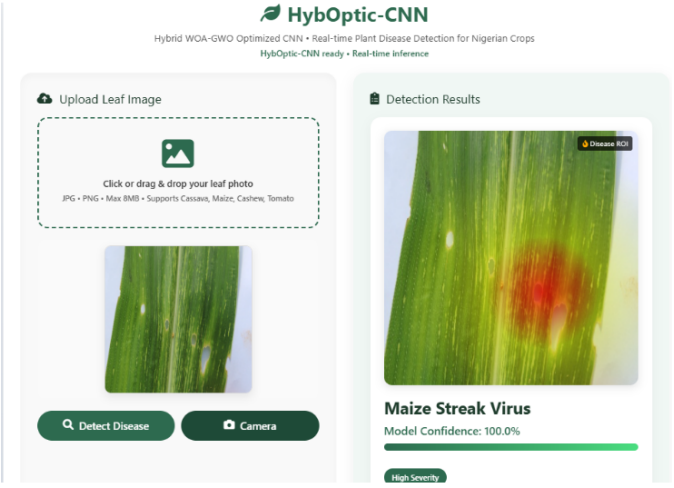

Figure 6: Maize Streak Virus classification result.

for Tomato Verticillium Wilt, Maize Streak Virus, Cassava Bacterial Blight, and Cashew Anthracnose. These demonstrations establish software integration only; they are not measurements of on-device computational efficiency or real-time edge performance.

Figures 5–8 illustrate the software workflow for four example images. The interface returned Tomato Verticillium Wilt, Maize Streak Virus, Cassava Bacterial Blight, and Cashew Anthracnose for the respective examples and displayed the associated softmax output values. These examples demonstrate that the trained classifier was successfully connected to the web interface; they are illustrative application outputs rather than an independent accuracy or latency benchmark.

For each case above, the model assigned a high softmax probability to the predicted class. These values are raw softmax outputs from the final layer and are presented as model confidence scores within the application interface; probability calibration was not separately performed. Figures 5–8 therefore demonstrate the end-to-end software workflow rather than an accuracy or hardware benchmark. Dedicated measurements of inference latency, memory use, energy consumption, FLOPs, and model size are reserved for deployment-focused evaluation.

Table 7: Single-run per-class validation metrics for the optimized HybOptic-CNN.

Crop	Disease class	Precision (%)	Recall (%)	F1-score (%)
Cashew	Anthracnose	95.8	95.5	95.6
Cashew	Gummosis	95.0	94.6	94.8
Cashew	Red Rust	92.5	92.1	92.3
Cassava	Bacterial Blight	96.7	96.3	96.5
Cassava	Mosaic	95.9	95.7	95.8
Maize	Leaf Blight	95.1	94.7	94.9
Maize	Streak Virus	95.9	95.3	95.6
Tomato	Leaf Blight	96.3	95.9	96.1
Tomato	Verticillium Wilt	94.8	94.2	94.5

Table 8: Model performance on the independent Nigerian hold-out set.

Metric	Result
Number of images	30
Represented disease classes	4
Accuracy (%)	93.3
Macro-precision (%)	93.3
Macro-recall (%)	93.3
Macro-F1 (%)	93.1

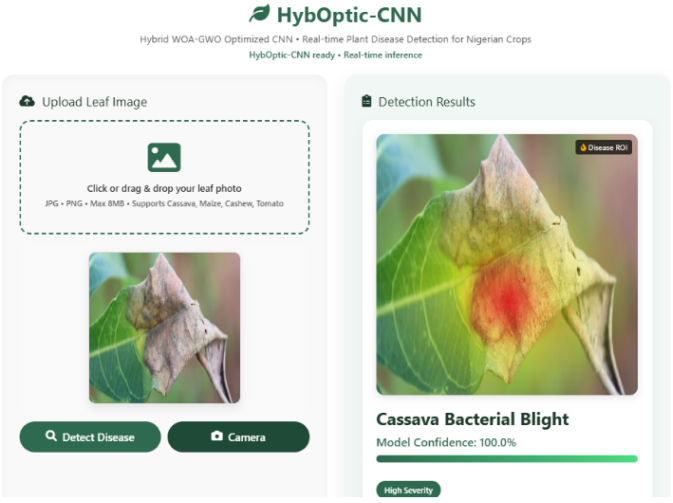

Figure 7: Cassava Bacterial Blight classification result.

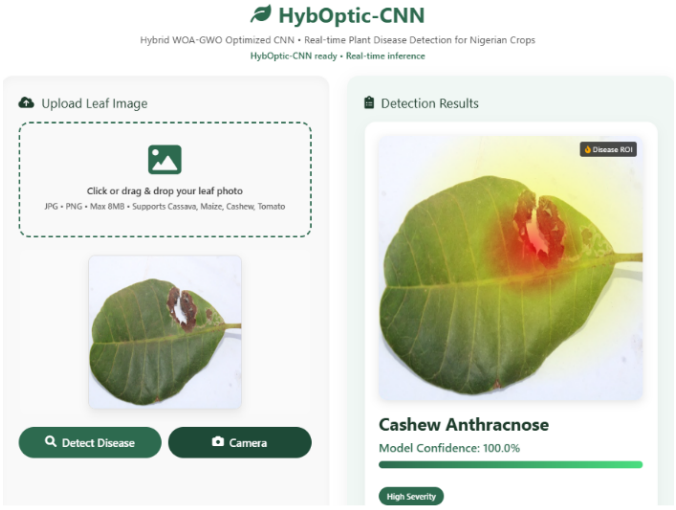

Figure 8: Cashew Anthracnose classification result.

4.6. *Study limitations*

Several limitations should be considered when interpreting the findings. First, the independent Nigerian hold-out contains 30 images and represents four of the nine target disease classes; it therefore provides a focused local-field transfer assessment rather than a complete nine-class or nationwide Nigerian benchmark. Second, the Nigerian images were collected from a single local area, and broader evaluation across multiple agro-ecological zones, seasons, crop varieties, disease stages, and imaging devices would strengthen external validity.

Third, the web application demonstrates functional software integration, whereas computational efficiency on smartphones and edge hardware requires dedicated measurement of model size, FLOPs, latency, memory use, and energy consumption. Finally, exact replication of the optimization experiment should be supported by the class-wise split manifest and the numerical WOA-GWO run-control settings included with the implementation code.

5. Conclusion and recommendations

This study introduced HybOptic-CNN, a CNN framework in which WOA-GWO hybrid optimization was used to train a nine-class plant disease classifier. During the 10-run repeated experiment, the optimized setting achieved 96.8 ± 0.4% validation accuracy, 95.2 ± 0.5% macro-precision, 94.9 ± 0.6% macro-recall, and 95.0 ± 0.5% macro-F1. The baseline achieved 90.3 ± 0.9% validation accuracy and 86.3 ± 1.2% macro-F1. Therefore, the optimized setting improved the mean validation accuracy by 6.5 percentage points (relative: +7.2%) and reduced the mean convergence epoch by 14.6 epochs (from 74.2 ± 1.5 to 59.6 ± 1.8 epochs). The single-run analysis with

class-wise performance also demonstrated that the optimized setting was superior across all nine considered classes.

Apart from the 10-run repeated validation analysis, 30 out of 152 local Nigerian field images were held out during training and used once the final model was selected. Independent test results on the 30-image Nigerian hold-out set achieved 93.3% accuracy and 93.1% macro-F1. This demonstrated that the final local model achieved good performance across the four categories of field images. The web application suggests that the model can be deployed in a web-based interactive user interface. To generalize the results to Nigerian field images, a similar independent-test design should be considered across locations, seasons, crops, cameras, and all nine target classes. Future work could benchmark the final model in terms of the number of FLOPs, inference time, memory footprint, and power consumption to assess its feasibility for on-device deployment.

5.1. Recommendations

Based on the results of this study, future work should evaluate HybOptic-CNN on smartphones and resource-constrained edge platforms and report model size, parameter count, FLOPs, inference latency, memory consumption, and energy requirements before real-time on-device operation is claimed. Future studies should also extend the present independent hold-out design using larger datasets collected from multiple Nigerian locations, seasons, crop varieties, disease stages, imaging devices, and environmental conditions, with adequate representation of all nine target disease classes. In addition, future implementations should preserve the separation between model-development and independent field-test data and should publish the class-wise split manifest, random seeds, WOA-GWO numerical settings, and evaluation code so that the reported procedure can be reproduced and extended.

Data availability

The data used in this study are available at https://data.mendeley.com/datasets/bwh3zbpkpv/1 and https://www.kaggle.com/datasets/rahimanshu/ccmt-plant-disease-dataset.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this manuscript.

Funding

The authors received no specific funding from any public, commercial, or not-for-profit organisation for this research.

References

[1] T. D. Salka, M. B. Hanafi, S. M. S. A. A. Rahman, D. B. M. Zulperi & Z. Omar, "Plant leaf disease detection and classification using convolution neural networks model: a review", Artificial Intelligence Review **58** (2025) 322. https://doi.org/10.1007/s10462-025-11234-6.

[2] A. Y. Ashurov, M. S. A. M. Al-Gaashani, N. A. Samee, R. Alkanhel, G. Atteia, H. A. Abdallah & M. S. A. Muthanna, "Enhancing plant disease detection through deep learning: a depthwise CNN with squeeze and excitation integration and residual skip connections", Frontiers in Plant Science **15** (2025) 1505857. https://doi.org/10.3389/fpls.2024.1505857.

[3] D. Devarajan, R. Allafi, M. Obayya & N. Nemri, "AI based real time disease diagnosis in plants using deep learning driven CNNs", Scientific Reports **16** (2026) 4587. https://doi.org/10.1038/s41598-025-34681-1.

[4] J. Chen, D. Zhang & Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification", Computers and Electronics in Agriculture **173** (2020) 105393. https://doi.org/10.1016/j.compag.2020.105393.

[5] C. Pal, S. Karmakar, I. Mukherjee & P. P. Chakrabarti, "A lightweight and explainable CNN model for empowering plant disease diagnosis", Scientific Reports **15** (2025) 30720. https://doi.org/10.1038/s41598-025-94083-1.

[6] R. Jothilakshmi & R. Sharanesh, "Automated plant disease detection using deep learning architectures with autonomous rover", International Journal of Recent Technology and Engineering **9** (2020) 248. https://doi.org/10.35940/IJRTE.B3469.079220.

[7] R. Deng, M. Tao, H. Xing, X. Yang, C. Liu, K. Liao & L. Qi, "Automatic diagnosis of rice diseases using deep learning", Frontiers in Plant Science **12** (2021) 701038. https://doi.org/10.3389/fpls.2021.701038.

[8] P. Pai, S. Amutha, S. Patil, T. Shobha, M. Basthikodi, B. M. A. Shafeeq & A. P. Gurpur, "Deep learning-based automatic diagnosis of rice leaf diseases using ensemble CNN models", Scientific Reports **15** (2025) 27690. https://doi.org/10.1038/s41598-025-13079-z.

[9] T. Nyawose, R. C. Maswanganyi & P. Khumalo, "A review on the detection of plant disease using machine learning and deep learning approaches", Journal of Imaging **11** (2025) 326. https://doi.org/10.3390/jimaging11100326.

[10] C. Dhakhinamoorthy, S. K. Mani, S. K. Mathivanan, S. Mohan, P. Jayagopal, S. Mallik & H. Qin, "Hybrid whale and gray wolf deep learning optimization algorithm for prediction of Alzheimer's disease", Mathematics **11** (2023) 1136. https://doi.org/10.3390/math11051136.

[11] M. Murugaiah & M. Ganesan, "A hybrid Grey Wolf-Whale optimization algorithm for classification of Corona virus genome sequences using deep learning", The International Arab Journal of Information Technology **20** (2023) 331. https://doi.org/10.34028/iajit/20/3/5.

[12] D. Thakur, S. Dangi & P. Lalwani, "A novel hybrid deep learning approach with GWO-WOA optimization technique for human activity recognition", Biomedical Signal Processing and Control **99** (2025) 106870. https://doi.org/10.1016/j.bspc.2024.106870.

[13] B. R. Ramana Reddy, G. Kalnoor, M. Devashish & P. S. K. Reddy, "Deep learning based mobile application for automated plant disease detection", IEEE Access **13** (2025) 107917. https://doi.org/10.1109/ACCESS.2025.3581099.

[14] J. J. Kennedy & G. Chelladurai, "AI-powered deep learning web application for automated plant disease diagnosis with rich visual analytics", Intelligent Agriculture **1** (2025) 12. https://doi.org/10.54963/ia.v1i2.1665.

[15] G. Karuna, C. Eshwar, Y. S. Bharath, D. Rohan, H. D. Sharma & S. M. N. Raja, "An automated system to detect plant disease using deep learning", E3S Web of Conferences **430** (2023) 01044. https://doi.org/10.1051/e3sconf/202343001044.

[16] S. Chandrasekaran, V. Rohan, B. N. Reddy & D. K. Krishna, *Enhanced deep learning model for automated plant disease classification*, 3rd International Conference on Optimization Techniques in the Field of Engineering (ICOFE), 2024, pp. 1–19. https://doi.org/10.2139/ssrn.5088981.

[17] R. Sujatha, S. Krishnan, J. M. Chatterjee & A. H. Gandomi, "Advancing plant leaf disease detection integrating machine learning and deep learning", Scientific Reports **15** (2025) 11552. https://doi.org/10.1038/s41598-024-72197-2.

[18] S. Wang, D. Xu, H. Liang, Y. Bai, X. Li, J. Zhou, C. Su & W. Wei, "Advances in deep learning applications for plant disease and pest detection: a review", Remote Sensing **17** (2025) 698. https://doi.org/10.3390/rs17040698.

[19] A. Upadhyay, N. S. Chandel, K. P. Singh, S. K. Chakraborty, B. M. Nandede, M. Kumar, A. Subeesh, K. Upendar, A. Salem & A. Elbeltagi, "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture", Artificial Intelligence Review **58** (2025) 92. https://doi.org/10.1007/s10462-024-11100-x.

[20] I. Pacal, I. Kunduracioglu, M. H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slany & R. Martinek, "A systematic review of deep learning techniques for plant diseases", Artificial Intelligence Review **57** (2024) 304. https://doi.org/10.1007/s10462-024-10944-7.

[21] M. Shoaib, A. Sadeghi-Niaraki, F. Ali, I. Hussain & S. Khalid, "Leveraging deep learning for plant disease and pest detection: a comprehensive review and future directions", Frontiers in Plant Science **16** (2025) 1538163. https://doi.org/10.3389/fpls.2025.1538163.

[22] Y. Borhani, J. Khoramdel & E. Najafi, "A deep learning based approach for automated plant disease classification using vision transformer", Scientific Reports **12** (2022) 11554. https://doi.org/10.1038/s41598-022-15163-0.

[23] D.-C. Rodríguez-Lira, D.-M. Córdova-Esparza, J. M. Álvarez-Alvarado, J. Terven, J.-A. Romero-González & J. Rodríguez-Reséndiz, "Trends in machine and deep learning techniques for plant disease identification: a systematic review", Agriculture **14** (2024) 2188. https://doi.org/10.3390/agriculture14122188.

[24] F. O. Isinkaye, M. O. Olusanya & P. K. Singh, "Deep learning and content-based filtering techniques for improving plant disease identification and treatment recommendations: a comprehensive review", Heliyon **10** (2024) e29583. https://doi.org/10.1016/j.heliyon.2024.e29583.

[25] M. Islam, A. K. M. Azad, S. E. Arman, S. A. Alyami & M. M. Hasan, "PlantCareNet: an advanced system to recognize plant diseases with dual-mode recommendations for prevention", Plant Methods **21** (2025) 52. https://doi.org/10.1186/s13007-025-01366-9.

[26] P. K. Mensah, V. Akoto-Adjepong, K. Adu, M. A. Ayidzoe, E. A. Bediako, O. Nyarko-Boateng, S. Boateng, E. F. Donkor, F. U. Bawah, N. S. Awarayi, P. Nimbe, I. K. Nti, M. Abdulai, R. R. Adjei, M. Opoku, S. Abdulai & F. Amu-Mensah, "CCMT: dataset for crop pest and disease detection", Data in Brief **49** (2023) 109306. https://doi.org/10.1016/j.dib.2023.109306.