RESEARCH PAPER

Optimized convolutional neural network using bacterial colony optimization for plant leaf disease detection

Ms. B. Jeevithapriya[1], Dr. K. R. Ananth[2]

[1]Research Scholar, Department of Computer Science, Nandha Arts and Science College (Autonomous), Erode, India. E-Mail: jeevithapriyact@gmail.com

[2]Associate Professor & Head, Department of Computer Science, Nandha Arts and Science College (Autonomous), Erode, India. E-mail: sapujaa@gmail.com

Abstract: The currently growing effects that plant diseases have on global agriculture require the creation of highly intelligent and accurate detection systems. Convolutional Neural Networks (CNNs), as deep learning, have proved effective in the detection of plant diseases based on images. The CNN performance, however, is very sensitive to hyperparameter tuning, which usually requires manual, sub-optimal tuning. The study suggests a new method to detect plant diseases with an optimized CNN architecture optimized by the Bacterial Colony Optimization (BCO). The BCO algorithm replicates the adaptive foraging behavior of bacterial colonies to automatically determine optimal CNN hyperparameters, such as the number of filters, kernel size, pooling methods, learning rate, and dropout probability of the CNN. The experiment conducted on the PlantVillage dataset demonstrated that the proposed BCO-CNN achieved an accuracy of 96.68% with a false alarm rate (FAR) of 4.05% outperforming other heuristic-fitted models such as particle swarm optimization (PSO) - CNN, CNN with support vector machine (SVM), and CNN, VGG16 in accuracy, precision, recall, and F1-score. The given work offers an automated, scalable approach to the accurate, early detection of the diseases of the plant, contributing to better yields of crops and sustainable agriculture.

Keywords: convolutional neural network; bacterial colony optimization; plant leaf disease detection; hyperparameters optimization;

How to cite this article: Jeevithapriya B, Ananth KR. Optimized convolutional neural network using bacterial colony optimization for plant leaf disease detection. Int J Drug Deliv Technol. 2026;16(17s): 24-35. DOI: 10.25258/ijddt.16.17s.4

1. Introduction

In the world economy, agriculture has been the backbone and an important pillar of food security. The population is constantly growing, and that is why the demand for agricultural products is constantly going up. Nonetheless, plant disease is one of the most common problems that endangers the productivity of agriculture. Crop diseases cut down on the yield, besides compromising the quality of the crops, thus causing losses to both farmers and the country in terms of revenue. Recent (Food and Agriculture Organization-FAO) publications point out that crop losses through plant diseases amount to 40 percent per year worldwide. Their prevention in advance and intervention to control the same is crucial [1]. However, conventional methods of disease detection in plants entail visual inspection by farmers, laboratory analysis, and chemical detectors. Although these techniques might be somewhat effective, they are tedious, labor-intensive, costly, and in most cases are not available in the remote or rural areas of farmlands [2].

The introduction of Artificial Intelligence (AI) and especially of deep learning (DL) to precision farming has been developing rapidly [3]. The class of DL models known as CNNs, which are especially suitable when it comes to processing image data, have proven to be of tremendous potential in the automatic identification of plant diseases[4]. Through analyzing the images of leaves, CNNs can correctly determine whether or not there is a disease, and what kind of one, in most cases, exceeding human skill. But CNNs are considerably sensitive to the architecture and hyperparameter settings of the CNNs. The learning capacity and generalization performance of the model are directly influenced by such factors as the number of convolutional layers, the sizes of filters, activation functions, pooling techniques, learning rates, and regularization methods. Tuning of these parameters is time-consuming and computationally demanding, which requires expert knowledge, and is practically very difficult [5].

In an attempt to mitigate this issue, researchers have resorted to optimization algorithms that allow for automatically exploring the most effective hyperparameter values [6]. Nature-inspired metaheuristic optimization algorithms have been used in growing numbers in this field because they are global searches and tend to avoid local minima. In this regard, methods based on swarm intelligence (SI) have emerged. Specifically, the BCO algorithm has developed into a strong evolutionary optimization method [7]. BCO provides an opportunity to tune the CNN model better in the context of CNN optimization to detect diseases in plants without running extensive manual optimization of them. By encoding the CNN hyperparameters as a structured vector with the utilization of BCO to evolve the structured vector, one can systematically search for the configuration with the most accurate classification results when presented with plant leaf micro hierarchies. In addition, the population-based structure of BCO guarantees diversity throughout the search process, where the chances are minimal of landing in suboptimal solutions [8, 9]. BCO with CNN, in addition to the higher accuracy of detection, also increases model scalability and flexibility in various crop brands and types of diseases.

The study proposes this framework that considers the representational strength in and optimization with BCO to identify the diseases in the plants based on the images of leaves. To train and test the proposed model, there are predefined benchmark datasets, namely, the PlantVillage dataset, which contains high-resolution images of healthy and diseased leaves of plants. By gradually optimizing the set of hyperparameters in a CNN in a sequential line search over a binary or real-valued vector, the BCO algorithm allows minimizing the model in a data-driven fashion. Although the suggested framework has excellent results on benchmark datasets, one of the most crucial aspects is that it applies to actual farming conditions. Field images also tend to contain uncontrolled lighting, complicated and cluttered backgrounds, overlaps of leaves, and partial occlusions, unlike the controlled environments of PlantVillage. These may cause noise and lower the recognition accuracy. As a solution to this, transfer learning, as proposed through the model, will optimize the BCO-CNN network with respect to smaller field-specific datasets to adapt to natural variations. Furthermore, the conditioning of synthetic data generation with GAN improves the resilience of the model, as it simulates various environmental factors, thereby alleviating the constraint of the datasets. The study can be undoubtedly useful both to scholarly science and in applications to increase food security and sustainable agricultural production. The results prove that the collaboration of CNN and BCO can be utilized to develop more precise, effective, and deliverable systems to detect agricultural diseases. This present work focuses on the following contributions:

- Introduced a new hybrid structure that tries to integrate CNN and BCO into an automated system for detecting the disease in plants.
- Devised a useful encoding scheme of how CNN hyperparameters could be represented as binary/real-valued data, which would then be optimized efficiently through the BCO algorithm.
- Used the optimized CNN model to carry out the right recognition of diverse plant diseases that apply leaf images to benchmark.
- Analyzed better accuracy, precision, recall, and F1-Score classification results than any optimized CNNs and alternative optimization algorithms.
- Saved on manual tuning task and training process with management of hyperparameters of CNN by fully automating the process on hyperparameter selection based on an intelligent global search method of the BCO.

The remainder of the paper is organized as follows: Section 2 reviews the related work on planet disease detection based on deep learning and metaheuristic methods. Section 3 presents the research methods, including CNN, BCO, and optimized BCO-CNN. Section 4 describes the experimental results, and Section 5 describes the conclusion with future works.

2. Related works

One of the greatest influences on the agricultural productivity and food security around the world is the diseases that affect plants. These diseases are critical and need to be detected early and precisely to avoid losses due to yield and guarantee sustainable farming. Conventionally, the diagnosis of diseases is based on manual observation of experts, and this process is lengthy, subjective, and prone to error, particularly when the symptoms of various diseases visually resemble. As digital agriculture is rapidly developing, the use of computer vision and DL methods has been proposed as promising for developing automated plant disease detection. As Table 1 indicates, the existing deep learning models to identify plant diseases are generally based on standard CNN or transfer learning models, but with several shortcomings (small datasets, poor optimization, and generalized across crops).

Table 1 : Reviews of deep learning approaches for planet disease detection

Ref. No.	Model	Optimization	Dataset(s)	Key Features	Performance	Limitations
M. Asghar et al. (2025) [10]	HPDC-Net (Hybrid Parallel Dilated CNN)	Lightweight CNN with hierarchical aggregation	Potato & Tomato Leaf	Reduced computational cost and efficient feature extraction	High accuracy	Limited evaluation lacks optimizer-based tuning
S. Gupta et al. (2025) [11]	VAE + GAN hybrid with Self-Attention, Residual & SR modules	GAN with perceptual and structural quality preservation	Real-field Eggplant dataset (1325 samples)	Generates real-like images; solves class imbalance	Fast convergence	Focused on one fruit type; computationally heavy
Gangadevi et al. (2025) [12]	FS-FRNet (Fruit Fly + Simulated Annealing Optimized	FOA + SA for hyperparameter tuning	Custom Plant Leaf	Wiener filter denoising and hybrid optimization	Improved convergence & Accuracy	High computational load; complex hybrid structure

	Faster R-CNN)					
I. Pacal et al. (2025) [13]	Ensemble of CNN + ViT	Adaptive threshold soft voting ensemble	PlantVillage & CD&S	Data augmentation	Excellent performance across datasets	High model complexity
D. S. Joseph et al. (2024) [14]	Benchmark DL architectures	Fixed hyperparameter tuning	Rice, Wheat & Maize	New benchmark datasets	Used 8 DL models; baseline comparison	Static parameters; lack of optimization
G. Shrestha et al. (2020) [15]	CNN-based model	None	Custom Dataset (15 cases)	Simple CNN architecture	Fast convergence rate	Small dataset; low generalization
Y. M. Abd Algani et al. (2023) [16]	ACO-CNN	ACO for CNN tuning	PlantVillage	Optimization-guided CNN learning	Improved accuracy & robustness	Premature convergence
P. Sharma et al. (2020) [17]	S-CNN (Segmented CNN)	Region segmentation	PlantVillage	Segmented vs full-image comparison	High accuracy	Limited to a single dataset; static architecture
S. M. Hassan et al. (2021) [18]	CNN family	Transfer learning	Multi-plant dataset (14 species, 38 diseases)	Fine-tuning batch size, epochs, dropout	High accuracy	Manual tuning required
J. Eunice et al. (2022) [19]	Transfer Learning CNNs	Fine-tuning of pre-trained models	PlantVillage (54,305 images, 38 classes)	Hyperparameter tuning & evaluation	High accuracy	Focused only on a large dataset
S. T. Y. Ramadan et al. (2025) [20]	GAN-augmented CNN	GAN-based data augmentation	Rice Leaf Disease Dataset	GANs generate synthetic images to handle imbalance	Enhanced results	Limited to the rice dataset
C. Wang et al. (2025) [21]	FHWD model with Wavelet & FFT loss	Multi-band frequency-domain constraint	PlantVillage (10 crop diseases)	Enhances lesion authenticity & texture quality	+7.25% accuracy gain	Needs validation on real-field datasets
S. Gupta et al. (2025) [11]	VAE + GAN hybrid with Self-Attention	GAN with perceptual and structural quality preservation	Real-field Eggplant dataset	Generates real-like images;	Enhancements to solve class imbalance	Focused on one fruit type; computationally heavy
A. A. Alatawi et al. (2022) [22]	VGG-16-based CNN	—	PlantVillage (15,915 images, 19 classes)	Deep feature extraction from leaf images	Optimal features are extracted	Lacks advanced augmentation; limited to static architecture
C. Zhang et al. (2025) [23]	Improved VGG16	Hyperparameter optimization	Potato leaf dataset	Lightweight attention mechanism	High accuracy with fast convergence	Evaluation limited to potato diseases
P. Hari et al. (2025) [24]	FDL-IWT + PMACNN	Weight adaptation based on local knowledge	Two public plant leaf disease datasets	Attention-based lightweight CNN	Best performance with high security	No on-device real-world test

Studies also exist where hybrid or ensemble techniques are used, but are computationally costly. GAN-based augmentation plans have proven to improve the variety of data, but tend to be cropline-based. Optimization-based models are more precise but too early convergent or too idealized. The benchmark models also possess the limitation of constant parameters, as well as the limitation of flexibility. Light CNNs such as HPDC-Net and FDL-IWT are fast, yet they lack robust optimization schemes. Thus, the issue of data imbalance and poor parameter tuning has not been addressed. In order to overcome them, the proposed optimized CNN with BCO offers the possibility of changing the parameter of choice and faster convergence. Other than that, GAN-based data augmentation makes the data more diverse and balanced. The blending will result in increased precision and strength as well as scalability of real-time identification of plant illnesses in various crops.

3. Problem definitions

In images of leaves, the analysis of plant diseases is a complex non-linear classification problem, which relies on the modification in texture, color, and illumination. The objective theory is used to model a discriminating function. $f_\theta: X \rightarrow Y$ that minimizes error on image features for the disease classes. Hierarchical feature extraction can assist CNNs to approximate these nonlinear mappings despite the fact that they need the optimal settings of hyperparameters are the learning parameters that regulate the convergence and generalization. Improper selection of hyperparameters brings about a local minimum, overfitting, and insensitivity to hidden data. The point, then, is to determine. The conventional tuning algorithms are both inefficient in computation as well as being inflexible. The theoretical model of the problem of plant disease recognition presented in the paper takes the form of an optimization-based learning problem, where the search is done by BCO that ensures high quality and fast convergence.

4. Research methods

4.1 GAN

Common classical techniques used to artificially inflate the training dataset of image classification problems to include artificial augmentation of the data include rotation, flipping, scaling, cropping, and translation. Though these changes increase generalization by introducing simple geometry variations, they do not introduce any new semantics or structure. Consequently, conventional augmentation models tend to lead to diminished diversity, this can restrict the generalizability of models in natural farms where variations in light levels, leaf area, and texture, as well as diseases and occlusion, can occur. GANs, on the other hand, offer self-directed and dynamically adaptive augmentation. GANs can be trained on the distribution of the original leaf plants and generate new and realistic, and high-quality synthetic samples, which are highly analogous to the actual diseased and healthy leaves. This aspect reacts directly to data imbalance and the small size of data sets, which are sensitive to plant disease detection. GAN is a complex type of DL model developed by Ian Goodfellow in the year 2014 [25]. GANs are an interesting and strong subtype of neural networks whose generative task is represented. They have two parts, namely, the Generator and the Discriminator. The task of the generator is to produce data that is quite similar to the real data. It consumes random noise as an input and creates samples. The discriminator has a task to discriminate between real data and fake data, such as that generated by the generator. A random vector noise (z) is used as an inputcreatese generator (G) that create samples $G(z)$ with a similar distribution to that of the training set. The discriminator D tries to learn to discern generated samples and real ones. The concept of a two-play minimax game, as described by the competitive engagement between the generator as well as the discriminator, is followed by,

$$\max_D \min_G V(G,D) = E_{x\sim P_{data}(x)}[log\, D(x)] + E_{z\sim P_{z(z)}}[log(1 - D(G(z)))] \quad (1)$$

where the probability distributions for the generated and the original samples are denoted by P_{data} and $P_z(z)$, respectively. The generator creates images $G(z)$ to trick D during training, the discriminator is taught to maximize $D(x)$ for images with $x \sim P_{data}$ in the objective function. By producing samples that appear identical to P_data, the generator instinctively attempts to deceive the discriminator as much as possible, while the discriminator enhances its capacity to differentiate between synthetic and real images.

4.2 CNN

The input layer, CL, MPL, FCL, and output layer make up the CNN architecture. The CNN's CL layer is in charge of identifying characteristics in the input images using multiple convolutional filters. The convolutional operation is carried out by these convolutional filters at each offset of the input image. Gradient descent training, which modifies the CL's parameters, should be used to optimize the weights in the CL. A ReLU is used to map the features that were taken out of the CL into feature space. The network's gradients and activations are normalized by a batch normalization layer (BNL), which sits between the CL and ReLU. The PL is used to preserve the most important details of the image while shrinking the size of the feature maps that were acquired via the CL. The CNN's training PL has neither weights nor biases. The retrieved characteristics from the CL and MPL are categorized into a specific CL class by the final layer, a fully connected classifier layer. hyperparameters to achieve the best network performance. The use of SGD has the drawback of having a large number of hyperparameters that affect network performance.

4.3 BCO

BCO is a novel swarm intelligence (SI) method introduced by Niu et al. (2012) [26]. BCO technique involves five simple processes, which include chemotaxis and communication, elimination and reproduction, and migration. Communication is always in existence throughout the entire lifetime of BCOs through chemotaxis. After a long period during which chemotaxis and communication take place between bacteria, the bacteria have two options available. They could either die of starvation or breed themselves in case they could acquire food by themselves. They may face dangerous experiences in a tough environment; taking risks or seeking space may result in people coming across hazardous situations [7].

Bacteria of high-energy will reproduce at a high rate and form new individuals during the removal process and reproduction and the unhealthy individuals will be replaced.

Energetic bacteria are good nutrient seekers. The final stage, a migration stage, implies that the bacteria can drift in the search range under some conditions. Chemotaxis and communication were employed throughout the whole BCO operation. The other two steps, however, are only completed in a particular situation, including completion of a particular number of iterations, the generation number generated randomly is less than a specific probability level, etc. Both the swimming and the tumbling missions are modeled by two different lifetimes. The tumbling process adds a stochastic direction to the swimming process. An idealizing searching director combined with a chaotic searching director predisposes the search orientation. The current location of individual bacteria is as follows:

$$Position_i(T) = Position_i(T-1) + C(i) \times [f_i \times (G_{best} - Position_i(T-1)) + (1-f_i) \times P_{best_i} - Position_i(T-1)) + turb_i] \quad (2)$$

Provided that their swimming process is not perturbed by the turbulence, the bacteria will move to the location where they are most at ease, and the location of every bacterium will be updated in the following way.

$$Position_i(T) = Position_i(T-1) + C(i) \times [f_i \times \left(G_{best} - Position_i(T-1)\right) + (1-f_i) \times \left(P_{best_i} - Position_i(T-1)\right)] \quad (3)$$

Where, $turb_i$ - turbulent direction variance value. $f_i \in \{0,1\}$. P_{best} is the personal best. G_{best} is the global best. $C(i)$ is the value of the chemotaxis step size is defined as follows,

$$C(i) = C_{min} + \left(\frac{Iter_{max} - Iter_j}{Iter_{max}}\right)^n (C_{max} - C_{min}) \quad (4)$$

Where, $Iter_{\max}$ is the maximum iteration, $Iter_j$- is the present iterations, respectively. n- the linearly decreasing method of the chemotaxis step.

4.4 Optimized CNN based on BCO

GA, ACO, and PSO tend to prematurely converge and are very responsive to the settings of the parameters. BCO resolves such problems by using chemotaxis to search and eliminate adaptively, as well as to disperse to preserve diversity. Its replication phase maintains elite solutions, and chemotaxis is an intelligent mutation to fine-tune. Also, dispersal recovers diversity and speeds up the process of convergence via efficient local exploitation. The application of BCO is critical in the improvement of the performance of CNNs to address the detection of planet disease. When it comes to the CNN architecture, hyperparameter tuning is usually done by empirical method or a grid/random search approach. Such manual or brute-force algorithms are computationally costly, result in inferior results, and are slow. This is where BCO can be used that automate the hyperparameter tuning procedure, increasing the overall accuracy of the model, convergence rates, and generalization. BCO is reused based on the way that groups of bacteria in nature tend to self-organize. Such operations can be translated into gaining, on-the-fly network architecture and evolutionary training when used to optimize CNN. The bacteria explore the hyperparameter space with an eye toward optimizing some fitness function, typically a measure of classification accuracy or of the loss experienced on a validation set. Bad applicants are shed off, and good ones are improved and cloned, just like in the natural selection process that builds towards global optimization. The combination of BCO and CNN is particularly effective when detecting planet diseases. Optimizing key parameters of the CNN guarantees that BCO makes use of the most discriminative features in the classification of pathogens such as early blight, late blight, bacterial spot, and healthy leaves. Also, it is possible to add BCO to tune the internal weights of the CNN layers, taking BCO to the end-to-end optimizer. Such ability brings better

Table 2. Details of datasets

Class Name	Original	Generated	Total
Spider mites two spotted spider	100	3,400	3,500
Bacterial Spot	2,127	1,373	3,500
Early Blight	1	2,500	3,500
Late Blight	1,909	1,591	3,500
Leaf Mold	952	2,548	3,500
Septoria Leaf Spot	1,771	1,729	3,500
Spider Mites Two Spotted Spider Mite	1,676	1,824	3,500
Target Spot	1,404	2,096	3,500
Mosaic Virus	373	3,127	3,500
Yellow Leaf Curl Virus	3,209	291	3,500
Healthy	1,591	1,909	3,500
Total count	**16,112**	**22,388**	**38,500**

Table 3. Parameter settings of BCO

Name of parameters	Notations	Values
Number of bacteria	S	50
Chemotaxis	N_c	100
Swim step	N_s	5
Reproductive value	N_{re}	5
Elimination and dispersal	N_{ed}	4
Probability of elimination	P_{ed}	0.25
Chemotaxis value (minimum)	C_{min}	0.01
Chemotaxis value(maximum)	C_{max}	0.2
Maximum iterations	Max_Iter	500
Elimination and dispersal steps	N_{ed}	4

Table 4. Parameters setting of CNN and GAN

Parameter	CNN	Parameter	GAN
Input	224×224	Optimizer	Adam
Pooling	Max	Learning rate (G, D)	0.0002
Learning rate	0.001	Dropout	0.25
Dropout	0.25	Batch Size	32
Batch Size	32	Epochs	200
Epochs	100		

stability to learning and convergence behavior in the training process. In contrast to gradient-based optimizers, which can easily be trapped in local minima, BCO can bypass local minima, particularly in a non-convex search space with high dimensions, which is common among deep learning models. To summarize, the findings indicate that BCO can be used to comprise a strong optimization backbone for planet village leaf disease detection by CNN. It also makes the architecture search and training of a model algorithmic, increases the performance of this model, and makes the parameter tuning process less reliant on the expertise of a person. This is due to the capability of biology-inspired exploration and exploitation ushered by BCO, a promising technique towards building intelligent, adaptable, and efficient disease classification systems in precision agriculture. The objective function is the quantitative function through which the BCO could explore the search space to find the most appropriate CNN hyperparameters. The objective function used in this work to assess the accuracy of predictions made by the model in the optimization is the Mean Squared Error (MSE), which is defined as follows,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i) \quad (5)$$

N is the number of samples, $y_i$is the actual and $\hat{y}_i$ is the predicted output. By minimizing MSE, the optimization process makes sure that the CNN gets as low a prediction error as possible, which will lead to a better detection error and stability of plant leaf disease classification.

5 Experimental analysis

Experimental findings give quantitative results in regards to model performance in different configurations. These are accuracy, precision, recall, F-measure, and false alarm rate (FAR) of performance metrics. By controlled experiments of varying activation functions, bias ranges, and weight initialization, researchers can show that the model is sensitive to these architectural and hyperparameter changes. A comparison of the results also indicates the success of data augmentation, which indicates whether the model has a better generalization when the data is more diverse. The MATLAB 2022b was used to implement the proposed method. The tomato dataset was collected from Plant Disease datasets (“https://www.kaggle.com/datasets/emmarex/plantdisease").
The average values of 30 independent runs with varying random seeds are used to report the obtained results. This will minimize the effects of the chance and provide a fair representation of the strength of the model. Also, 80 % of the data is employed in the training and the rest 20 % in the test purposes. Tables 1,2, and 3 show details of datasets and parameter settings.

5.1 Data augmentation

GANs are applied to augment the tomato data in plant PlantVillage dataset due to the inherent imbalance in classes of the dataset [27-29]. A GAN produces images of synthesized leaves depending on their classes, which means that minority classes can be augmented with enough synthetic data, and majority ones can be trained with the minimal number of synthetic images. This GAN-based method expands the volume of data, adds natural differences in the texture of the leaves and disease spots, and enhances the generalization capability of the CNN [30]. By producing more examples of minority classes, GANs help in achieving better class distribution, which in turn enhances the model’s ability to detect all disease types more effectively.

5.2 Performance measures

Performance measures such as accuracy, precision, recall, F1-score, and false alarm rate (FAR) play a critical role in assessing the effectiveness of classification models.

- Accuracy is the percentage of a currently classified forecast and is computed as given below

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

- Precision is the ratio of good results that were really right and is computed as follows:

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

- Recall is a measurement of the ratio of the actual identified positives that were not false, and is found as follows:

$$Recall = \frac{TP}{TP+TN} \quad (8)$$

- F-Measure is the collective measure and harmonic mean of precision and recall, which is defined as follows,

$$\boldsymbol{F-Measure = 2 \times \frac{(precision \times recall)}{(precision+recall)}} \quad (9)$$

- FAR is the ratio between the healthy leaves and the false rate of diseased leaves. It speaks about false positives, a case when the model is more likely to predict that there is a disease where there is no such disease.

$$FAR = \frac{FP}{FP+TN} \quad (10)$$

Here, *TP, TN, FP, and FN* are the symbols of true positive, true negative, false positive, and false negative, respectively**.**

5.3 Results analysis

The experiment to test the optimized CNN using BCO was run over different combinations of the biases, weights, and activation functions with the Planet disease data set. Tables 4 and 5 show the performance results based on accuracy before and after data augmentation. Figures 1 and 4 show training

Table 5. Performance analysis of the tomato dataset

Bias	Weight	Activation function	Accuracy	Precision	Recall	F-measure	FAR
[0,1]	[-1,1]	ReLU	94.56	93.87	94.12	94.00	6.13
		Sigmoid	93.78	93.56	94.23	93.89	6.44
		Tanh	95.34	94.21	93.98	94.09	5.79
		Leaky ReLU	94.67	94.78	94.32	94.55	5.22
		Softmax	93.31	93.15	93.07	93.11	6.85
		Swish	**95.62**	**95.60**	**94.76**	**95.32**	**4.11**
	[-0.5,0.5]	ReLU	93.87	93.45	93.78	93.61	6.55
		Sigmoid	94.23	94.12	93.89	94.00	5.88
		Tanh	94.56	94.34	94.12	94.23	5.66
		Leaky ReLU	95.12	94.89	94.56	94.72	5.11
		Softmax	93.45	93.23	93.11	93.17	6.77
		Swish	95.34	95.21	95.12	95.16	4.79
	[0,1]	ReLU	94.78	94.56	94.34	94.45	5.44
		Sigmoid	93.98	93.78	94.12	93.95	6.22
		Tanh	95.23	95.12	94.89	95.00	4.88
		Leaky ReLU	94.34	94.12	94.56	94.34	5.88
		Softmax	93.67	93.45	93.23	93.34	6.55
		Swish	95.45	95.34	95.23	95.28	4.66

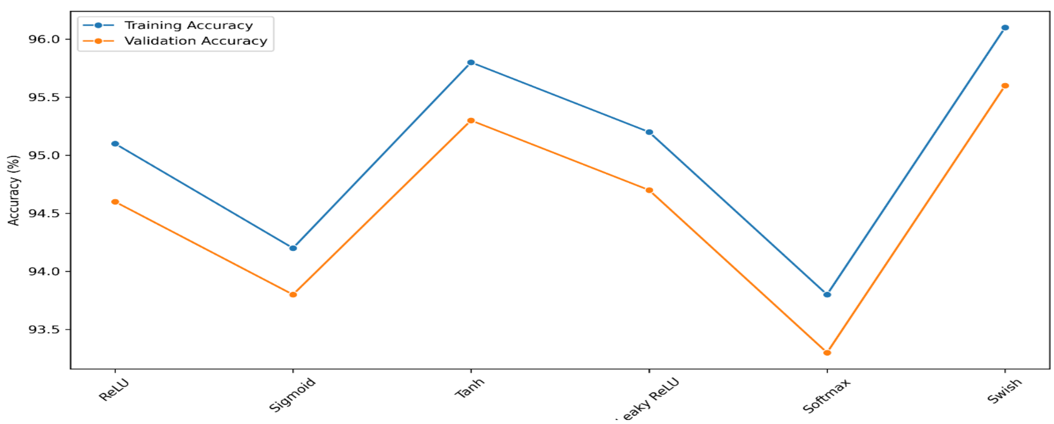

Figure. 1 Training and validation accuracy for the tomato dataset

and validation accuracy before and after data augmentations. Figures 2 and 5 show the training and validation loss comparisons. Figures 3 and 6 show the ROC analysis before and after augmentations. Table 6 and Figure 7 show the performance analysis based on accuracy and error comparisons with state-of-the-art methods. The best performance was obtained with the Swish activation function with a bias range of [0,1] and a weight range of [-1,1] without data augmentation, according to which, an accuracy of 95.62%, the precision of 95.60%, the recall of 94.76, and the F-measure of 95.32 were obtained, and the minimal false acceptance rate (FAR) was achieved (4.11%). In relative terms, the traditional ReLU and Softmax functionalities showed decreased results, whereby Softmax had the lowest accuracy result of 93.31 which implies that Swish increased accuracy by a margin of 2.31 and 0.84 over Softmax and ReLU, respectively. Generally, Swish and Tanh also showed stronger generalization on average across all combinations than other activation functions, even though it is admitted that this is due to a smaller proportion of generalization increasing values than with other activation functions. On carrying out data augmentation, the performance of the model improved remarkably in terms of the metrics. With the Swish activation function, a bias of [0, 1], it obtained the highest accuracy of 96.68%, an accuracy increase of 1.06 percent compared with its performance without data augmentation. On the same note, augmentation benefited the ReLU function as well by increasing by 1.61 percent (94.78% to 96.39%). Interestingly, Softmax was still the worst performing in both of these settings, but it also had an accuracy increase of around 1.97%, which also implies that data augmentation indeed makes all activation functions generalize better on unseen data. It is worthwhile to note that Swish had the greatest advantage in F-measure (95.32% to 95.25%) and had the lowest FAR of 4.11 percent without augmentation and 5.19 percent with augmentation, proving that Swish can lower false predictions even in a wide range of data variance. Comparing the average of all combinations of parameters, Swish performed best with a mean accuracy of about 95.56%, Tanh fell second with 94.84%, and Softmax was about 93.72%. The swish activation function also showed a consistent performance with the rest being the best in terms of precision, recall, and F-measure, showing that it can be relied upon to conserve features related to classes that are important in classifying plant village leaves that are diseased. The optimized CNN using BCO produced a significant change of 1.8-3 % accuracy and a reduction of overall FAR of more than 2 % with respect to normal CNN parameters that were not optimized. In general, the addition of BCO in the CNN structure effected a huge boost in the learning performance as well as classification in the detection of planet village leaf disease. Data augmentation further enhanced these advantages as it increased the aspects of model generalization, especially in situations in which Swish and Leaky ReLU activation functions were used. These results confirm the fact that BCO-based optimization would not only tune weights and biases towards more convergence but also

Table 6. Performance analysis after data augmented

Bias	Weight	Activation function	Accuracy	Precision	Recall	F-measure	FAR
		ReLU	96.39	91.72	91.36	93.31	6.16
		Sigmoid	95.85	95.88	94.11	92.89	5.18

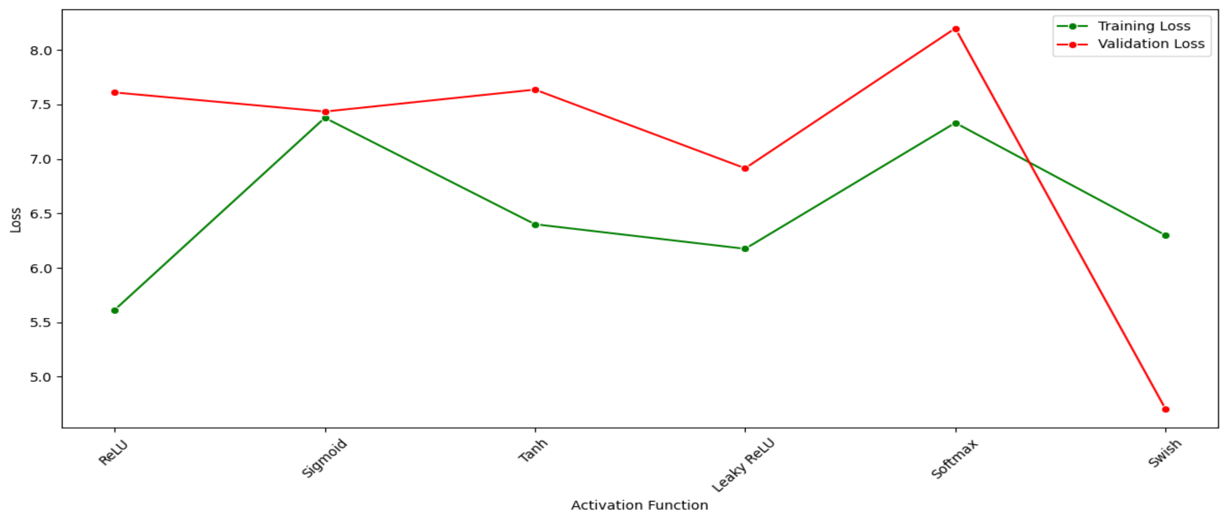

Figure. 2 Training and validation loss of the tomato dataset

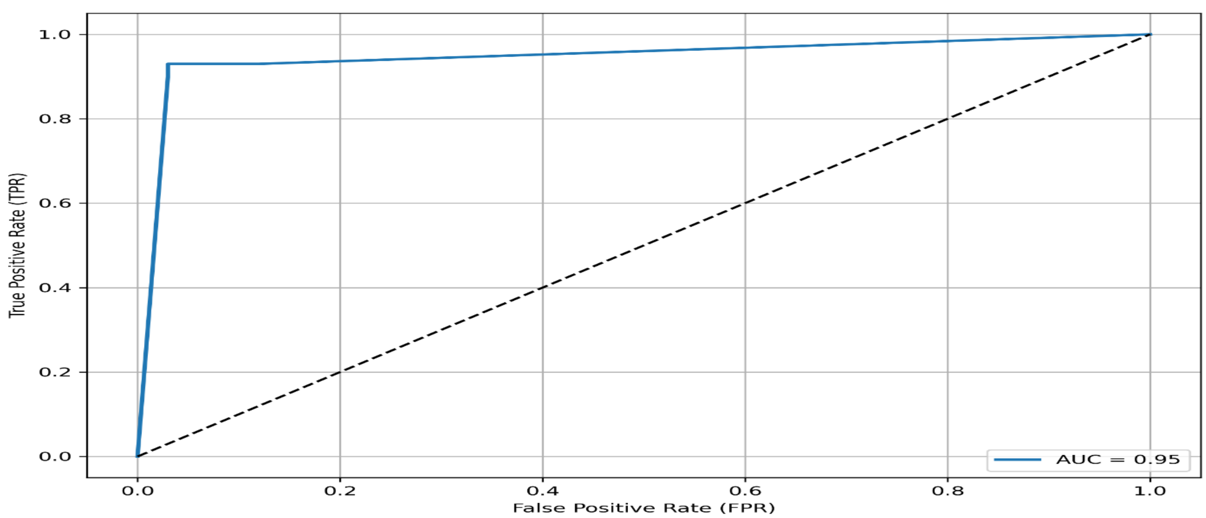

Figure. 3 ROC curve for the tomato dataset

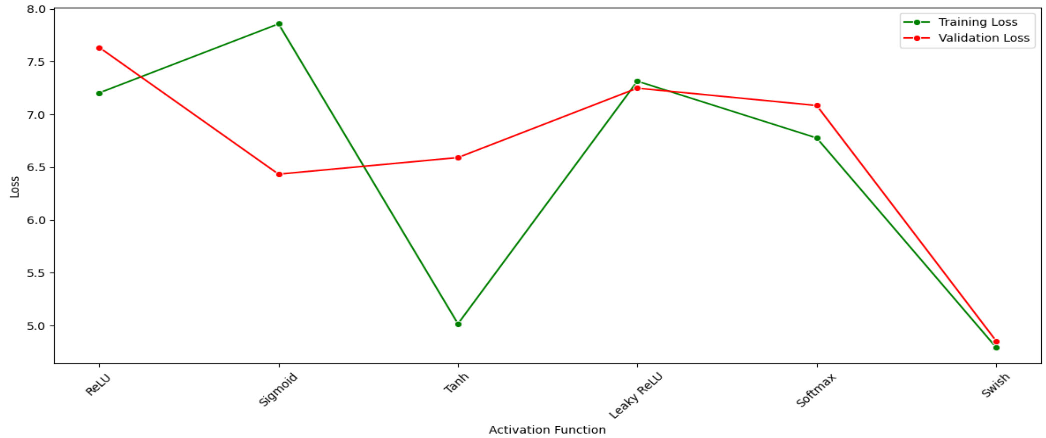

Figure. 5 Training and validation loss on the tomato dataset after data augmentation

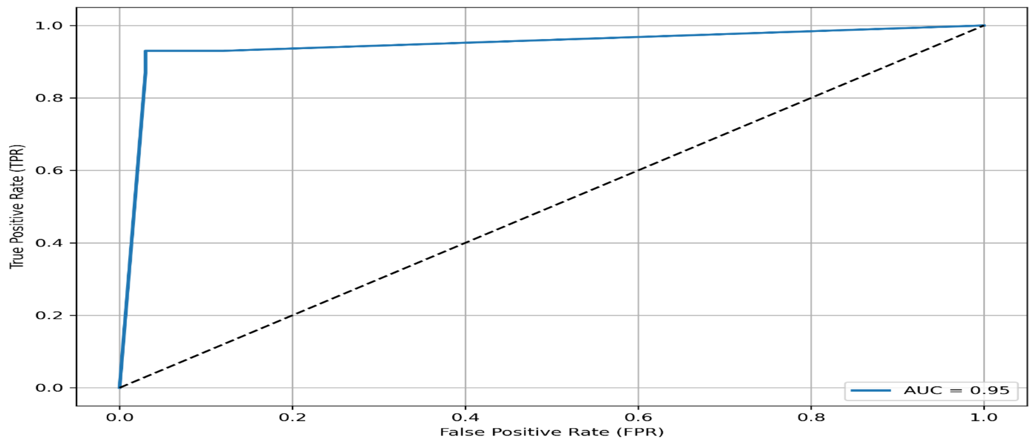

Figure. 6 ROC curve for the tomato dataset after data augmentation

align the architecture towards providing superior results in real-world agricultural image classification applications. The bar chart shown provides a user-friendly graph comparison of the two types of performance variables of Accuracy (%) and FAR (%), of the six methods of planet village leaf disease detection. All methods employ variations of the CNN model, with some of them further augmented with an optimization algorithm or combination models. The result can be visually checked with the help of the two bars representing each of the methods and judging at once about their predictive success and reliability. The highest accuracy of 96.68% and the lowest FAR of 4.05% have been the most commendable characteristics of the suggested technique that combines CNN and BCO. It implies that BCO can well adjust the weights and biases of CNN, enabling more accurate feature learning and classification. Turning on Swish activation functions and application of data augmentation also enhance the capacity of the model to generalize the training data. On the contrary, the bare CNN model containing no optimization yields poor performance, 91.87% accuracy with a high FAR (8.12%). This shows clearly how unoptimized the networks could not handle the intricacies of disease classification in the leaf images. CNN-based methods are computationally expensive, need a huge number of annotated datasets, and are susceptible to overfitting small or imbalanced data. Next on the scale of optimization is the GA-optimized CNN with 95.31 % accuracy and 5.44 % FAR, next to the PSO-based CNN, which recorded 94.22 % accuracy and 5.78% FAR. GA-CNN is more effective in improving parameter tuning, yet it is both resource-consuming and premature. PSO-CNN optimizes better but is more complex, and it has the dangers of slow convergence or local optima. Likewise, GA-CNN is more effective in improving parameter tuning, yet it is both resource-consuming and premature. In spite of the fact that both perform decently well, BCO also outperforms the other two, indicating the competitive advantages of swarm-intelligence-based bacteria modeling. Other hybrid algorithms, such as CNN + SVM and transfer learning with VGG16, also manage to achieve moderate results of 92.45 % and 93.57 % respectively. They, however, have larger FAR values than the optimization-based CNNs, and this means lesser reliability.

Table 7. Performance comparisons with state-of-the-art methods

Study	Optimizer	Accuracy
Present	BCO	96.68
CNN [15]	Adam	91.87
PSO-CNN[31]	PSO	94.22
GAN+AE[11]	-	95.74
CNN + SVM[32]	Manual	92.45

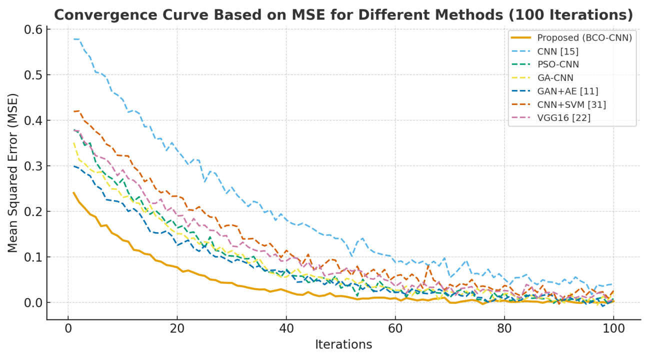

Figure. 7 Error comparison of detection methods

VGG16[22]	Manual	93.57
ANN[33]	-	89

A combination of CNN and SVM tends to enhance classification but complicate the model and have issues with high-dimensional features. Although highly accurate, VGG16 is very deep and computationally intensive and is not as applicable in small datasets and the real-time plant disease detection. Broadly, the bar chart illustrates eloquently how optimization, particularly optimization with BCO, plays a crucial role in increasing the efficiency of the CNN model in detecting the plant village leaf diseases. The lower FAR and higher accuracy set BCO-CNN as the best and strongest of the compared techniques in a clear way.

The convergence character of the various optimization-based CNN models based on MSE after 100 iterations is shown in Figure 7. The proposed BCO-optimized CNN is the fastest and smoothest converging approach than other ones. Every model begins with fairly high values of MSE but after the first 30-40 steps, the error in the BCO-based model decreases, and the error level holds steady to below 0.02. This means that it learns efficiently and can tune its parameters better, due to the good exploration and exploitation trade-off of BCO, and the dynamics of its adaptive population. Conversely, GA and PSO have slow convergence and high oscillations, indicating premature convergence and weak stability. Furthermore, the BCO-CNN curve is always lower than the others during the training process, and this is an indication of higher robustness and less predisposition to overfitting. The models that are optimized by using GA and ACO achieve satisfactory performance after 60-70 iterations, whereas the traditional CNN and VGG16 demonstrate a greater residual error because of the unchanged hyperparameters and the absence of adaptive optimization. On balance, the convergence curve empirically supports the fact that the BCO-optimized CNN is faster convergent, has lower training error, and is more computationally efficient than traditional optimization methods, which confirms its usefulness in plant disease detection.

6 Conclusions

In the present work, a CNN model was optimized with BCO, to improve disease detection of planet village leaves. With the integration of BCO, which is a metaheuristic algorithm based on the behavior of bacterial colonies, the model successfully optimized the parameters of CNN, leading to better convergence, robustness, and classification behavior. The evaluation of the experimental results obtained on the PlanetVillage dataset with and without data augmentation has shown that the BCO-optimized CNN always gave better results. The best performance was obtained by Swish activation an accuracy of 95.62% without augmentation and 96.68 % with augmentation. In addition, the model resulted in upsurges in precision, recall, F-measure, and the noticeable decrease in the FAR demonstrated the efficiency of the BCO strategy to the alternative capabilities of feature learning and classification quality. The conclusion proves the thesis BCO, not only speeds up convergence, but also leads to improved generalization, especially when combined with data augmentation. The proposed solution in this research is scalable and intelligent, as it is now possible to transfer it to other crops and types of disease as a prevention measure and to help precision farming.

References

[1] A. Upadhyay *et al.*, "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," *Artificial Intelligence Review,* vol. 58, no. 3, p. 92, 2025.

[2] A. Das, F. Pathan, J. R. Jim, M. M. Kabir, and M. Mridha, "Deep learning-based classification, detection, and segmentation of tomato leaf diseases: A state-of-the-art review," *Artificial Intelligence in Agriculture,* 2025.

[3] S. Janarthan, S. Thuseethan, S. Rajasegarar, and J. Yearwood, "Family-based plant disease characterization using deep neural networks," *Multimedia Tools and Applications,* pp. 1-23, 2025.

[4] R. Sujatha, S. Krishnan, J. M. Chatterjee, and A. H. Gandomi, "Advancing plant leaf disease detection integrating machine learning and deep learning," *Scientific Reports,* vol. 15, no. 1, p. 11552, 2025.

[5] W. Hussain *et al.*, "Ensemble genetic and CNN model-based image classification by enhancing hyperparameter tuning," *Scientific Reports,* vol. 15, no. 1, p. 1003, 2025.

[6] A. Kaissar, A. B. Nassif, B. Soudan, and M. Injadat, "Enhancing CNN-based network intrusion detection through hyperparameter optimization," *Intelligent Systems with Applications,* vol. 26, p. 200528, 2025.

[7] B. Sivasakthi, K. Preetha, and D. Selvanayagi, "Osteoporosis disease detection using optimized Elman recurrent neural network based on hybrid bacterial colony optimization and tabu search algorithm," *International Research Journal of Multidisciplinary Technovation,* vol. 7, no. 1, pp. 1-16, 2025.

[8] B. Sivasakthi and D. Selvanayagi, "Prediction of osteoporosis disease using enhanced Elman recurrent neural network with bacterial colony optimization," in *Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2022*: Springer, 2023, pp. 211-220.

[9] M. Arunadevi and V. Sathya, "DDoS Attack Detection using Back Propagation Neural Network Optimized by Bacterial Colony Optimization," *International Journal of Intelligent Engineering & Systems,* vol. 16, no. 5, 2023.

[10] M. Asghar *et al.*, "A lightweight hybrid model for scalable and robust plant leaf disease classification," *Scientific Reports,* vol. 15, no. 1, p. 32353, 2025.

[11] S. Gupta, A. K. Tripathi, and N. Lewis, "Pre-trained noise based unsupervised GAN for fruit disease classification in imbalanced datasets," *Pattern Analysis and Applications,* vol. 28, no. 2, p. 39, 2025.

[12] E. Gangadevi, B. O. Soufiane, B. Balusamy, F. Khan, and M. Getahun, "A novel hybrid fruit fly and simulated annealing optimized faster R-CNN for detection and classification of tomato plant leaf diseases," *Scientific Reports,* vol. 15, no. 1, p. 16571, 2025.

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

[14] D. S. Joseph, P. M. Pawar, and K. Chakradeo, "Real-time plant disease dataset development and detection of plant disease using deep learning," *Ieee Access,* vol. 12, pp. 16310-16333, 2024.

[15] G. Shrestha, M. Das, and N. Dey, "Plant disease detection using CNN," in *2020 IEEE applied signal processing conference (ASPCON)*, 2020: IEEE, pp. 109-113.

[16] Y. M. Abd Algani, O. J. M. Caro, L. M. R. Bravo, C. Kaur, M. S. Al Ansari, and B. K. Bala, "Leaf disease identification and classification using optimized deep learning," *Measurement: Sensors,* vol. 25, p. 100643, 2023.

[17] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," *Information Processing in Agriculture,* vol. 7, no. 4, pp. 566-574, 2020.

[18] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of plant-leaf diseases using CNN and transfer-learning approach," *Electronics,* vol. 10, no. 12, p. 1388, 2021.

[19] J. Eunice, D. E. Popescu, M. K. Chowdary, and J. Hemanth, "Deep learning-based leaf disease detection in crops using images for agricultural applications," *Agronomy,* vol. 12, no. 10, p. 2395, 2022.

[20] S. T. Y. Ramadan, M. S. Islam, T. Sakib, N. Sharmin, M. M. Rahman, and M. M. Rahman, "Image-based rice leaf disease detection using CNN and generative adversarial network," *Neural Computing and Applications,* vol. 37, no. 1, pp. 439-456, 2025.

[21] C. Wang, Y. Xia, L. Xia, Q. Wang, and L. Gu, "Dual discriminator GAN-based synthetic crop disease image generation for precise crop disease identification," *Plant Methods,* vol. 21, no. 1, p. 46, 2025.

[22] A. A. Alatawi, S. M. Alomani, N. I. Alhawiti, and M. Ayaz, "Plant disease detection using AI based VGG-16 model," *International Journal of Advanced Computer Science and Applications,* vol. 13, no. 4, 2022.

[23] C. Zhang, S. Wang, C. Wang, H. Wang, Y. Du, and Z. Zong, "Research on a Potato Leaf Disease Diagnosis System Based on Deep Learning," *Agriculture,* vol. 15, no. 4, p. 424, 2025.

[24] P. Hari and M. P. Singh, "Adaptive knowledge transfer using federated deep learning for plant disease detection," *Computers and Electronics in Agriculture,* vol. 229, p. 109720, 2025.

[25] I. J. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems,* vol. 27, 2014.

[26] B. Niu and H. Wang, "Bacterial colony optimization," *Discrete Dynamics in Nature and Society,* vol. 2012, 2012.

[27] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE signal processing magazine,* vol. 35, no. 1, pp. 53-65, 2018.

[28] A. Biswas *et al.*, "Generative adversarial networks for data augmentation," in *Data Driven Approaches on Medical Imaging*: Springer, 2023, pp. 159-177.

[29] Z. ur Rahman, M. S. M. Asaari, H. Ibrahim, I. S. Z. Abidin, and M. K. Ishak, "Generative adversarial networks (GANs) for image augmentation in farming: A review," *IEEE Access,* 2024.

[30] G. Han, D. K. P. Asiedu, and K. E. Bennin, "Plant disease detection with generative adversarial networks," *Heliyon,* vol. 11, no. 7, 2025.

[31] A. Elaraby, W. Hamdy, and M. Alruwaili, "Optimization of deep learning model for plant disease detection using particle swarm optimizer," *Computers, Materials & Continua,* vol. 71, no. 2, 2022.

[32] S. Kalaivani, C. Tharini, T. S. Viswa, K. F. Sara, and S. Abinaya, "ResNet-based classification for leaf disease detection," *Journal of The Institution of Engineers (India): Series B,* vol. 106, no. 1, pp. 1-14, 2025.

[33] T. N. Pham, L. Van Tran, and S. V. T. Dao, "Early disease classification of mango leaves using feed-

forward neural network and hybrid metaheuristic feature selection," *IEEE access,* vol. 8, pp. 189960-189973, 2020.