Modified particle swarm optimization (MPSO) optimized CNN's hyperparameters for classification

Murinto [a,1,*], Sri Winiarti [a,2]

[a] Department of Informatic, Universitas Ahmad Dahlan Yogyakarta, Kampus 4 UAD Ringroad Selatan, Yogyakarta, 55167, Indonesia
[1] murintokusno@tif.uad.ac.id; [2] sri.winiarti@tif.uad.ac.id
* corresponding author

ARTICLE INFO

Article history
Received September 7, 2023
Revised September 10, 2024
Accepted December 28, 2024
Available online February 28, 2025

Keywords
Batik motif
CNN
Classification
Hyperparameter optimization
MPSO

ABSTRACT

This paper proposes a convolutional neural network architectural design approach using the modified particle swarm optimization (MPSO) algorithm. Adjusting hyper-parameters and searching for optimal network architecture from convolutional neural networks (CNN) is an interesting challenge. Network performance and increasing the efficiency of learning models on certain problems depend on setting hyperparameter values, resulting in large and complex search spaces in their exploration. The use of heuristic-based searches allows for this type of problem, where the main contribution in this research is to apply the MPSO algorithm to find the optimal parameters of CNN, including the number of convolution layers, the filters used in the convolution process, the number of convolution filters and the batch size. The parameters obtained using MPSO are kept in the same condition in each convolution layer, and the objective function is evaluated by MPSO, which is given by classification rate. The optimized architecture is implemented in the Batik motif database. The research found that the proposed model produced the best results, with a classification rate higher than 94%, showing good results compared to other state-of-the-art approaches. This research demonstrates the performance of the MPSO algorithm in optimizing CNN architectures, highlighting its potential for improving image recognition tasks.

1. Introduction

Industry is considered a major factor in sustainable development in several countries because of its high impact on the people's economy. Continuous monitoring of production played an important role in the early industrial world. Various fields in the industry, including the field of Agriculture and the home Industry (e.g., Batik and Food). The traditional method of inspection of industrial production depends on visual observation with the help of human experience and training to detect production disturbances. The traditional method approach is only based on cognitive and psychological aspects, which can lead to errors due to illusions [1]. To assist decision-makers in intelligent systems (artificial intelligence/AI), recognition of plant diseases based on intelligent system techniques has been proposed in several literatures [2]–[5].

In the context of AI, deep learning is an important part of machine learning that is in great demand and has become increasingly important in recent years [6], [7]. Deep learning has been used in various fields of application, such as hyperspectral image analysis [8]–[10]. The advantages of deep learning include that this approach is based on neural networks that can exploit feature hierarchies and their interactions. In the deep learning extraction process, feature selection and classification can be identified

through an in-depth architectural optimization process. The most common architecture is the convolutional neural network (CNN), as shown in [11]–[16].

In computer vision (CV) applications, CNN has shown better performance. Model results are often reused to solve problems in new problems using methods such as transfer learning [17]–[19]. Transfer learning involves a pre-trained model and some training from further layers using the new dataset. When building a new model from scratch, one of the main challenges is choosing the right number of convolution layers, kernel size, and number of output channels. Research related to this field has resulted in many architectures, including VGG16 [20], Inception [21], ResNet [22], and DenseNet [23]. In practice, it is not easy to evaluate network hyperparameters. In fact, the number of layers, the number of neurons per layer, or the different connections between layers are important elements and are essentially determined through intuition or through a series of test or error calculations, which takes a lot of time.

Several approaches developed for hyperparameter optimization include random search (RS) [24], Bayes optimization (BO) [25], and metaheuristic algorithms [26]. Metaheuristics have been used successfully to solve optimization problems in science, engineering, and the industry. Several studies have been carried out well, including using a Genetic Algorithm (GA) [27], [28]Particle Swarm Optimization (PSO) [29], [30], Modified Particle Swarm Optimization (MoPSO) [31], [32]. Models that use traditional PSO particle positions are changed through random multiplication of layers from personal best (Pbest) or global solutions (Gbest). The PSO algorithm has many advantages, but it also has disadvantages. It quickly falls into the local optimum, has a low convergence speed in the iterative process, and loses population diversity. A modified particle swarm optimization proposed by Murinto *et al.* [33] will be used to overcome this.

Deep neural networks (DNN) can be grouped into feed-forward neural networks (FCNN) and deep convolutional neural networks (CNN). One of the very good algorithms in deep learning (DL) is convolutional neural networks (CNN). CNN is an artificial neural network used in image processing and recognition. The first part of CNN is the actual convolution, a simple mathematical operation specifically used for image feature extraction. The next part is the pooling layer, followed by the activation layer and the fully connected layer, which is the final layer of the CNN architecture. Hyperparameters in CNN are variables that determine the network architecture, namely the number of channels, number of layers, kernel/filter size, padding, stride, and layer pooling parameters, as well as variables that determine how the network is trained, namely learning rate, dropout rate, normalization, regulation function (L1, L2), type of activation function (Rectified Linear Unit/ReLU, Sigmoid, Tanh, etc.), batch size. The particle swarm optimization (PSO) algorithm is an evolutionary algorithm that uses a population of candidate solutions to reach an optimal solution in solving problems.

This research aims to introduce a CNN optimization approach using the Modified PSO algorithm, which contributes to deep learning by leveraging MPSO to determine optimal hyperparameters efficiently. The model was validated on the Batik motif dataset, achieving classification accuracy.

2. Method

In this research, an optimization approach is used, and the MPSO algorithm is applied to optimize the CNN architecture parameters, denoted as MPSO-CNN. The aim is to select the most relevant parameters influencing the determination of good CNN performance and then apply the MPSO algorithm to find these optimal parameters. The parameters for optimization were selected after evaluating the performance of a CNN with an experimental study, where the parameters were changed manually. As explained above, different CNN parameter values produce varying results for the same task, for this reason, the goal here is to find the optimal architecture. The parameters chosen to optimize here are the number of convolutional layers, filter size or filter dimensions used in each convolution operation, number of filters to extract the next feature map (number of convolution filters), total batch size (this value represents the number of images inserted into the CNN in each block).

The general methodology of the proposal here is shown in Fig. 1; the "training and optimization" blocks are the most important part of the whole process, where CNN is initialized for the integration of optimization parameters through the application of the MPSO algorithm. In this process, MPSO is initialized through parameters given for execution and particle generation. Each particle is a possible solution, and the position has optimized parameters so that each solution describes a complete CNN training. Maintaining the Integrity of the Specifications.

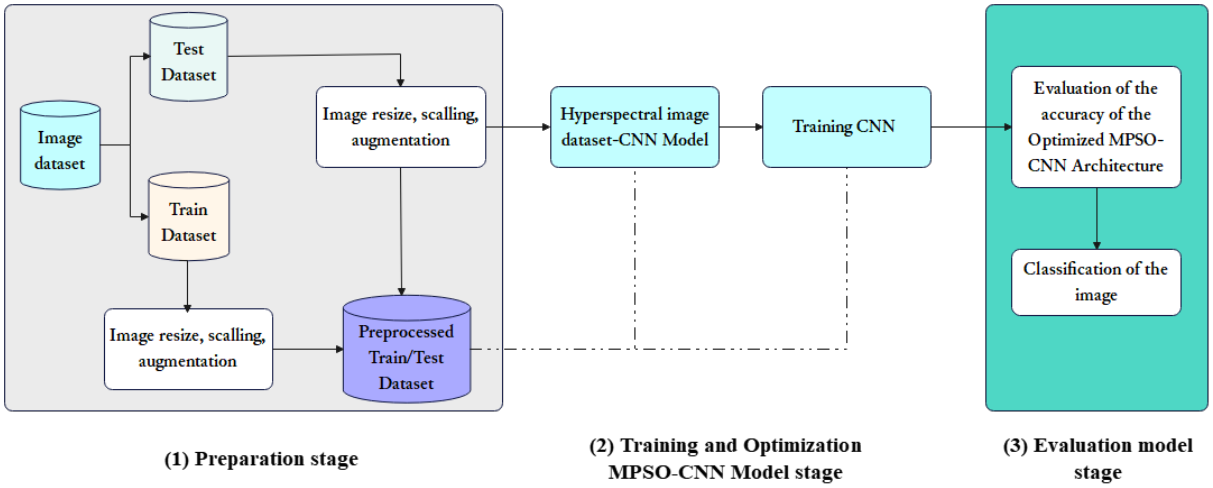

Fig. 1. Research Methodology MPSO-CNN Architecture

The training process is an iterative cycle that ends when the particles generated through MPSO are evaluated for each generation. The calculation cost is higher and depends on database size, particle size, number of iterations of PSO, and number of particles in each iteration. The steps for CNN optimization using the MPSO algorithm are shown in Fig. 2.

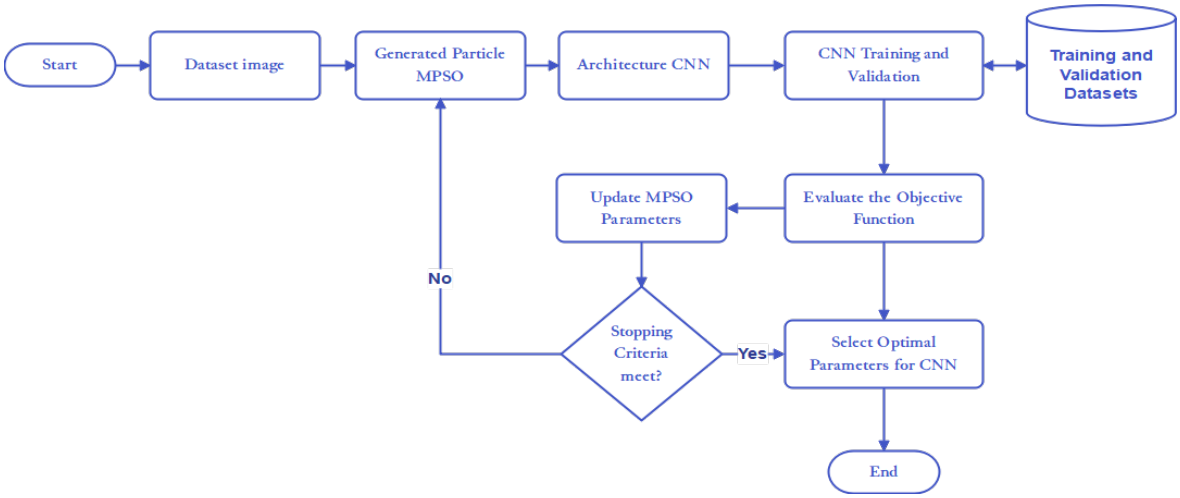

Fig. 2. Optimized Hyperparameter MPSO in CNN Architecture

The first step in the process involves selecting a suitable database for training the convolutional neural network (CNN). This study obtained the dataset from Kaggle, where images were processed and classified using CNN. Selecting an appropriate dataset is crucial for ensuring accurate model training and validation. Next, the particle population for the Modified Particle Swarm Optimization (MPSO) algorithm is generated. This includes determining key parameters such as the number of iterations, particles, inertial weights, cognitive constants (W1), and social constants (W2). These parameters are vital in guiding the optimization process to find the most suitable CNN architecture.

Once the MPSO parameters are set, the CNN architecture is initialized with optimized parameters obtained through MPSO. This includes defining the number of convolution layers, filter size, number of convolution filters, and batch size. After initialization, CNN is ready to process the selected database and begin training. During the training and validation phase, CNN reads and processes the input dataset by dividing it into training, validation, and testing subsets. The model undergoes iterative training, measuring its performance based on the classification rate. This classification rate is then assigned to the MPSO algorithm as part of the objective function.

The objective function is evaluated to determine the best model configuration. In this study, the classification rate serves as the only objective function. The MPSO algorithm continuously assesses different CNN configurations to find the optimal parameter settings that maximize classification accuracy. Following this, the MPSO parameters, specifically the velocity and position of particles, are updated. These updates depend on the best-known positions in the search space, including each particle's personal best position (Pbest) and the global best position (Gbest) of the entire swarm. The process repeats iteratively, evaluating all particles and updating their positions until the stopping criterion is met, which, in this case, is determined by the predefined number of iterations. This ensures the algorithm explores and converges toward the most optimal CNN architecture. Finally, the optimal solution is selected, where the best-performing particle represents the optimal CNN configuration. This final Gbest value signifies the best set of parameters for the CNN model, ensuring high classification accuracy and efficient deep learning performance.

2.1. Modified Particle Swarm Optimization (MPSO)

The standard particle swarm optimization (PSO) technique, first introduced by Eberhart and Kennedy (1995), is a stochastic optimization technique that is similar to the behavior of a flock of birds (birds flocking) or the sociological behavior of a group of humans. The basic idea of PSO is to involve a scenario where a flock of birds is searching for food sources in an area. All the birds don't know exactly where the food is, but with each iteration, they will find out how far the food will be from its original position. The best strategy will be followed by the bird closest to the food and also from the previous best position achieved. PSO is built with the optimization concept through a particle swarm. Each particle is in a position in the search space with a fitness value, evaluated through the fitness function to optimize each particle representing the quality of that position. All particles fly through the multidimensional search space by adjusting their position based on their own experience and that of their neighbors. PSO consists of a group (swarm) of particles randomly initialized as points in n-dimensional space in search of the optimal solution to an optimization problem. In PSO, the population is also known as a swarm; candidate solutions are coded as particles in the search space. PSO starts with a random initialization of the population of particles. Particles move through the search space in search of the optimal solution by updating each particle's position based on its own experience and that of the particles around it. During the movement, the current position of particle i is represented by a vector $X_i = X_{i1}, X_{i2}, \dots X_{iD}$, where D is the dimension of the search space. The velocity of a particle i is represented as $V_i = V_{i1}, V_{i2}, \dots V_{iD}$. Whereas the best previous position of a particle is recorded as the personal best and is named as Pbest and the best position obtained by the swarm as long as the global best is named as Gbest. PSO seeks the optimal solution by updating the position and velocity of each particle through equations (1) and (2).

$$v_{id}^{t+1} = v_{id}^{t} + c_1 * r_{1i} * (p_{id} - x_{id}^{t}) + c_2 * r_{2i} * \left(p_{gd} - x_{id}^{t}\right) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where t is the t^{th} iteration in the evolutionary process. $d \in D$ denotes the d^{th} dimension in the search space c_1 and c_2 are acceleration constants, representing the weighting of the stochastic acceleration term pushing each particle towards the $pbest$ (p_{id}) and $gbest$ (p_{gd}) positions. While r_{1i} and r_{2i} are random numbers that have a uniform distribution in the range [0,1]. The variables p_{id} and r_{gd} represent the elements of pbest and best in the d-dimensional. Vid is constrained by predefined maximum velocity

for modified particle swarm optimization algorithm (MPSO), $v_{max,d}\ to\ [-v_{max}, v_{max}]$, $pbest_i = (pbest_i^1, pbest_i^2, \ldots, pbest_i^d)$, representing the previous best position (best fitness value) of the particle to i. Whereas $gbest = (gbest^1, gbest^2, .. gbest^d)$ represents the previous best position of the population. The development of standard particle swarm optimization (PSO) was carried out by researchers due to the weaknesses of the PSO technique through proposing different strategies to increase its effectiveness, efficiency and robustness in dealing with optimization problems. The development includes four groups, namely: initialization and modification of parameters, particle position mutations, topological structures and integration with other techniques (hybrid).

3. Results and Discussion

3.1. Batik Dataset

The data used in this research uses image data of batik motifs originating from the North Coastal Region of Java. The results of the data collection are 6 classes of batik motifs from the North Coastal Region of Java and 1 negative class (when you do not recognize other than the six specified motifs). These motifs consist of Buketan, Singa Barong, Mega Mendung, Seven Rupa, Liong, Jlamprang and Negative class (when you don't recognize other than the six motifs above). The total amount of data is 805 images of batik motifs. The image dataset comes from Kaggle (https://www.kaggle.com/datasets/ilyamfsl/batik-pesisir-utara-jawa-with-negative-class-v2). The image data is then subjected to pre-processing, namely data augmentation, including zoom, random flip, random brightness, rotation, random distortion, and skew. The function of the augmentation process is to add new images from existing images. Another function of augmentation is to reduce the occurrence of overfitting during the testing process. Ultimately, it produced 500 images in the 6 batik motif classes and 700 in the Negative class. Fig. 3 shows the 6 batik motifs used in this research.

Fig. 3. Batik Motives (a) Mega Mendung (b) Singa Barong (c) Liong Pekalongan (d) Jlamprang (e) Buketan (f) Tujuh Rupa

3.2. Experiment Parameters

In this experiment, the CNN parameter settings involve the learning function, the activation function in the classification layer, the non-linear activation function, and the number of epochs. Meanwhile, the exact parameters in MPSO are the number of particles, number of iterations, inertial weights, cognitive and social constants, and position initialization using a logistic map. The dynamic parameters optimized using MPSO are the number of convolution layers, filters, and batch size.

3.3. PSO-CNN Hyperparameter Optimization Process

The hyperparameter optimization approach in this study is written as MPSO-CNN, which consists of implementing a particle with 4 places, with one position for each parameter optimized. Table 1 displays

the details of the particle composition where the 1 is X1 place corresponds to the number of layers with a search space from 1 to n.

Table 1. Setting CNN Parameters

No.	Parameter	Values
1.	Epochs	10
2.	Activation function	Softmax
3.	Non-linearity activation function	ReLU
4.	Learning function	Adam

The method can produce an architecture with a minimum of one layer and a maximum of n layers, for this study $n = 3$ was used. Place 2 is X2 which represents the number of convolution filters for characteristic extraction, with a search space from 32 to 18 filters. 3rd place is X3 is the filter size; search space from 1 to 4 where this value represents the position, the value obtained is mapped with the value from Table 2 to get the filter size (i.e. the particle generates a value of 1 which represents a filter size of [3 x 3], to get a value of the 2 filter sizes will be [5 x 5] and so on, accordingly, for each value.

Table 2. Setting Parameter MPSO

No.	Parameter	Values
1.	Particles	6
2.	Iteration	20
3.	Inertia Weight (W)	0.1
4.	Social Constant (C1)	2
5.	Cognitive Constant (C2)	2

The last position is X4 which represents the batch size (x4). Initialization involves a search space range from 32 to 256. This optimization process maintains parameter consistency between layers in the same state. After execution, MPSO generates a particle with 3 convolutional layers (X1), 50 filters (X2), a filter dimension of 3 x 3 (X3), and a batch size of 50 (X4). the same number of filters (X2) and filer size (X3) will be applied to the three convolution layers of the CNN. Fig. 4 shows and Setting parameter in the MPSO-CNN approach shown in Table 3.

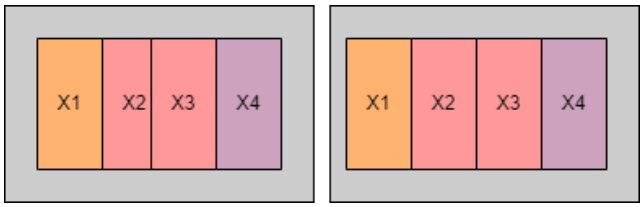

Fig. 4. Particle Structure in MPSO-CNN

Table 3. Setting Parameter in the MPSO-CNN Approach

Particle	Hyperparameter	Search Space
X1	Number of Convolutional layers	[1, 3]
X2	Number of filters	[32, 128]
X3	Filter Size	[1, 4]
X4	Batch Size	[40, 256]

In this process, the classification rate (precision) gives the objective function, which returns the CNN after being trained with the parameters generated by the PSO.

3.4. Model Testing

3.4.1. CNN Model Testing

Table 4 shows the results of tests on the data testing the batik image dataset and the CNN model confusion matrix [34].

Table 4. Confusion Matrix CNN Model

Motive	Precision	Recall	F1-Score	Support
Buketan	0.8113	0.8600	0.8350	50
Jamplang	0.9091	0.8000	0.8511	50
Liong	0.8000	0.4000	0.5333	50
Mega Mendung	0.6866	0.9200	0.7863	50
Negatif	0.6495	0.8873	0.7500	71
Singa Barong	0.7818	0.8600	0.8190	50
Tujuh Rupa	0.9667	0.5800	0.7250	50
Accuracy			**0.7665**	371

The accuracy of the CNN model when used to classify batik image datasets is 77%. Confusion Matrix show as Fig. 5.

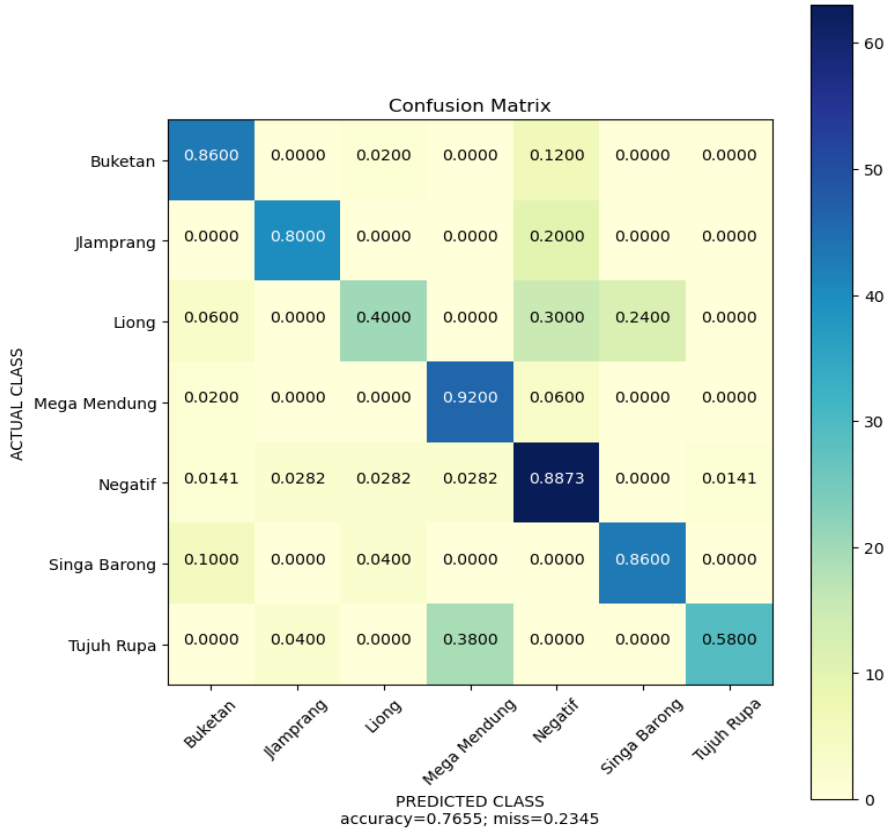

Fig. 5. Confusion Matrix

3.4.2. MPSO-CNN Model Testing

Table 5 shows the testing results on the batik dataset testing data and the MPSO-CNN model confusion matrix. The accuracy of the MPSO-CNN model when used to classify the batik motive dataset is 94% with a layer number of 3, a filter size of [7 x 7], and a Batch size of 256. From the results of tests on 2 models, the CNN model and the MPSO-CNN Model, the highest accuracy was obtained when classifying batik images using the MPSO-CNN model, namely 94%. The increased level of accuracy when compared to ordinary CNN models indicates that the use of optimized architecture has a good effect on the level of accuracy obtained.

Table 5. Confusion Matrix MPSO-CNNModel

Motive	Precision	Recall	F1-Score	Support
Buketan	0.7101	0.9800	0.9238	50
Jamplang	0.7959	0.7800	0.8878	50
Liong	0.8696	0.4000	0.7488	50
Mega Mendung	0.8929	1.0000	0.9484	50
Negatif	0.7556	0.9577	0.9487	71
Singa Barong	0.9574	0.9000	0.9288	50
Tujuh Rupa	0.9730	0.7200	0.9296	50
Accuracy			**0.94032**	371

4. Conclusion

This paper proposes an algorithm-modified particle swarm optimization (MPSO) for optimized hyperparameters in CNN. This also helps the PSO algorithm to continue searching other areas in the practical settlement space. Compared to different PSO variants, the PSO proposed in this paper shows that MPSO has better stability, quality of final completion, and convergence speed. From the results of tests carried out on 2 models, namely the CNN model and the MPSO-CNN model, it was found that the highest accuracy was obtained when classifying batik images using the MPSO-CNN model VGG16, namely 94%. The increased level of accuracy when compared to ordinary CNN models indicates that the use of MPSO for optimized hyperparameters has a good effect on the level of accuracy obtained. As further research, we plan to implement the MPSO algorithm that results in classification images in the Agricultural Industry, especially for the classification of chili diseases based on leaves.

Acknowledgment

The authors thank RISTEKDIKTI KEMDIKBUD for providing grants through a fundamental research (PRF) scheme in 2023.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.
Funding statement. RISTEKDIKTI KEMDIKBUD funded this research by providing grants through a fundamental research (PRF) scheme in 2023.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, no. February, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.

[2] S. Kumar, B. Sharma, V. K. Sharma, H. Sharma, and J. C. Bansal, "Plant leaf disease identification using exponential spider monkey optimization," *Sustain. Comput. Informatics Syst.*, vol. 28, 2020, doi: 10.1016/j.suscom.2018.10.004.

[3] A. Waheed, M. Goyal, D. Gupta, A. Khanna, A. E. Hassanien, and H. M. Pandey, "An optimized dense convolutional neural network model for disease recognition and classification in corn leaf," *Comput. Electron. Agric.*, vol. 175, no. January, p. 105456, 2020, doi: 10.1016/j.compag.2020.105456.

[4] D. Rahadiyan, S. Hartati, Wahyono, and A. P. Nugroho, "Feature aggregation for nutrient deficiency identification in chili based on machine learning," *Artif. Intell. Agric.*, vol. 8, pp. 77–90, Jun. 2023, doi: 10.1016/j.aiia.2023.04.001.

[5] E. Elbasi *et al.*, "Artificial Intelligence Technology in the Agricultural Sector: A Systematic Literature Review," *IEEE Access*, vol. 11, pp. 171–202, 2023, doi: 10.1109/ACCESS.2022.3232485.

[6] T. A. Pham, V. Q. Tran, and H. L. T. Vu, "Evolution of Deep Neural Network Architecture Using Particle Swarm Optimization to Improve the Performance in Determining the Friction Angle of Soil," *Math. Probl. Eng.*, vol. 2021, pp. 19–22, 2021, doi: 10.1155/2021/5570945.

[7] N. Selvam, Y. Nagesa, and F. Negesa, "Deep Learning Approach with Optimization Algorithm for Reducing the Training and Testing Time in SAR Image Detection and Recognition," *Indian J. Sci. Technol.*, vol. 15, no. 9, pp. 371–385, 2022, doi: 10.17485/ijst/v15i9.1266.

[8] X. Liu, C. Zhang, Z. Cai, J. Yang, Z. Zhou, and X. Gong, "Continuous particle swarm optimization-based deep learning architecture search for hyperspectral image classification," *Remote Sens.*, vol. 13, no. 6, pp. 1–22, 2021, doi: 10.3390/rs13061082.

[9] G. Zhang, S. Zhao, W. Li, Q. Du, Q. Ran, and R. Tao, "HTD-Net: A deep convolutional neural network for target detection in hyperspectral imagery," *Remote Sens.*, vol. 12, no. 9, pp. 1–21, 2020, doi: 10.3390/RS12091489.

[10] L. Mukku and J. Thomas, "Deep learning-based cervical lesion segmentation in colposcopic images," *Appl. Eng. Technol.*, vol. 3, no. 1, pp. 16–25, Apr. 2024, doi: 10.31763/aet.v3i1.1345.

[11] H. K. Jeon, S. Kim, J. Edwin, and C. S. Yang, "Sea fog identification from GOCI images using CNN transfer learning models," *Electron.*, vol. 9, no. 2, pp. 1–9, 2020, doi: 10.3390/electronics9020311.

[12] T. Majeed, R. Rashid, D. Ali, and A. Asaad, "Issues associated with deploying CNN transfer learning to detect COVID-19 from chest X-rays," *Phys. Eng. Sci. Med.*, vol. 43, no. 4, pp. 1289–1303, 2020, doi: 10.1007/s13246-020-00934-8.

[13] J. Geldmacher, C. Yerkes, and Y. Zhao, "Convolutional neural networks for feature extraction and automated target recognition in synthetic aperture radar images," *CEUR Workshop Proc.*, vol. 2819, pp. 86–91, 2020. [Online]. Available at: https://apps.dtic.mil/sti/trecms/pdf/AD1114517.pdf.

[14] D. P. Ismi and N. Khoirunnisa, "Enhancing the performance of heart arrhythmia prediction model using Convolutional Neural Network based architectures," *Sci. Inf. Technol. Lett.*, vol. 5, no. 2, pp. 1–12, 2024, [Online]. Available at: https://pubs2.ascee.org/index.php/sitech/article/view/1794.

[15] H. Jayadianti, B. A. Arianti, N. H. Cahyana, S. Saifullah, and R. Dreżewski, "Improving sentiment analysis on PeduliLindungi comments: a comparative study with CNN-Word2Vec and integrated negation handling," *Sci. Inf. Technol. Lett.*, vol. 4, no. 2, pp. 75–89, Nov. 2023, doi: 10.31763/sitech.v4i2.1184.

[16] A. H. Pratomo, N. H. Cahyana, and S. N. Indrawati, "Optimizing CNN hyperparameters with genetic algorithms for face mask usage classification," *Sci. Inf. Technol. Lett.*, vol. 4, no. 1, pp. 54–64, May 2023, doi: 10.31763/sitech.v4i1.1182.

[17] M. Masum, H. Shahriar, and H. M. Haddad, "A Transfer Learning with Deep Neural Network Approach for Network Intrusion Detection," vol. 12, no. 1, pp. 1087–1095, 2021, doi: 10.20533/ijicr.2042.4655.2021.0132.

[18] T. B. Shahi, C. Sitaula, A. Neupane, and W. Guo, "Fruit classification using attention-based MobileNetV2 for industrial applications," *PLoS One*, vol. 17, no. 2, p. e0264586, Feb. 2022, doi: 10.1371/journal.pone.0264586.

[19] H. Ali Khan, W. Jue, M. Mushtaq, and M. Umer Mushtaq, "Brain tumor classification in MRI image using convolutional neural network," *Math. Biosci. Eng.*, vol. 17, no. 5, pp. 6203–6216, 2020, doi: 10.3934/mbe.2020328.

[20] Z. P. Jiang, Y. Y. Liu, Z. E. Shao, and K. W. Huang, "An improved VGG16 model for pneumonia image classification," *Appl. Sci.*, vol. 11, no. 23, 2021, doi: 10.3390/app112311185.

[21] S. W. P. Listio, "Performance of Deep Learning Inception Model and MobileNet Model on Gender Prediction Through Eye Image," *Sinkron*, vol. 7, no. 4, pp. 2593–2601, 2022, doi: 10.33395/sinkron.v7i4.11887.

[22] R. U. Khan, X. Zhang, R. Kumar, and E. O. Aboagye, "Evaluating the performance of ResNet model based on image recognition," *ACM Int. Conf. Proceeding Ser.*, pp. 86–90, 2018, doi: 10.1145/3194452.3194461.

[23] N. Hasan, Y. Bao, A. Shawon, and Y. Huang, "DenseNet Convolutional Neural Networks Application for Predicting COVID-19 Using CT Image," *SN Comput. Sci.*, vol. 2, no. 5, pp. 1–11, 2021, doi: 10.1007/s42979-021-00782-7.

[24] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012, [Online]. Available at: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

[25] S. Rana, S. Jasola, and R. Kumar, "A review on particle swarm optimization algorithms and their applications to data clustering," *Artif. Intell. Rev.*, vol. 35, no. 3, pp. 211–222, 2011, doi: 10.1007/s10462-010-9191-9.

[26] J. R. Challapalli and N. Devarakonda, "A novel approach for optimization of convolution neural network with hybrid particle swarm and grey wolf algorithm for classification of Indian classical dances," *Knowl. Inf. Syst.*, vol. 64, no. 9, pp. 2411–2434, 2022, doi: 10.1007/s10115-022-01707-3.

[27] S. Oh, J. Yoon, Y. Choi, Y. A. Jung, and J. Kim, "Genetic Algorithm for the Optimization of a Building Power Consumption Prediction Model," *Electron.*, vol. 11, no. 21, 2022, doi: 10.3390/electronics11213591.

[28] R. D. Logan *et al.*, "Hyperspectral imaging and machine learning for monitoring produce ripeness," p. 25, 2020, doi: 10.1117/12.2560968.

[29] X. Zhang, X. Wang, Q. Kang, and J. Cheng, "Differential mutation and novel social learning particle swarm optimization algorithm," *Inf. Sci. (Ny).*, vol. 480, pp. 109–129, 2019, doi: 10.1016/j.ins.2018.12.030.

[30] E. Çomak, "A particle swarm optimizer with modified velocity update and adaptive diversity regulation," *Expert Syst.*, vol. 36, no. 1, pp. 1–18, 2019, doi: 10.1111/exsy.12330.

[31] Murinto, A. Prahara, and E. I. H. Ujianto, "Multilevel Thresholding Image Segmentation Based-Logarithm Decreasing Inertia Weight Particle Swarm Optimization," *Int. J. Adv. Soft Comput. its Appl.*, vol. 14, no. 3, pp. 64–77, 2022, doi: 10.15849/IJASCA.221128.05.

[32] W. H. Bangyal, H. T. Rauf, H. Batool, S. A. Bangyal, J. Ahmed, and S. Pervaiz, "An improved Particle Swarm Optimization algorithm with Chi-Square mutation strategy," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 3, pp. 481–491, 2019, doi: 10.14569/IJACSA.2019.0100362.

[33] Murinto, A. Harjoko, S. Hartati, and P. Danoedoro, "Modified Particle Swarm Optimization With Chaos-Based Particle Initialization and Logarithmic Decreasing Inertia Weight," *ICIC Express Lett.*, vol. 16, no. 2, pp. 195–203, 2022. [Online]. Available at: http://www.icicel.org/ell/contents/2022/2/el-16-02-11.pdf.

[34] I. M. Wismadi, D. C. Khrisne, and I. M. A. Suyadnya, "Detecting the Ripeness of Harvest-Ready Dragon Fruit using Smaller VGGNet-Like Network," *J. Electr. Electron. Informatics*, vol. 3, no. 2, p. 35, 2020, doi: 10.24843/jeei.2019.v03.i02.p01.