Optimizing deep learning models for plant leaf disease classification using nature-inspired algorithms

Avinesh Culloo, Avinash Bhunjun, Geerish Suddul
Department of Business Informatics and Software Engineering, School of Innovative Technologies and Engineering, University of Technology, Mauritius (UTM), Port Louis, Mauritius

Article Info

Article history:

Received Apr 27, 2026
Revised Jul 2, 2026
Accepted Jul 13, 2026

Keywords:

Computer vision
Leaf disease classification
Nature-inspired algorithm
Optimization
Transfer learning

ABSTRACT

Plant diseases greatly affect agricultural production, especially in developing countries, where prompt diagnosis can be quite challenging due to the limited availability of experts in real-time. Deep learning techniques for image analysis is gaining popularity and are increasingly considered an alternative to traditional manual inspection of plants. This research presents the evaluation of plant leaf disease detection system based on a convolutional neural network (CNN) optimized with different nature-inspired algorithms. The backbone model is based on the EfficientNet-B0 pretrained on ImageNet. Therefore, transfer learning is used to adapt the model to an updated PlantVillage dataset. Experiments have been conducted with multiple nature-inspired algorithms to improve generalisation and training efficiency of the prediction model. Different data preparation techniques have been carefully applied to the dataset, creating a unified approach to ensure consistency in the preprocessing pipeline for the training, validation, and testing phases. Our experiments indicate that application of the grey wolf optimizer (GWO) for tuning key hyperparameters of the model, including dropout, learning rates, and weight decay produced the best results, with an accuracy around 99.45%.

Corresponding Author:

Geerish Suddul
Department of Business Informatics and Software Engineering, School of Innovative Technologies
University of Technology, Mauritius (UTM)
La Tour Koenig, Pointe-aux-Sables, Port Louis, Mauritius
Email: g.suddul@utm.ac.mu

1. INTRODUCTION

Agriculture remains a vital sector for ensuring food security and economic sustainability, particularly in developing and island nations such as Mauritius. Improving agricultural productivity and crop health is therefore essential to support global initiatives such as the United Nations Sustainable Development Goal 2, which focuses on ensuring food security, encouraging sustainable agriculture and ending hunger [1]. However, crop production is frequently affected by plant diseases, which can significantly reduce yield and quality if not identified at an early stage. In many agricultural settings, disease diagnosis relies on manual examination of the plant by farmers or experts in agriculture. This process is often time consuming, subjective, and not always accessible to smallholder farmers, especially in remote or resource-limited environments [2], [3].

With recent advances in artificial intelligence, computer vision has emerged as a promising tool for supporting agricultural decision making. Image-based disease detection systems allow plant health to be assessed directly from leaf images, minimising the need for manual inspection. Convolutional neural networks (CNNs) demonstrate the ability to automatically learn relevant features from raw image data indicating its strong performance in computer vision tasks. Previous studies show that CNN-based models can effectively identify disease symptoms such as discoloration, lesions, and irregular leaf textures making them a suitable approach for the plant leaf disease detection [4]–[9].

Although CNN-based models have demonstrated strong performance, several challenges remain. Deep learning models require large amounts of labeled data for effective training and may be prone to overfitting, particularly when trained on datasets under controlled conditions. To mitigate these limitations, techniques such as transfer learning and optimisation approaches have been utilised to enhance model generalization and training efficiency. Additionally, data preparation and preprocessing are essential for computer vision systems, as these steps play a critical role in determining overall model performance [10].

The PlantVillage dataset [11] is commonly used in digital agriculture for the automation of plant disease prediction consisting of 14 crop categories and 26 diseases. We focused on a subset of the dataset, known as the 'plant village dataset (updated)' from Kaggle [12], comprising of nine main crop species, namely apple, bell pepper, cherry, grape, peach, strawberry, tomato, potato, and corn (maize). Each crop category contains multiple classes representing either specific plant diseases or healthy plant conditions. The problem is therefore a classification task with multiple classes, where each image is assigned to its corresponding class label such as tomoto_bacterial_spot, tomato_early_blight, tomato_late_blight, and tomato_healthy.

Our proposed method combines a convolutional neural network with optimisation techniques inspired by natural processes to enhance model performance. Additionally, standardized preprocessing and evaluation strategies are employed to assess classification accuracy and robustness. Standard performance metrics, along with comparative experiments, are adopted to analyse the effectiveness of the proposed approach. The primary contributions of this work are firstly the application of metaheuristic algorithms for hyperparameter tuning of the model and secondly, a comprehensive evaluation and comparison of the proposed approach using benchmark datasets.

The rest of this paper is organised as: section 2 shows the methods used in this study, including the CNN architecture, optimisation strategy, dataset preparation, and evaluation metrics. The experimental results and discussion are presented in section 3. Section 4, concludes with the paper and outlines directions for future work.

2. METHOD

This research introduces a deep learning framework for the categorisation of plant diseases, combining a CNN with nature-inspired optimisation algorithms. The objective is to find out which nature-inspired algorithm performs best for improving deep learning models that are used to identify plant diseases in images. These algorithms are specifically utilised to enhance hyperparameter tuning and model generalization [13], [14]. The methodology is divided into four main parts, as depicted in Figure 1. First a CNN is used as the main model for image classification. Second, a set of algorithms based on nature is used to find the best values for the most important hyperparameters of the deep learning model. Third, the steps for getting the dataset ready with specific pre-processing steps are performed to ensure the data is strong and consistent during training. Finally, we use standard performance evaluation metrics to test and compare the suggested optimisation strategies.

2.1. Convolutional neural network

CNNs are used frequently for image classification tasks given their capacity to independently learn spatial and structural features from raw image data [15]. Traditional machine learning methods depend on features that are designed manually, but CNNs do both feature extraction and classification from start to finish. This makes them well suited for plant disease identification, where visual symptoms such as spots, lesions, and color variations appear at different spatial scales on leaf images. A CNN usually consists of convolutional layers, activation functions, pooling layers, and fully connected layers [16]. In a convolutional layer, learnable filters are applied to the input image to retrieve local patterns. The convolution operation can be expressed as:

$$z^{\{(k)\}} = w^{\{(k)\}} * x + b^{\{(k)\}} \tag{1}$$

where x denotes the input image, $w^{(k)}$ represents the k-th convolutional filter, $b^{(k)}$ is the bias term, and $z^{(k)}$ denotes the resulting feature map produced by the convolution operation. The symbol (*) represents the convolution operator. An activation function is used after convolution to make the network non-linear. This study uses the rectified linear unit (ReLU) because it is simple and works well in deep neural networks [17]. It is defined as:

$$f(z) = \max(0, z) \quad (2)$$

As the network depth increases, convolutional and pooling layers progressively extract higher-level feature representations. After that, these features are compressed and sent to fully connected layers for classification. The final output layer applies the softmax function to transform the network outputs into class probabilities, ensuring that the predicted probabilities across all disease classes sum to one [16]. This enables multi-class plant disease classification by selecting the class with the highest predicted probability.

EfficientNet is a group of CNN architectures that uses a compound scaling strategy to scale the depth, width, and input resolution of the network in a balanced way [18]. This design enables EfficientNet models to achieve strong classification performance while maintaining computational efficiency. Among the different variants, EfficientNet-B0 represents the baseline architecture and provides an effective trade-off between accuracy and model complexity, making it suitable for plant disease classification tasks using images. In this study, EfficientNet-B0 pretrained on the ImageNet dataset [19] is employed as the CNN backbone as it has a good balance between classification accuracy and computational efficiency. Transfer learning is used by, firstly, freezing the pretrained layers and training the classifier head to make the model fit the target dataset. Subsequently, all layers are unfrozen and fine-tuned using a smaller learning rate to further improve performance and generalisation on plant disease images. Pooling layers are often used in CNN architectures to keep important information while lowering the spatial resolution of feature maps. This operation improves the computing and makes it less likely that translations will change the results. Pooling operations are built into the network blocks in EfficientNet, which helps with effective feature abstraction. During training, the CNN's parameters are learned by minimising a classification loss function through gradient-based optimisation. This study employs cross-entropy loss alongside backpropagation to update the network weights, facilitating the acquisition of discriminative features by the model, for better classification.

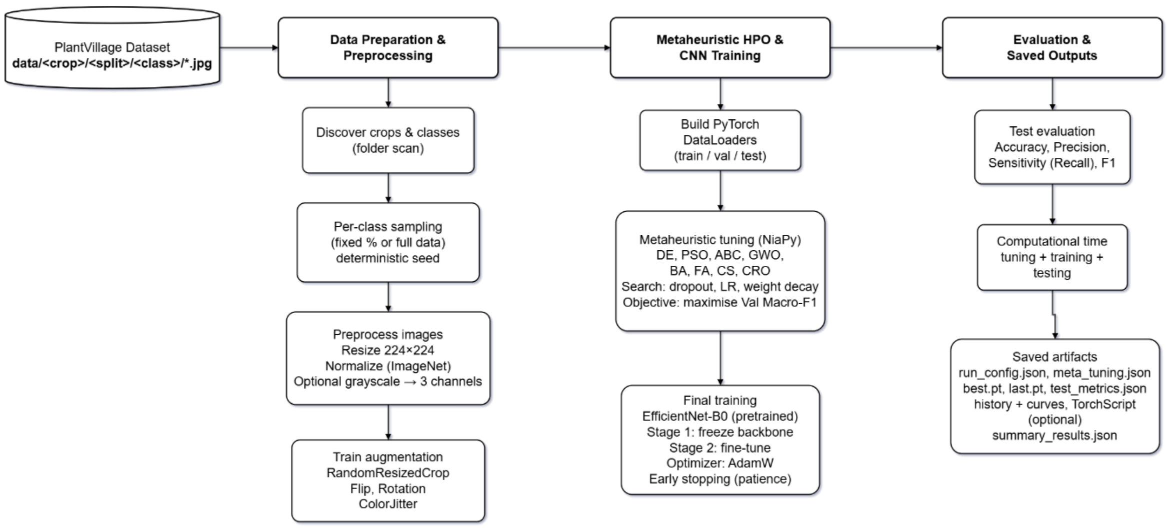

Figure 1. Methodology for plant disease detection using nature inspired algorithm

2.2. Nature inspired optimisation algorithms

The impact of metaheuristic optimization approach on CNN performance is assessed using a diverse set of nature-inspired algorithms, encompassing both evolutionary and swarm intelligence approaches. Each algorithm differs in its exploration-exploitation strategies and search dynamics, enabling a thorough comparison of their effectiveness in hyperparameters for plant disease classification. The selected nature-inspired algorithms are:

- Differential evolution (DE) [20]
- Particle swarm optimization (PSO) [21]
- Grey wolf optimizer (GWO) [22]
- Bat algorithm (BA) [23]
- Firefly algorithm (FA) [24]
- Cuckoo search (CS) [25]
- Coral reef optimization (CRO) [26]
- Artificial bee colony (ABC) [27]

DE and PSO are often utilized to solve problems that need to be optimized, such as tuning the learning rate and weight decay in deep learning. GWO is well known for maintaining a balance between exploration and exploitation. Other approaches, including BA, FA, and CS, are capable of preventing premature convergence and reduce the risk of being trapped in the local optima. CRO promotes competitive reproduction mechanisms by increasing population diversity. Finally, ABC is a bio-inspired, multi-agent probabilistic search algorithm that employs a metaphor based on the foraging intelligence of honeybee swarms. The metaheuristic algorithms were implemented using the NiaPy framework and combined with the EfficientNet-B0 model for the hyperparameter tuning. The comparative performance of each algorithm is analysed in section 3.

2.3. Dataset

The plant village updated dataset consists of a mixture of healthy and diseased leaves across nine different crops for a total of 67,118 high quality plant leaves, as listed in Table 1. The dataset is organized in such a way that each crop is stored in its own folder and then split into train, test, and validation sets. Within each split, the leaf pictures are classified into different folders named after the disease with an additional folder containing the healthy leaves. The folders are arranged to allow the loading of the dataset in a standard way which is practical for the machine learning. Figure 2 shows an example of tomato sample images across six classes in the training split which the model needs to train from.

Table 1. Distribution of plant disease images across different crop categories with split ratio

Crop	Train	Validation	Test	Total	Split ratio (train/val/test)
Apple	7,771	1,747	196	9,714	80.00%/17.98%/2.02%
Bell Pepper	3,901	877	98	4,876	80.00%/17.99%/2.01%
Cherry	3,509	788	89	4,386	80.00%/17.97%/2.03%
Corn (Maize)	7,316	1,645	188	9,149	79.97%/17.98%/2.05%
Grape	7,222	1,623	182	9,027	80.00%/17.98%/2.02%
Peach	3,566	801	90	4,457	80.01%/17.97%/2.02%
Potato	5,702	1,282	144	7,128	79.99%/17.99%/2.02%
Strawberry	3,598	809	91	4,498	79.99%/17.99%/2.02%
Tomato	11,108	2,495	280	13,883	80.01%/17.97%/2.02%
Total	53,693	12,067	1,358	67,118	

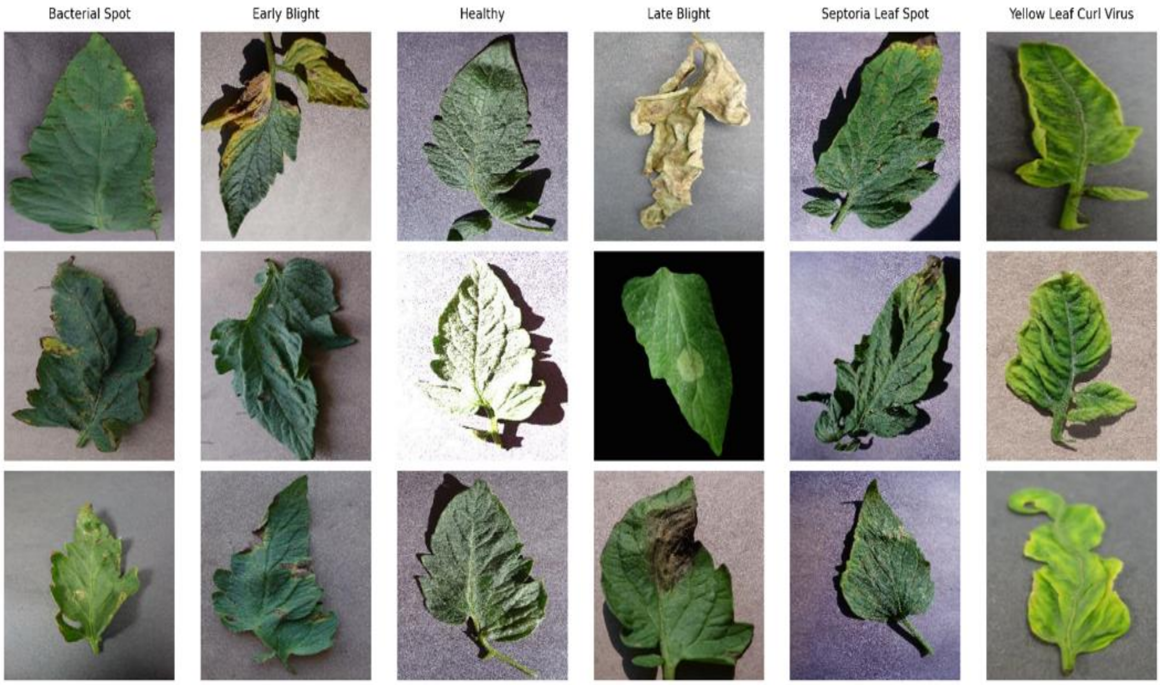

Figure 2. Visual differences of the training samples for tomato leaves across six different classes (starting from left: bacterial spot, early blight, healthy, late blight, septoria leaf spot and yellow leaf curl virus)

2.4. Data preprocessing

The loading of the data is standardised so that the model always uses the same type of input. The data preprocessing pipeline is kept simple and consistent for the model testing and validation. Each image input to the computer vision model is first converted to a 3-channel grayscaled image then transformed into tensors and the ImageNet mean and standard deviation is used for normalising the input. This preprocessing step is fundamental to keeping the training stable since the EfficientNet-B0 backbone were trained on ImageNet.

For training of the model, the same core steps from the data preprocessing are applied with additional image augmentation to improve generalisation. Instead of using a fixed resize, the image is randomly resized around 224 pixels and randomly cropped with a lower bound of 0.8 and upper bound of 1.0. This allows the model to learn small changes in the image frame while keeping part of the image visible. In addition, the image is randomly horizontally flipped with a probability of 50% and randomly rotated with a range of degrees between -15 and +15. On top of these random flipping and rotation, to mimic natural variations in lighting and camera conditions, the brightness and the contrast are each jittered by 0.15. Since in the data preprocessing step, the image is converted to grayscale, the jitter is limited to contrast and brightness only to be consistent. Through these preprocessing and data augmentation, the variety of training examples have been used, without changing the underlying classes. These data preprocessing techniques are expected to reduce the overfitting and also improving performance of the model on unseen test images [28].

2.5. Performance metrics

To evaluate the performance of the model, metrics are used which are described as:

– Accuracy: the performance metric that measures the proportion of correct predictions the model makes across the entire test set.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3}$$

– Precision: the proportion of cases the model labels as positive that are actually positive. It shows how well the model reduces false positives (that is, avoids false alarms).

$$Precision = \frac{TP}{(TP + FP)} \tag{4}$$

– Recall/sensitivity (true positive rate): the percentage of real positives that have been identified correctly. It shows the extent to which the model finds all the right cases and doesn't miss any (false negatives).

$$Recall = \frac{TP}{(TP + FN)} \tag{5}$$

– F1-score: the average of precision and recall. It is especially helpful when class frequencies are not perfectly balanced because it rewards models that find a good balance between precision and recall instead of just optimising one of them.

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{6}$$

where TP is true positive, TN is true negative, FN is false negative, and FP is false positive.

3. RESULTS AND DISCUSSION

Table 2 presents the summary of the results for all the combinations of metaheuristic with CNN experiments performed in this study. The same EfficientNet-B0 (pretrained) backbone was used throughout and the metaheuristics were applied to tune key training hyperparameters (dropout, learning rates, and weight decay) using validation Macro-F1 as the objective. Overall, all approaches achieved very strong performance (above 99% across the main metrics). This demonstrates that transfer learning with EfficientNet-B0 is highly effective for this PlantVillage classification setup.

Firstly, DE produced a strong baseline with 99.21% accuracy and a 99.21% Macro-F1 score, completing in approximately 32.97 minutes. Building on this, PSO delivered a noticeable improvement, reaching 99.37% accuracy and 99.36% Macro-F1, while also being one of the most time-efficient runs at around 29.15 minutes. Furthermore, the GWO achieved the best overall results in Table 2, recording the highest accuracy (99.45%) and Macro-F1 (99.46%), which indicates the most consistent performance across classes, with a total runtime of roughly 31.19 minutes.

Moreover, both the BA and CS achieved very similar mid-range performance (about 99.29% accuracy and 99.28% Macro-F1), although BA was slightly faster overall (29.93 minutes) compared with CS (32.29 minutes). The ABC delivered results comparable to the DE baseline (around 99.21% accuracy and 99.20% Macro-F1) with a total time of 30.35 minutes. In contrast, the FA produced the lowest Macro-F1 in Table 2 (99.18%) and required a longer total runtime (34.15 minutes), suggesting that under the same tuning budget it was less effective at locating the best hyperparameter region.

Finally, the CRO approach achieved performance that was very close to PSO (99.36% Macro-F1) but at a substantially higher computational cost (45.66 minutes), making it a less practical choice when runtime is a concern. In summary, GWO provided the best overall predictive performance in this study, while PSO offered a very competitive alternative with excellent results and the most efficient runtime among the top-performing methods.

Table 2. Performance comparison of EfficientNet-B0 combined with different nature-inspired optimisation algorithms

Model	Computational (min)	Accuracy	Precision (Positive predicted value)	Recall/sensitivity (True positive rate)	F1-Score
EfficientNet-B0+DE	32.97	0.9921	0.9924	0.9920	0.9921
EfficientNet-B0+PSO	29.15	0.9937	0.9940	0.9934	0.9936
EfficientNet-B0+GWO	31.19	0.9945	0.9949	0.9944	0.9946
EfficientNet-B0+BA	29.93	0.9929	0.9929	0.9927	0.9928
EfficientNet-B0+FA	34.15	0.9921	0.9925	0.9915	0.9918
EfficientNet-B0+CS	32.29	0.9929	0.9929	0.9927	0.9928
EfficientNet-B0+CRO	45.66	0.9937	0.9938	0.9935	0.9936
EfficientNet-B0+ABC	30.35	0.9921	0.9921	0.9921	0.9920

3.1. Comparative analysis

According to the results in Table 3, our suggested EfficientNet-B0 model with metaheuristic tuning GWO has the best overall performance, with a test accuracy of 99.45%. This shows that the selected model works better than other methods for detecting plant diseases from images when it comes to classifying them.

Table 3. Comparative analysis of existing models

Reference	Architecture	Accuracy metric (%)
Sladojevic *et al.* [4]	CaffeNet (Modified AlexNet)	96.30
Mohanty *et al.* [5]	GoogleLeNet	99.35
Picon *et al.* [6]	Deep residual neural network (Modified ResNet-50)	96.0
Zhang *et al.* [7]	GPDCNN (Modified AlexNet)	94.65
Fuentes *et al.* [8]	Faster R-CNN (with VGG-16)	83.0 (mAP)
Our hybrid model	EfficientNet-B0+GWO	99.45

4. CONCLUSION

This study shows that nature-inspired optimization algorithms have strong potential to support plant disease detection. This in turn can assist in improving crop health and productivity in Small Island Development States. By enabling faster and more accessible diagnosis from images without always relying on expert inspection, such systems can help farmers respond earlier, reduce unnecessary losses, and potentially improve profitability. In a wider context, this direction aligns well with UN SDG 2 (Zero Hunger) by contributing to more resilient and efficient agricultural practices.

The most effective configuration amongst all evaluated approaches was the EfficientNet-B0 paired with the GWO to tune key hyperparameters, including dropout, learning rates, and weight decay. The test performance achieved is a strong indication that using metaheuristic-driven tuning approaches can significantly improve model generalization. In addition to using nature inspired algorithm for hyperparameter fine tuning, the preprocessing strategy with image resizing, flipping, rotating, grayscaling and jittering improved the robustness of the model to variations in viewpoint and leaf orientation. This better reflects realworld conditions where imaging environments are difficult to control.

To further advance the plant disease classification model, future works must focus on improving the robustness of the model against a broad range of crops, disease types, as well as under diverse imaging conditions such as cluttered background, motion blur and varying camera quality. Currently, the model was trained on a dataset with a limited nine plant species, which can be expanded further to include more species. Also, more diverse field images will make the model more dependable in real-world settings. Moreover, further research can also explore deployment strategies, particularly lightweight inferences for mobile devices. Clear user documentation and guidelines must also be developed to build trust and support adoption.

FUNDING INFORMATION
Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT
This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Avinesh Culloo	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	
Avinash Bhunjun	✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	
Geerish Suddul	✓	✓		✓					✓	✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT
Authors state no conflict of interest.

DATA AVAILABILITY
Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] United Nations, *Transforming our world: the 2030 agenda for sustainable development*, A/RES/70/1. New York, NY: United Nations, 2015. [Online]. Available: https://documents.un.org/doc/undoc/gen/n15/291/89/pdf/n1529189.pdf

[2] FAO, IFAD, UNICEF, WFP, and WHO, *The state of food security and nutrition in the world 2017: building resilience for peace and food security*. Rome: FAO, 2017. [Online]. Available: https://www.fao.org/3/I7695EN/i7695en.pdf

[3] S. Savary, A. Ficke, J. N. Aubertot, and C. Hollier, “Plant diseases and food security: challenges and opportunities,” *Plant Pathology*, vol. 61, no. 1, pp. 3–16, 2012.

[4] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, “Deep neural networks based recognition of plant diseases by leaf image classification,” *Computational Intelligence and Neuroscience*, vol. 2016, no. 1, pp. 1–11, 2016, doi: 10.1155/2016/3289801.

[5] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[6] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, “Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild,” *Computers and Electronics in Agriculture*, vol. 161, pp. 280–290, 2019, doi: 10.1016/j.compag.2018.04.002.

[7] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi, “Cucumber leaf disease identification with global pooling dilated convolutional neural network,” *Computers and Electronics in Agriculture*, vol. 162, pp. 422–430, 2019, doi: 10.1016/j.compag.2019.03.012.

[8] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, “A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition,” *Sensors*, vol. 17, no. 9, p. 2022, 2017, doi: 10.3390/s17092022.

[9] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, “A comparative study of fine-tuning deep learning models for plant disease identification,” *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[10] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[11] D. P. Hughes and M. Salathé, “An open access repository of images on plant health to enable the development of mobile disease diagnostics,” *arXiv:*1511.08060, 2015.

[12] T. Sharma, “Plant village dataset (Updated).” *Kaggle*. Accessed Jan. 27, 2026 [Online]. Available: https://www.kaggle.com/datasets/tushar5harma/plant-village-dataset-updated

[13] G. Suddul, K. Dookhitram, and K. Tameswar, “Optimizing convolutional neural networks with nature-inspired algorithms for diabetic retinopathy classification,” in *Proceedings of International Conference on Advanced Communications and Machine Intelligence*, 2026, pp. 245–255, doi: 10.1007/978-981-96-5589-2_25.

[14] K. Tameswar, G. Suddul, and K. Dookhitram, “A hybrid deep learning approach with genetic and coral reefs metaheuristics for enhanced defect detection in software,” *International Journal of Information Management Data Insights*, vol. 2, no. 2, p. 100105, Nov. 2022, doi: 10.1016/j.jjimei.2022.100105.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems (NeurIPS/NIPS 2012)*, 2012, pp. 1097–1105.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, 1st ed. Cambridge, MA: MIT Press, 2016.

[17] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010, pp. 807–814.
[18] M. Tan and Q. Le, “EfficientNet: rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 6105–6114.
[19] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “ImageNet: a large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
[20] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.
[21] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.
[23] X.-S. Yang, “A new metaheuristic bat-inspired algorithm,” in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 65–74, doi: 10.1007/978-3-642-12538-6_6.
[24] X.-S. Yang, “Firefly algorithms for multimodal optimization,” in *4th International Symposium on Stochastic Algorithms: Foundations and Applications (SAGA 2009)*, 2009, pp. 169–178, doi: 10.1007/978-3-642-04944-6_14.
[25] X.-S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in *2009 World Congress on Nature and Biologically Inspired Computing (NABIC 2009)*, 2009, pp. 210–214, doi: 10.1109/NABIC.2009.5393690.
[26] S. Salcedo-Sanz, J. D. Ser, I. Landa-Torres, S. Gil-López, and J. A. Portilla-Figueras, “The coral reefs optimization algorithm: a novel metaheuristic algorithm for solving optimization problems,” *Scientific World Journal*, vol. 2014, no. 1, p. 739768, 2014, doi: 10.1155/2014/739768.
[27] D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” Technical Report (TR06), 2005. [Online]. Available: https://abc.erciyes.edu.tr/pub/tr06_2005.pdf
[28] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, p. 181, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

BIOGRAPHIES OF AUTHORS

Avinesh Culloo is currently pursuing his master’s degree in the MAIML Programme at the University of Technology, Mauritius, while working as an Application Developer at Dayforce Mauritius. He is certified in Microsoft AI Fundamentals and is also an AWS Certified Cloud Practitioner. His research and project interests include machine learning, deep learning, and generative AI, and he has worked on several projects in these areas. He can be contacted at email: avineshculloo@gmail.com.

Avinash Bhunjun holds a bachelor’s degree in Information Technology from the University of Mauritius and a Postgraduate Diploma in Web and Mobile Application Development from the Open University. He works as an Analyst and Developer in a textile company, specializing in ERP systems and digital solutions. He is currently pursuing an M.Sc. in Artificial Intelligence with Machine Learning at the University of Technology, Mauritius, and has experience working on machine learning projects. He can be contacted at email: avibhunjun@gmail.com.

Geerish Suddul received his Ph.D. from the University of Technology, Mauritius (UTM). He is currently a senior lecturer at the UTM, in the Department of Business Informatics and Software Engineering under the School of Innovative Technologies and Engineering. He has been actively involved in research and teaching since 2005, and currently his research work focuses on different aspects of machine learning such as computer vision and natural language processing. He can be contacted at email: g.suddul@utm.ac.mu.