Journal of Digital Food, Energy & Water Systems [JD-FEWS]

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

A Spider Wasp Optimizer-Based Deep Learning Framework for Efficient Citrus Disease Detection

Abisola Ayomide OLAYIWOLA
Olabisi Onabanjo University, Ago-Iwoye, Nigeria; olayiwola.abisola@oouagoiwoye.edu.ng

Olufemi Olayanju AWODOYE
Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

Dare Samseedeen OLAYIWOLA
Ladoke Akintola University of Technology, Ogbomoso, Nigeria.

Ajibola Oluwafemi OYEDEJI
University of Johannesburg, Johannesburg, 2006, South Africa.

Olukunle Ebenezer OYEBODE
University of Ilesa, Nigeria

Received: 31 May 2025
Review: 26 June 2025
Accepted: 10 July 2025
Published: 20 July 2025

Abstract—Managing citrus diseases is important for lowering crop losses and raising the economic value of citrus output. To provide a novel approach for the identification and classification of three significant citrus diseases—Citrus Canker, Citrus Greening, and Citrus Black Spot—this study uses a Deep Convolutional Neural Network (DCNN) optimized using the Spider Wasp Optimizer (SWO). Traditional disease diagnosis methods heavily rely on expert visual inspection, which is often subjective and time-consuming. To overcome these drawbacks, the proposed SWO-DCNN model automates hyperparameter tuning, improving classification accuracy and reducing computation time. Citrus image datasets containing both healthy and infected samples were pre-processed using grayscale conversion, normalization, and augmentation, and then trained using a 10-fold cross-validation technique. Performance evaluations based on sensitivity, specificity, false positive rate, accuracy, and identification time show that the SWO-DCNN outperforms the conventional DCNN in every disease category. With accuracies of 96.22%, 96.51%, 95.70%, and 97.04% for the classification of Black Spot, Greening, Canker, and overall healthy/non-healthy, respectively, the SWO-DCNN significantly reduced false positive rates and recognition times. This paper contributes to knowledge by presenting the Spider Wasp Optimizer, a hyperparameter tuning technique for deep learning models used to identify agricultural diseases. The SWO-DCNN framework offers a dependable and scalable approach for automated citrus disease classification by enhancing model performance and computational efficiency. This innovation supports precision farming initiatives and provides a reliable alternative to traditional diagnostic methods, which may improve export quality control and reduce citrus farming's financial losses.

Keywords— Citrus Disease Detection, Deep Convolutional Neural Network (DCNN), Spider Wasp Optimizer (SWO), Hyperparameter Optimization

1 Introduction

Reducing crop losses and increasing agricultural productivity depend on efficiently managing pests and diseases. Infections and physical defects on fruit peels drastically reduce market value in citrus cultivation, resulting in trade restrictions in extreme situations.

Citrus canker, citrus greening, and citrus black spot (CBS) are the most destructive of these diseases because of their aggressive infection rates and long-term economic impact. These diseases significantly reduce fruit quality and yield, often rendering produce unmarketable for export [1-2]. For example, citrus greening (Huanglongbing) has been linked to widespread losses in Asia, Africa, and the Americas, leading to severe economic consequences for

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

citrus producers [3]. Even minor infections or physical defects on fruit peels drastically reduce market value and may result in trade restrictions imposed by importing countries [4].

Historically, disease diagnosis in citrus production has relied on manual visual inspection and microscopic analysis, methods which, while useful, are often subjective, time-consuming, and dependent on expert availability [5]. These conventional approaches may not scale efficiently in large orchards or regions experiencing labor shortages, leading to inconsistent sorting and undetected spread of infections.

Automated image-based categorization methods are now the main emphasis due to computer vision and machine learning (ML) developments. Deep learning models, particularly Convolutional Neural Networks (CNNs), have been successfully used to detect citrus disease from images of fruit and leaves [6]. These systems overcome the drawbacks of human sorting, like inconsistent standards and low productivity [7].

The biggest challenge that remains despite CNNs' advancements in plant disease identification is their inability to recognize numerous diseases or multiple instances of the same disease in a single image. This multi-class classification problem is significantly influenced by the hyperparameter choices made during neural network training [8-9]. Hyperparameters such as learning rates, batch sizes, and weight initializations must be carefully considered to produce high-performing models. Many metaheuristic algorithms have been studied to automate and optimize this process, including Genetic Algorithms [10], Parameter Setting-Free Harmony Search (PSF-HS) [11], Particle Swarm Optimization (PSO) [9], and Multi-level Particle Swarm Optimization (MPSO) [12]. Despite improving model performance, these techniques are often computationally demanding and challenging to implement.

Convolutional neural networks (CNNs), a recent advancement in deep learning, have significantly improved the capacity to recognize and categorize plant diseases. Ref [13] conducted a comprehensive analysis of 121 papers that used CNNs and found significant trends and gaps in literature. [14] demonstrated the potential of edge computing in agriculture using real-time cloud-based systems that used AWS DeepLens, which produced realistic deployments with an accuracy of 98.78%. As an extension of these developments, [15] proposed PlantXViT, a hybrid Vision Transformer–CNN model that surpassed 93.00% accuracy across multiple datasets. [16] employed unsupervised deep learning with multispectral imaging for powdery mildew identification, and emphasized the significance of spectral data.

Studies have also been conducted on specific crops. Though [17] investigated deep learning techniques for apple leaf disease classification, [18] looked at performance trade-offs across multiple deep learning models. Ref [19] introduced a 14-layer Deep CNN trained on a large dataset of 147,500 images containing a no-leaf class and 58 disease categories. The model's accuracy was 99.97% using techniques like neural style transfer, Generative Adversarial Network (GAN), and picture augmentation.

Citrus crops have received a lot of attention. A customized Self-Structured CNN outperformed MobileNet in terms of accuracy and efficiency (99.00%), according to [20]. [21] developed a dense CNN that can recognize 27 disease categories in six crops with a cross-validation accuracy of 99.58%. Ref [6] trained a CNN with 94.55% accuracy using the Citrus and PlantVillage datasets. Furthermore, a Convolutional Neural Network Long Short-Term Memory (CNN–LSTM) hybrid model outperformed K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and standalone CNNs in citrus illness classification, achieving 96.00% accuracy. Advanced sensor technology improved the detection even more. [22] used hyperspectral cameras and stacked autoencoders installed on Unmanned Aerial Vehicles (UAVs) to detect Huanglongbing (HLB) with 99.72% accuracy. Additionally, [23] used hyperspectral imaging and Partial Least Squares Discriminant Analysis (PLS-DA) to identify HLB with 96.40% accuracy.

[24] took citrus photos at three different stages of pest infestation and then used different optimizers to analyze four CNN models. Visual Geometry Group-16 (VGG-16) with stochastic gradient descent (SGD) performed best in the early stages of infestation, while AlexNet with SGD performed exceptionally well in the later stages, achieving an accuracy of up to 99.34%. The study confirmed the effectiveness of CNNs in controlling pests.

Multimodal strategies have also shown promise. [25] developed a soft attention-based fusion model that classified nutrient deficiencies and HLB with 97.89% accuracy by combining Red, Green, Blue (RGB), and hyperspectral data. Deep CNN frameworks with different picture sizes and severity-level classification using VGGNet were able to distinguish between healthy and sick citrus fruits with up to 99.00% accuracy, according to [26-27]. [28] achieved 99.84% accuracy in classifying eight peel states in Ruby Red grapefruit using hyperspectral imaging and VGG-16. To increase accuracy, [29] divided diseased areas before classifying them. Similarly, CNN-based citrus disease diagnosis was shown to outperform conventional techniques like KNN and SVM by [30]. Additional developments include a two-stage CNN developed by [31] with a 94.37% accuracy rate in detecting black spots, canker, and HLB. [32] used segmented citrus leaf photos and obtained 96.00% accuracy.

Despite these successes, hyperparameter tuning remains a challenge. Therefore, this study integrated the Spider Wasp Optimizer (SWO) with a Deep CNN to enhance the detection of three key citrus diseases: canker, greening, and black spot.

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

Recent advancements in metaheuristic optimization have led to the development of algorithms to improve convergence speed, robustness, and search-space diversity in high-dimensional problems. The Spider Wasp Optimizer (SWO) has garnered significant attention due to its biologically inspired design and effective balance between exploration and exploitation [33].

SWO is inspired by the hunting and reproductive behaviour of solitary spider wasps (Pompilidae family), which paralyse spiders, lay eggs on them, and use them as hosts for larval development. This natural predator-prey interaction has been translated into a mathematical framework for solving optimization problems. The algorithm functions in two main phases:

1. Exploration (Paralysing Behaviour):
   Spider wasps randomly explore the search space to discover high-potential regions, maintaining diversity and preventing premature convergence to local optima.
2. Exploitation (Egg-Laying Behaviour):
   Once a promising region is identified, the optimizer intensifies its search around that area, refining solutions for improved precision and faster convergence.

This dual-phase mechanism enables a dynamic transition between global search and local refinement, addressing a key challenge in metaheuristic optimization [33]. SWO incorporates Bayesian-inspired probabilistic learning, allowing it to refine its belief about the best regions in the search space based on previous evaluations. This approach reduces the number of evaluations needed to reach optimal solutions, making it suitable for complex, high-dimensional tasks such as hyperparameter tuning in deep learning models [34-35]. Compared to classical metaheuristics like Harmony Search, Genetic Algorithms (GA), and Particle Swarm Optimization (PSO), SWO demonstrates several unique advantages:

- Balanced Dual Behaviour**:** Its biologically inspired two-phase mechanism ensures a more effective balance between exploration and exploitation [36].
- Probabilistic Decision Making**:** It adapts its strategy based on population feedback using Bayesian principles.
- High-dimensional Optimization Efficiency: SWO excels in tuning complex parameters such as learning rates, number of layers, batch sizes, and neuron counts in deep networks [37].

Several enhanced variants of SWO have emerged, expanding its applicability:

- Multiple-strategy SWO (MS-SWO): Integrated Lévy flights and adaptive mechanisms to improve performance on engineering design problems [38].
- Boosted SWO (BSWO): Improved feature selection in high-dimensional datasets [35].
- Binary SWO: Achieved robust classification accuracy for intrusion detection in Industrial IoT applications [39].

These adaptations confirm the algorithm’s relevance for neural network training and other tasks involving nonlinear, high-dimensional search spaces. Despite its promising capabilities, SWO remains underexplored in the field of agricultural image analysis. This study represents one of the first efforts to apply SWO for plant disease classification using deep learning. Specifically, the SWO-DCNN model was used to optimize critical hyperparameters of a Deep Convolutional Neural Network (DCNN) for citrus disease detection. The optimizer dynamically adjusted parameters such as learning rate, number of convolutional layers, number of neurons, and weight initialization schemes. To achieve these goals, the study aims to:

(i) develop an enhanced Deep Convolutional Neural Network (DCNN) using the Spider Wasp Optimizer (SWO) for optimal hyperparameter tuning.
(ii) design a robust citrus fruit disease detection and classification system based on the proposed SWO-DCNN model.
(iii) implement the system using MATLAB R2020a.
(iv) analyse the system's accuracy, sensitivity, specificity, false positive rate, and average recognition time.

Organization of the Paper

The remainder of this paper is structured as follows:

Section 2 presents the methodology adopted, including image acquisition, preprocessing, model formulation, and the integration of the Spider Wasp Optimizer with the DCNN.

Section 3 discusses the implementation of the proposed technique and presents the evaluation metrics and experimental results.

Section 4 provides a detailed discussion of the findings, comparisons with existing techniques, and implications of the results.

Section 5 concludes the paper and offers suggestions for future research directions.

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

2 Methodology

The methodology adopted in this study comprises four main stages and are outlined as follows: Data acquisition, data pre-processing, model formulation, and performance evaluation. Pre-processing entails removing noise and other undesirable components from the citrus photos by filtering, cropping, normalizing, and converting them to greyscale. Deep Convolution Neural Network was utilized to identify and categorize infected citrus images from non-infected citrus images, and Spider Wasp Optimizer was utilized to choose valuable features from the extracted features. Performance metrics like sensitivity, specificity, false positive rate, and overall accuracy were used to assess the outcome.

2.1 Acquisition of Citrus Images

The citrus fruit disease dataset used in this study was obtained from the publicly available Kaggle repository titled "Orange Diseases Dataset" by Jonathan Silva. This dataset comprises high-quality images captured under real-world agricultural conditions, including four clearly labeled categories: Citrus Black Spot, Citrus Greening, Citrus Canker, and Healthy samples. Its structured organization, visual clarity, and accurate annotations reinforce the dataset's credibility, making it suitable for supervised machine learning applications.

A total of 2,500 images were curated for this study, comprising 1,000 healthy samples and 1,500 diseased samples distributed equally across the three disease categories. To enhance intra-class variability and reduce overfitting, data augmentation techniques such as horizontal flipping, ±15° rotation, and contrast adjustment were applied. All images were resized to 600×600 pixels without altering their content. After augmentation and cleaning, the dataset comprised 2,500 images: 500 for each disease class and 1,000 healthy samples, as shown in Table 1.

A 10-fold stratified cross-validation approach was used to ensure balanced training and testing, effectively addressing the mild class disparity and supporting generalizable model performance. The validity of the dataset is supported by recent scholarly work. For instance, [40] developed a CNN-based citrus fruit disease diagnosis system and validated its effectiveness using the same dataset, achieving high classification performance. Similarly, [41] employed CNN-extracted features in conjunction with traditional machine learning classifiers to distinguish between lemon and orange diseases using this dataset, demonstrating its suitability for diverse classification approaches. These studies confirm the dataset's reliability, structured labeling, and compatibility with modern deep learning pipelines.

Table 1: Class-wise Distribution of Citrus Image Dataset After Augmentation

Class	**Number of Images**
Citrus Black Spot	500
Citrus Greening	500
Citrus Canker	500
Healthy (No Disease)	1000
Total	**2500**

2.2 Pre-processing of Citrus Images

To perform pre-processing, the coloured image was converted to grayscale, and the citrus vectors were normalized by taking the average and deducting it from each vector. This was done to purge the citrus images of noise and other undesirable components. The images were transformed into black-and-white, or grayscale, images with pixel values ranging from 0 to 255. In MATLAB, each grayscale image was expressed and saved as a matrix, which was then transformed into a vector image for use in subsequent procedures. To facilitate the normalization process, a citrus vector conversion was made.

A histogram equalization system was applied during the normalization process to improve contrast and increase the intensity range of the converted grayscale images. This improved the grayscale images' brightness so that the structure of each citrus fruit could be seen more clearly. Any shared characteristics among the citrus images were eliminated during the normalization process, leaving each one with its distinct traits. The common features were discovered by averaging the citrus vectors across the whole training set (citrus images). The average citrus vector was then subtracted from each citrus vector to create a normalized citrus vector.

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

2.3 Formulation of Spider Wasp Optimizer based Deep Convolutional Neural Network (DCNN)

The Spider Wasp Optimizer (SWO) employs a probabilistic model to represent the behaviour of candidate solutions in the search space. This model estimates the likelihood that a given set of hyperparameters will yield high-performing results. The probabilistic approach enables the optimizer to adjust its search strategy dynamically by balancing exploration and exploitation during each iteration [33].

In particular, SWO leverages concepts from Bayesian optimization, a technique that uses probabilistic surrogate models (e.g., Gaussian processes) to guide the selection of the next promising solution based on past observations [35, 42]. Unlike grid search or random search, Bayesian optimization intelligently samples the hyperparameter space to reduce the number of evaluations needed to find an optimum. This makes it well-suited for tuning deep learning models where training is computationally expensive.

In the SWO framework, the optimizer updates its belief about the best regions of the search space after evaluating each candidate. This adaptive learning mechanism helps accelerate convergence and ensures efficient exploration of complex, high-dimensional hyperparameter spaces. The following is the basic procedure for using Spider Wasp optimizer to optimize DCNN:

1. The search space for hyperparameters was defined. The search space contained all the hyperparameters that could be changed for a specific DCNN model. For example, learning rate and weight parameters.
2. The optimizer was constructed using a starting set of hyperparameters chosen at random from the search space.
3. The DCNN model was trained using the chosen hyperparameters, and the performance metric (such as accuracy or loss) was computed.
4. The optimizer adjusted its probabilistic model in response to the DCNN model's performance.
5. This update chose which set of hyperparameters to evaluate next based on the optimizer's existing probabilistic model.
6. Steps three and four are repeated until the optimizer converges on the ideal set of hyperparameters.

One of the main advantages of the Spider Wasp optimizer is its ability to adjust multiple hyperparameters simultaneously. This is important because the interactions between different hyperparameters often affect the performance of DCNN models.

2.4 CNN Parameter Selection using Spider Wasp Optimizer

Achieving optimal performance in deep learning applications requires the careful tuning of hyperparameters such as the number of layers, number of neurons per layer, learning rate, activation functions, and weight initialization schemes. These hyperparameters significantly influence deep convolutional neural networks' learning dynamics and generalization ability (DCNNs) [43-44].

Manual tuning or exhaustive methods like grid search can be inefficient and computationally expensive, particularly for high-dimensional and non-linear search spaces. Consequently, population-based metaheuristic optimization techniques—such as the Spider Wasp Optimizer (SWO)—have emerged as effective tools for automatic hyperparameter optimization due to their gradient-free nature and balance between exploration and exploitation [33, 38].

The following steps outline how the Spider Wasp optimizer was used to choose CNN's parameters:

(i) Define the search space: The first step involved defining the hyperparameter search space. This required setting ranges for each hyperparameter that could be adjusted, such as the learning rate and weight parameters [44].

(ii) Configure the optimizer: A starting set of hyperparameters had to be entered into the Spider Wasp optimizer. From the designated search area, these were selected at random [33].

(iii) Train the CNN model: The original set of hyperparameters was used to train the CNN model on the training dataset. This required adjusting the weights per the optimizer's method and moving data forward and backward across the network.

(iv) Determine the performance metric: After the model was trained, its performance was evaluated using a validation dataset. A performance metric, such as accuracy or mean squared error (MSE), was computed to achieve this [45].

(v) Update the optimizer: The Spider Wasp optimizer adjusted its probabilistic model of the hyperparameters' behavior by choosing the subsequent set of hyperparameters to assess using the present probabilistic model.

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

(vi) Steps iii–v were repeated until the optimizer reached a consensus on the ideal set of hyperparameters [34-35].

(vii) Test the finished model: After the optimal hyperparameters were identified, the final CNN model was trained using the selected hyperparameters on the entire training dataset. Finally, an alternative test dataset assessed the model's performance [46-47].

2.5 Feature Extraction and Classification using SWO-DCNN

SWO's chosen hyperparameters were used to train the DCNN model. This involved moving data through the network both forward and backward and modifying the weights according to the optimizer's algorithm. Performance metrics, such as accuracy or mean squared error (MSE), were used to assess the model's performance on a validation dataset. Using the full training dataset and the ideal hyperparameters identified by the SWO, the final DCNN model was trained. Lastly, a different test data set was used to evaluate the model's performance. It should be noted that the optimizer guides the search process during training by using a probabilistic model of the hyperparameters' behavior. Compared to other optimization methods that use a grid search or random search approach, the optimizer can explore the search space more effectively. The block diagram that depicts the developed system's process flow is shown in Figure 1, while the flowchart illustrating the trained and tested citrus using SWO-DCNN is depicted in Figure 2.

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

Algorithm 1: Spider Wasp Optimized based CNN

Input:
CNN parameters, such as weights, layers, and filters,
N: the size of the initial population,
N_{min}: The minimum size of the population,
CR: The rate of crossover,
TR: The threshold for hunting and maturing behaviour
t_{max}: Maximum number of generations
Output:
Optimized CNN parameters $\overrightarrow{SW}^*$
Step 1: Initialization
For $i = 1,2,\dots,N,$ initialize N female wasp individuals $\overrightarrow{SW_i}$ for using
$\overrightarrow{SW_i}(t) = \vec{L} + \vec{\rho} \times (\vec{H} - \vec{L})$
Where:
t is the generation index
$\vec{L},\vec{H}: Lower\ and\ upper\ bounds\ of\ the\ parameter\ space$
$\vec{\rho}: A\ D - dimensional\ random\ vector\ generated\ using\ Roulette\ Wheel\ Selection$
Step 2: Fitness Evaluation
Evaluate the fitness of each, $\overrightarrow{SW_i}$ and identify the best individual $\overrightarrow{SW}^*$
Step 3:
Set $t = 1$ (initialize general counter)
Step 4: while $t < t_{max}, do$:
Step 5:
Generate ρ_6 using roulette-based probabilistic control
Step 6
If $\rho_6 < TR$: Hunting and Nesting Behaviour
For $i = 1\ to\ N, do$
- Apply hunting/nesting update strategies to $\overrightarrow{SW_i}$
- Compute fitness f($\overrightarrow{SW_i}$)
- Increment generation count: $t = t + 1$

End For
Step 7
Else: Mating Behaviour
For $i = 1\ to\ N, do$
- Select male partner $\overrightarrow{SW_m}$
- Apply uniform crossover:

$$\overrightarrow{SW_i}\ (t+1) = Crossover(\overrightarrow{SW_i}(t), \overrightarrow{SW_m}(t), CR)$$

- Increment generation count: $t = t + 1$

End For
End if
Step 8: Memory Saving and Population Update
Update Population Size
N = N_{min}+(N-N_{min}) $\times K$
Where k is a decay factor to reduce population gradually and avoid local optima
End While
Step 9: Return best solution $\overrightarrow{SW}^*$ as the optimal CNN parameters.

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

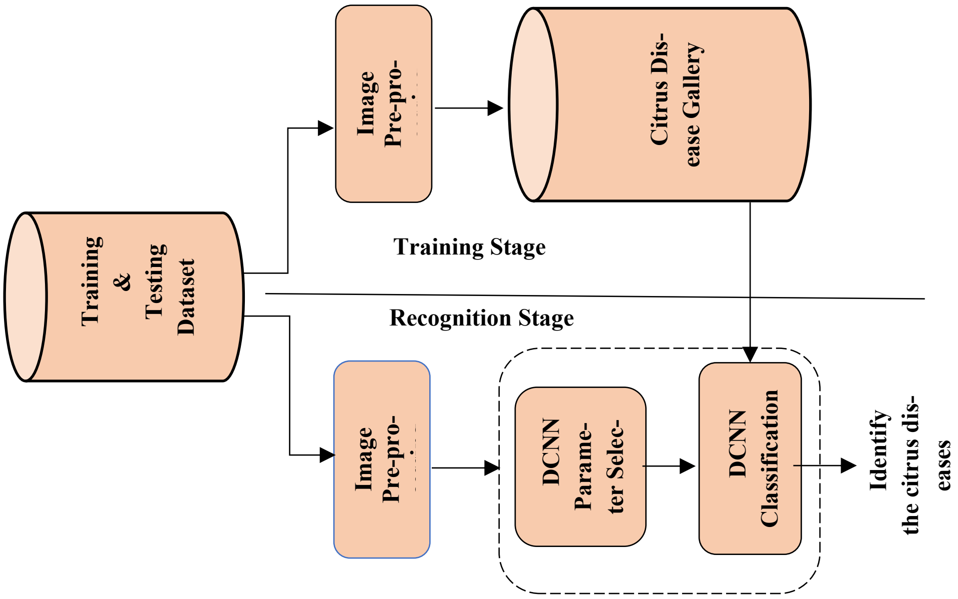

Fig. 1: The block diagram representing the process flow of the developed system

3 Implementation of Developed Technique for Citrus Fruit Disease

An online database of citrus fruit disease data was used to create an interactive Graphic User Interface (GUI) application. MATLAB (2020a) toolboxes for image processing, deep learning, and optimization were used in GUI's design. The implementation was done on a computer system with a particular configuration using the MATLAB software package.

3.1 Evaluation Metrics

The accuracy of the developed system is the ability to detect a citrus fruit with a disease or exclude a citrus fruit without a disease, and it is usually described in terms of sensitivity, specificity, and false positive rate (FPR).

Accuracy: is the proportion of citrus fruits with or without abnormalities (e.g., Citrus Canker, Citrus Greening, or Citrus Black Spot) that the system was able to correctly identify, and is given by equation 1:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \text{ x } 100\% \qquad (1)$$

Sensitivity is the proportion of citrus fruit with abnormalities that the system can correctly identify. Sensitivity is defined in equation 2

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \text{ x } 100\ \% \qquad (2)$$

Specificity shows the system's capacity for identifying non-diseased citrus fruits. Specificity is defined in equation 3

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \text{ x } 100\% \qquad (3)$$

The False Positive Rate (FPR) is the proportion of healthy citrus fruits wrongly classified as diseased to the proportion of all healthy samples citrus fruits.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \text{ x } 100\% \qquad (4)$$

3.1.1 Performance of the system using the BS dataset

Table 2 displays the results of the DCNN and SWO-DCNN methods using Black spot datasets. The table shows that the DCNN method had a false positive rate of 8.74%, an accuracy of 93.90% at 69.02 seconds, a sensitivity of 95.02%, and a specificity of 91.26%. At 46.19 seconds, the SWO-DCNN approach also had an accuracy of 96.22%, a sensitivity of 96.68%, a specificity of 95.15%, and a false positive rate of 4.85%. The SWO-DCNN method

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

was better than the DCNN method when it came to recognition accuracy, sensitivity, specificity, and the rate of false positives.

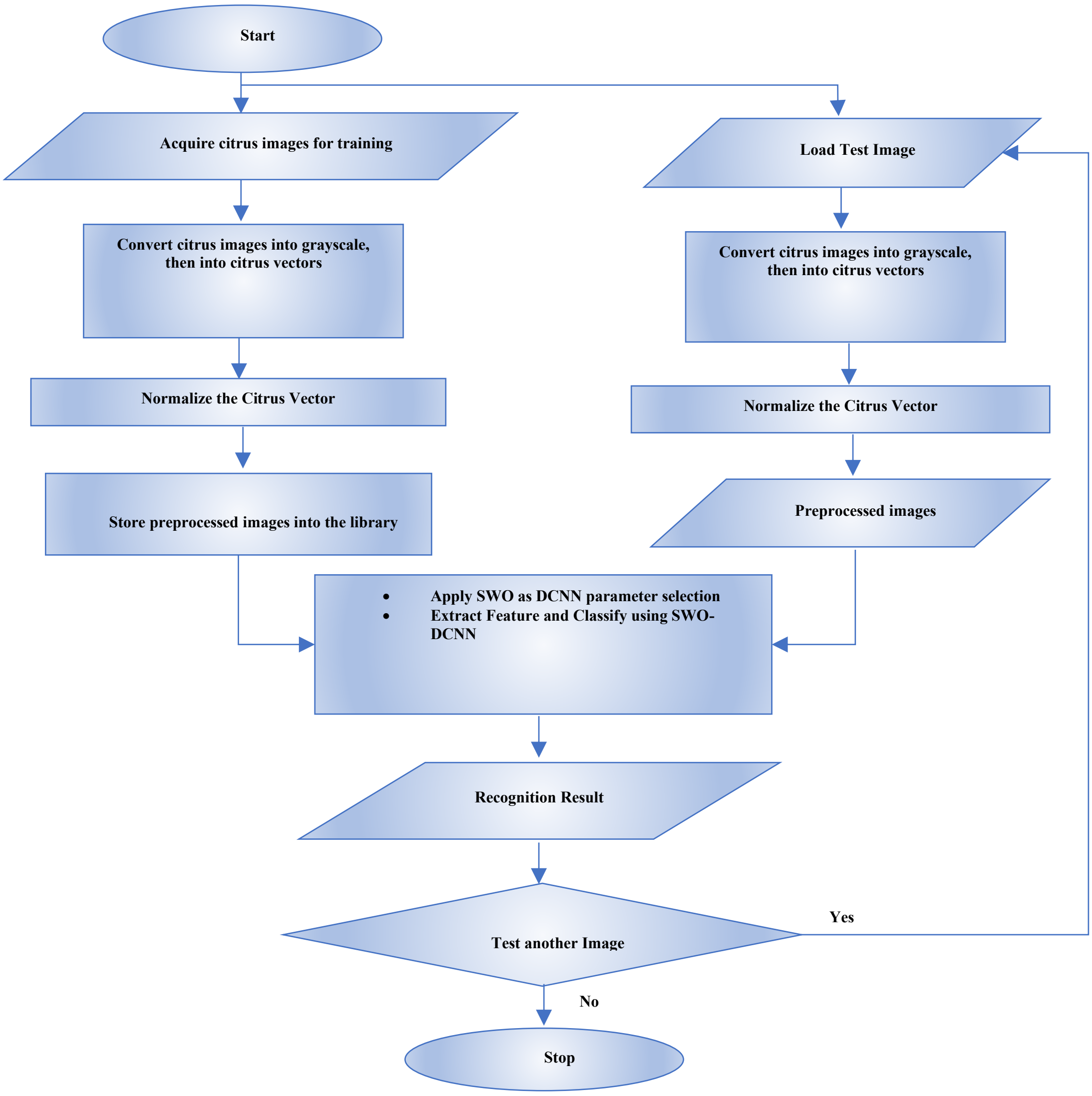

Fig. 2. Flowchart of the SWO-DCNN

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

3.1.2 Performance of the system using the Greening dataset

Table 2: Findings using BS datasets from the DCNN and SWO-DCNN techniques

Technique	FPR (%)	Specificity (%)	Sensitivity (%)	Accuracy (%)	Recognition Time (seconds)
DCNN	8.74	91.26	95.02	93.9	69.02
SWO-DCNN	4.85	95.15	96.68	96.22	46.19

Table 3 shows the results of the DCNN and SWO-DCNN methods using Greening datasets and performance markers. The table says that the DCNN method had a false positive rate of 7.98%, a sensitivity of 95.81%, a specificity of 92.02%, and an accuracy of 94.68% at 68.90 seconds. The SWO-DCNN approach also has a sensitivity of 97.12%, a specificity of 95.09%, an accuracy of 96.51%, and a false positive rate of 4.91% at 45.89 seconds. This finding shows that the SWO-DCNN method was superior to the DCNN method in terms of recognizing things, being sensitive, specific, and having a low false positive rate.

3.1.3 Performance of the system using the Canker (CCK) dataset

Table 4 shows the results of the DCNN and SWO-DCNN methods using Canker datasets concerning the performance indicators. The table shows that the DCNN method had a false positive rate of 9.52%, an accuracy of 93.41% at 68.59 seconds, a sensitivity of 94.67%, and a specificity of 90.48%. The SWO-DCNN method got similar results in 46.44 seconds, with a false positive rate of 5.71%, a sensitivity of 96.31%, a specificity of 94.29%, and an accuracy of 95.70%. Table 2a shows that the SWO-DCNN method was better than the DCNN method in terms of false positive rate, sensitivity, specificity, and recognition accuracy.

3.1.4 Evaluation of Results using the Healthy and Non-Healthy dataset

Table 5 shows the results of the DCNN and SWO-DCNN methods using Healthy and Non-Healthy datasets and performance indicators. The table shows that the DCNN method has a sensitivity of 95.85%, a specificity of 90.88%, an accuracy of 94.36%, and a false positive rate of 9.12% at 202.17 seconds. The SWO-DCNN approach, on the other hand, had a false positive rate of 4.66%, a sensitivity of 97.77%, a specificity of 95.34%, and an accuracy of 97.04% in 136.86 seconds. Table 4.2b shows that the SWO-DCNN method did better than the DCNN method in terms of false positive rate, sensitivity, specificity, and recognition accuracy.

4 Discussion of Results

This part discusses the experimental data and the citrus disease detection and classification system's overall recognition time, accuracy, FPR, sensitivity, and specificity. Table 6 shows the combined results for SWO-DCNN and DCNN based on the datasets used.

Table 3: Findings using Greening datasets from DCNN and SWO-DCNN techniques

Technique	FPR (%)	Specificity (%)	Sensitivity (%)	Accuracy (%)	Recognition Time (seconds)
DCNN	7.98	92.02	95.81	94.68	68.90
SWO-DCNN	4.91	95.09	97.12	96.51	45.89

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

Table 4: Findings using Canker (CCK) datasets from DCNN and SWO-DCNN techniques

Technique	FPR (%)	Specificity (%)	Sensitivity (%)	Accuracy (%)	Recognition Time (seconds)
DCNN	9.52	90.48	94.67	93.41	68.59
SWO-DCNN	5.71	94.29	96.31	95.70	46.44

Table 5: Findings using Healthy and Non-Healthy datasets from DCNN and SWO-DCNN techniques

Technique	FPR (%)	Specificity (%)	Sensitivity (%)	Accuracy (%)	Recognition Time (seconds)
DCNN	9.12	90.88	95.85	94.36	202.17
SWO-DCNN	4.66	95.34	97.77	97.04	136.86

Table 6: Findings of SWO-DCNN and DCNN combined according to the datasets

Technique	Black spot	Greening	Canker	Healthy/non-healthy
Accuracy (%)				
DCNN	93.90	94.68	93.41	94.36
SWO-DCNN	96.22	96.51	95.70	97.04
Sensitivity (%)				
DCNN	95.02	95.81	94.67	95.85
SWO-DCNN	96.68	97.12	96.31	99.29
Specificity (%)				
DCNN	91.26	92.02	90.48	90.88
SWO-DCNN	95.24	97.92	97.37	95.00
Recognition time (sec)				
DCNN	69.02	68.90	68.59	202.17
SWO-DCNN	74.94	49.52	72.27	167.94

4.1 Performance Evaluation of Recognition Rates

Figures 3, 4, 5, 6, and 7show that the SWO-DCNN method did better than the DCNN in terms of accuracy, specificity, and sensitivity for all dataset categories utilized in this study. Table 6 shows that the SWO-DCNN method was better at recognizing the black spot, greening, canker, and healthy/non-healthy datasets than the DCNN method by 4.81%, 8.33%, 7.89%, and 7.33%, respectively. The recognition accuracy increased, resulting in improved performance because SWO adjusted the DCNN parameters to make them more discriminated against.

Also, the SWO-DCNN technique had a greater specificity of 9.00%, 8.04%, 10.53%, and 3.57% for the black spot, greening, canker, and healthy/non-healthy datasets than the DCNN technique did. Also, the SWO-DCNN method was 3.81%, 8.33%, 8.33%, and 8.73% more sensitive than the DCNN method for black spot, greening, canker, and healthy/non-healthy datasets.

The adaptive threshold of SWO-DCNN is responsible for the technique's superior performance over DCNN in terms of sensitivity, specificity, and FPR. Additionally, this supported the findings of [47], who noted that choosing the right parameters could improve the recognition accuracy rate. According to [48], parameters tuned using the SWO algorithm improved the classification accuracy rate, and the SWO provides more accuracy than the current method. By using parameter selection, [46] were able to attain high classification rates and very high discriminating parameters.

Considering the outcome, the combination of the DCNN and SWO approach improves accuracy, specificity, sensitivity, and FPR for every dataset employed in the research. This suggests that, compared to the current DCNN methodologies, the SWO-DCNN technique produced higher-quality results. Hence, the SWO-DCNN technique did better than the DCNN technique on the measures listed above when it came to finding and classifying citrus diseases.

Journal of Digital Food, Energy & Water Systems [JD-FEWS]

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

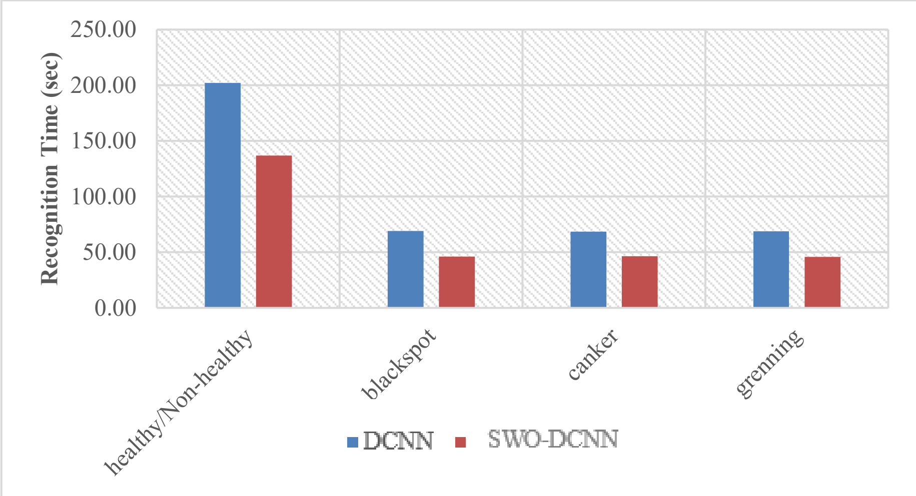

Fig. 3: Comparing the Citrus disease detection and classification system's total recognition time

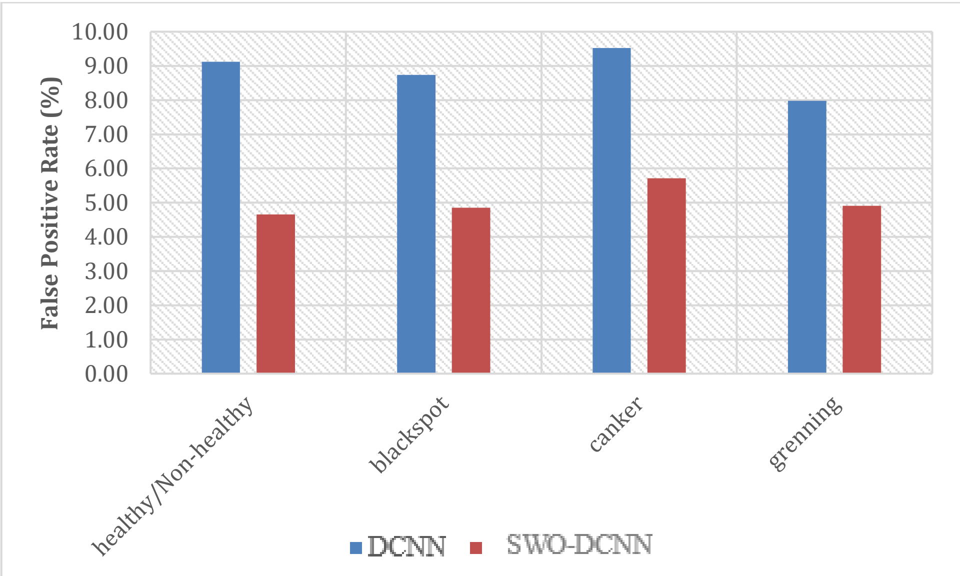

Fig. 4: FPR comparison for citrus disease classification and detection systems

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

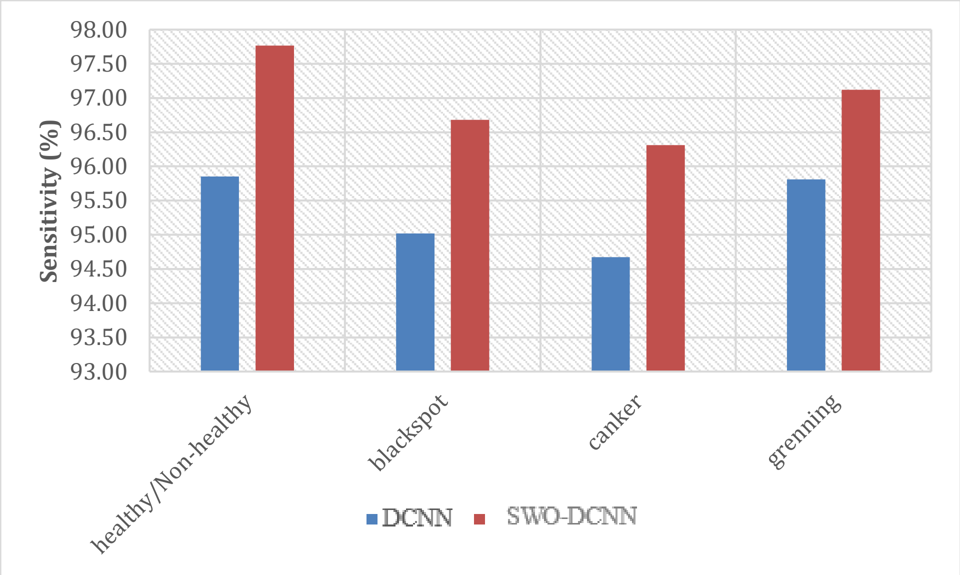

Fig. 5: Sensitivity comparison of citrus disease classification and detection systems

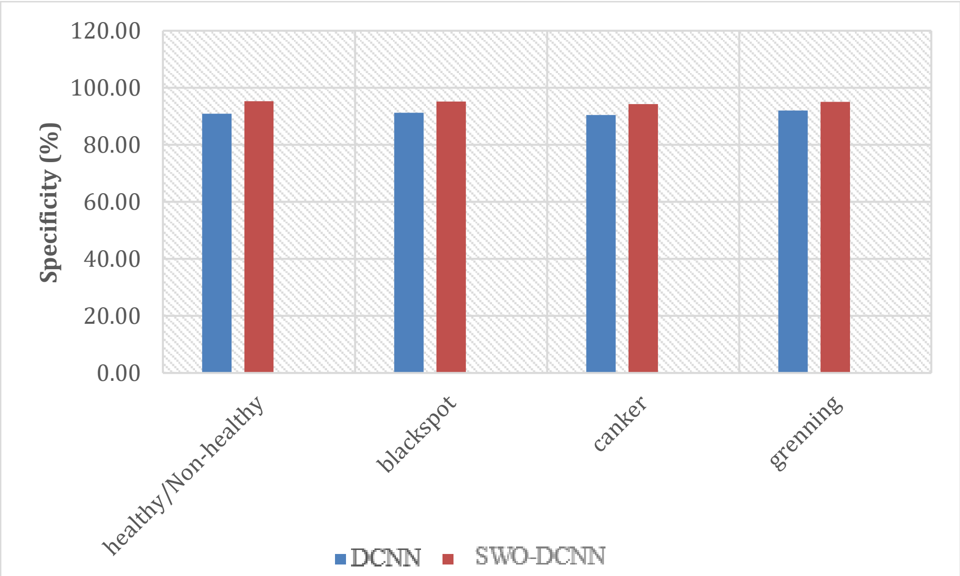

Fig. 6: Comparison of Citrus Disease Detection and Categorization System Specificity

Journal of Digital Food, Energy & Water Systems [JD-FEWS]

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

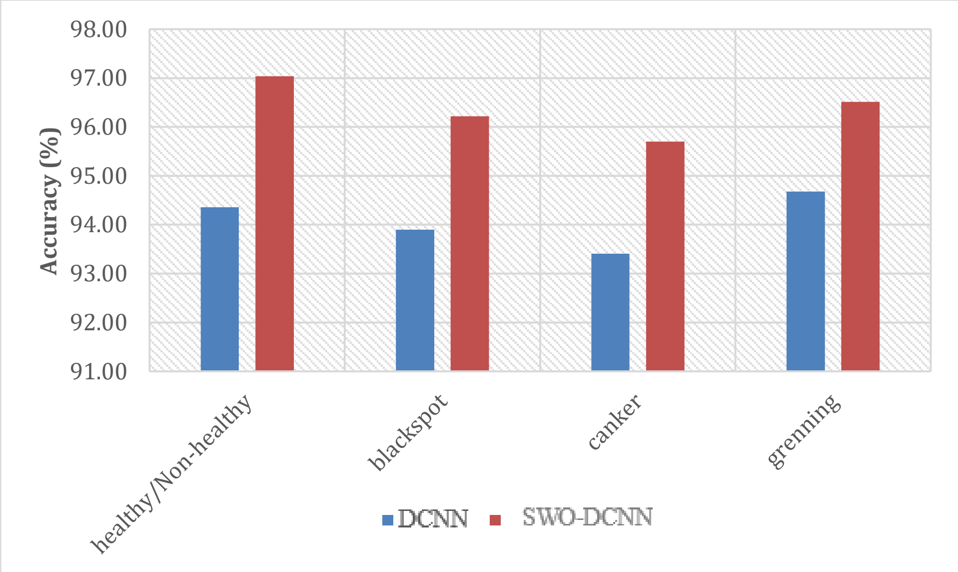

Fig. 7: Comparison of Citrus Disease Detection and Classification System Recognition Accuracy

4.2 Comparative Analysis with Other Optimizers

Additional tests were carried out to evaluate the effectiveness of the proposed Spider Wasp Optimizer (SWO). For these tests, the same DCNN architecture was used along with the following optimizers: Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Bayesian Optimization (BO).

Each optimizer was applied to tune the hyperparameters of the DCNN. The tuned parameters included learning rate, batch size, number of neurons, and weight initialization. The performance was evaluated under identical conditions. It was also performed on the same citrus disease datasets using four metrics: Accuracy, Sensitivity, Specificity, and False Positive Rate (FPR). The result of this analysis is shown in Table 7.

Table 7. Performance Comparison of SWO vs. GA, PSO, and BO

Optimizer	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**FPR (%)**	**Avg. Time (s)**
SWO-DCNN	97.04	97.77	95.34	4.66	136.86
GA-DCNN	95.23	94.22	92.22	7.78	169.30
PSO-DCNN	95.75	95.21	93.72	6.28	157.64
BO-DCNN	96.14	96.02	94.90	5.10	146.84

The SWO-DCNN outperformed all other optimizers regarding classification accuracy, sensitivity, and FPR, while maintaining a relatively shorter computation time. Bayesian Optimization showed competitive results, but the SWO was more consistent across all performance indicators. Genetic Algorithm had the longest computation time.

These results align with findings by [33], [42], and [47], who noted the superior convergence behavior and adaptability of the SWO in high-dimensional search spaces.

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

5 Policy and Practical Implications

The integration of the Spider Wasp Optimizer (SWO) with a Deep Convolutional Neural Network (DCNN) for citrus disease classification presents several practical and policy-relevant contributions to precision agriculture and food system resilience.

From a practical standpoint, the proposed SWO-DCNN model provides a fast, accurate, and automated method for detecting major citrus diseases—Citrus Black Spot, Citrus Greening, and Citrus Canker. Early and precise identification of these diseases reduces the need for broad-spectrum pesticide use, lowers production losses, and enhances crop quality. These improvements translate into higher yields and better market access, especially for export-grade citrus fruits where phytosanitary compliance is mandatory. In terms of economic contribution, timely disease detection minimizes crop rejection rates and reduces the financial burden of post-infection treatments. By improving disease surveillance, farmers can reduce yield losses—often estimated at 20–40% in severely infected fields—thus improving profitability. Additionally, the use of AI-based decision support systems reduces reliance on manual labour and expert inspections, enabling scalable disease monitoring in large orchards. This aligns with ongoing efforts to reduce production costs while maintaining high-quality standards in the agri-food supply chain.

The beneficiaries of this model include:

- Smallholder and commercial citrus farmers benefit from reduced losses and higher productivity.
- Agri-tech companies and researchers who can incorporate the SWO-DCNN framework into mobile apps or drone-based surveillance tools.
- Government agencies and policymakers can use technology to inform phytosanitary policies, enhance food security, and ensure compliance with international trade standards.
- Exporters and food processors benefit from improved fruit grading and disease-free produce, ensuring market competitiveness.

From a policy perspective, this research supports digital agriculture transformation policies, such as those articulated by the FAO, African Union, and national agricultural innovation strategies. Integrating AI-optimized disease detection into national agricultural extension programs can enhance decision-making and promote sustainable farming practices. Furthermore, public investment in open-access datasets and infrastructure to deploy such technologies in rural settings would amplify the impact.

6 Limitations of the Study

While the proposed Spider Wasp Optimizer (SWO)–based Deep Convolutional Neural Network (DCNN) model demonstrated high accuracy in detecting citrus diseases, several limitations should be acknowledged:

1. **Model Sensitivity to Image Quality**: The deep learning model's accuracy is dependent on the clarity, resolution, and proper annotation of input images. Images with low resolution, occlusion, or mixed disease symptoms may challenge the classifier's ability to correctly label instances, potentially increasing false positives or negatives.

2. **Limited Disease Categories**: The current model was trained to detect only three citrus diseases—Citrus Black Spot, Citrus Greening, and Citrus Canker—alongside healthy samples. Other prevalent citrus diseases or overlapping conditions were not included, restricting the model's utility for broader diagnostic tasks.

3. **Real-time Deployment Challenges**: The system has not yet been field-tested in real-time scenarios using camera feeds or integrated with agricultural IoT platforms. The current results are based on static image datasets and cross-validation, leaving room for further validation in operational agricultural settings.

7 Conclusion

This research evaluated the essential parameters of the SWO-DCNN technique for a citrus disease detection and classification system. The evaluation of the devised technique included 1,790 images that were divided into four groups: black spot (BS), greening (GS), canker (CCK), and healthy/non-healthy. The created SWO-DCNN was used to train and test these images at varied threshold settings.

The new SWO-DCNN method had better identification accuracy, fewer false positives, higher sensitivity, shorter computation time, and higher specificity in all the tests. This result explains that in terms of accuracy, false positive rate, sensitivity, computational time, and specificity, the developed technique outperformed the other techniques considered in this study. The SWO-DCNN technique can be used to deal with challenges in trying to detect and classify citrus diseases. Based on the results of this study, the following are suggested:

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

i. Other feature extraction and fusion algorithms can also be introduced in the future to further examine the system's performance and possibly improve upon the results obtained.
ii. Aquilla Optimizer algorithm could be hybridized with other high-convergence speed algorithms, such as the Reptile Search Algorithm (RSA), and the Dwarf Mangoose Optimization Algorithm (DMOA).

8 References

[1] Bové JM. Huanglongbing: A destructive, newly emerging, century-old disease of citrus. *J Plant Pathol*. 2006;88(1):7–37. Available from: http://www.jstor.org/stable/41998278.
[2] Gottwald TR. Current epidemiological understanding of citrus Huanglongbing. *Annu Rev Phytopathol*. 2010;48:119–39. doi:10.1146/annurev-phyto-073009-114418.
[3] Wang N, Trivedi P. Citrus Huanglongbing: A newly relevant disease presents unprecedented challenges. *Phytopathology*. 2013;103(7):652–65. doi:10.1094/PHYTO-12-12-0331-RVW.
[4] da Graça JV, Douhan GW, Halbert SE, Keremane ML, Lee RF, Vidalakis G, Zhao H. Huanglongbing: An overview of a complex pathosystem ravaging the world's citrus. *J Integr Plant Biol*. 2016;58(4):373–87. doi:10.1111/jipb.12437.
[5] Gent DH, Grove GG, Nelson ME, Wolfenbarger SN, Woods JL. Corrigendum: Crop damage caused by powdery mildew on hop and its relationship to late season management. Plant Pathol. 2015;64(1). doi:10.1111/ppa.12312.
[6] Khattak A, Asghar MU, Batool U, Asghar MZ, Ullah H, Al-Rakhami M, Gumaei A. Automatic detection of citrus fruit and leaves diseases using deep neural network model. IEEE Access. 2021; 9:112942–54. doi:10.1109/ACCESS.2021.3096895.
[7] Panchal AV, Patel SC, Bagyalakshmi K, Kumar P, Khan IR, Soni M. Image-based plant diseases detection using deep learning. Mater Today Proc. 2023; 80:3500–6. doi: 10.1016/j.matpr.2021.07.281.
[8] Arsenovic M, Karanovic M, Sladojevic S, Anderla A, Stefanovic D. Solving current limitations of deep learning-based approaches for plant disease detection. Symmetry. 2019;11(7):939. doi:10.3390/sym11070939.
[9] Duth PS, Bhat SG. Disease classification in citrus leaf using deep learning. In: 2022 IEEE International Conference on Data Science and Information System (ICDSIS); 2022 Jul. p. 1–5. doi:10.1109/ICDSIS55133.2022.9915847.
[10] Moussafir M, Chaibi H, Saadane R, Chehri A, Rharras AE, Jeon G. Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks. Plant Soil. 2022;479(1):251–66. doi:10.1007/s11104-022-05513-2.
[11] Son Y, Lee SB, Kim H, Song ES, Huh H, Czosnyka M, Kim D. J. Automated artifact elimination of physiological signals using a deep belief network: an application for continuously measured arterial blood pressure waveforms. Inf Sci (Ny). 2018; 456:145–58. doi: 10.1016/j.ins.2018.05.018.
[12] Singh UP, Chouhan SS, Jain S, Jain S. Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. IEEE Access. 2019; 7:43721–9. doi:10.1109/ACCESS.2019.2907383.
[13] Abade A, Ferreira PA, de Barros Vidal F. Plant diseases recognition on images using convolutional neural networks: a systematic review. Comput Electron Agric. 2021; 185:106125. doi: 10.1016/j.compag.2021.106125.
[14] Khan A, Nawaz U, Ulhaq A, Robinson RW. Real-time plant health assessment via implementing cloud-based scalable transfer learning on AWS DeepLens. PLoS One. 2020;15(12): e0243243. doi: 10.1371/journal.pone.0243243.
[15] Thakur PS, Chaturvedi S, Khanna P, Sheorey T, Ojha A. Vision transformer meets convolutional neural network for plant disease classification. Ecol Inform. 2023; 77:102245. doi: 10.1016/j.ecoinf.2023.102245.
[16] Benfenati A, Causin P, Oberti R, Stefanello G. Unsupervised deep learning techniques for powdery mildew recognition based on multispectral imaging. arXiv [Preprint]. 2021. arXiv:2112.11242. doi:10.48550/arXiv.2112.11242.
[17] Doutoum AS, Tugrul B. A systematic review of deep learning techniques for apple leaf diseases classification and detection. PeerJ Comput Sci. 2025;11:e2655. doi:10.7717/peerj-cs.2655.
[18] Sankhe SR, Ambhaikar A. Plant disease detection and classification techniques: a review. Multiagent Grid Syst. 2024;20(3–4):265–82. doi:10.1177/15741702241304087.
[19] Pandian JA, Kumar VD, Geman O, Hnatiuc M, Arif M, Kanchanadevi K. Plant disease detection using deep convolutional neural network. Appl Sci. 2022;12(14):6982. doi:10.3390/app12146982.
[20] Barman U, Choudhury RD, Sahu D, Barman GG. Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease. Comput Electron Agric. 2020; 177:105661. doi: 10.1016/j.compag.2020.105661.
[21] Tiwari V, Joshi RC, Dutta MK. Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. Ecol Inform. 2021; 63:101289. doi: 10.1016/j.ecoinf.2021.101289.
[22] Deng X, Zhu Z, Yang J, Zheng Z, Huang Z, Yin X, Lan Y. Detection of citrus huanglongbing based on multi-input neural network model of UAV hyperspectral remote sensing. Remote Sens. 2020;12(17):2678. doi:10.3390/rs12172678.
[23] Mei H, Deng X, Hong T, Luo X. Early detection and grading of citrus huanglongbing using hyperspectral imaging technique. Trans Chin Soc Agric Eng. 2014;30(9):140–7.

Journal of Digital Food, Energy & Water Systems [JD-FEWS]

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529
[24] Hadipour-Rokni R, Asli-Ardeh EA, Jahanbakhshi A, Sabzi S. Intelligent detection of citrus fruit pests using machine vision system and convolutional neural network through transfer learning technique. Comput Biol Med. 2023; 155:106611. doi: 10.1016/j.compbiomed.2023.106611.
[25] Yang D, Wang F, Hu Y, Lan Y, Deng X. Citrus huanglongbing detection based on multi-modal feature fusion learning. Front Plant Sci. 2021; 12:809506. doi:10.3389/fpls.2021.809506.
[26] Dhiman P, Kukreja V, Kaur A. Citrus fruits classification and evaluation using deep convolution neural networks: an input layer resizing approach. In: 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO); 2021. p. 1–4. doi:10.1109/ICRITO51393.2021.9596357.

[27] Dhiman P, Kukreja V, Manoharan P, Kaur A, Kamruzzaman MM, Dhaou IB, Iwendi C. A novel deep learning model for detection of severity level of the disease in citrus fruits. Electronics. 2022;11(3):495. doi:10.3390/electronics11030495.
[28] Yadav PK, Burks T, Frederick Q, Qin J, Kim M, Ritenour MA. Citrus disease detection using convolution neural network generated features and Softmax classifier on hyperspectral image data. Front Plant Sci. 2022;13:1043712. doi:10.3389/fpls.2022.1043712.
[29] Sharma P, Abrol P. Multi-component image analysis for citrus disease detection using convolutional neural networks. Crop Prot. 2025; 193:107181. doi: 10.1016/j.cropro.2025.107181.
[30] Negi A, Kumar K. Classification and detection of citrus diseases using deep learning. In: Data Science and Its Applications. Chapman and Hall/CRC; 2021. p. 63–85. doi:10.1201/9781003102380-4.
[31] Syed-Ab-Rahman SF, Hesamian MH, Prasad M. Citrus disease detection and classification using end-to-end anchor-based deep learning model. Appl Intell. 2022;52(1):927–38. doi:10.1007/s10489-021-02452-w.
[32] Çetiner H. Citrus disease detection and classification based on convolution deep neural network. Microprocess Microsyst. 2022; 95:104687. doi: 10.1016/j.micpro.2022.104687.
[33] Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M. Spider wasp optimizer: a novel meta-heuristic optimization algorithm. Artif Intell Rev. 2023;56(10):11675–738. doi:10.1007/s10462-023-10446-y.
[34] Narayanan R, Ganesh N. A comprehensive review of metaheuristics for hyperparameter optimization in machine learning. In: Metaheuristics for Machine Learning: Algorithms and Applications. 2024. p. 37–72. doi: 10.1002/9781394233953.ch2.
[35] Mohamed EA, Braik MS, Al-Betar MA, Awadallah MA. Boosted spider wasp optimizer for high-dimensional feature selection. J Bionic Eng. 2024;21(5):2424–59. doi:10.1007/s42235-024-00558-8.
[36] Abdel-Basset M, Mohamed R, Hezam IM, Sallam K, Hameed IA. An enhanced spider wasp optimization algorithm for multilevel thresholding-based medical image segmentation. Evol Syst. 2024;15(6):2249–71. doi:10.1007/s12530-024-09614-4.
[37] Baskaran NK, Pratap B, Bansal S. Hyperparameter tuning of convolutional neural networks using nature-inspired metaheuristic algorithms for image classification. In: Nature-inspired Metaheuristic Algorithms. CRC Press; 2025. p. 119–55. doi:10.1201/9781003612858.
[38] Sui J, Tian Z, Wang Z. Multiple strategies improved spider wasp optimization for engineering optimization problem solving. Sci Rep. 2024;14(1):29048. doi:10.1038/s41598-024-78589-8.
[39] Shtayat MBM, Hasan MK, Budhati AK, Solaiman R, Islam S, Pandey B, Saeed MMA. An improved binary spider wasp optimization algorithm for intrusion detection for industrial Internet of Things. IEEE Open J Commun Soc. 2024; 6:2926–44. doi:10.1109/OJCOMS.2024.3421647.
[40] Huang Z, Jiang X, Huang S, Qin S, Yang S. An efficient convolutional neural network-based diagnosis system for citrus fruit diseases. Front Genet. 2023; 14:1253934. doi:10.3389/fgene.2023.1253934.
[41] Arifin KN, Rupa SA, Anwar MM, Jahan I. Lemon and orange disease classification using CNN-extracted features and machine learning classifier. arXiv [Preprint]. 2024. arXiv:2408.14206. doi:10.1145/3723178.3723199.
[42] Das H, Das S, Gourisaria MK, Khan SB, Almusharraf A, Alharbi AI, TR M. Enhancing software fault prediction through feature selection with spider wasp optimization algorithm. IEEE Access. 2024. doi:10.1109/ACCESS.2024.3435333.
[43] Bengio Y. Practical recommendations for gradient-based training of deep architecture. In: Neural Networks: Tricks of the Trade. 2nd ed. Berlin, Heidelberg: Springer; 2012. p. 437–78. doi:10.1007/978-3-642-35289-8_26.
[44] Hutter F, Kotthoff L, Vanschoren J. Automated machine learning: methods, systems, challenges. Springer Nature; 2019. p. 219. Available from: http://library.oapen.org/handle/20.500.12657/23012.
[45] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. Vol. 1, No. 2. Cambridge: MIT Press; 2016.
[46] Zhao M, Shi P, Xu X, Xu X, Liu W, Yang H. Improving the accuracy of an R-CNN-based crack identification system using different preprocessing algorithms. Sensors. 2022;22(18):7089. doi:10.3390/s220807089.
[47] Subramanian M, Shanmugavadivel K, Nandhini PS. On fine-tuning deep learning models using transfer learning and hyper-parameters optimization for disease identification in maize leaves. Neural Comput Appl. 2022;34(16):13951–68. doi:10.1007/s00521-022-07246-w.
[48] Mohakud R, Dash R. Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection. J King Saud Univ Comput Inf Sci. 2022;34(8):6280–91. doi: 10.1016/j.jksuci.2021.05.012.

9 Authors

JDFEWS 6 (1): 24-41, 2025
ISSN 2709-4529

Dr. A. A. Olayiwola is a Lecturer in the Department of Computer Engineering at Olabisi Onabanjo University, Ogun State. She holds a Ph.D. in Intelligent Systems Engineering from Obafemi Awolowo University and has over a decade of experience spanning academia and international research collaborations. Her areas of expertise include Artificial Intelligence, Natural Language Processing, and Software Engineering. ORCID: 0009-0008-2982-4720.

Engr. Dr. O. O. Awodoye is an eminent faculty member in the Department of Computer Engineering, Faculty of Engineering and Technology at Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria. His areas of specialization include Biometrics, Pattern Recognition, Artificial Intelligence (AI) and Machine Learning. He can be communicated via email at ooawodoye50@lautech.edu.ng. ORCID: 0009-0006-1163-0472.

Engr. Dr. D.S. Olayiwola has Ph.D., M.Sc. and BSc. In Computer Engineering from Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria. His areas of specialization include Biometrics, Pattern Recognition, Artificial Intelligence (AI) and Machine Learning.

Ajibola Oyedeji is a Lecturer at the Department of Computer Engineering, Olabisi Onabanjo University, Nigeria. He received his B.Sc. and M.S degrees in Computer Engineering from the Olabisi Onabanjo University, Nigeria and Eastern Mediterranean University, TRNC, respectively. He conducts his research in artificial intelligence, computer vision, IoT and wireless sensor networks. ORCID iD: https://orcid.org/0000-0002-0180-492X

Dr. E. O. Oyebode is a Senior Lecturer in the Department of Computer Science, Faculty of Computing, University of Ilesa, Osun State, Nigeria. His areas of specialization include Cloud Computing, Computer Networks, Cyber Security and Knowledge-based Systems. He has some articles and conference publications published in various research platforms worldwide. He can be communicated via email at ebenezer_oyebode@unilesa.edu.ng.