Tomato Leaf Disease Classification with Optimized Hyperparameter: A DenseNet-PSO Approach

Cynthia Ayu Dwi Lestari[1], Syaiful Anam[2]*, and Umu Sa'adah[3]

[1,2,3] Brawijaya University, Malang, Indonesia
*syaiful@ub.ac.id

Abstract. Tomato stands out as one of the foremost horticultural crops worldwide, including in Indonesia that profoundly influences the agricultural economy. Tomato leaf disease poses a critical and direct threat to quality, yield, and overall production of tomato crops, demanding swift attention and action. Manual detection of tomato leaf diseases not only consumes significant time, effort, and financial resources, but also demands specialized expertise and skills, hampering the urgency and effectiveness of disease control measures. Utilizing CNN, a robust AI approach known for its high precision in image classification, serves as a valuable tool for the automated identification of diseases in tomato leaves in large datasets. DenseNet is a type of CNN architecture that establishes connections between each layer and all subsequent layers, ensuring comprehensive connectivity throughout the network. This approach effectively minimizes the distance of gradient flow during backpropagation, addressing and alleviating issues related to vanishing gradients. Notably, hyperparameter optimization shows significant potential for improving the performance of CNN. In this work, we propose an innovative algorithm driven by Particle Swarm Optimization (PSO) to swiftly and efficiently converge on optimal hyperparameter configurations, including number neuron in fully connected layer, dropout rate, learning rate, activation function and optimizer for DenseNet architecture. Through this optimization, we aim to boost the classification accuracy of the DenseNet architecture in distinguishing among nine distinct types of tomato leaf diseases. The proposed DenseNet-PSO achieves up to 7.67% improvement in accuracy.

Keywords: Tomato leaf disease, Hyperparameter, CNN, DenseNet, PSO

1 Introduction

Tomato stands as a cornerstone of global horticulture, playing a pivotal role in agricultural economies, with Indonesia being no exception. Its widespread cultivation significantly shapes and influences the agricultural landscape, contributing significantly to the economic fabric of the nation [1]. Agricultural productivity of tomatoes is increasing in the agricultural sector along with the high demand for nutrient-rich tomatoes, whether in their fresh or processed product [2]. In Indonesia, tomato production has shown substantial growth, with an annual average increase of 11.60%, rising from

K. Dharmawan and N. A. Sanjaya ER (eds.), *Proceedings of the First International Conference on Applied Mathematics, Statistics, and Computing (ICAMSAC 2023)*, Advances in Computer Science Research 110,
https://doi.org/10.2991/978-94-6463-413-6_23

20,000 tons in 1972 to an impressive 1.16 million tons in 2022 [3]. Meanwhile, this promising progress poses challenges for farmers and agricultural experts, particularly in terms of combatting diseases that affect tomato plants. These diseases represent a significant threat, as they can substantially diminish tomato yield and quality, which potentially resulting in losses as high as 40% of the total harvest [4].

Manual detection of tomato leaf diseases not only consumes significant time, effort, and financial resources but also demands specialized expertise and skills [5]. Utilizing advanced technology-based solutions offers significant potential for enhancing the rapid and precise detection of diseased tomato leaves. One exemplary application of the latest technology is the use of drones that equipped with cameras or sensors capable of discerning variations in color or texture in diseased tomato leaves with a high level of detail, even from altitudes that are hard to access for humans [6]. The image data gathered by these drones can be analyzed using algorithms and Artificial Intelligence (AI) techniques to classify the types of diseases that attack tomato leaves, so that farmers can obtain relevant information regarding the types of tomato leaf diseases that exist and their severity.

The rapid development of technology based on AI and computer vision has significantly improved the detection and classification of objects. Image processing using machine learning techniques, such as SVM [7], Random Forest [8], KNN [9], and Naive Bayes [10], have been applied to classify diseases on tomato leaves, but they tend to struggle with large datasets and numerous disease categories due to the challenge of identifying critical features [6]. Recently, Deep Learning (DL), a subset of machine learning, facilitates robust feature learning and achieves state-of-the-art performance on various image classification tasks. One of the most dominant and frequently used techniques in tasks such as image classification, pattern, object, face, and motion recognition is Convolutional Neural Networks (CNN) [11]. The utilization of CNN has been extensively investigated in the agricultural domain for diverse applications, including the identification and categorization of plant pests and disease [12]–[15], crop yield estimation [16], product quality monitoring [17], and others. Several well-known CNN-based architectures, such as AlexNet [18], GoogLeNet [19], VGGNet [20], Inception V3 [21], MobileNet V2 [14], ResNet [22], and DenseNet [23] have been used in several studies with high accuracy. Gehlot and Saini (2020) employed the architecture of AlexNet, VGG16, GoogleNet, DenseNet-121, and ResNet101 to classify nine distinct diseases affecting tomato leaves. The results indicate that DenseNet-121 achieves the highest accuracy rate, reaching 99.97% and has a smaller model size compared to other architectures [24]. DenseNet-121 has demonstrated remarkable performance while maintaining a more efficient use of memory and processing resources compared to other architecture. Its primary strength lies in effectively addressing the issue of vanishing gradients. This not only alleviates the training challenges associated with deep learning models but also facilitates the reuse of features across layers. Additionally, DenseNet-121 achieves a reduction in parameter usage when compared to other prevalent deep learning architectures [25].

The careful selection and configuration of hyperparameters plays a crucial role in determining the effectiveness of CNN. If hyperparameters are incorrectly chosen, CNN performance may suffer, leading to inaccurate results, as the loss function cannot be appropriately minimized. Generally, hyperparameter selection is done based on trial-and-error process. The utilization of trial-and-error techniques for hyperparameter tuning may require a significant amount of time and effort to achieve the optimal model. Additionally, these processes are influenced by historical experience, pre-existing knowledge, and the individual preferences of the network designer, frequently result in a locally optimal model, deviating significantly from globally optimal hyperparameter configurations [26].

Swarm Intelligence (SI) is one of the alternatives in navigating the complexity of CNNs to obtain the best hyperparameters. Inspired by natural group behavior, these algorithms exhibit the capability to solve complex problems. They are scalable, adaptable, and have collective robustness and individual simplicity. Moreover, they can overcome local optima and can be used for both continuous and discrete optimization problems. Particle Swarm Optimization (PSO) is a type of SI that has several advantages over conventional optimizers in CNNs. Utilizing PSO in conjunction with CNN leads to a reduction in the training process's epoch count and its reliance on GPU systems, making it more cost-effective and resource-efficient compared to other algorithms [27]. Arie et al. propose CNNPSO, a method that combines CNNs with PSO to find the optimal architecture for MNIST classification [28]. Junior and Yen introduce a variable-length PSO algorithm for finding optimal architectures in DNNs, which can be applied to various classification tasks [29]. Anam introduces an optimized approach to segment leaf spot disease using the K-means algorithm, which is enhanced through PSO [30]. Utama focuses on hyperparameter tuning of CNNs for multivariate time-series analysis and shows that PSO-CNN produces architectures that exhibit superior performance compared to regular CNNs [31]. Wang et.al. introduces cPSO-CNN, a variant of PSO that uses a confidence function and vector acceleration coefficients to enhance exploration capability and adapt to the range of CNN hyperparameters [32].

This article explores the optimal combination of hyperparameters, including the number of neurons in fully connected layers, dropout rate, learning rate, activation function, and optimizer type of DenseNet121 architecture using Particle Swarm Optimization (PSO). The performance of the DenseNet with the best hyperparameter configuration obtained through PSO is compared with default DenseNet-121 in classifying tomato leaf diseases.

2 Research Methods

This section outlines the methods utilized in developing the proposed methodology for classifying diseases in tomato leaves. The strategy includes optimizing hyperparameters of DenseNet through PSO, aiming to achieve highest classification accuracy. The assessment of model performance involves metrics such as accuracy, macro-precision, macro-recall, and macro-F1-score.

2.1 DenseNet

DenseNet forms connections between each layer in a feed-forward manner, where each layer obtains feature-maps as inputs from all the layers that precede it. Subsequently, these feature-maps are utilized in all the following layers of the network. DenseNet architecture effectively reduces gradient flow distances in backpropagation and mitigating vanishing gradient issues. It enhances feature propagation, encourages feature reuse, and significantly decreases the parameter count. In this architecture, each layer (l^{th} layer) receives inputs from all prior convolutional blocks and sends its feature-maps to all the following layers, commencing from $L - l$ onwards, resulting in $\frac{L(L+1)}{2}$ connections in L-layer network. Due to this dense connectivity pattern, the architecture is referred to as a Dense Convolutional Neural Network [33].

2.2 Particle Swarm Optimization (PSO)

PSO is an iterative optimization technique inspired by the collective movement observed in social animals. PSO operates as a search method based on a population, wherein particles collaborate to discover optimal solutions within the search space, with each particle representing a potential solution to minimize error or maximize accuracy. Two key factors primarily influence the performance of the particle: its position and velocity. The velocity of the particle plays a crucial role in optimization, relying on both the particle's individual learning and knowledge acquired from neighboring particles. The position encapsulates the particle's current solution or potential solution to the optimization problem [34]. The velocity and position equation are provided below,

$$v_i^{t+1} = \omega . v_i^t + c_1 r_1 \left(p_{best,i}^t - x_i^t\right) + c_2 r_2 (g_{best}^t - x_i^t) \tag{1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

where ω is the inertial coefficient, c_1 and c_2 are acceleration coefficient, r_1 and r_2 are random numbers produced in every iteration, falling within the range of [0, 1], $p_{best,i}^t$ denotes the personal or local best position of particle i at iteration t, g_{best}^t denotes the globally best position within the entire particle swarm.

2.3 Hyperparameter Optimization with PSO

The optimization of the DenseNet architecture for enhanced accuracy is primarily focused on five key hyperparameters: the number of neurons in the fully connected layer, dropout rate, learning rate, as well as the selection of activation function and optimizer. PSO is employed for this optimization task, with each particle in the PSO algorithm representing a distinct combination of these hyperparameters within specified constraints or search space. The objective of PSO is to systematically choose hyperparameter configurations that maximize accuracy of DenseNet architecture. Initially, a swarm or population of random particles is initialized, each corresponding to a hyperparameter of DenseNet architecture. These hyperparameters are employed for the training and testing of DenseNet model, resulting in the generation of a scalar error value for the fitness calculation of each particle. In each iteration of the PSO, three consecutive stages are typically executed. Initially, each particle undergoes the evaluation of its fitness function, and then the swarm updates using the best fitness values as both local and global bests. Finally, in each iteration, the velocity and position of every particle are updated. This iterative process continues until the specified termination conditions are met.

2.4 Evaluation Metrics

The quantitative assessment of the DenseNet model's effectiveness in classifying tomato leaf diseases encompasses metrics like True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP denotes the number of accurately classified positive images, TN signifies the number of correctly classified negative images, FP denotes the count of inaccurately classified positive images, and FN denotes the count of images that should be positive but were misclassified as negative. In multiclass classification, indices like TP_k, FP_k, and FN_k are used for each class and there is no use of TN_k as each sample can only be classified as one of the classes. The evaluation employs a macro approach, treating each class equally in the average calculation, regardless of population differences. This method avoids weighting based on class frequency. Some of the evaluation tools used are:

Accuracy measures the overall correctness of the model by determining the percentage of accurately classified images in relation to the total number of images tested.

$$Accuracy = \frac{\sum TP_k}{\sum TP_k + \sum FP_k + \sum FN_k}. \tag{3}$$

Macro-Precision calculates the average precision across all classes with equal weighting. Precision is the proportion of instances correctly classified among the overall instances predicted as positive for a specific class.

$$Precision_k = \frac{TP_k}{TP_k + FP_k},$$
$$MacroAvgPrecision\ (MAP) = \frac{\sum_{k=1}^{K} Precision_k}{K}. \tag{4}$$

Macro-Recall calculates the average recall across all classes with equal weighting. Recall evaluates the model's capacity to accurately identify all positive instances within a class.

$$Recall_k = \frac{TP_k}{TP_k + FN_k}, \tag{5}$$
$$MacroAvgRecall\ (MAR) = \frac{\sum_{k=1}^{K} Recall_k}{K}.$$

Macro-F1-Score calculates the harmonic mean between precision and recall for each class, offering a balanced evaluation that takes into account both false positives and false negatives. The objective is to achieve a balance between precision and recall.

$$Macro\ F1 - Score = \frac{2 \times MAP \times MAR}{MAP + MAR}. \tag{6}$$

3 Experimental and Result

3.1 Datasets

In this research, we obtained the dataset from a publicly accessible source on Kaggle [35], and it consists of a total of 11,000 images. The dataset exhibits an even distribution, with each of the ten classes containing 1,100 images. The representative image of each class is shown in Fig. 1. All available images are in .jpg and RGB format. The provided dataset has been partitioned into training and testing data, maintaining a ratio of 90:10.

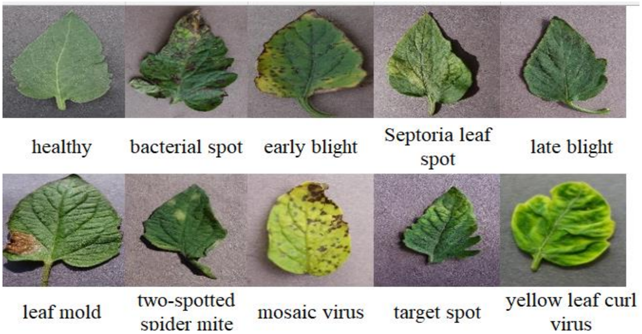

Fig. 1. Representative images of tomato leaf diseases for each class

3.2 Data Preprocessing

The original dataset had images sized at 256×256 pixels. To align with requirements of DenseNet-121 architecture, which mandates a 224×224 pixel input size, all dataset images were resized accordingly. Data augmentation methods were implemented to enrich the diversity of the dataset, ensuring better model generalization and robustness. These augmentations included random rotation, shear, shift, flip and zoom with detail in Tabel 1.

Table 1. Data augmentation technique

Technique	Value
Rescale	1./255
Rotation	20^0
Shear range	20^0
Vertical shift	20%
Horizontal shift	20%
Zoom	20%
Horizontal flip	True
Vertical flip	True
Fill mode	Nearest

3.3 Data Processing

At this phase, the dataset is partitioned into training and validation data. Although the initial dataset was initially segregated into training and validation data with a predefined ratio of 90:10, for this study, the division will be adjusted to an 80:20 ratio. The PSO algorithm implemented with the parameters listed in Table 2 to search for the optimal configuration of DenseNet hyperparameters within the specified search space boundaries outlined in Table 3.

Table 2. Parameter of PSO.

Parameter	Value
c_1, c_2	1.494
ω	0.792
Number of particles	10
Maximum iteration	20
Number of iterations for convergency criteria	10
Number of experiments	15

Table 3. The search space boundaries for DenseNet hyperparameters using PSO.

Hyperparameter	Search Space
Number of neurons in the fully connected layer	128, 256, 512, 1024
Learning rate	0.1, 0.01, 0.001, 0.0001
Dropout rate	10%, 20%, 30%, 40%, 50%
Activation function	ReLU, Tanh, Sigmoid
Optimizer	SGD, ADAM

3.4 Evaluation

The optimized hyperparameters obtained through the PSO technique are outlined in the presented Table 4. This table encapsulates the refined configuration achieved through the iterative PSO process, showcasing the values of key hyperparameters, such as the number of neurons in the fully connected layer, dropout rate, learning rate, activation function, and optimizer type. These optimized hyperparameters are instrumental in enhancing the performance of the DenseNet ultimately influencing the accuracy and efficiency of image classification, particularly in the context of detecting tomato leaf diseases.

Table 4. Optimized DenseNet hyperparameter based on PSO.

Hyperparameter	Value
Number of neurons in the fully connected layer	256
Learning rate	0.0001
Dropout rate	30%
Activation function	Tanh
Optimizer	ADAM

As shown in Table 5, the accuracy of DenseNet-PSO shows an improvement of 7.67% compared to the conventional DenseNet-121 with an accuracy of 93.73%, macro-precision of 93.19%, macro-recall of 92.87%, and macro-F1-score of 92.72%. In execution time, DenseNet121-PSO consume longer time than conventional DenseNet121, but this could have happened due to hardware limitations and the additional layer in the fully connected layer introduced during the optimization process. Nevertheless, it is crucial to highlight that, in the context of memory usage, DenseNet121-PSO necessitates a smaller amount of memory. This indicates that, despite the longer execution time, the PSO-optimized model is more memory-efficient, potentially offering advantages in scenarios where memory resources are a critical consideration.

Table 5. Comparation of the proposed model with conventional DenseNet121

	DenseNet121		DenseNet121-PSO	
	Avg	Std	Avg	Std
Accuracy	0.86058	0.02255	**0.93732**	0.02845
Macro-precision	0.85932	0.02532	0.93198	0.02813
Macro-recall	0.85521	0.02310	0.92879	0.02774
Macro-F1-score	0.85685	0.0216	0.92725	0.02832
Time computation (second)	954	33	1288.546	41.56
Memory (MB)	18002.17	1782.73	11646.8	1591.06

Table 6. DenseNet with optimized hyperparameters classification result for each disease class

Class	Precision	Recall	F1-score
Tomato Bacterial Spot	0.93	0.88	0.90
Tomato Early Blight	0.96	0.78	0.86
Tomato Healthy	0.85	0.94	0.89
Tomato Late Blight	0.97	0.91	0.94
Tomato Leaf Mold	0.89	0.95	0.91
Tomato Septoria Leaf Spot	0.83	0.92	0.88
Tomato Target Spot	0.86	0.83	0.84
Tomato Two Spotted Spider Mite	0.99	0.96	0.97
Tomato Yellow Leaf Curl Virus	0.94	0.99	0.97
Tomato Mosaic virus	0.89	0.98	0.93
accuracy			**0.94**
macro avg	**0.93**	**0.93**	**0.93**
weighted avg	0.93	0.93	0.93

In this research, there are nine categories of tomato leaves affected by diseases and one category of tomato leaves in a healthy state. The results of assessing the multiclass classification of diseases affecting tomato leaves through DenseNet-PSO are displayed in Table 6. In detail, evaluation metrics are provided for ten different classes in tomato leaves. For example, for the class "Tomato Early Blight" with a precision of 0.96, recall of 0.78, and an F1-score of 0.86 are obtained. High precision indicates that the model has a good level of accuracy in classifying samples as "Tomato Early Blight". This is beneficial in avoiding errors in classifying false negatives as "Tomato Early Blight". The lower recall value indicates that the model fails to correctly identify some samples that are actually "Tomato Early Blight". This could be due to variations in visual characteristics of this disease, making it more challenging for the model to accurately recognize it. Similar interpretations can be made for each class of disease in the table. Furthermore, there is an overall accuracy metric indicating the percentage of correct predictions out of all samples. In this case, an accuracy rate of 0.94 is obtained, meaning that the model correctly classifies approximately 94% of all samples.

4 Conclusion

We proposed a technique for optimizing DenseNet hyperparameters using the PSO algorithm for the classifying nine different kinds of diseases affecting tomato leaves. The outcomes indicated an enhancement in the accuracy rate with the proposed DenseNet-PSO, showing an improvement up to 7.67% compared to the conventional DenseNet-121. These outstanding metrics, with an accuracy of 93.73%, macro-precision of 93.19%, macro-recall of 92.87%, and macro-F1-score of 92.72%, validate the potential of our approach. This demonstrates the efficacy of the proposed model in processing and extracting features from tomato leaf data with the aim of classification. Thus, the model holds significant potential in providing superior solutions for farmers in identifying and classifying tomato leaf diseases. Consequently, farmers can enhance crop quality, reduce losses due to plant diseases, and adopt a more sustainable agricultural approach.

Acknowledgements

We express our gratitude to the High-Performance Computing (HPC) AI-Center Brawijaya University for providing this research facility.

References

1. Trivedi, N.K., Gautam, V., Anand, A., Aljahdali, H.M., Villar, S.G., Anand, D., Goyal, N, Kadry, S.: Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network. Sensors 21, 7987 (2021).
2. Adhikari, P., Oh, Y., Panthee, D.R.: Current Status of Early Blight Resistance in Tomato : An Update. International Journal of Molecular Sciences 18, 2019 (2017).
3. BPS Homepage: Produksi Tanaman Sayuran, https://www.bps.go.id/, last accessed 2023/08/12.
4. FAO Homepage: Leveraging agricultural automation for transforming agrifood systems, https://www.fao.org/, last accessed 2023/08/12.
5. David, H.E., Ramalakshmi, K., Gunasekaran, H., Venkatesan R.: Literature review of disease detection in tomato leaf using deep learning techniques. In: 7th International

Conference on Advanced Computing and Communication Systems (ICACCS), pp. 274–278. IEEE, India (2021).
6. Thangaraj, R., Anandamurugan, S., Pandiyan, P., Kaliappan, V. K.: Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion. J. Plant Dis. Prot 129(3), 469–488 (2022).
7. Zaka-ud-din, W., Aziz, M., Rashid, S., Ismail, J.: Classification of Disease in Tomato Plants Leaf Using Image Segmentation and SVM. Tech. Journal, Univ. Eng. Technol 23(2), 81–88 (2018).
8. Annabel, L.S.P., Muthulakshmi, V.: AI-Powered Image-Based Tomato Leaf Disease Detection. In: Proceedings of the 3rd International Conference on I-SMAC IoT in Social, Mobile, Analytics and Cloud, I-SMAC, pp. 506–511. IEEE (2019).
9. Nasution, A.S., Alvin, A., Siregar, A.T., Sinaga, M.S.: KNN Algorithm for Identification of Tomato Disease Based on Image Segmentation Using Enhanced K-Means Clustering. Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control 7(3), 299-308 (2022).
10. Gadede, H.D., Kirange, D.K.: Machine Learning Approach towards Tomato Leaf Disease Classification. International Journal of Advanced Trends in Computer Science and Engineering 9(1), 490-495 (2020).
11. Tugrul, B., Elfatimi, E., Eryigit, R.: Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review. Agriculture 12(8), 1192 (2022).
12. Rekha, K.S., Phaneendra, H.D., Gandha, B.S., Rohan, H., Niranjan, B.S. and Badrinat, C., 2022, October. Disease Detection in Tomato Plants Using CNN. In: 3rd Global Conference for Advancement in Technology (GCAT), pp. 1-6. IEEE (2022).
13. Attallah, O.: Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection. Horticulturae 9(2), 149 (2023).
14. Zaki, S.Z.M., Zulkifley, M.A., Stofa, M.M., Kamari, N.A.M. Mohamed, N.A.: Classification of tomato leaf diseases using MobileNet v2. IAES International Journal of Artificial Intelligence 9(2), 290 (2020).
15. Widiyanto, S., Fitrianto, R., Wardani, D.T.: Implementation of convolutional neural network method for classification of diseases in tomato leaves. In: 4th International Conference on Informatics and Computing (ICIC), pp. 1-5. IEEE (2019).
16. Yogapriya, G., Kumari, R.S., Suganthi, S., Agalya, K., Elangovan, T., Pandian, D.S.: Crop Yield Identification Using CNN. In: International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), pp. 1-4. IEEE (2023).
17. Wang, T., Chen, Y., Qiao, M., Snoussi, H.: A fast and robust convolutional neural network-based defect detection model in product quality control. The International Journal of Advanced Manufacturing Technology 94, 3465-3471 (2018).
18. Chen, H.C., Widodo, A.M., Wisnujati, A., Rahaman, M., Lin, J.C.W., Chen, L., Weng, C.E.: AlexNet convolutional neural network for disease detection and classification of tomato leaf. Electronics 11(6), 951 (2022).
19. Maeda-Gutiérrez, V., Galván-Tejada, C.E., Zanella-Calzada, L.A., Celaya-Padilla, J.M., Galván-Tejada, J.I., Gamboa-Rosales, H., Luna-Garcia, H., Magallanes-Quintanar, R., Guerrero Mendez, C.A., Olvera-Olvera, C.A.: Comparison of convolutional neural network architectures for classification of tomato plant diseases. Applied Sciences 10(4), 1245 (2020).
20. Nguyen, T.H., Nguyen, T.N., Ngo, B.V.: A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease. AgriEngineering 4(4), 871-887 (2022).

21. Saeed, A., Abdel-Aziz, A.A., Mossad, A., Abdelhamid, M.A., Alkhaled, A.Y., Mayhoub, M.: Smart Detection of Tomato Leaf Diseases Using Transfer Learning-Based Convolutional Neural Networks. Agriculture 13(1), 139 (2023).
22. Zhang, K., Wu, Q., Liu, A., Meng, X., 2018. Can deep learning identify tomato leaf disease. Advances in multimedia, (2018).
23. Bakr, M., Abdel-Gaber, S., Nasr, M., Hazman, M.: Tomato disease detection model based on densenet and transfer learning. Applied Computer Science 18(2), 56-70 (2022).
24. Gehlot, M., Saini, M.L.: Analysis of different CNN architectures for tomato leaf disease classification. In: 5th IEEE international conference on recent advances and innovations in engineering (ICRAIE), pp. 1-6. IEEE (2020).
25. Ali Albelwi, S., Hyperparameter Optimization of Deep Convolutional Neural Networks Architectures for Object Recognition (Doctoral dissertation). University of Bridgeport. Connecticut (2018).
26. Guo, Y., Li, J.Y., Zhan, Z.H.: Efficient hyperparameter optimization for convolution neural networks in deep learning: A distributed particle swarm optimization approach. Cybernetics and Systems 52(1), 36-57 (2020).
27. Isuwa, J., Abdullahi, M., Sahabi Ali, Y., Abdulrahim, A.: Hybrid particle swarm optimization with sequential one point flipping algorithm for feature selection. Concurrency and Computation: Practice and Experience 34(25), 7239 (2022).
28. Syulistyo, A.R., Purnomo, D.M.J., Rachmadi, M.F., Wibowo, A.: Particle swarm optimization (PSO) for training optimization on convolutional neural network (CNN). Jurnal Ilmu Komputer dan Informasi 9(1), 52-58 (2016).
29. Junior, F.E.F., Yen, G.G.: Particle swarm optimization of deep neural networks architectures for image classification. Swarm and Evolutionary Computation 49, 62-74 (2019).
30. Anam, S.: Segmentation of leaf spots disease in apple plants using particle swarm optimization and K-means algorithm. Journal of Physics: Conference Series 1562(1), 012011 (2020).
31. Utama, A.B.P., Wibawa, A.P., Muladi, M., Nafalski, A.: PSO based Hyperparameter tuning of CNN Multivariate Time-Series Analysis. Jurnal Online Informatika 7(2), 193-202 (2022).
32. Wang, Y., Zhang, H., Zhang, G.: cPSO-CNN: An efficient PSO-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. Swarm and Evolutionary Computation 49, 114-123 (2019).
33. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4700-4708. IEEE (2017).
34. Gupta, A., Singhal, R., Kumar, R.: Optimizing Neural Network Hyperparameters with Swarm Intelligences for Commercial Buildings Load Classification. In: 18th India Council International Conference (INDICON), pp. 1-6. IEEE (2021).
35. Kaggle: Tomato Leaf Disease Detection Tomato Leaf Disease Detection using CNN, https://www.kaggle.com/kaustubhb999/tomatoleaf, last accessed 2023/10/22.