JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

METAHEURISTIC HYPERPARAMETER OPTIMIZATION FOR DEEP LEARNING: A COMPARATIVE STUDY

Zulfahmi Syahputra[1*] ; Romi Fadillah Rahmat[2]

Computer System[1]
Universitas Pembangunan Panca Budi, Medan, Indonesia[1]
www.pancabudi.ac.id[1]
zulfahmi@dosen.pancabudi.ac.id[*]

Faculty of Computer Science and Information Technology[2]
Universitas Sumatera Utara, Medan, Indonesia[2]
www.usu.ac.id[2]
romi.fadillah@usu.ac.id

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract— *One of the primary challenges in developing effective deep learning models lies in identifying optimal hyperparameter configurations, particularly within high-dimensional and complex search spaces. Traditional tuning strategies, including grid search and random search, are widely known to be computationally inefficient and frequently yield suboptimal outcomes. To address this limitation, this study presents a comparative analysis of two metaheuristic algorithms — Particle Swarm Optimization (PSO) and Dwarf Mongoose Optimization (DMO) — as advanced alternatives for hyperparameter tuning in deep learning models trained on the CIFAR-10 dataset. Both algorithms were rigorously assessed using a comprehensive set of performance metrics, namely accuracy, precision, recall, and F1-score, supplemented by confusion matrix analysis to capture class-level behavior. Experimental findings confirm that both approaches yield substantial improvements in model performance. Notably, PSO demonstrated superior results, achieving a validation accuracy of 89.15% and a test accuracy of 87.82%, while DMO reached a final accuracy of 86.38%. In terms of optimization behavior, PSO exhibited greater convergence stability and more uniform class-level performance, whereas DMO proved more effective in broadly exploring the search space. Overall, this study reinforces the potential of metaheuristic-based optimization as a robust framework for hyperparameter tuning and underscores the critical role of optimization stability in achieving reliable model generalization.*

Keywords: *Deep Learning, Dwarf Mongoose Optimization, Hyperparameter Optimization, Model Performance, Particle Swarm Optimization.*

Intisari— *Salah satu tantangan utama dalam pengembangan model deep learning adalah menentukan konfigurasi hiperparameter yang optimal, khususnya ketika berhadapan dengan ruang pencarian yang luas dan kompleks. Pendekatan tradisional seperti grid search dan random search terbukti kurang efisien karena cenderung menghasilkan solusi yang tidak optimal. Untuk mengatasi keterbatasan tersebut, penelitian ini membandingkan dua algoritma metaheuristik Particle Swarm Optimization (PSO) dan Dwarf Mongoose Optimization (DMO) sebagai strategi alternatif dalam pencarian hiperparameter terbaik pada model deep learning berbasis dataset CIFAR-10. Kinerja kedua algoritma dievaluasi secara komprehensif menggunakan sejumlah metrik, yakni akurasi, presisi, recall, dan F1-score, serta diperkuat dengan analisis confusion matrix untuk menilai performa per kelas. Hasil yang diperoleh membuktikan bahwa penerapan kedua metode tersebut memberikan peningkatan performa yang nyata. Secara khusus, PSO unggul dengan akurasi validasi mencapai 89,15% dan akurasi uji sebesar 87,82%, sedangkan DMO menghasilkan akurasi akhir 86,38%. Dari sisi karakteristik, PSO terbukti lebih konsisten dan stabil dalam proses konvergensi, sementara DMO* *memperlihatkan kemampuan eksplorasi ruang pencarian yang lebih agresif. Secara keseluruhan, studi ini memperkuat argumen bahwa algoritma metaheuristik merupakan solusi yang menjanjikan untuk optimisasi hiperparameter, sekaligus menggarisbawahi peran krusial stabilitas dalam memastikan kemampuan generalisasi model yang baik.*

Kata Kunci*: Pembelajaran Mendalam, Optimasi Dwarf Mongoose, Optimasi Hyperparameter, Performa Model, Optimasi Particle Swarm.*

INTRODUCTION

Over the past decade, deep learning models have proven to be remarkably capable in handling a broad spectrum of complex data analysis tasks, including image classification, predictive modeling, and high-dimensional pattern recognition [1], [2]. The fundamental advantage of this approach stems from its capacity to construct layered feature representations through multi-level architectures and nonlinear activation functions, enabling efficient learning from large-scale, high-dimensional datasets. Nevertheless, the success of deep learning models in real-world applications is not solely dictated by network architecture design it is equally dependent on the appropriate selection of training hyperparameters. A significant obstacle in deploying deep learning systems is the high sensitivity of model performance to hyperparameter settings [3]. Parameters such as the learning rate and optimization configurations play a decisive role in governing training stability and the model's ability to generalize. Poorly chosen values can lead to slow convergence and unstable training dynamics [4]. Even minor variations in hyperparameter values can lead to significant differences in model outcomes, highlighting the critical role of hyperparameter tuning in deep learning workflows.

In practice, hyperparameter selection is still frequently performed using manual trial-and-error approaches. Such methods are not only subjective but also computationally inefficient, especially as the number of hyperparameters increases and the search space expands. Conventional search strategies such as grid search and random search provide more systematic alternatives; however, both suffer from fundamental limitations in scalability and efficiency [4], [5]. These limitations become increasingly critical when training deep learning models on large datasets under constrained computational resources. These challenges have driven the development of more adaptive and efficient hyperparameter optimization approaches capable of handling high-dimensional search spaces [6]. One promising direction is the use of metaheuristic algorithms [7]. Metaheuristics are inherently designed to strike a balance between exploration and exploitation within the search space [8], [9], enabling the discovery of near-optimal solutions without exhaustively evaluating the entire search space. This characteristic makes metaheuristics particularly suitable for optimizing deep learning hyperparameters, which often involve nonlinear and complex optimization landscapes.

Particle Swarm Optimization (PSO) is one of the most widely adopted metaheuristic algorithms for solving complex optimization problems. PSO models collective intelligence through a population of interacting particles that iteratively move toward optimal solutions based on individual and global experiences. Its simplicity and fast convergence have led to extensive adoption in machine learning optimization tasks, including deep learning hyperparameter tuning [10], [11]. However, several studies indicate that PSO is susceptible to premature convergence, especially in highly complex search spaces [12]. As an alternative, Dwarf Mongoose Optimization (DMO) has recently been introduced as a swarm-based metaheuristic inspired by the cooperative behavior of mongoose colonies in foraging and territorial defense [13]. DMO incorporates role-based population dynamics to enhance the balance between global exploration and local exploitation [14]. Recent studies have demonstrated the applicability of DMO and its variants to various optimization problems, including feature selection, global optimization, and engineering optimization [15], [16]. Nevertheless, its application to deep learning hyperparameter optimization remains relatively limited, and comparative evaluations against well-established metaheuristics such as PSO are still scarce [17].

Although numerous studies have explored metaheuristic-based hyperparameter optimization, a gap remains in controlled comparative evaluations between established algorithms and emerging approaches [8], [9]. Similar observations regarding performance sensitivity across algorithmic configurations have also been reported in optimization studies examining distance-based clustering dynamics. Furthermore, prior research often emphasizes improvements in final accuracy without systematically analyzing the role of optimization strategies in shaping model

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

performance. As a result, the relative contributions of different metaheuristic algorithms to deep learning performance remain insufficiently understood.

To address this gap, this study presents a comprehensive comparative framework for deep learning hyperparameter optimization using Particle Swarm Optimization (PSO) and Dwarf Mongoose Optimization (DMO). Unlike previous studies that primarily focus on accuracy improvement, this research systematically evaluates optimization behavior, convergence characteristics, classification performance, and computational effectiveness under controlled experimental settings. Furthermore, the study investigates how different metaheuristic search dynamics influence hyperparameter selection and model generalization performance on the CIFAR-10 dataset. Through this controlled experimental design, the study aims to provide practical insights into selecting appropriate hyperparameter optimization strategies while contributing empirical evidence to the growing body of research on metaheuristic applications in deep learning.

Based on this motivation, the main contributions of this study are threefold. First, this study provides one of the early controlled comparative evaluations between a well-established metaheuristic algorithm (PSO) and an emerging swarm intelligence approach (DMO) for deep learning hyperparameter optimization on the CIFAR-10 dataset. Second, the study introduces a structured evaluation framework incorporating convergence analysis, classification metrics, confusion matrix analysis, and computational performance assessment to comprehensively examine optimization effectiveness. Third, this research provides empirical insights into the influence of metaheuristic search behavior on hyperparameter configuration and deep learning generalization capability.

MATERIALS AND METHODS

Dataset and Data Splitting

This study utilizes the CIFAR-10 dataset [18], which is a widely adopted benchmark in deep learning research for image classification tasks. The dataset consists of 60,000 color images with a spatial resolution of 32×32 pixels, evenly distributed across 10 object categories. All pixel values were normalized into the range [0,1] to ensure stable data distribution, stabilize the training process, and improve convergence behavior during model training. In addition, lightweight data augmentation techniques, including random horizontal flipping and random cropping, were applied during training to improve model generalization and reduce overfitting.

The CIFAR-10 dataset is divided into two primary subsets: training and testing data. The training subset contains 50,000 images, while the testing subset includes 10,000 images that are completely isolated from both the training and hyperparameter optimization stages. This separation is intended to preserve the objectivity of the final performance evaluation. To support hyperparameter optimization, the training data is further partitioned into training and validation subsets using an 80:20 split ratio [18]. The partitioning is performed through a random splitting mechanism to maintain representative data distribution across subsets. The validation subset serves as a performance reference during the hyperparameter optimization process, whereas the training subset is used to update model parameters at each optimization iteration. Within the hyperparameter optimization framework, the validation set functions as the fitness evaluation basis for the Particle Swarm Optimization (PSO) and Dwarf Mongoose Optimization (DMO) algorithms. Each candidate hyperparameter configuration is assessed based on model performance on the validation data, enabling the optimization process to identify configurations that generalize well to unseen samples. This strategy is employed to reduce the risk of overfitting to the training data during the search process.

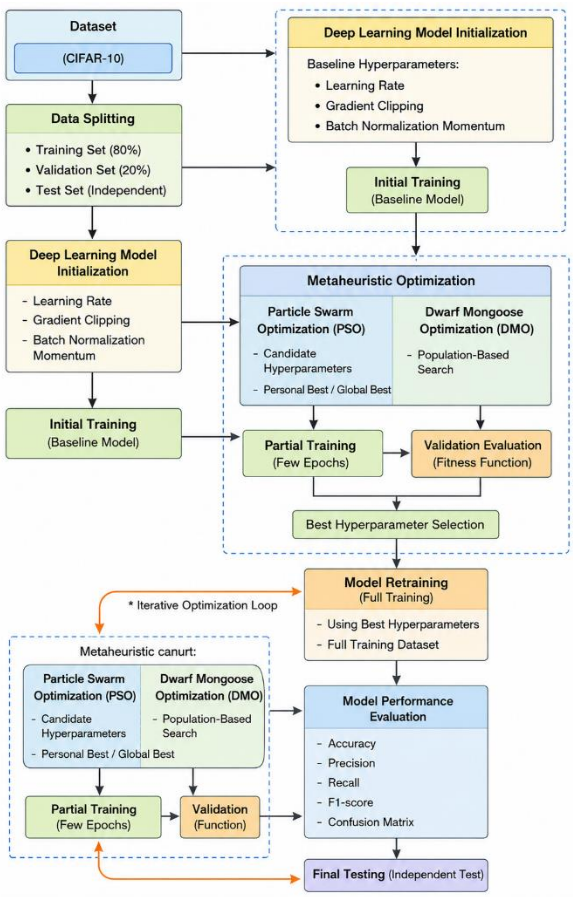
After completing the optimization stage, the deep learning model is retrained using the best-performing hyperparameter configuration identified by each optimization algorithm. The final training phase leverages the full training dataset prior to evaluation on the independent test set. This final evaluation step provides an unbiased estimate of the model's generalization capability on previously unseen data. The overall data partitioning scheme is designed to ensure a fair and leakage-free comparison between optimization methods. By maintaining strict separation between training, validation, and testing subsets, the evaluation results more accurately reflect the true contribution of hyperparameter optimization to deep learning model performance, enabling a controlled and objective comparison between Particle Swarm Optimization and Dwarf Mongoose Optimization.

Proposed Optimization Architecture

To provide a structured representation of the optimization workflow, this study proposes an optimization architecture that illustrates the

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

process of hyperparameter adjustment within the deep learning framework. The proposed architecture explicitly defines the interaction between dataset partitioning, model training, metaheuristic optimization algorithms, and performance evaluation, while preserving the original neural network architecture. This formulation serves as a conceptual abstraction of the hyperparameter optimization mechanism implemented throughout the experimental pipeline.



Source: (Research Results 2026)
Figure 1. Proposed Optimization Architecture

Figure 1 presents the proposed optimization architecture that governs the overall experimental workflow in this study, starting from dataset preparation to final evaluation. Initially, the CIFAR-10 dataset is partitioned into three subsets: training data, validation data, and an independent test set. The training and validation subsets are used during the optimization process, while the test set is strictly reserved for final evaluation to ensure unbiased performance assessment. In the initial stage, the deep learning model is initialized using baseline hyperparameter configurations, including the learning rate, gradient clipping threshold, and batch normalization momentum. The model is then trained using the training dataset to obtain the initial model performance, which serves as a baseline for subsequent comparisons. Subsequently, hyperparameter optimization is performed using a metaheuristic optimization approach involving Particle Swarm Optimization (PSO) and Dwarf Mongoose Optimization (DMO). These algorithms are executed independently to explore the hyperparameter search space in a comprehensive and adaptive manner. In PSO, each particle represents a candidate solution that evolves based on individual experience (*personal best*) and collective knowledge (*global best*). In contrast, DMO simulates the cooperative behavior of mongoose populations to balance exploration and exploitation during the search process.

Each candidate hyperparameter configuration generated by these algorithms undergoes partial training over a limited number of epochs using the training dataset. The resulting model is then evaluated on the validation dataset, and the validation accuracy is used as the fitness function. This fitness evaluation process is performed iteratively, enabling both PSO and DMO to update their search positions until convergence criteria or iteration limits are reached. After completing the optimization process, the system performs the selection of the best solution by identifying the hyperparameter configuration with the highest fitness value. The selected configuration is subsequently used to retrain the model using the full training dataset, producing the final optimized model. In the next stage, the optimized model undergoes final model evaluation using multiple performance metrics, including accuracy, precision, recall, F1-score, and confusion matrix, to provide a comprehensive assessment of classification performance. Finally, the model is evaluated on an independent test dataset during the final testing phase to ensure objective and unbiased performance measurement. The proposed architecture explicitly enforces a strict separation between training, validation, and testing phases, thereby preventing data leakage and ensuring the reliability and validity of the experimental results.

Deep Learning Model Architecture

The deep learning model employed in this study is a Convolutional Neural Network (CNN) specifically designed for CIFAR-10 image classification [2]. The input dimension of the network is 32×32×3, corresponding to RGB images in the CIFAR-10 dataset. The architecture consists of three convolutional blocks, where each block includes a convolutional layer with a 3×3 kernel, followed by batch normalization, ReLU activation,

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

and a 2×2 max-pooling operation. The convolutional layers use 32, 64, and 128 filters, respectively, to progressively capture hierarchical feature representations. The extracted features are then flattened and passed through a fully connected layer with 256 neurons, followed by a dropout layer with a rate of 0.5 to mitigate overfitting. The final classification layer consists of 10 output neurons with a softmax activation function.

To ensure an objective evaluation of the proposed optimization methods, a conventional CNN architecture was deliberately adopted as the experimental backbone. Although advanced architectures such as ResNet have demonstrated superior classification performance on CIFAR-10, their deeper network structures and residual learning mechanisms introduce additional architectural complexity that may affect optimization behavior. Since the primary objective of this study is to compare the effectiveness of PSO and DMO for hyperparameter optimization rather than to evaluate different neural network architectures, employing a lightweight CNN provides a controlled experimental setting in which the influence of the optimization algorithms can be examined more objectively. The model is trained using the Adam optimizer with categorical cross-entropy as the loss function. The same network architecture and training configuration are maintained throughout all experiments to ensure a fair comparison between the two optimization methods.

Hyperparameter Search Space

To support the hyperparameter optimization process, a set of key training parameters and their corresponding search ranges are defined. These hyperparameters are selected based on their critical role in controlling training stability and convergence behavior in deep learning models. The optimization focuses on three primary parameters: learning rate, gradient clipping value, and batch normalization momentum. The learning rate controls the step size of parameter updates and significantly affects convergence speed and stability. Gradient clipping is applied to prevent gradient explosion, which is particularly important in convolutional neural networks with complex optimization landscapes. Batch normalization momentum is included to regulate the stability of feature distribution during training.

The selected learning rate interval was intentionally constrained to provide a stable and controlled search space for hyperparameter optimization. A relatively narrow range allows the optimization algorithms to perform fine-grained exploration around practically relevant learning rate values while reducing the likelihood of unstable training behavior associated with excessively large or excessively small learning rates. This design is consistent with the objective of this study, which is to compare the effectiveness of PSO and DMO under identical experimental conditions rather than to perform an exhaustive search over a broad hyperparameter space. In contrast, the batch size is kept fixed at 64 throughout all experiments to maintain consistency and ensure a fair comparison between optimization strategies.

Table 1. Hyperparameter Search Space

Parameter	Range
Learning Rate	[0.0002, 0.0005]
Gradient Clipping	[0.95, 1.05]
BatchNorm Momentum	[0.05, 0.15]
Batch Size	Fixed (64)

Source: (Research Results, 2026)

Table 2. Baseline and Optimized Hyperparameter Configurations

Parameter	Baseline	PSO	DMO
Learning Rate	0.001	0.00032	0.00041
Gradient Clipping	1.0	0.98	1.02
BatchNorm Momentum	0.10	0.08	0.12
Batch Size	64	64	64

Source: (Research Results, 2026)

Table 2 presents the comparison between the baseline hyperparameter configuration and the optimized configurations generated by PSO and DMO. The results indicate that both optimization methods successfully identified more suitable parameter combinations compared to the manually defined baseline configuration, contributing to improved convergence stability and classification performance.

RESULTS AND DISCUSSION

This section presents the experimental findings and provides a comprehensive discussion on the effectiveness of metaheuristic-based hyperparameter optimization. The analysis extends beyond final model accuracy by incorporating baseline model comparison, convergence behavior, training stability, and class-wise performance distribution. Such a multi-perspective evaluation enables a more holistic interpretation of optimization outcomes, offering not only descriptive results but also conceptual insights into the learning dynamics of deep learning models under hyperparameter optimization. The discussion is structured progressively to ensure analytical clarity. It begins with a comparison between the baseline model and optimized models,

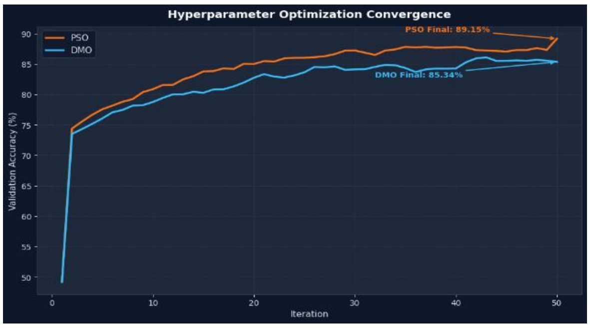
VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

followed by an examination of optimization behavior, training dynamics analysis, quantitative performance evaluation, and a fine-grained class-wise performance investigation. This layered analytical framework facilitates a systematic interpretation of the results and strengthens the validity of the experimental conclusions.

Hyperparameter Optimization Behavior

The convergence analysis of the hyperparameter optimization process reveals that both metaheuristic algorithms effectively explore the search space. The convergence trajectories exhibit a rapid decline in fitness values during the early iterations, indicating strong global exploration capabilities for identifying high-quality candidate solutions. In this study, the fitness value is defined as the validation accuracy obtained from partial training, ensuring that the optimization process reflects the model's generalization performance. Particle Swarm Optimization (PSO) demonstrates a more stable convergence profile with minimal oscillations after the initial exploration phase. This stability can be attributed to the global-best learning mechanism, which enables particles to consistently move toward promising regions in the search space. As a result, PSO converges more smoothly toward high-quality hyperparameter configurations.

Conversely, Dwarf Mongoose Optimization (DMO) exhibits more dynamic convergence behavior due to its role-based population structure, which promotes exploration diversity but introduces higher variability during optimization. While this dynamic behavior enhances the ability to escape local optima, it may lead to less stable convergence compared to PSO. Overall, both algorithms successfully identify improved hyperparameter configurations compared to the baseline model, demonstrating the effectiveness of metaheuristic-based optimization in navigating complex and high-dimensional search spaces.

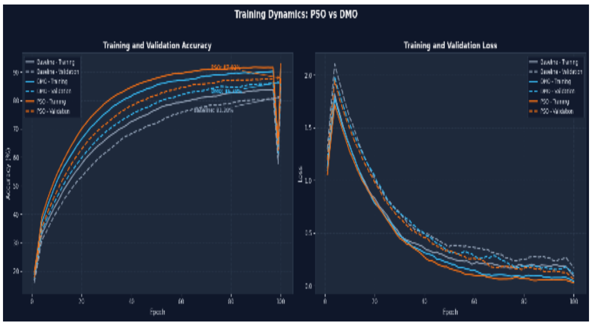


Source: (Research Results, 2026)
Figure 2. Convergence Curves of Hyperparameter Optimization Using PSO and DMO

Convergence curves of hyperparameter optimization using Particle Swarm Optimization (PSO) and Dwarf Mongoose Optimization (DMO). The fitness value represents validation accuracy obtained from partial training at each iteration. PSO demonstrates more stable convergence and achieves higher final performance compared to DMO.

Training Dynamics

The training dynamics of the model are analyzed through accuracy and loss curves on both training and validation datasets, as illustrated in Figure 3. The results show that both optimized hyperparameter configurations exhibit stable convergence patterns, with a consistent increase in accuracy as the number of training epochs increases. The PSO optimized configuration demonstrates a faster convergence rate during the early stages of training, indicating that the selected hyperparameters guide the model toward a more stable region of the loss landscape. This results in a smoother learning process and quicker performance improvement. In contrast, the DMO optimized configuration exhibits a more gradual convergence pattern, reflecting its exploratory optimization behavior. Despite this slower progression, the model maintains stable performance throughout the training process and achieves competitive results compared to PSO. Overall, both optimization strategies significantly improve training stability and convergence behavior compared to the baseline model, confirming the effectiveness of metaheuristic-based hyperparameter optimization.



Source: (Research Results, 2026)
Figure 3. Training Curves PSO & DMO

Comparative Performance Analysis

The final performance evaluation demonstrates that both optimization strategies yield competitive classification outcomes compared to the baseline model. The hyperparameter configuration identified by Particle Swarm Optimization (PSO) achieved a peak validation

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

accuracy of 89.15% and a final test accuracy of 87.82%. In contrast, the configuration obtained through DMO achieved a maximum validation accuracy of 87.82% coincidentally identical to PSO's final test accuracy and a final test accuracy of 86.38%. Both results significantly outperform the baseline configuration, confirming the effectiveness of metaheuristic based hyperparameter optimization. In addition to accuracy, the evaluation of classification metrics indicates that both methods maintain relatively balanced performance distributions across classes. PSO achieved a macro precision of 0.88 and an F1-score of 0.87, slightly surpassing DMO, which exhibited values in the range of 0.86–0.87. These results highlight the effectiveness of both metaheuristic optimizers, while suggesting a modest performance advantage for PSO in terms of classification robustness and stability.

Table 3. Model Performance Comparison

Method	Best Accuracy (%)	Final Accuracy (%)	Precision	Recall	F1-score
Baseline	83.45	81.20	0.81	0.80	0.80
PSO	89.15	87.82	0.88	0.87	0.87
DMO	87.82	86.38	0.87	0.86	0.86

Source: (Research Results, 2026)

The relatively narrow performance gap between PSO and DMO suggests that both optimization algorithms converge toward closely related regions within the hyperparameter search space. However, the consistent advantage demonstrated by PSO indicates that the stability of the optimization trajectory plays a critical role in achieving superior model performance and generalization capability.

Table 4. Computational Cost Comparison

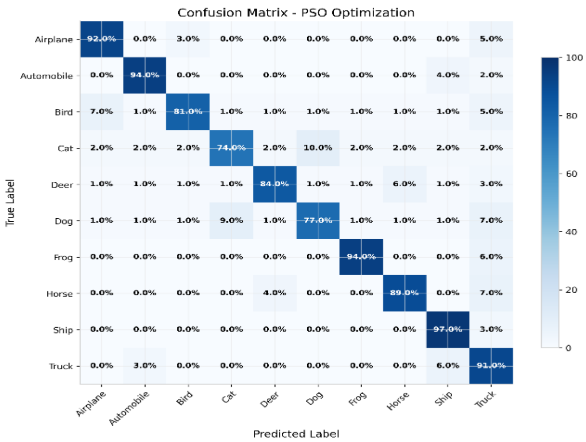
Method	Iterations	Avg Training Time (min)	Convergence Behavior
PSO	50	45.2	Stable
DMO	50	58.6	More Dynamic

Source: (Research Results, 2026)

In terms of computational cost, PSO demonstrated faster convergence and lower average training time compared to DMO. This indicates that the more stable optimization trajectory of PSO contributes not only to higher classification performance but also to improved computational efficiency. Although DMO provides broader exploration capability, its population-based search dynamics introduce higher optimization overhead during the optimization process.

Class-wise Performance Analysis

A confusion matrix-based evaluation was conducted to analyze class-level classification performance, as illustrated in Figure 4. This analysis provides a more granular understanding of model generalization beyond aggregate accuracy metrics by revealing the distribution of predictions across individual classes. The confusion matrix of the PSO-optimized model demonstrates strong performance on classes with distinctive visual features, such as ship (97.06%), automobile (94.29%), and frog (94.03%). These results indicate that the optimized hyperparameters enable the model to effectively capture discriminative patterns in visually separable categories. In contrast, lower performance is observed in classes with higher intra-class variability and visual similarity, such as cat (74.38%) and bird (81.93%), which are more prone to misclassification due to overlapping feature representations and complex textures. The dominance of diagonal values in the confusion matrix indicates consistent classification performance across most classes, confirming that metaheuristic-based hyperparameter optimization improves both overall accuracy and class-level generalization.

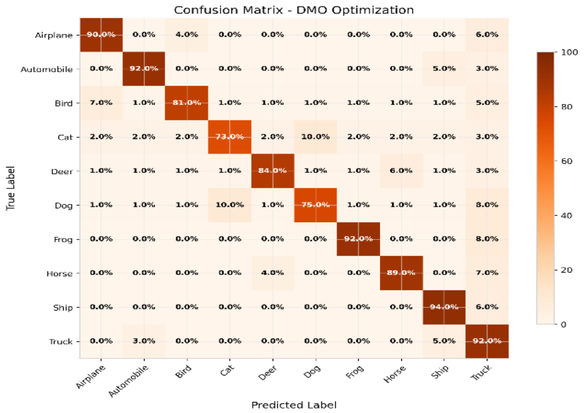


Source : (Research Results, 2026)
Figure 4. Confusion Matrix PSO

Error distribution analysis shows that most misclassifications occur between visually similar class pairs, such as cat–dog and automobile–truck, which is a well-known challenge in the CIFAR-10 dataset. The model optimized using Dwarf Mongoose Optimization (DMO) exhibits a comparable performance pattern, achieving high accuracy on classes such as ship (94.31%) and automobile (92.81%). However, several classes show slightly lower accuracy compared to the PSO-based model, indicating that differences in

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

optimization dynamics influence the distribution of class-wise predictions and overall consistency.



Source : (Research Results, 2026)

Figure 5. Confusion Matrix DMO

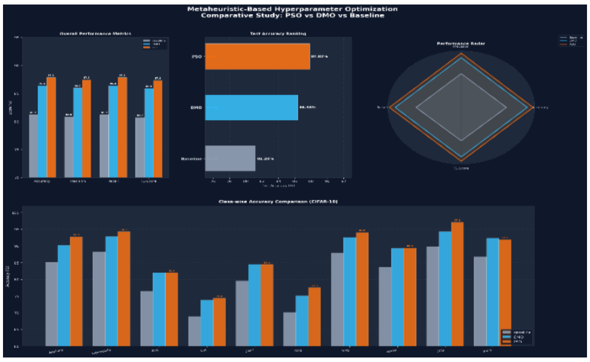
To enable a more structured comparison, class-wise classification accuracy is summarized in Table 5. The results indicate that PSO generally achieves higher performance on classes with clearly defined object structures, such as ship, with a performance margin of 2.74%. In contrast, DMO demonstrates localized advantages in certain classes, including truck and deer, suggesting its ability to explore alternative regions of the solution space. Although the overall performance gap between the two methods remains relatively small, their class-wise performance distributions reveal distinct patterns. These differences indicate that optimization dynamics influence not only global accuracy but also the consistency and reliability of predictions across different object categories. This further confirms that optimization strategies contribute differently to class-level generalization, beyond what is reflected in aggregate performance metrics.

Table 5. Class-wise Classification Accuracy Comparison Between Models Optimized Using PSO and DMO

Class	PSO (%)	DMO (%)	Difference (%)
Airplane	92.75	90.12	+2.63
Automobile	94.29	92.81	+1.48
Bird	81.93	81.93	0
Cat	74.38	73.81	+0.57
Deer	84.39	84.46	-0.08
Dog	77.53	75.08	+2.45
Frog	94.03	92.53	+1.49
Horse	89.33	89.23	+0.10
Ship	97.06	94.31	+2.74
Truck	91.91	92.35	-0.44

Source: (Research Results, 2026)

The findings suggest that PSO maintains a more consistent performance distribution across the majority of classes, while DMO exhibits greater variability in class-wise outcomes. This discrepancy reflects the distinct exploration dynamics of the two algorithms during the optimization process.



Source : (Research Results, 2026)

Figure 6. Class-Wise Comparison of Classification Accuracy

Discussion

This study demonstrates that metaheuristic-based hyperparameter optimization significantly enhances the performance of deep learning models. Particle Swarm Optimization (PSO) exhibits stronger exploitation capabilities with a more stable convergence trajectory, while Dwarf Mongoose Optimization (DMO) provides broader exploration through its dynamic population-based search mechanism. The results indicate that the effectiveness of metaheuristic optimizers is influenced not only by the final accuracy achieved but also by the stability of the optimization trajectory. A more stable convergence path contributes to improved consistency in model performance across both training and evaluation phases.

Furthermore, confusion matrix analysis reveals that certain performance limitations are inherently associated with the intrinsic complexity of the dataset rather than the optimization strategy alone. Classes with high visual similarity remain challenging, even under optimized hyperparameter configurations. Overall, the findings highlight that optimization stability plays a more critical role than exploration diversity in constrained hyperparameter search spaces. In this context, PSO demonstrates a consistent advantage due to its stable convergence behavior, whereas DMO remains a promising alternative for problems requiring broader exploration of more complex or less constrained search spaces.

Although this study focuses specifically on comparative analysis between Particle Swarm Optimization (PSO) and Dwarf Mongoose Optimization (DMO), other widely used hyperparameter optimization methods such as Grid

VOL. 12. NO. 1 AUGUST 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v12i1.8269

Search, Random Search, Bayesian Optimization, and Hyperband were not included in the experimental evaluation. This limitation is primarily due to computational constraints and the study's focus on analyzing swarm-based metaheuristic optimization behavior under controlled experimental settings. Future research may extend the comparative framework by incorporating additional optimization strategies and multiple benchmark datasets to provide broader benchmarking and evaluation perspectives.

CONCLUSION

This study investigates the effectiveness of metaheuristic-based hyperparameter optimization in improving deep learning performance. The experimental results demonstrate that both Particle Swarm Optimization and Dwarf Mongoose Optimization are capable of identifying hyperparameter configurations that significantly enhance model stability and classification accuracy. PSO exhibits superior performance in terms of convergence stability and consistency across classes, achieving a final test accuracy of 87.82%. In contrast, DMO provides broader exploration of the search space but yields slightly lower final performance. Confusion matrix analysis further reveals that classification limitations are influenced not only by the optimization strategy but also by the intrinsic complexity of the dataset, particularly in visually similar classes. Overall, the findings confirm that metaheuristic-based optimization offers a robust alternative for hyperparameter tuning in deep learning, especially in complex search environments. Future research may extend this comparative study by incorporating additional hyperparameter optimization methods, such as Bayesian Optimization and Hyperband, as well as evaluating the proposed approach on multiple benchmark datasets and more advanced deep learning architectures to provide a broader assessment of its robustness and generalization capability.

REFERENCE

[1] T. Talaei Khoei, H. Ould Slimane, and N. Kaabouch, "Deep Learning: Systematic Review, Models, Challenges, and Research Directions," *Neural Comput. Appl.*, vol. 35, pp. 23103–23124, 2023, doi: 10.1007/s00521-023-08957-4.

[2] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A Review of Convolutional Neural Networks in Computer Vision," *Artif. Intell. Rev.*, vol. 57, p. 99, 2024, doi: 10.1007/s10462-024-10721-6.

[3] A. Chen, Z. Ren, M. Wang, H. Chen, H. Leng, and S. Liu, "A Surrogate-Assisted Highly Cooperative Coevolutionary Algorithm for Hyperparameter Optimization in Deep Convolutional Neural Networks," *Appl. Soft Comput.*, vol. 147, p. 110794, 2023, doi: 10.1016/j.asoc.2023.110794.

[4] B. Bischl *et al.*, "Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 13, no. 2, p. e1484, 2023, doi: 10.1002/widm.1484.

[5] I. Amerini, and L. Palagi, "Tuning parameters of deep neural network training algorithms pays off: a computational study," TOP, vol. 32, no. 3, pp. 579–620, Oct. 2024, doi: 10.1007/s11750-024-00683-x.

[6] T. Maumela, F. Nelwamondo, and T. Marwala, "Population based training and federated learning frameworks for hyperparameter optimisation and ML unfairness using Ulimisana Optimisation Algorithm," *Inf. Sci. (N. Y).*, vol. 612, pp. 132–150, 2022, doi: 10.1016/j.ins.2022.08.091.

[7] A. Dushatskiy, A. Chebykin, T. Alderliesten, and P. Bosman, "Multi-Objective Population Based Training," in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, 2023, pp. 8969–8989.

[8] F. Zito, E.-G. Talbi, C. Cavallaro, V. Cutello, and M. Pavone, "Metaheuristics in Automated Machine Learning: Strategies for Optimization," *Intelligent Systems with Applications*, vol. 26, p. 200532, 2025, doi: 10.1016/j.iswa.2025.200532.

[9] D. Gutiérrez-Avilés, M. J. Jiménez-Navarro, J. F. Torres, and F. Martínez-Álvarez, "MetaGen: A Framework for Metaheuristic Development and Hyperparameter Optimization in Machine and Deep Learning," *Neurocomputing*, vol. 637, p. 130046, 2025, doi: 10.1016/j.neucom.2025.130046.

[10] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle Swarm Optimization: A Comprehensive Survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.

[11] T.-Y. Hong and C.-C. Chen, "Hyperparameter Optimization for Convolutional Neural Network by Opposite-Based Particle Swarm Optimization and an Empirical Study of

Photomask Defect Classification," *Appl. Soft Comput.*, vol. 148, p. 110904, 2023, doi: 10.1016/j.asoc.2023.110904.
[12] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, pp. 2531–2561, 2022, doi: 10.1007/s11831-021-09694-4.
[13] J. O. Agushaka and others, "Dwarf mongoose optimization algorithm: A new bio-inspired metaheuristic," *Expert Syst. Appl.*, 2022.
[14] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf Mongoose Optimization Algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 391, p. 114570, 2022, doi: 10.1016/j.cma.2022.114570.
[15] I. Al-Shourbaji *et al.*, "Artificial Ecosystem-Based Optimization with Dwarf Mongoose Optimization for Feature Selection and Global Optimization Problems," *International Journal of Computational Intelligence Systems*, vol. 16, p. 102, 2023, doi: 10.1007/s44196-023-00279-6.
[16] S. Fu, H. Huang, C. Ma, J. Wei, Y. Li, and Y. Fu, "Improved Dwarf Mongoose Optimization Algorithm Using Novel Nonlinear Control and Exploration Strategies," *Expert Syst. Appl.*, vol. 233, p. 120904, 2023, doi: 10.1016/j.eswa.2023.120904.
[17] G. Moustafa *et al.*, "An Enhanced Dwarf Mongoose Optimization Algorithm for Solving Engineering Problems," *Mathematics*, vol. 11, no. 15, p. 3297, 2023, doi: 10.3390/math11153297.
[18] S. Aslam and A. B. Nassif, "Deep Learning Based CIFAR-10 Classification," in *2023 Advances in Science and Engineering Technology International Conferences (ASET)*, IEEE, 2023. doi: 10.1109/ASET56582.2023.10180767.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri