Bulletin of Electrical Engineering and Informatics
Vol. 13, No. 2, April 2024, pp. 1268~1275
ISSN: 2302-9285, DOI: 10.11591/eei.v13i2.6112

Convolution neural network hyperparameter optimization using modified particle swarm optimization

Muhammad Munsarif, Muhammad Sam'an, Andrian Fahrezi
Department of Informatics, Universitas Muhammadiyah Semarang, Semarang, Indonesia

Article Info

Article history:

Received Feb 28, 2023
Revised Jul 16, 2023
Accepted Sep 11, 2023

Keywords:

Convolutional neural network
Handwritten digit
Hyperparameter optimization
Modified National Institute of Standards and Technology
Particle swarm optimization

ABSTRACT

Based on the literature review, a convolutional neural network (CNN) is one of the deep learning techniques most often used for classification problems, especially image classification. Various approaches have been proposed to improve accuracy performance. In CNN architecture, parameter determination is very influential on accuracy performance. Particle swarm optimization (PSO) is a type of metaheuristic algorithm widely used for hyperparameter optimization. PSO convergence is faster than genetic algorithm (GA) and attracts many researchers for further studies such as genetic algorithms and ant colony. In PSO, determining the value of the weight parameter is very influential on accuracy. Therefore, this paper proposes CNN hyperparameter optimization using modified PSO with linearly decreasing randomized weight. The experiments use the modified National Institute of Standards and Technology (MNIST) dataset. The accuracy of the proposed method is superior, and the execution time is slower to random search. In epoch 1, epoch 3, and epoch 5, the proposed method is superior to baseline CNN, linearly decreasing weight PSO (LDW-PSO), and RL-based optimization algorithm (ROA). Meanwhile, the accuracy performance of the proposed method is superior to previous studies, namely LeNet-1, LeNet-2, LeNet-3, PCANet-2, RANDNet-2, CAE-1, CAE-2, and bee colony. Otherwise, lost to PSO-CNN, distributed PSO (DPSO), recurrent CNN, and CNN-PSO. However, the four methods have a complex architecture and wasteful execution time.

Corresponding Author:

Muhammad Munsarif
Department of Informatics, Universitas Muhammadiyah Semarang
Jl. Kedung Mundu Raya No. 18, Semarang, 50273 Jawa Tengah, Indonesia
Email: m.munsarif@unimus.ac.id

1. INTRODUCTION

Literally, convolutional neural network (CNN) is recognized as one of the most powerful deep learning models for accurately predicting handwritten recognition with high accuracy [1], [2]. Compared to other deep learning models such as deep neural network (DNN), recurrent neural network (RNN), and artificial neural network (ANN) in image classification [3]–[5], CNN exhibits higher accuracy and faster execution time. The CNN architecture consists of two main components Figure 1: the convolution layer used for feature extraction and the Fully connected layer used for the classification process. The feature extraction process plays a crucial role in determining prediction accuracy, leading many researchers to explore various CNN architectural models that can achieve optimal performance [6]. One particular focus of research on CNN architectural models is the optimization of parameters using a hyperparameter approach [7].

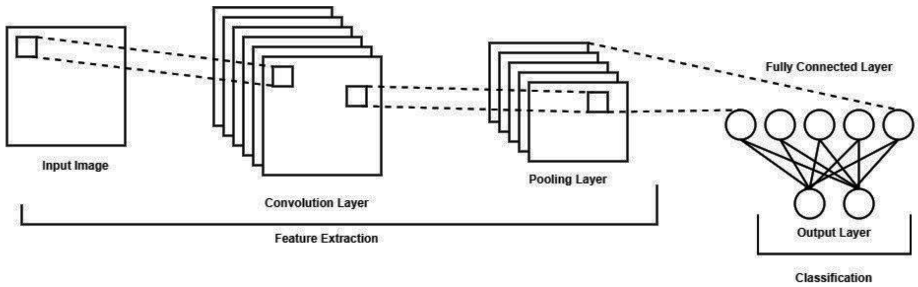


Figure 1. The block of CNN architecture

The convolution layer consists of a kernel that detects features of a certain size (usually 3×3 or 5×5). Each filter is convolved over all images with a specific stride value (typically 1 or 2). The presence of multiple convolution layers facilitates easier identification of deeper features. In CNN, the first convolution layer detects the edges of image pixels, while the second convolution layer detects simpler shapes and deeper features [8]. The output of the convolution process is connected through a non-linear activation function, commonly using the rectified linear unit (ReLU) as the default choice. Dimension reduction and preservation of data quality are achieved through the utilization of the MaxPooling layer, as depicted in Figure 2. Finally, a fully connected layer is added to the network as a classifier, often employing the multi-layer perceptron as the default choice.

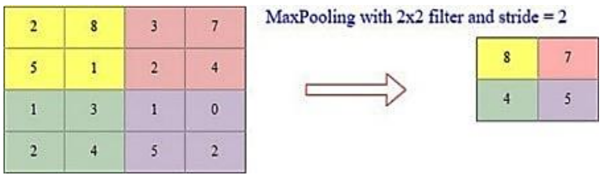


Figure 2. Max pooling with filter 2×2 and stride=2

The determination of optimal parameter values significantly impacts the performance of a CNN model. The importance of selecting appropriate hyperparameters in CNN models has been demonstrated in previous research. However, manual hyperparameter selection methods have limitations due to their time-consuming nature and lack of significant performance improvements [9]. Therefore, an automated approach to hyperparameter selection is necessary.

One effective automated approach for hyperparameter selection is the utilization of metaheuristic algorithms such as particle swarm optimization (PSO). PSO and other metaheuristic algorithms draw inspiration from natural phenomena and have been successfully applied to optimize CNN models [10], [11]. By leveraging PSO, researchers can efficiently explore the parameter search space and adaptively adjust particle positions to achieve optimal solutions [12], [13]. The strength of PSO lies in its ability to find optimal solutions relatively quickly [12]. Using PSO, researchers can improve the performance of CNN models through automated and efficient hyperparameter selection [14]. This algorithm helps overcome the limitations of manual selection methods, unlocks the potential to optimize complex parameters, produces superior, and effective CNN models.

In order to enhance the convergence performance of CNN models, researchers [15] have proposed an effective method known as linearly decreasing weight PSO (LDW-PSO) for optimizing hyperparameters. LDW-PSO utilizes parameter weights within the PSO algorithm to achieve higher efficiency. However, manual weight configuration has its limitations, including subjectivity and the lengthy optimization time required to reach optimal solutions. To overcome these limitations, an alternative approach called linearly decreasing randomized weight PSO (LDRW-PSO) has been introduced. LDRW-PSO automatically adjusts the weights using a linearly decreasing pattern, thereby improving the efficiency of hyperparameter optimization in CNN models. Additionally, LDRW-PSO expands the exploration of the parameter space by incorporating additional optimizer parameters such as Adamax, Nadam, RMSprop, Adadelta, Adgrad, and the "eLu" activation function. The performance of the proposed model is evaluated using metrics such as accuracy and execution time. By adopting the effective approach of LDRW-PSO, this research aims to address the limitations of manual weight configuration and optimize the hyperparameter selection process in CNN models.

The outline of this paper is as follows; section 2 describes PSO, CNN and related research. Section 3 describes method and section 4 describes the experiment and its results. The final section 5 describes concludes this paper.

2. RELATED WORK

CNN's performance improvement has been widely studied. AlexNet type CNN architecture to improve accuracy and manual configuration of new parameters [16]. On the other hand, configuring the parameters using the algorithm on the CNN architecture is already done. Genetic algorithm [17], evolutionary algorithm [18], and PSO [19] was used for hyperparameter optimization. Using the CIFAR-10, they claim that the CNN classification based on hyperparameter optimization reaches 18.53% for 13 layers and 22.5% for 8 layers. However, this approach is more computationally wasteful. The accuracy of 98.97% with modified National Institute of Standards and Technology (MNIST) (when learning 5th epoch) and 73.40% with CIFAR-10 (when learning 10th epoch) by using new Q-learning RL-based optimization algorithm (ROA) for CNN hyperparameter optimization [20].

This article uses filter size, kernel, optimizer, batch normalization, dense, and activation functions for optimization hyperparameters. Parameter configuration automatically using distributed PSO (DPSO) can get the best CNN model globally with an accuracy of 99.25% for PSO and 99.2% for DPSO [21]. Parameter configuration can increase accuracy from 0.7% to 5.7% on Alexnet-CNN [22]. With CIFAR-10 dataset, the hyperparameter optimization approach using PSO with three mechanisms, namely vectorization acceleration coefficient, compound normal significant distribution, and linear estimation scheme, can produce a classification error of 8.67% [23]. 15 particles and 10 iterations on CNN hyperparameter optimization using PSO achieved 0.87% error testing with MNIST handwritten digit [24]. 30 executions on CNN hyperparameter optimization obtained the best loss value of 0.04651067 [25]. Hybrid multi-level PSO-CNN produces good performance with 99.13% accuracy for MINST dataset [26]. Based on the related work, it can be seen that PSO is proven to improve CNN performance, especially hyperparameter optimization. Therefore, this article investigates evolving CNN based on LDRW-PSO hyperparameter optimization.

3. METHOD

Evolutionary algorithms (EA) have been shown to perform well for optimization problems. PSO as an EA that is often used for parameter optimization. The PSO algorithm contains several steps in detail Algorithm 1 where iter as the number of iterations, N as total number of particles. Each particle can update its position and velocity. Personal best (pBest) is defined by the best position. The best particle position whole group is defined by gBest and pBest each particle moves towards its new position, which is close to gBest and pBest, so that the optimal solution is found. The velocity (v) in (1) can update the particles move.

$$v_{id}(t+1) = w \times v_{id}(t) + c_p \times r_p \times (pBest_{id} - x_{id}(t)) + c_g \times r_g(gBest_{id} - x_{id}(t)) \quad (1)$$

where v_{id} as velocity of $i-th$ particle in d-dimension, x as the current particle position, c_p and c_g as predefined constants called acceleration factors, r_p and r_g as random number in [0,1]. Next, the update position is formulated as (2):

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

w as a weight parameter has a significant effect on convergence. w is formulated as (3):

$$w(iter) = \max(iter)\frac{\max(iter)-iter}{\max(iter)}(\max w - \min w) + \min w \quad (3)$$

where $\max(iter)$ is the maximum number of iterations the PSO is allowed to continue, $\max w$ and $\min w$ are the maximum and minimum weights to be specified.

Algorithm 1. Pseudocede of PSO

```
Input: iter =0, v dan x of all particles, pBest and gBest
while iter ≤ max(iter) do
 for i = 1 to N do
 for j = 1 to D do
 Update the velocity and position of the particles by using (1) and (2)
 end
```

```
  Calculate the evaluation value of particle i
  if f(x_i^{iter+1}) < f(pBest_i^{iter+1}) then
  pBest_i^{iter+1} = x_i^{iter+1}
  end
  end
  k = argmin f (pBest_i^{iter+1})
  if (x_i^{iter+1}) < f(pBest_i^{iter+1}) then
  pBest_i^{iter+1} = x_i^{iter+1}
  end
  t = t + 1
end
```

In order to design a CNN architecture based on hyperparameter optimization, this article uses improved PSO with linearly decreasing randomized weight. The CNN architecture of the LeNet-5 type is enhanced by optimizing parameters such as filter size, kernel, optimizer, dense, and activation functions. This article also optimizes batch normalization and epochs for the best accuracy. In detail, Table 1 shows the configuration of the hyperparameter values.

Table 1. The configuration of hyperparameter value in CNN architecture

Hyperparameter	Baseline	Optimization value
Number of filters in C1	6	4 – 100
Number of filters in C2	16	4 – 100
Size of kernel in C1	5	3, 5, 7
Size of kernel in C2	5	3, 5, 7
Activation function in C1	“sigmoid”	“sigmoid”, “tanh”, “relu”, “selu”, “elu”
Activation function in C2	“sigmoid”	“sigmoid”, “tanh”, “relu”, “selu”, “elu”
Activation function in FC1	“sigmoid”	“sigmoid”, “tanh”, “relu”, “selu”, “elu”
Activation function in FC2	“sigmoid”	“sigmoid”, “tanh”, “relu”, “selu”, “elu”
Number of neurons in FC1	120	4 – 200
Number of neurons in FC2	84	4 – 200
Batch size in the training	10	10 – 100
Optimizer	SGD	SGD, Adam, RMSprop, Adadelta, Adagrad, Nadam, and Adamax

The accuracy value is a fitness function to determine the optimal hyperparameter value. The flow chart of the proposed approach is shown in Figure 3. The experimental process begins by adjusting the position and speed of each particle. Secondly, every particle is executed by CNN. The position, speed, pBest, and gBest are updated based on the accuracy of step 2. This process is repeated until the best gBest parameter is found.

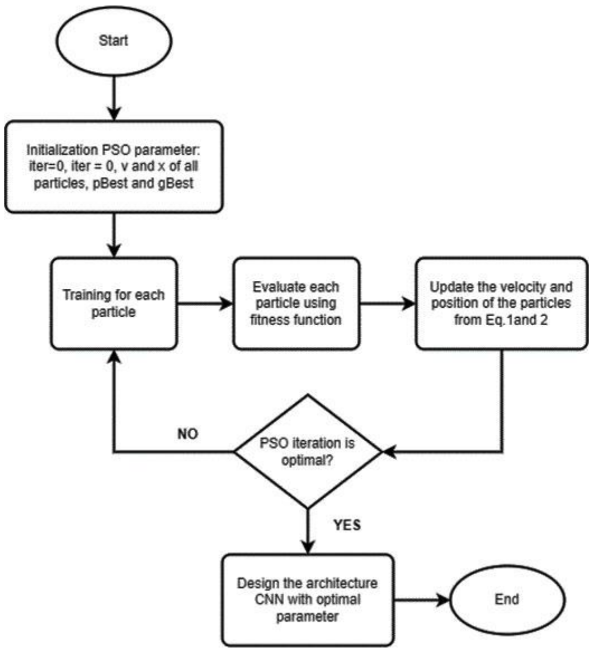


Figure 3. The flowchart of proposed method

4. RESULTS AND DISCUSSION

This paper uses a handwritten size of 28×28 pixels grayscale from 0-9 divided by 60.000 training and 10.000 testing on the MNIST dataset to evaluate the proposed algorithm. A sample of MNIST dataset is shown in Figure 4. In this experiment, LDRW-PSO is set swarm size and the number of iterations is 10, acceleration factors (c1 and c2) are 2.0. In the experiment, optimization was carried out with LDRW-PSO every 5 epochs, and learning was done using the obtained parameters. A random search algorithm is also presented to compare the prediction accuracy and execution time shown in Figure 5.

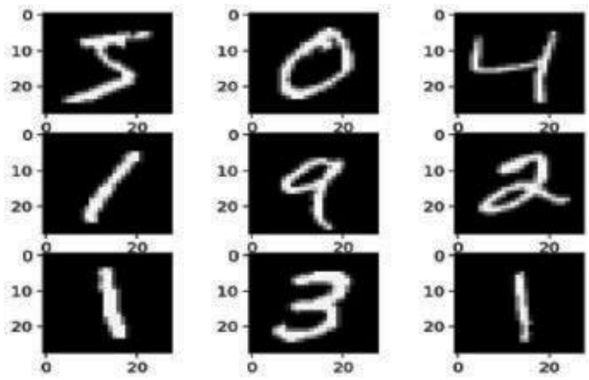
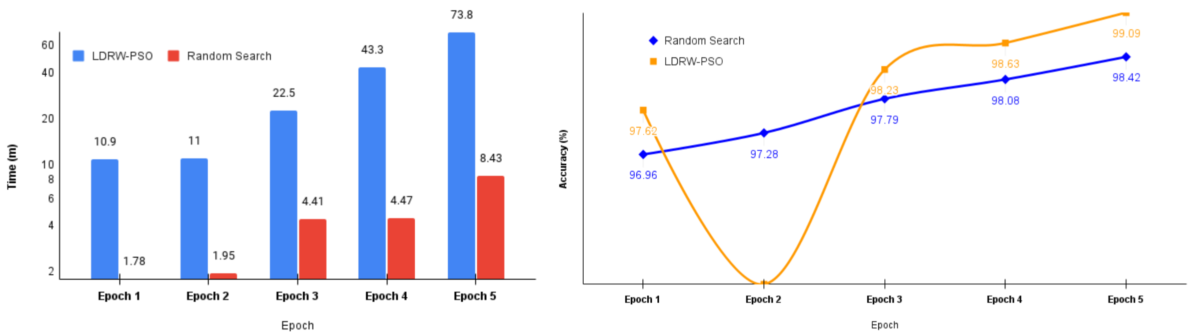


Figure 4. A sample of handwritten digit

Figure 5(a) shows that LDRW-PSO is superior to random search on accuracy performance, while Figure 5(b) shows that LDRW-PSO is slower than random search on execution time performance. Accuracy performance is affected by hyperparameter optimization, where Adamax optimizes LDRW-PSO and Nadam optimizes random search. Based on convergence, Adamax is more optimized than Nadam.



(a) (b)

Figure 5. The comparison performance of LDRW-PSO and random search; (a) time execution and (b) accuracy

Figure 6 illustrates the accuracy of CNN optimized by LDRW-PSO vs the accuracy of CNN optimized by the existing algorithm (baseline CNN, LDW-PSO [10], and ROA [15]). LDRW-PSO outperformed the existing algorithm in epoch 1, epoch 2, and epoch 5. The best accuracy on CNN optimized by LDRW-PSO when learning 5 epochs is 99.09%, which is better than 97.62%, 98.95%, and 98.97% on CNN without optimization, CNN optimized by LDW-PSO and ROA, respectively.

Furthermore, this paper also compares the proposed method vs previous studies are detailed in Table 2. PSO-CNN that automatically searches for meaningful CNNs architectures with PSO algorithm, DPSO, LeNet-1, LeNet-4, LeNet-5, recurrent CNN that incorporates recurrent connections into each convolutional layer, PCANet-2, RANDNet-2 that employed the principal component, binary hashing, and block-wise histograms analysis to the deep learning network, CAE-1 and CAE-2 that present an approach for training deterministic auto-encoders, PSO-CNN and bee colony, LDW-PSO, ROA. Table 2 shows that the proposed method produces an accuracy of 99.09% and can be improved with more learning epochs. This accuracy can compete with previous studies except for psoCNN, which creates an accuracy of 99.56%. PSO-CNN has a more complex architecture and complicated coding strategy so that it consumes extravagant execution time. DPSO produces an accuracy of 99.20% with almost the same characteristics as PSO-CNN, except in the mixed variables. In addition, recurrent CNN has a test accuracy of 99.69% but cannot produce the best configuration, and the hyperparameter range is limited, meaning that this model is not optimized because it uses repeated connections. LeNet-1, LeNet-4, and LeNet-5 obtained a testing accuracy of 98.30%, 98.90%, and 99.05%, respectively. These networks have a small number of parameters with different types of a last-layer classifier. Principal component, binary hashing, block-wise histogram analysis used for deep learning network on PCANet-2 resulted in 98.96% accuracy, and RANDNet-2 yielded 98.73% accuracy. The auto-encoder deterministic learning or CAE-1 and CAE-2 yielded 97.17%, and 97.52%, respectively. Accuracy on PSO-CNN is 99.13%. This model uses 100 epochs and limited hyperparameters. i.e., convolution layers number is 2, and convolution kernel size is 1-8, number of kernels is in the range 1-128, neurons numbers in the fully connected layer is the range 1-300.

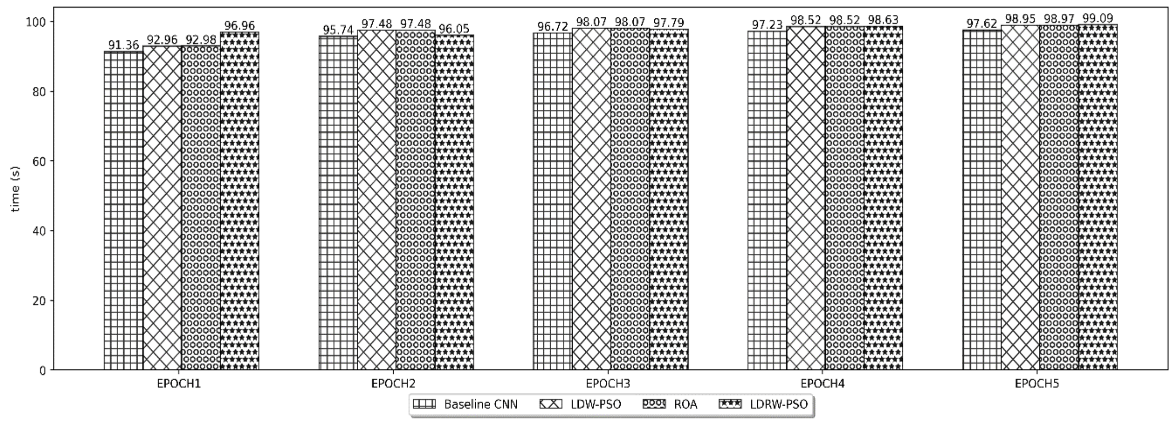


Figure 6. The comparison accuracy of proposed algorithm and existing algorithms

Table 2. The comparison of MINST classification accuracy of the state-of-the-art methods

Method	Accuracy (%)
PSO-CNN [13]	99.56
DPSO [16]	99.20
LeNet-1 [27]	98.30
LeNet-4 [22]	98.90
LeNet-5 [22]	99.05
Recurrent CNN [28]	99.69
PSONet-2 [29]	98.94
RANDNet-2 [24]	98.73
CAE-1 [30]	97.17
CAE-2 [25]	97.52
CNN-PSO [19]	99.13
Bee colony [19]	98.98
LDW-PSO [10]	98.85
ROA [15]	98.97
Proposed method	99.09

5. CONCLUSION

This paper proposes randomized weights for PSO, i.e., LDRW-PSO on CNN hyperparameter optimization. Based on experiments using the MNIST dataset. The accuracy of the proposed method is superior, and the execution time is slower to random search. In epoch 1, epoch 3, and epoch 5, the proposed method is superior to baseline CNN, LDW-PSO, and ROA. Meanwhile, the accuracy performance of the proposed method is superior to previous studies, namely LeNet-1, LeNet-2, LeNet-3, PCANet-2, RANDNet-2, CAE-1, CAE-2, and bee colony. Otherwise, lost to PSO-CNN, DPSO, recurrent CNN, and CNN-PSO. However, the four methods have a complex architecture and wasteful execution time.

REFERENCES

[1] S. Ali, Z. Shaukat, M. Azeem, Z. Sakhawat, T. Mahmood, and K. ur Rehman, "An efficient and improved scheme for handwritten digit recognition based on convolutional neural network," *SN Applied Sciences*, vol. 1, no. 9, p. 1125, Sep. 2019, doi: 10.1007/s42452-019-1161-5.

[2] M. A. Hossain and M. M. Ali, "Recognition of Handwritten Digit using Convolutional Neural Network (CNN)," *Global Journal of Computer Science and Technology*, pp. 27–33, May 2019, doi: 10.34257/GJCSTDVOL19IS2PG27.

[3] J. Qiao, G. Wang, W. Li, and M. Chen, "An adaptive deep Q-learning strategy for handwritten digit recognition," *Neural Networks*, vol. 107, pp. 61–71, Nov. 2018, doi: 10.1016/j.neunet.2018.02.010.

[4] S. Jain and R. Chauhan, "Recognition of Handwritten Digits Using DNN, CNN, and RNN," in *Communications in Computer and Information Science*, 2018, pp. 239–248, doi: 10.1007/978-981-13-1810-8_24.

[5] M. M. A. Ghosh and A. Y. Maghari, "A Comparative Study on Handwriting Digit Recognition Using Neural Networks," in *2017 International Conference on Promising Electronic Technologies (ICPET)*, Oct. 2017, pp. 77–81, doi: 10.1109/ICPET.2017.20.

[6] T. Ghosh, M.-H.-Z. Abedin, H. Al Banna, N. Mumenin, and M. A. Yousuf, "Performance Analysis of State of the Art Convolutional Neural Network Architectures in Bangla Handwritten Character Recognition," *Pattern Recognition and Image Analysis*, vol. 31, no. 1, pp. 60–71, Jan. 2021, doi: 10.1134/S1054661821010089.

[7] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)," *Sensors*, vol. 20, no. 12, pp. 1–18, Jun. 2020, doi: 10.3390/s20123344.

[8] A. K. Sharma, P. Thakkar, D. M. Adhyaru, and T. H. Zaveri, "Handwritten Gujarati Character Recognition Using Structural Decomposition Technique," *Pattern Recognition and Image Analysis*, vol. 29, no. 2, pp. 325–338, Apr. 2019, doi: 10.1134/S1054661819010061.

[9] N. Q. Ann, D. Pebrianti, M. F. Abas, and L. Bayuaji, "Automated-tuned hyper-parameter deep neural network by using arithmetic optimization algorithm for Lorenz chaotic system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 2, pp. 2167–6176, Apr. 2023, doi: 10.11591/ijece.v13i2.pp2167-2176.

[10] I. Cholissodin and S. Sutrisno, "Prediction of rainfall using improved deep learning with particle swarm optimization," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2498–2504, Oct. 2020, doi: 10.12928/telkomnika.v18i5.14665.

[11] M. Ashikuzzaman, W. Akram, M. M. I. Anik, M. Hasan, M. S. Ali, and T. Jabid, "PSO-ANN in preventing traffic collisions: a comparative study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, pp. 1796–1803, Dec. 2021, doi: 10.11591/ijeecs.v24.i3.pp1796-1803.

[12] W. Hu and G. G. Yen, "Adaptive Multiobjective Particle Swarm Optimization Based on Parallel Cell Coordinate System," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 1–18, Feb. 2015, doi: 10.1109/TEVC.2013.2296151.

[13] G. L. F. da Silva, T. L. A. Valente, A. C. Silva, A. C. de Paiva, and M. Gattass, "Convolutional neural network-based PSO for lung nodule false positive reduction on CT images," *Computer Methods and Programs in Biomedicine*, vol. 162, pp. 109–118, Aug. 2018, doi: 10.1016/j.cmpb.2018.05.006.

[14] Wei-Chang Yeh, "New Parameter-Free Simplified Swarm Optimization for Artificial Neural Network Training and its Application in the Prediction of Time Series," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 661–665, Apr. 2013, doi: 10.1109/TNNLS.2012.2232678.

[15] T. Serizawa and H. Fujita, "Optimization of Convolutional Neural Network Using the Linearly Decreasing Weight Particle Swarm Optimization," *arXiv preprint arXiv:2001.05670*, 2020, doi: 10.48550/arXiv.2001.05670.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[17] S. Loussaief and A. Abdelkrim, "Convolutional Neural Network Hyper-Parameters Optimization based on Genetic Algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 252–266, 2018, doi: 10.14569/IJACSA.2018.091031.

[18] F. E. F. Junior and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification," *Swarm and Evolutionary Computation*, vol. 49, pp. 62–74, Sep. 2019, doi: 10.1016/j.swevo.2019.05.010.

[19] T. Sinha, A. Haidar, and B. Verma, "Particle Swarm Optimization Based Approach for Finding Optimal Values of Convolutional Neural Network Parameters," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2018, pp. 1–6, doi: 10.1109/CEC.2018.8477728.

[20] F. M. Talaat and S. A. Gamel, "RL based hyper-parameters optimization algorithm (ROA) for convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 10, pp. 13349–13359, Oct. 2023, doi: 10.1007/s12652-022-03788-y.

[21] Y. Guo, J.-Y. Li, and Z.-H. Zhan, "Efficient Hyperparameter Optimization for Convolution Neural Networks in Deep Learning: A Distributed Particle Swarm Optimization Approach," *Cybernetics and Systems*, vol. 52, no. 1, pp. 36–57, Jan. 2021, doi: 10.1080/01969722.2020.1827797.

[22] T. Yamasaki, T. Honma, and K. Aizawa, "Efficient Optimization of Convolutional Neural Networks Using Particle Swarm Optimization," in *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, Apr. 2017, pp. 70–73, doi: 10.1109/BigMM.2017.69.

[23] 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.

[24] Z. Fouad, M. Alfonse, M. Roushdy, and A.-B. M. Salem, "Hyper-parameter optimization of convolutional neural network based on particle swarm optimization algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 6, pp. 3377–3384, Dec. 2021, doi: 10.11591/eei.v10i6.3257.

[25] A. Gaspar, D. Oliva, E. Cuevas, D. Zaldívar, M. Pérez, and G. Pajares, "Hyperparameter Optimization in a Convolutional Neural Network Using Metaheuristic Algorithms," in *Stud. Comput. Intell.*, 2021, pp. 37–59, doi: 10.1007/978-3-030-70542-8_2.

[26] P. Singh, S. Chaudhury, and B. K. Panigrahi, "Hybrid MPSO-CNN: Multi-level Particle Swarm optimized hyperparameters of Convolutional Neural Network," *Swarm and Evolutionary Computation*, vol. 63, p. 100863, Jun. 2021, doi: 10.1016/j.swevo.2021.100863.

[27] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[28] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3367–3375.

[29] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A Simple Deep Learning Baseline for Image Classification?," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015, doi: 10.1109/TIP.2015.2475625.
[30] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th international conference on international conference on machine learning*, 2011, pp. 833–840.

BIOGRAPHIES OF AUTHORS

Muhammad Munsarif received the Master Degree in Computer science in 2002 and Ph.D. degree in 2023 from Dian Nuswantoro University (UDINUS). Currently, he is a lecturer in Informatics Engineering at Muhammadiyah University, Semarang (UNIMUS). His research interests include computer vision, data science, and technopreneuership. He can be contacted at email: m.munsarif@unimus.ac.id.

Muhammad Sam'an received Bachelor Degree from Universitas Negeri Semarang and Master Degree from Universitas Diponegoro in Mathematics 2010 and 2016 respectively and now, he is studying in the postgraduate student, Faculty of Technology Management and Business, Universiti Tun Hussein Onn Malaysia (UTHM). His research interests are in optimization, fuzzy mathematics and computational mathematics. He can be contacted at email: muhammad.92sam@gmail.com.

Andrian Fahrezi is currently student of Informatics Engineering at Muhammadiyah University, Semarang (UNIMUS). His research interests include software engineering and data mining. He can be contacted at email: andrianfahrezi12@gmail.com.