IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 2, April 2025, pp. 1377~1385
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i2.pp1377-1385

Exploring DenseNet architectures with particle swarm optimization: efficient tomato leaf disease detection

Cynthia Ayu Dwi Lestari, Syaiful Anam, Umu Sa'adah
Department of Mathematics, Faculty of Mathematics and Science, Universitas Brawijaya, Malang, Indonesia

Article Info

Article history:

Received Mar 8, 2024
Revised Nov 16, 2024
Accepted Nov 24, 2024

Keywords:

Classification
Convolutional neural network
Deep learning
Hyperparameter
Image processing
Optimization

ABSTRACT

The critical challenge of tomato leaf disease demands effective solutions surpassing manual detection limitations, ensuring rapid intervention, optimal crop health, and maximizing yield for farmers. DenseNet, a convolutional neural network (CNN) architecture, is lauded for its adept handling of gradient flow issues by extensive interlayer connectivity. Its application holds significant promise in tackling the intricate task of identifying tomato leaf diseases. This research introduces an innovative methodology employing particle swarm optimization (PSO) to fine-tune the DenseNet architecture and hyperparameter. The proposed approach efficiently converges on optimal configurations encompassing parameters, such as the number of layers in dense blocks, growth rates, dropout rates, activation functions, and optimizers tailored for DenseNet. The DenseNet-PSO model achieves remarkable accuracy and precision in classifying various tomato leaf diseases, outperforming alternative architectures in total parameters, computational efficiency, and overall performance compared with six other architecture models. These outcomes elucidate DenseNet-PSO's efficacy in tomato leaf disease detection and demonstrate.

Corresponding Author:

Syaiful Anam
Department of Mathematics, Faculty of Mathematics and Science, Universitas Brawijaya
Malang, East Java, Indonesia
Email: syaiful@ub.ac.id

1. INTRODUCTION

Tomatoes (*Solanum Lycopersicum*) have garnered recognition as a "functional food" due to their rich composition of bioactive compounds, which confer health benefits extending beyond basic nutritional value. This attribute positions tomatoes as a crucial contributor to global food security and economic prosperity [1]. The Food and Agriculture Organization of the United Nations (FAO) reports that global tomato production reached 186 million tons in 2022 [2], solidifying its status as the sixth most abundant vegetable crop worldwide [3]. In Indonesia, the production of tomatoes has experienced significant expansion, witnessing an average annual growth rate of 11.60%, reaching 1.16 million tons in 2022 [4]. However, this impressive trajectory is continually threatened by the persistent challenge of tomato leaf diseases. These pathologies, caused by a diverse array of fungal, bacterial, and viral agents, have the potential to inflict substantial damage, potentially reducing crop yields by up to 40% [5]. Consequently, the timely and accurate diagnosis of these diseases is paramount for safeguarding food security and maintaining economic stability.

Traditionally, identifying tomato leaf diseases relied heavily on visual inspection by farmers or trained personnel. However, this approach is inherently time-consuming, labor-intensive, and necessitates a level of expertise and experience that is often scarce in resource-limited environments [6]. Fortunately, advancements in artificial intelligence (AI) and machine learning have ushered in a new era of disease detection methodologies that promise enhanced efficiency and reliability. In recent years, image processing techniques utilizing various classifiers have emerged as powerful tools, offering automated and objective analysis of leaf images [7]. Machine learning algorithms, such as support vector machines (SVM) [8] and random forests [9] have shown potential in disease classification, but they often struggle with large datasets and diverse disease categories. Additionally, extracting critical features from complex leaf images poses challenges for these algorithms, limiting their generalizability and reliability [7].

Deep learning, a subfield of AI, has recently gained significant traction due to its ability to automatically learn intricate features from data [10]. Within this domain, convolutional neural networks (CNNs) have emerged as a transformative force in image classification, demonstrating remarkable efficacy in tasks such as plant disease and pest identification [11]–[13], crop yield estimation [14], [15], and product quality assessment [16]. CNNs have evolved from pioneering architectures like AlexNet [17] and visual geometry group network (VGGNet) [18] to more efficient models such as GoogLeNet [19] and Inception-v3 [20]. The pursuit of lightweight models suitable for mobile applications led to the development of EfficientNet [21] and MobileNet V2 [22], which achieved impressive performance. A pivotal advancement in deep learning architecture came with the introduction of ResNet [23] which incorporated residual connections to facilitate the training of deeper networks and mitigate the vanishing gradient problem. Building upon these innovations, DenseNet [24] emerged as a powerful contender, boasting exceptional accuracy of 99.97% on the ImageNet dataset while maintaining a comparatively modest model size. Studies have demonstrated that variants such as DenseNet-201 consistently outperform other architectures like ResNet-50 and Inception-v3 in this domain [25]. The ability of DenseNet to effectively learn relevant features from images and classify them accurately has been a key factor in its success in plant disease classification tasks. Its effectiveness arises from its capability to address vanishing gradients and its distinctive characteristic of reusing features across layers, thereby notably decreasing memory and processing requirements [26].

The performance of DenseNet architectures is primarily influenced by two key parameters: the number of layers within dense blocks (L) and the growth rate (k). Increasing L generally enhances accuracy and the ability to learn complex features but also escalates model complexity and computational cost [27]. Conversely, employing a smaller L in certain scenarios can yield similar or superior accuracy while providing added benefits in terms of efficiency and efficacy. This can be attributed to factors such as reduced overfitting risk and lower computational requirements, facilitating faster training and deployment processes [28]. The growth rate (k) in DenseNet architectures governs the number of new feature maps added in each dense block layer. While higher k values offer richer feature representations and potentially improved performance, they also contribute to increased model size and complexity. Conversely, excessively low k values risk underfitting, while excessively high k may lead to overfitting and memory constraints [29]. Therefore, it is essential to explore various combinations L and k values and carefully consider the trade-off between accuracy, efficiency, and computational cost to achieve an optimal DenseNet architecture tailored to the specific task and dataset.

Achieving peak performance with DenseNet requires precise architectural adjustments and hyperparameter fine-tuning. However, manual tuning these hyperparameters is a tedious and time-consuming process, often leading to suboptimal results. Studies on [30] have demonstrated the computational challenges inherent in identifying optimal hyperparameter settings for CNN models, highlighting the limitations of manual tuning approaches. Considering these challenges, particle swarm optimization (PSO) emerges as a promising alternative. Inspired by the collective intelligence of natural swarms, PSO possesses a well-established capability to navigate complex search spaces and identify globally optimal configurations [31]. This approach transcends the limitations of manual tuning by automating the exploration of various DenseNet architecture and hyperparameter variations. PSO facilitates the discovery of an architecture that is tailored to the specific challenges of tomato leaf disease classification by dynamically adjusting these parameters. The effectiveness of PSO-based optimization for deep learning models has been demonstrated in numerous studies, encompassing tasks like modified national institute of standards and technology (MNIST) classification [32] and leaf spot disease segmentation [33]. For instance, one study introduced a novel PSO algorithm tailored for searching optimal architectures in deep CNNs, employing variable-length particles with exceptional efficacy [34]. Another study proposed a PSO-based method for evolving deep CNNs, leveraging PSO's capability to tackle optimization challenges devoid of domain knowledge [35]. Variants like cPSO-CNN further augment exploration capabilities, leading to even more effective hyperparameter tuning [36]. Notably, PSO-CNN architectures consistently outperform regular CNNs, showcasing the potential of this approach for significantly enhancing deep learning model performance. This research proposes a novel approach that leverages the power of PSO to optimize both the hyperparameters and architecture of DenseNet for tomato leaf disease classification. By integrating PSO with DenseNet, this study aims to achieve several key benefits: i) enhanced accuracy and robustness: optimizing architecture and hyperparameters with PSO has the potential to significantly improve the accuracy and generalization of DenseNet, leading to more accurate and reliable disease detection; ii) reduced training time and memory consumption: PSO offers a faster way to explore different configurations, cutting down on training time and computational resources compared to manual tuning; and iii) automated architecture exploration: this approach goes beyond hyperparameter tuning, actively probing diverse DenseNet architecture configurations to autonomously identify the most efficient structure.

This innovative approach not only unlocks possibilities for advancements in disease identification but also lays the groundwork for making DenseNets better for classifying tomato leaf diseases. It could lead to simpler and more accurate models than traditional methods. Although using PSO in deep learning has shown promise in other areas, its potential in optimizing DenseNets for agricultural disease classification hasn't been explored much. Ultimately, this research helps improve food security in regions with limited resources by making disease detection better and could be applied to different crops in the future.

2. METHOD

2.1. Dataset

The experimental data originated from a publicly available dataset [37], comprising 11,000 images categorized into 10 classes. These classes encompassed nine distinct tomato leaf disease pathologies and a dedicated class for healthy leaves. Visual representations for each class are provided in Figure 1, where Figure 1(a) shows a healthy, Figure 1(b) shows a bacteria spot, Figure 1(c) shows a early blight, Figure 1(d) shows a leaf mold, Figure 1(e) shows a late blight, Figure 1(f) shows a septoria leaf spot, Figure 1(g) shows a two-spotted spider mite, Figure 1(h) shows a mosaic virus, Figure 1(i) shows a target spot, and Figure 1(j) shows a yellow leaf curl virus. The dataset comprises images in .jpg format with a uniform resolution of 256×256 pixels and utilizes the RGB color space. The dataset has been meticulously divided into training and testing subsets, maintaining an 80:20 ratio to facilitate model training and evaluation.

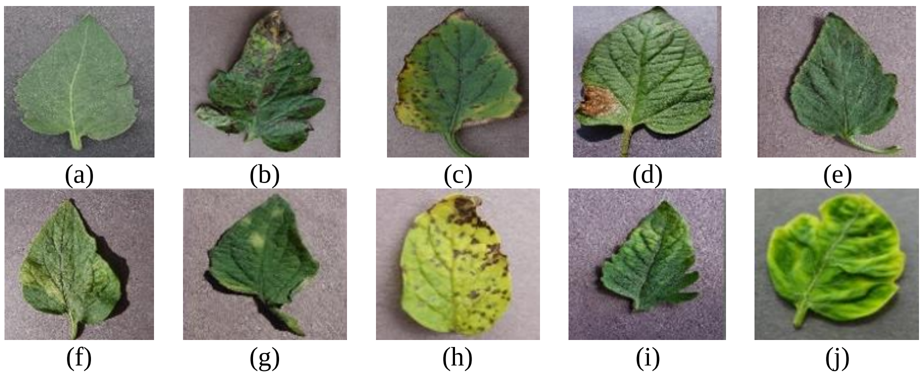

Figure 1. Visual representations for each class of tomato leaf diseases of (a) healthy, (b) bacteria spot, (c) early blight, (d) leaf mold, (e) late blight, (f) septoria leaf spot, (g) two-spotted spider mite, (h) mosaic virus, (i) target spot, and (j) yellow leaf curl virus

2.2. Data preprocessing

Data preprocessing assumes a pivotal role in readying image data for deep learning models, employing diverse techniques to refine data quality, boost model performance, and enhance accuracy. In this study, we resized all images to a uniform size of 224×224 pixels to match the input requirements of the architecture models used. This step ensures that all images are processed consistently and efficiently by the models. Furthermore, we implemented data augmentation techniques to enrich the dataset and improve model generalization. Data augmentation emerges as a prevalent strategy for image classification tasks, generating additional images through diverse transformation methods to address limited training data. Table 1 provides a detailed overview of the data augmentation parameters used.

2.3. Model

DenseNet serves as the cornerstone of this investigation due to its renowned effectiveness in feature reuse and dense connectivity. Each layer in a DenseNet leverages features from all preceding layers as input, while its own features contribute to all subsequent layers in the network. Its core architecture revolves around dense blocks, meticulously structured with 3×3 kernel convolutional layers and batch normalization to ensure stability. In the dense blocks, each layer benefits from access to all preceding feature maps, promoting comprehensive information flow and facilitating feature reuse. Each layer in the dense block produces k feature maps post-convolution, with k representing the growth rate hyperparameter. This parameter dictates the quantity of new feature maps incorporated in each layer, necessitating meticulous exploration to achieve a harmonious equilibrium between expressive capability and model complexity. The number of layers within each dense L is equally significant, influencing the richness of extracted features and potential accuracy gains. However, excessive layering poses a risk of overfitting, wherein the model memorizes training data rather than learning to generalize effectively. Transition layers elegantly connect these blocks, employing 1×1 convolutions and pooling to manage spatial dimensions and overall model size. Its structure is batch normalization (BatchNorm)+activation function+1×1 convolution+2×2 AvgPooling. Additionally, global average pooling replaces traditional fully connected layers to enhance the model's generalization capabilities. This is followed by a single output neuron with softmax activation, which provides class probabilities for the classification task.

The PSO-guided optimization process commences with the formation of a particle swarm, each representing a potential configuration for DenseNet. These particles encapsulate hyperparameter values, including the number of layers in dense block, growth rate, dropout rate, activation function, and optimizer. Initial values are drawn from predefined boundaries in Table 2 to ensure exploration within feasible limits. The particles performance evaluated through metrics such as training accuracy, drives the iterative optimization process, which is governed by initial parameters specified in Table 3. As the swarm iterates through the search space, continuous evaluation propels collective movement towards an optimal configuration until the predefined number of iterations progressively converge towards the most effective DenseNet architecture for the task at hand. The position (x_i) and velocity (v_i) of each particle are updated using (1) and (2).

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

$$v_i^{t+1} = \omega . v_i^t + c_1 r_1 \left(p_{best,i}^t - x_i^t\right) + c_2 r_2 (g_{best}^t - x_i^t) \tag{2}$$

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.

Table 1. Data augmentation technique

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

Table 2. The search space bounds for the DenseNet architecture and hyperparameters utilizing PSO

Hyperparameter	Search space
Number of layers in 1st block	[1, 6]
Number of layers in 2nd block	[1, 12]
Number of layers in 3rd block	[1, 48]
Number of layers in 4th block	[1, 32]
Growth rate	12, 16,2 4, 32, 48
Dropout rate	10%, 20%, 30%, 40%, 50%
Activation function	ReLU, Tanh, Sigmoid
Optimizer	SGD, ADAM

Table 3. Initial parameters

Parameter	Value
c_1, c_2	1.494
ω	0.792
Number of particles	10
Batch size	128
Loss function	Categorical cross-entropy
Maximum iteration	20
Number of iterations for convergency criteria	10
Number of experiments	20

2.4. Evaluation

The study adopts a macro-based evaluation approach, treating each disease class with equal importance regardless of the number of images per class. This method ensures unbiased assessment despite potential imbalances in the dataset. Several key metrics were employed:

– Accuracy: reflects the overall model performance by measuring the percentage of correctly classified images across all classes. A higher accuracy signifies better overall prediction capability.

$$Accuracy = \frac{\sum TP_k}{\sum TP_k + \sum FP_k + \sum FN_k} \quad (3)$$

– Macro-precision: represents the average precision across all classes. Precision indicates the proportion of true positives within the model's predictions for each specific disease. It gauges how accurate the model's positive predictions are:

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

– Macro-recall: captures the average recall across all classes. Recall quantifies the model's effectiveness in identifying all genuine cases within each disease class. A higher recall value indicates better detection of true positives with fewer missed cases.

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

– Macro-F1-score: calculates the harmonic mean of precision and recall for each class, providing a balanced assessment that considers both false positives and false negatives. A high F1 score indicates a good balance between precision and recall, signifying the model can accurately predict both the presence and absence of diseases.

$$Macro\ F1 - score = \frac{2 \times MAP \times MAR}{MAP + MAR} \quad (6)$$

3. RESULT AND DISCUSSION

3.1. DenseNet-PSO architecture

The PSO approach has been applied to optimize the performance and efficiency of the DenseNet architecture by finding the optimal parameters by conducting 20 trials. The documentation of the optimal architecture and hyperparameters of DenseNet-PSO along with the comparison with the default hyperparameters of other DenseNet architecture types are presented in Table 4. DenseNet-PSO features a lighter and simpler structure compared to other DenseNet architectures. The overall structure of DenseNet-PSO is detailed in Figure 2, presenting a comprehensive scheme comprising bottleneck layers, dense blocks, and transition layers. The DenseNet-PSO architecture implements a more conservative growth rate compared to its predecessors, yielding a more compact parameter space and enhancing both memory efficiency and computational performance. This model also employs a lower dropout rate, which mitigates overfitting through selective neuronal deactivation during the training phase, fostering generalized learning while preserving model stability. Furthermore, this lower dropout rate contributes to the model's training stability by attenuating excessive fluctuations often associated with high dropout implementations. DenseNet-PSO employs the same activation function and optimizer as other DenseNet architectures. DenseNet-PSO capitalizes on rectified linear unit (ReLU) capacity for non-linear representation learning and gradient preservation along with utilizing Adam's adaptive learning rate adjustment contributes to faster model convergence during the training process.

3.2. Comparison of DenseNet-PSO model with other architectures

The DenseNet-PSO model in this section is compared with other architectural models that have been used or proposed in previous research, such as DenseNet-121, DenseNet-169, DenseNet-201, ResNet-101, InceptionV3, and MobileNet. This comparative analysis encompasses a comprehensive evaluation of model efficiency and performance metrics. Specifically, we examine total parameters, computation time, and storage memory requirements to assess computational efficiency. Additionally, we evaluate model performance through key metrics such as accuracy, precision, recall, and F1-score.

3.2.1. Comparison based on total parameters, time computation, and memory usage

Based on the comparison listed in Table 5, the DenseNet-PSO model stands out compared to the other six models because it has significantly fewer total parameters. This shows the efficiency of the model in utilizing memory and computational resources. DenseNet-PSO showed superior performance in the overall evaluation behind its simplicity with lower complexity. Computation times across all models, including DenseNet-PSO, are notably similar, with minimal variations in mean and standard deviation, indicating consistent and stable prediction and training times. Additionally, DenseNet-PSO's lower storage memory requirements enhance its efficiency, making it particularly suitable for deployment on devices with limited memory resources. This combination of compact design, high performance, and resource efficiency positions DenseNet-PSO as a standout model among its peers.

Table 4. Comparison of architecture and hyperparameters

Hyperparameter	DenseNet-PSO	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-161
Number of layers in 1st block	3	6	6	6	6
Number of layers in 2nd block	9	12	12	12	12
Number of layers in 3rd block	32	24	32	48	36
Number of layers in 4th block	17	16	32	32	24
Growth rate	12	32	32	32	48
Dropout rate	0.1	0.5	0.5	0.5	0.5
Activation function	ReLU	ReLU	ReLU	ReLU	ReLU
Optimizer	Adam	Adam	Adam	Adam	Adam

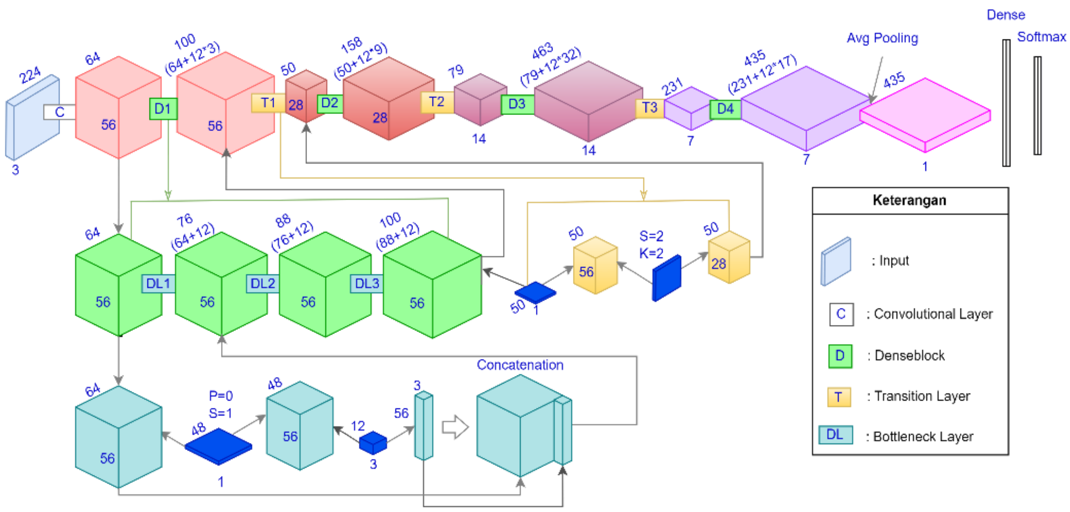

Figure 2. Full schematic representation of DenseNet-PSO

Table 5. Comparison based on total parameters, time computation, and memory usage

Model	Total Parameter	Total time computation (seconds)		Total memory usage (MB)	
		Avg	Std	Avg	Std
DenseNet-PSO	1, 257, 027	749.6868	2.3490	13,620.09	3.6840
DenseNet-121	7, 047, 754	752.4470	5.7617	16,243.04	0.7864
DenseNet-169	12, 659, 530	750.2726	4.1358	16,509.17	21.3546
DenseNet-201	18, 341, 194	755.9802	1.7955	16,077.62	18.8820
ResNet-101	42, 678, 666	749.8854	5.2668	16,252.44	2.3335
InceptionV3	21, 823, 274	753.9185	2.4259	16,197.90	59.9983
MobileNet	3, 239, 114	748.6338	1.5583	15,923.28	104.279

3.2.2. Comparison based on evaluation metrics

DenseNet-PSO emerges as the standout model, demonstrating superior and stable performance across evaluation metrics compared to the six other models tested. As visualized in Figure 3 and detailed in

Table 6, it achieved the highest average accuracy of 97.39% (±0.74%), along with top macro-precision (97.47%), macro-recall (97.39%), and macro-F1-score (97.38%). These results not only showcase the model's exceptional accuracy and balanced performance in disease detection but also highlight its efficiency, as it accomplishes this with fewer parameters than its counterparts. The consistently low standard deviations across metrics indicate stable and resilient performance, highlighting DenseNet-PSO's ability to maintain high accuracy across varied datasets.

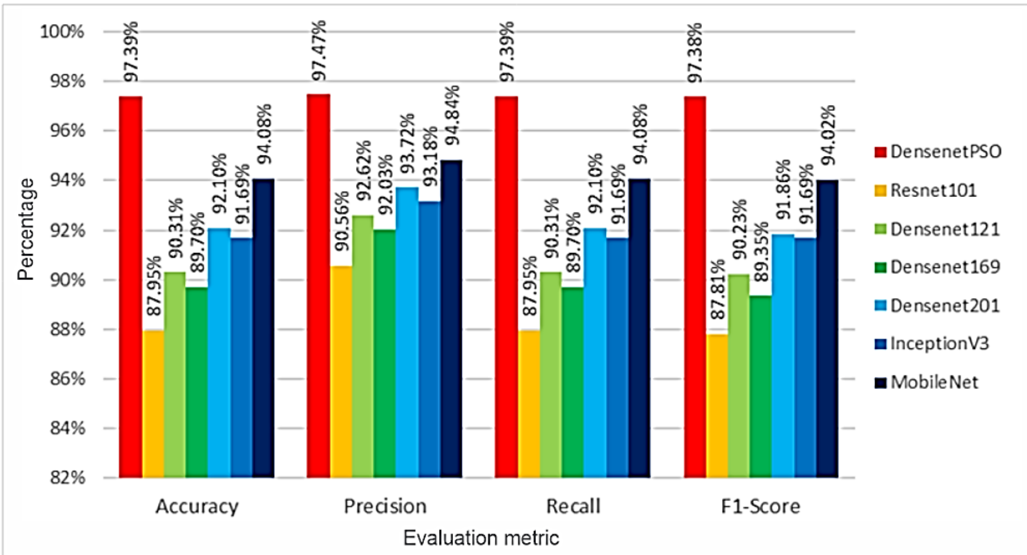

Figure 3. Comparison chart of average evaluation metric results

Table 6. Comparison based on evaluation metrics

Model	Accuracy		Macro-precision		Macro-recall		Macro-F1-score	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
DenseNet-PSO	0.9739	0.0074	0.9747	0.0069	0.9739	0.0074	0.9738	0.0074
DenseNet-121	0.9031	0.0485	0.9262	0.0316	0.9031	0.0485	0.9023	0.0493
DenseNet-169	0.8970	0.0638	0.9203	0.0398	0.8970	0.0638	0.8935	0.0676
DenseNet-201	0.9210	0.0571	0.9372	0.0380	0.9210	0.0571	0.9186	0.0608
ResNet-101	0.8795	0.0563	0.9056	0.0393	0.8795	0.0563	0.8781	0.0571
InceptionV3	0.9169	0.0531	0.9318	0.0395	0.9169	0.0531	0.9169	0.0527
MobileNet	0.9408	0.0312	0.9484	0.0248	0.9408	0.0312	0.9402	0.0323

4. CONCLUSION

This research investigated the effectiveness of DenseNet-PSO, a model optimized using the PSO algorithm for classifying tomato leaf diseases. The model achieved an impressive overall accuracy of 97.39% and consistently outperformed six other architectures in terms of various metrics, including macro-precision, macro-recall, macro-F1-score, total parameters, computational time, and storage memory. These findings suggest DenseNet-PSO's potential for robust performance, efficient resource utilization, and reduced overfitting, leading to reliable generalization capabilities. The implementation of this model in agriculture holds the promise of transforming disease diagnosis, empowering informed decision-making, and ultimately enhancing crop quality, minimizing losses, and fostering a more sustainable agricultural paradigm. This research paves the way for utilizing advanced deep learning models like DenseNet-PSO to address crucial challenges in agriculture and contribute towards ensuring global food security. However, further and in-depth studies may be needed to confirm its generalization performance across diverse datasets and real-world scenarios, especially regarding its robustness to overfitting. It's worth noting that the PSO optimization process may encounter challenges related to local optima, especially under constraints such as limited particles or iterations due to computational resources. Future research could focus on enhancing PSO's exploration capabilities or exploring alternative optimization algorithms to overcome such limitations effectively.

ACKNOWLEDGEMENTS

The authors express their gratitude to the High-Performance Computing (HPC) AI-Center of Universitas Brawijaya for their support in providing the essential research infrastructure utilized in this study.

REFERENCES

[1] S. R. Wyngaard and M. Kissinger, "Tomatoes from the desert: environmental footprints and sustainability potential in a changing world," *Frontiers in Sustainable Food Systems*, vol. 6, Oct. 2022, doi: 10.3389/fsufs.2022.994920.

[2] FAO, "Agricultural production statistics 2000–2022," *Food and Agriculture Organization*. [Online]. Available: http://www.fao.org/3/cc9205en/cc9205en.pdf

[3] M. H. Najim, S. K. Abdulateef, and A. H. Alasadi, "Early detection of tomato leaf diseases based on deep learning techniques," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 1, pp. 509-515, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp509-515.

[4] BPS, "Production of vegetable and seasonal fruits by crop type (in indonesian: produksi tanaman sayuran dan buah–buahan semusim menurut jenis tanaman)," *Badan Pusat Statistik*, 2023. [Online]. Available: https://www.bps.go.id/id/statistics-table/3/VFV4MmQxaG9kakZrVUdWeEx6aDFUMnN6WmpocVp6MDkjMw==/produksi-tanaman-sayuran-dan-buahbuahan-semusim-menurut-jenis-tanaman--2023.html?year=2022

[5] FAO, "Climate change fans spread of pests and threatens plants and crops, new fao study," *Food and Agriculture Organization*, 2021. [Online]. Available: https://www.fao.org/newsroom/detail/Climate-change-fans-spread-of-pests-and-threatens-plants-and-crops-new-FAO-study/en

[6] H. E. David, K. Ramalakshmi, H. Gunasekaran, and R. Venkatesan, "Literature review of disease detection in tomato leaf using deep learning techniques," in *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, Mar. 2021, pp. 274–278, doi: 10.1109/ICACCS51430.2021.9441714.

[7] R. Thangaraj, S. Anandamurugan, P. Pandiyan, and V. K. Kaliappan, "Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion," *Journal of Plant Diseases and Protection*, vol. 129, no. 3, pp. 469–488, Jun. 2022, doi: 10.1007/s41348-021-00500-8.

[8] Z. ud Din, S. M. Adnan, R. W. Ahmad, S. Aziz, W. Ismail, and J. Iqbal, "Classification of tomato plants' leaf diseases using image segmentation and svm," *Technical Journal*, vol. 23, no. 2, pp. 81–88, 2018.

[9] L. S. P. Annabel and V. Muthulakshmi, "AI-powered image-based tomato leaf disease detection," in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, IEEE, Dec. 2019, pp. 506–511, doi: 10.1109/I-SMAC47947.2019.9032621.

[10] L. Alzubaidi *et al.*, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[11] K. S. Rekha, H. D. Phaneendra, B. . S. Gandha, H. Rohan, B. N. S. Niranjan, and C. Badrinat, "Disease detection in tomato plants using cnn," in *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, IEEE, Oct. 2022, pp. 1–6, doi: 10.1109/GCAT55367.2022.9972186.

[12] O. Attallah, "Tomato leaf disease classification via compact convolutional neural networks with transfer learning and feature selection," *Horticulturae*, vol. 9, no. 2, Jan. 2023, doi: 10.3390/horticulturae9020149.

[13] S. Widiyanto, R. Fitrianto, and D. T. Wardani, "Implementation of convolutional neural network method for classification of diseases in tomato leaves," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, IEEE, Oct. 2019, pp. 1–5, doi: 10.1109/ICIC47613.2019.8985909.

[14] G. Yogapriya, R. S. Kumari, S. Suganthi, K. Agalya, T. Elangovan, and D. S. Pandian, "Crop yield identification using cnn," in *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF)*, IEEE, 2023, pp. 1–4, doi: 10.1109/ICECONF57129.2023.10084304.

[15] F. F. Haque, A. Abdelgawad, V. P. Yanambaka, and K. Yelamarthi, "Crop yield prediction using deep neural network," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, IEEE, Jun. 2020, pp. 1–4, doi: 10.1109/WF-IoT48130.2020.9221298.

[16] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9–12, pp. 3465–3471, Feb. 2018, doi: 10.1007/s00170-017-0882-0.

[17] H.-C. Chen *et al.*, "AlexNet convolutional neural network for disease detection and classification of tomato leaf," *Electronics*, vol. 11, no. 6, Mar. 2022, doi: 10.3390/electronics11060951.

[18] T.-H. Nguyen, T.-N. Nguyen, and B.-V. Ngo, "A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease," *AgriEngineering*, vol. 4, no. 4, pp. 871–887, Oct. 2022, doi: 10.3390/agriengineering4040056.

[19] V. Maeda-Gutiérrez *et al.*, "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Applied Sciences*, vol. 10, no. 4, Feb. 2020, doi: 10.3390/app10041245.

[20] A. Saeed, A. A. Abdel-Aziz, A. Mossad, M. A. Abdelhamid, A. Y. Alkhaled, and M. Mayhoub, "Smart detection of tomato leaf diseases using transfer learning-based convolutional neural networks," *Agriculture*, vol. 13, no. 1, 2023, doi: 10.3390/agriculture13010139.

[21] N. C. Kundur and P. B. Mallikarjuna, "Insect pest image detection and classification using deep learning," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 9, 2022, doi: 10.14569/IJACSA.2022.0130947.

[22] S. Z. M. Zaki, M. A. Zulkifley, M. M. Stofa, N. A. M. Kamari, and N. A. Mohamed, "Classification of tomato leaf diseases using mobilenet v2," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 2, pp. 290-296, Jun. 2020, doi: 10.11591/ijai.v9.i2.pp290-296.

[23] K. Zhang, Q. Wu, A. Liu, and X. Meng, "Can deep learning identify tomato leaf disease?," *Advances in Multimedia*, vol. 2018, pp. 1–10, Sep. 2018, doi: 10.1155/2018/6710865.

[24] M. Bakr, S. Abdel-Gaber, M. Nasr, and M. Hazman, "Tomato disease detection model based on densenet and transfer learning," *Applied Computer Science*, vol. 18, no. 2, pp. 56–70, Jun. 2022, doi: 10.35784/acs-2022-13.

[25] W. Albattah, M. Nawaz, A. Javed, M. Masood, and S. Albahli, "A novel deep learning method for detection and classification of plant diseases," *Complex & Intelligent Systems*, vol. 8, no. 1, pp. 507–524, Feb. 2022, doi: 10.1007/s40747-021-00536-1.

[26] M. Gehlot and M. L. Saini, "Analysis of different cnn architectures for tomato leaf disease classification," in *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, IEEE, Dec. 2020, pp. 1–6, doi: 10.1109/ICRAIE51050.2020.9358279.

[27] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
[28] X. Feng, H. Yao, and S. Zhang, "An efficient way to refine densenet," *Signal, Image and Video Processing*, vol. 13, no. 5, pp. 959–965, Jul. 2019, doi: 10.1007/s11760-019-01433-4.
[29] G. Pleiss, D. Chen, G. Huang, T. Li, L. van der Maaten, and K. Q. Weinberger, "Memory-efficient implementation of densenets," *arXiv-Computer Science*, pp. 1-8, Jul. 2017.
[30] O. N. Oyelade and A. E. Ezugwu, "A comparative performance study of random-grid model for hyperparameters selection in detection of abnormalities in digital breast images," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 13, Jun. 2022, doi: 10.1002/cpe.6914.
[31] J. Isuwa, M. Abdullahi, Y. S. Ali, and A. Abdulrahim, "Hybrid particle swarm optimization with sequential one point flipping algorithm for feature selection," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 25, Nov. 2022, doi: 10.1002/cpe.7239.
[32] A. R. Syulistyo, D. M. J. Purnomo, M. F. Rachmadi, and A. Wibowo, "Particle swarm optimization (PSO) for training optimization on convolutional neural network (CNN)," *Jurnal Ilmu Komputer dan Informasi*, vol. 9, no. 1, Feb. 2016, doi: 10.21609/jiki.v9i1.366.
[33] S. Anam, "Segmentation of leaf spots disease in apple plants using particle swarm optimization and k-means algorithm," *Journal of Physics: Conference Series*, vol. 1562, no. 1, Jun. 2020, doi: 10.1088/1742-6596/1562/1/012011.
[34] D. Elhani, A. C. Megherbi, A. Zitouni, F. Dornaika, S. Sbaa, and A. Taleb-Ahmed, "Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification," *Expert Systems with Applications*, vol. 229, Nov. 2023, doi: 10.1016/j.eswa.2023.120411.
[35] B. Wang, B. Xue, and M. Zhang, "Particle swarm optimisation for evolving deep neural networks for image classification by evolving and stacking transferable blocks," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, Jul. 2020, pp. 1–8, doi: 10.1109/CEC48606.2020.9185541.
[36] Y. Wang, H. Zhang, and G. Zhang, "CPSO-cnn: an efficient pso-based algorithm for fine-tuning hyper-parameters of convolutional neural networks," *Swarm and Evolutionary Computation*, vol. 49, pp. 114–123, Sep. 2019, doi: 10.1016/j.swevo.2019.06.002.
[37] B. Kaustubh, "Tomato leaf disease detection-tomato leaf disease detection using cnn," *Kaggle*, 2020. [Online]. Available: https://www.kaggle.com/datasets/kaustubhh999/tomatoleaf

BIOGRAPHIES OF AUTHORS

Cynthia Ayu Dwi Lestari received her master's degree in Department of Mathematics, Faculty of Mathematics and Science from Universitas Brawijaya, Indonesia in 2024 with a focus on artificial intelligence and data science. She also holds a bachelor's degree in mathematics from Universitas Brawijaya, which she completed in 2020. Her current research interests lie in the areas of data mining, machine learning, deep learning, and metaheuristics. Her keen interest lies in the practical application of these techniques to address real-world challenges, aiming to play a role in the progress of machine learning and artificial intelligence. She can be contacted by email at: cynthiaayu14@gmail.com.

Syaiful Anam received a doctor of natural science and mathematics degree from Yamaguchi University, Japan in 2015. He also received his bachelor's degree in mathematics from Universitas Brawijaya, Indonesia in 2001 and his master degree from Sepuluh Nopember Institute of Technology, Indonesia in 2006. He is currently an Assistant Professor at Department of Mathematics, Faculty of Mathematics and Science, Universitas Brawijaya, Malang, Indonesia. His research includes data science, computational intelligence, machine learning, digital image processing, and computer vision. He has published over 35 papers in international journals and conferences. He can be contacted at email: syaiful@ub.ac.id.

Umu Sa'adah holds doctor of statistics degree (Dr.) from Universitas Gadjah Mada, Indonesia, in 2015. Prior to that, she completed her bachelor's degree (Dra) in mathematics at the same university in 1993, followed by a master's degree (M.Si.) in mathematics in 2002, also from Universitas Gadjah Mada, Indonesia. She is currently an Associate Professor at Department of Mathematics, Faculty of Mathematics and Science in Universitas Brawijaya, Malang, Indonesia. Her research areas of interest include data mining, machine learning, bootstrap, computational statistics, and actuaria. She can be contacted at email: u.saadah@ub.ac.id.