Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 14, No. 3, 2026, 341-350.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Metaheuristic Hyperparameter Optimization of Deep Learning and CNN Model for EEG-Based Emotion Classification

Yashar Chehardah Cheriki Gheisari and Hadi Grailu*

Faculty of Electrical Engineering, Shahrood University of Technology, Shahrood, Iran.

Article Info

Article History:
Received 29 December 2025
Revised 14 March 2026
Accepted 02 May 2026

DOI:10.22044/jadm.2026.17322.2872

Keywords:
Brain Signals, Emotion Recognition, EEG Signal Processing, Metaheuristic Algorithm.

**Corresponding author: grailu@shahroodut.ac.ir (H. Grailu).*

Abstract

Human emotion recognition based on electroencephalogram signals remains a significant challenge in computational neuroscience and artificial intelligence. Convolutional neural networks have been widely adopted to address this challenge due to their strong capabilities in feature extraction and representation learning. Empirical tuning of hyperparameters, however, is often inefficient and prone to suboptimal solutions. To overcome this limitation, the Grey Wolf Optimizer and Whale Optimization Algorithm—two swarm intelligence–based metaheuristic methods—were employed to refine the architecture and learning hyperparameters of a baseline convolutional neural network. Performance was evaluated on the SEED EEG dataset for emotion recognition. Experimental results showed that the optimized model achieved an accuracy of 86% compared to 83% for the baseline, with the confusion matrix confirming reduced misclassification errors and improved recognition of emotional states. Beyond accuracy, the optimization process also lowered the number of trainable parameters and computational overhead, thereby enhancing efficiency. These findings highlight that swarm intelligence–based methods provide superior exploration and exploitation capabilities, enabling systematic hyperparameter tuning and delivering a balanced engineering approach that combines predictive performance with resource efficiency.

1. Introduction

Electroencephalography (EEG) signals generated by the central nervous system are among the most significant biological data for analyzing and recognizing human emotions. Compared to other physiological data, these signals possess the unique ability to reflect emotional states in an unbiased manner, which has granted them a prominent role in computational neuroscience and artificial intelligence. Within this domain, convolutional neural networks (CNNs) are widely used for EEG-based emotion recognition due to their strong ability to extract complex features. Nevertheless, the design of their architecture and learning process poses numerous challenges [1–4].

A fundamental challenge in emotion recognition with deep learning models is the tuning and optimization of numerous hyperparameters. Traditionally, researchers have relied on prior knowledge and trial-and-error strategies. These approaches introduce human bias and are time-consuming, as they depend on manual searches across large hyperparameter spaces. Consequently, researchers have sought alternative strategies to overcome these limitations. In recent years, metaheuristic algorithms have emerged as a promising solution, offering systematic approaches to model design and hyperparameter tuning while achieving notable improvements in both predictive performance and learning efficiency. According to the "No Free Lunch" theorem, no single algorithm can guarantee optimal performance across all optimization problems. Therefore, selecting an appropriate method depends strongly on the dimensionality and complexity of the task. For small-scale problems with limited hyperparameters, exact algorithms remain efficient and reliable, as they guarantee convergence to the optimal solution. However, as dimensionality increases and the number of hyperparameters grows, the exponential rise in computational time renders such methods impractical. Under these conditions, metaheuristic algorithms, which are stochastic and approximation-based in nature, provide a balanced trade-off between global exploration and local exploitation. They enable near-optimal solutions that achieve both accuracy and efficiency within reasonable time frames [5].

Addressing the challenges of emotion recognition based on brain signals often relies on convolutional neural networks (CNNs). The architecture of these networks, along with the tuning of their learning hyperparameters, requires careful optimization to ensure robust performance and reliable outcomes. A critical component of this optimization is the design of cost functions, where researchers have generally employed two main strategies for problem formulation: binary encoding and the traditional approach [6,7]. These strategies are primarily distinguished by the type of cost function employed. Binary encoding reduces the dimensionality of the optimization problem, thereby simplifying the search process; however, this simplification comes at the cost of excluding a significant portion of the valid solution space [6].

In contrast, the traditional approach preserves higher dimensionality, providing metaheuristic algorithms with a broader search space and enabling more profound exploration of potential solutions. While binary encoding offers efficiency through reduced complexity, the traditional approach maintains flexibility and comprehensiveness, though it may increase computational demands. Consequently, metaheuristic algorithms serve as the primary tools for solving largescale problems, while traditional approaches play a complementary role in design and optimization.

To the best of the authors' knowledge, a comprehensive cost function for the meta-heuristic optimization of baseline deep learning models in EEG-based emotion recognition has not been proposed. Therefore, in this paper, while reviewing the latest findings from existing survey studies, we design and implement a comprehensive cost function for the meta-heuristic optimization of a baseline CNN model for emotion recognition, and present our results through a fair comparison with existing works.

2. Literature Review

2.1. EEG-Based Emotion Recognition

Recent studies have employed multiple approaches to measure emotions, including questionnaires, physical signals, and physiological signals. Among these, EEG, which is a type of physiological signal, has gained particular prominence due to its involuntary nature and high temporal resolution [4]. However, the noisy and nonlinear nature of brain signals makes it challenging to extract discriminative features. In this regard, previous studies have primarily focused on classical feature extraction techniques and conventional machine learning models [2].

In recent years, the advent of deep learning (DL) models has emerged as a prominent approach in EEG-based emotion recognition, owing to their inherent ability to hierarchically learn features from raw data. A review of the literature reveals that architectures such as convolutional neural networks (CNN), recurrent neural networks (RNN), autoencoders (AE), generative adversarial networks (GAN), and graph neural networks (GNN) have made the most significant contributions to improving classification accuracy [3].

In this study, we adopt a discrete emotion approach and focus on three distinct categories: negative, neutral, and positive emotions.

2.2. Feature Extraction and Deep Learning Models

Differential Entropy (DE) features extracted from theta, alpha, beta, and gamma frequency bands have consistently been reported as reliable indicators for emotion analysis using EEG signals [8,9].

Convolutional Neural Networks (CNNs) are among the most widely adopted architectures for EEG-based emotion recognition in deep learning. Two-dimensional CNNs utilize convolutional layers, activation functions, and optimization algorithms such as Adam. With these components, they can capture complex spatial and temporal patterns from multichannel EEG data [3].

Nevertheless, designing optimal architectures and tuning hyperparameters remains a challenging task, often requiring costly human trial-and-error [5,10].

To evaluate CNN performance, common metrics include accuracy, confusion matrix, and loss function. Additional measures such as the number of trainable parameters, computational complexity

(FLOPs), and memory requirements are also employed. Furthermore, K-Fold cross-validation is frequently used to ensure the reliability and generalizability of results [11]. Baseline CNN architectures, such as the CCNN implemented in Torcheeg, have also been introduced in prior studies as starting points for EEG-based emotion recognition, providing a foundation for further optimization and experimentation [12].

2.3. Metaheuristic Algorithms for Hyperparameter Optimization

In recent years, metaheuristic algorithms have been introduced as innovative approaches for optimizing deep learning architectures. Inspired by natural behaviors, these algorithms are designed to balance exploration and exploitation. As a result, they can escape local optima and achieve more efficient solutions. Prior studies have applied metaheuristics in two main ways: (a) optimizing hyperparameters of the learning process , such as learning rate and batch size, and (b) optimizing CNN architectures using binary encoding strategies. Previous research has investigated metaheuristic algorithms—including PSO, GA, GWO, and WOA—for CNN hyperparameter optimization [5,13,14].

2.4. Summary and Research Gap

Empirical studies have demonstrated that integrating CNNs with metaheuristic algorithms enhances classification accuracy, reduces computational time, and improves practical efficiency in emotion recognition. However, despite these advances, challenges remain regarding scalability, computational overhead, and the risk of overfitting in high-dimensional spaces.

Table 1. Identified research gaps in existing EEG emotion recognition models and directions for structural improvements [4].

Research Directions and Priorities
• Uncertainty in performance
• Exhaustive use of available datasets
• Tested on cleaned and pre-processed data
• Lack of adaptivity
• Lack of explainability
• Non-uniformity in EEG segment length selection
• Limited usage of hyperparameter tuning
• Limited usage of fusion techniques

Moreover, existing works have typically focused on either learning hyperparameters or architecture optimization separately [5]. According to Table 1, "Lack of adaptivity" and "Limited usage of hyperparameter tuning" are key limitations in this field. This study addresses these limitations by proposing a comprehensive cost function and utilizing GWO and WOA metaheuristic algorithms for simultaneous optimization of both aspects. This highlights the need for approaches that jointly optimize both aspects, which is the contribution of the present study.

3. Proposed Method

This section introduces the innovations of the present research to address some of the existing research gaps. The primary objective is to optimize the performance of a Convolutional Neural Network (CNN). Therefore, selecting a credible baseline methodology with accessible and documented data, architecture, and training/validation procedures is essential for defining the cost function and search space for optimization.

3.1. Dataset and Preprocessing

In this study, we utilized the SEED emotional EEG dataset [15], which comprises data from 15 participants. Each participant completed three sessions, with each session consisting of 15 trials equally distributed across three emotional states (positive, neutral, and negative). Specifically, each session contained five trials for each of the three emotional classes, ensuring an inherent balance across the dataset. We employed the officially released preprocessed version of the SEED dataset. In this version, the 62-channel EEG signals are downsampled from the original 1000 Hz to 200 Hz and cleaned via band-pass filtering. The signals were segmented into non-overlapping 1-second windows (200 samples). This segmentation was performed independently for each session, preserving a single emotional label per session and ensuring a balanced distribution of samples across them after windowing. Differential Entropy (DE) features were extracted from the alpha, beta, gamma, and theta frequency bands and subsequently normalized (Figure 1(a)). Using the TorchEEG toolbox, we constructed two-dimensional feature maps, as described in [12] and [8]. Each DE feature vector was spatially mapped according to the 10–20 electrode system, organizing the data into structured feature pages, consistent with Figure 1(b).

3.2. Baseline CNN Architecture

The baseline model was a Continuous Convolutional Neural Network (Figure 1 (c)) [8], trained on 9×9-pixel input images. The architecture consists of four convolutional blocks with a kernel size of 4×4, each followed by a ReLU activation.

After the convolutional layers, a fully connected layer with 1024 neurons and a Dropout rate of 0.5

was applied. The final output layer contained three neurons, corresponding to the classification of positive, negative, and neutral emotions.

This explicit specification of both image and kernel sizes resolves the apparent inconsistency between the figure and the textual description.

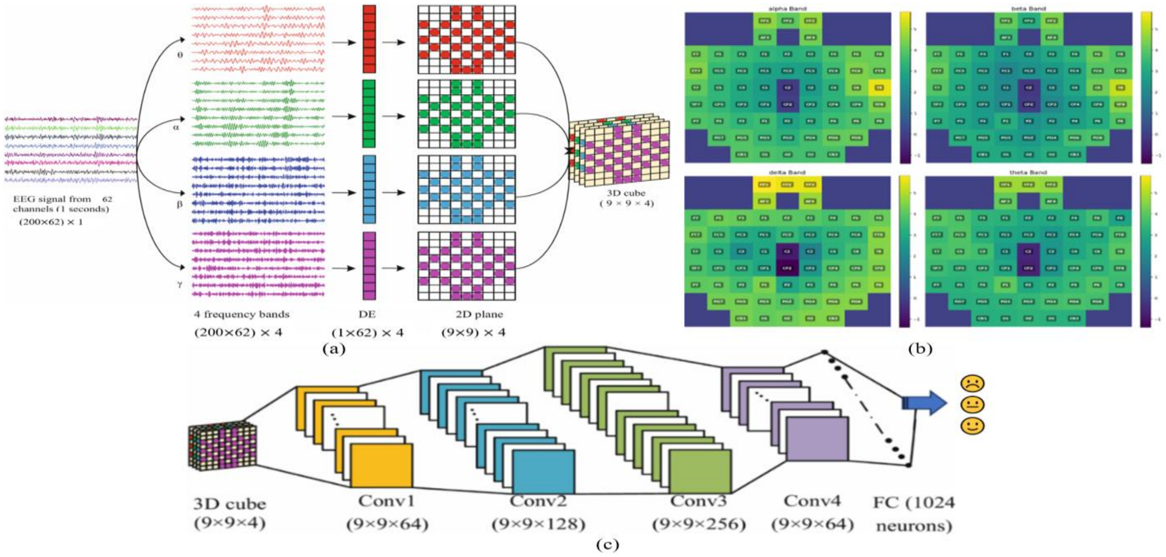

Figure 1. (a) Visual summary of data preparation steps for data loader [8,12]. (b) Differential Entropy (DE) feature maps across frequency space within a 1 second interval . (c) The developed structure of the CCNN model by TorchEEG [12].

3.3. Training and Validation

Training was conducted with a batch size of 64, using the Adam optimizer and the Cross-Entropy loss function. For validation, the 5-Fold KFoldGroupByTrial method provided in TorchEEG was employed. This approach divides each trial into k periods; each period serves as the test set in one of the k experiments, while the remaining subsets form the training set. While this strategy ensures fold-level consistency and trial-wise separation, it does not prevent data from the same subject from appearing in both training and validation folds. Consequently, the reported baseline accuracy of approximately 83% [12] may be optimistic, and future work should consider subject-level grouping to eliminate potential data leakage.

3.4. Cost Function and Metaheuristic Algorithms

Figure 2 illustrates the overarching role of the comprehensive cost function in the sentiment analysis model optimization process. In scenarios where deep learning architects typically conduct architecture tuning and training empirically, error magnitudes and overall model accuracy on balanced datasets are conventionally employed as primary metrics in final reports.

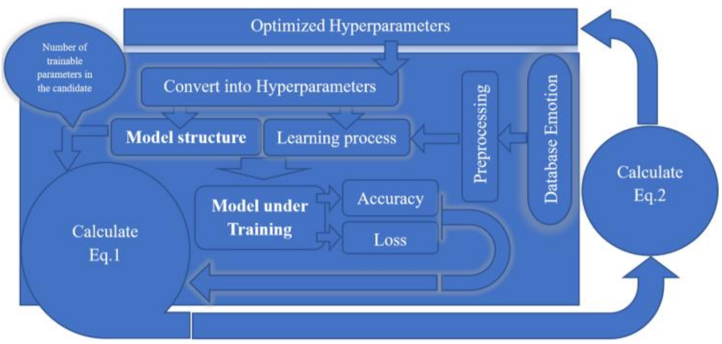

Figure 2. proposed method.

Recognizing that practitioners manually mitigate excessive architectural complexity in deep learning models, we have incorporated this pragmatic constraint directly into our cost function design. Consequently, both error and accuracy metrics are integrated with equal weighting within the comprehensive cost function's definition. Ultimately, this manual mitigation strategy has been formally formulated as our proposed approach. To address the identified research gap (Section 2.4), our proposed method (Figure 2) introduces a comprehensive cost function (1) that synergistically combines the mean loss with the complement of the mean accuracy obtained through K-fold validation. This cost function was formulated as an optimization problem, enabling metaheuristic algorithms to refine both the network structure and the learning process. The hyperparameters considered for optimization include the learning rate, weight decay, batch size, number of neurons per layer, and activation function type. To prevent excessive complexity, the cost function included an infinite penalty constraint on the number of trainable parameters, ensuring that any configuration exceeding the baseline complexity was immediately rejected by the optimizer.

$$\text{Cost}^{\theta} = \begin{cases} \dfrac{1}{k}\sum_{i=1}^{k=i}\left(\text{loss}_i^{\theta} + (1-\text{Acc}_i^{\theta})\right), & P_{\theta} \le P_{\text{base}} \\ +\infty, & P_{\theta} > P_{\text{base}} \end{cases} \quad (1)$$

$$\textit{Metaheuristic Algorithms}\,(GWO\ ,\ WOA):$$
$$\theta^{*} = \arg\min_{\theta\in\Omega} \text{Cost}^{\theta} \quad (2)$$

i : Num of Fold CV

P^{θ} : Number of trainable parameters in the candidate model

P_{base} : Parameter count of the baseline model

θ : Opitmization parameters

θ^{*} : Best Optimized parameters

Ω : Search space

To solve the formulated optimization problem (2), two metaheuristic algorithms were employed: the Grey Wolf Optimizer (GWO) and the Whale Optimization Algorithm (WOA). Their coding and implementation were based on references [16-18], which provided the algorithmic framework. These algorithms were selected because each embodies distinct natural inspirations and search strategies, allowing us to benchmark their relative strengths in refining the network structure and hyperparameters. GWO, inspired by the cooperative hunting behavior and hierarchical leadership of grey wolves, is known for its strong exploitation ability and fast convergence. These properties make it well-suited for large-scale optimization tasks. WOA is inspired by the humpback whale's bubble-net hunting strategy. Whales encircle prey using bubble nets and then attack with either a circular or spiral motion, as depicted in Figure 3 [13].

WOA comprises two main phases for encircling prey, where the whale moves towards the best known position. The distance to this position is calculated by cost function (3), and the whale's position is updated via position update rule (4). Control parameters like $\vec{A}$ and $\vec{C}$ are defined considering a linearly decreasing coefficient a (from 2 to 0) as per parameter definition (5). Following this, the algorithm probabilistically switches between two movement types (50% chance each) for the bubble-net attack (Exploitation Phase): circular motion and a logarithmic spiral, governed by bubble-net attack (6). For enhanced Exploration, when $|\vec{A}| > 1$, the whale moves towards a random member of the population as defined by exploration rule (7), aiding escape from local optima.

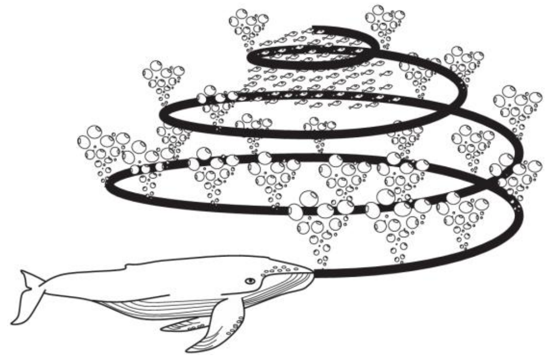

Figure 3. Bubble-net feeding behavior of humpback whales.

$$\vec{D} = \left|\vec{C}.\overrightarrow{X^{*}}(t) - \vec{X}(t)\right| \quad (3)$$

$\vec{X}(t)$: Position Vector

$\overrightarrow{X^{*}}(t)$: Best Solution Position Vector

$\vec{C}$: Random Value Between 0 and 2

$$\vec{X}(t+1) = \overrightarrow{X^{*}}(t) - \vec{A}.\vec{D} \quad (4)$$

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a}$$
$$a = 2 - L \times \left(\frac{2}{Max_{itr}}\right)$$
$$\vec{C} = 2.\vec{r} \quad (5)$$

$\vec{A}$: Random vector between -1 and 1

$\vec{a}$: Linear decrease from 2 to 0

L is the iteration number

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^{*}}(t) - \vec{A}.\vec{D} & if \quad p < 0.5 \\ \vec{D'}.e^{bl}\cos(2\pi l) + \overrightarrow{X^{*}}(t) & if \quad p \ge 0.5 \end{cases}$$
$$\vec{D'} = \left|\overrightarrow{X^{*}}(t) - \vec{X}(t)\right|$$
$$\vec{D} = \left|\vec{C}.\overrightarrow{X^{*}}(t) - \vec{X}(t)\right| \quad (6)$$

l is a random number in $[-1,1]$ and b is a constant for logarithmic spiral shape

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D}$$
$$\vec{D} = \left|\vec{C}.\vec{X}_{rand} - \vec{X}(t)\right| \quad (7)$$

Table 2 provides a comparative summary of GWO and WOA, highlighting differences in exploitation, exploration, convergence trends, and their ability to reach the global optimum. This comparison establishes a clear rationale for employing both algorithms in the present study, enabling an empirical evaluation of their performance on the proposed optimization problem.

Table 2. Comparative characteristics of Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA) [5].

MH Algorithm	GWO	WOA
Exploitation ability	Very high	High
Exploration ability	High	Medium
Convergence trend	Very fast	Medium
Ability of finding global optimum	Very high	High
Year	2014	2016

3.5. Search space

According to Tables 3 and 4, the hyperparameter search spaces were defined to guide the metaheuristic optimization process. The selected ranges aim to balance model flexibility with computational feasibility.

3.5.1. Model structure

The number of filters for 2D convolutional layers 1 to 4, as introduced as the model structure hyperparameters (according to Table 3), was limited to between 32 and 256, covering compact to moderately large feature depths. This enables the exploration of lightweight architectures and deeper feature extraction while avoiding excessive complexity.

The fully connected layer size ranged from 128 to 1024, allowing the optimizer to test both compact and expressive representations.

Eleven activation functions were included (e.g., GELU, SiLU, LeakyReLU, PReLU, ELU, Softplus, Tanh, Softsign, Sigmoid, ReLU, SELU), providing diversity in nonlinear transformations to improve gradient flow and generalization.

Table 3. Model structure hyperparameters for metaheuristic optimization

Model structure	$\mathbf{n}:[\mathbf{lb},\mathbf{ub}]$	Descriptions and Bounds
N (for convolutional layers)	$[1,\ 7]$	$Conv2D(N)_n = [32,48,64,96,128,196,256]$
Fully connected neurons	$[1,\ 7]$	$N_n = [128,196,256,384,512,768,1024]$
Convolution activation	$[1,\ 11]$	$N_n =$ [GELU, SiLU, LeakyReLU, PReLU, ELU, Softplus, Tanh, Softsign, Sigmoid, ReLU, SELU]
Fully connected activation	$[1,\ 11]$	$N_n =$ [GELU, SiLU, Softplus, ELU, Tanh, Softsign, LeakyReLU, PreLU, ReLU, Sigmoid, SELU]

3.5.2. Learning process

As detailed in Table 4, hyperparameter tuning was performed across a defined range of values. The learning rate was varied between 0.0001 and 0.01, representing conservative to moderately aggressive update steps. Weight decay ranged from 0.0001 to 0.05, allowing exploration from minimal to stronger regularization.

The batch size was defined as 2^{n+4} (with $n \in [1,2,3,4]$), offering flexibility in training dynamics. This configuration allows for noisy gradient updates that enhance generalization, as well as stable convergence with larger batches.

Table 4. Learning process hyperparameters for metaheuristic optimization.

Learning process	$\mathbf{n}:[\mathbf{lb},\mathbf{ub}]$	Descriptions and Bounds
Learning rate	$[1,\ 20]$	$\mathrm{lr}_n = [0.0001,...,0.01]$
weight decay	$[1,\ 24]$	$\mathrm{WD}_n = [0.0001,...,0.05]$
Batch size	$[1,\ 4]$	$Batch\ Size_n = 2^{n+4}$

Overall, Tables 3 and 4 highlights that the search spaces were broad enough to allow meaningful exploration, yet constrained to prevent impractical architectures or unstable training. This design ensured that the metaheuristic algorithms could optimize effectively without exceeding resource limits.

Since several hyperparameters, such as activation functions, batch size, and the number of filters, are discrete variables, the continuous outputs of the metaheuristic algorithms (GWO and WOA) were rounded to the nearest valid values. This rounding strategy maintained compatibility with the discrete nature of the hyperparameters while preserving the algorithms' exploration and exploitation capabilities.

4. Results

The objective of this section is to present the simulation outcomes of the proposed method and to examine the performance of metaheuristic algorithms in optimizing the CNN model for EEG-based emotion recognition. By defining a comprehensive cost function—designed to minimize error and complement accuracy under K-Fold validation—the optimization process was guided toward architectures and hyperparameters that improved classification performance compared to the baseline reference model.

4.1. Optimization Results

For both metaheuristic algorithms (GWO and WOA), the population size was fixed at 15 agents, and the optimization process was executed for 10 iterations. Although the number of iterations may appear limited, experimental observations revealed that the optimization trajectories converged rapidly, with no significant changes in the hyperparameters or cost function values beyond the tenth iteration. Consequently, the optimization was terminated early due to premature convergence, ensuring computational efficiency without compromising performance. This approach provided an optimal balance between exploration and exploitation while avoiding redundant evaluations. Multiple independent runs confirmed that this early convergence behavior was consistent across different initializations, further validating the robustness of the chosen settings.

The optimization trajectories, as illustrated in Figure 4, highlight apparent differences between the Grey Wolf Optimizer (GWO) and the Whale Optimization Algorithm (WOA) in minimizing the cost function. GWO demonstrated rapid convergence, with the cost function decreasing sharply from 1.7537 in the first iteration to 0.5058 in the tenth iteration. This reflects its strong exploitation capability and efficient refinement of the CNN architecture and learning parameters. In contrast, WOA exhibited slower but more stable convergence, reducing the cost function from 1.3762 to 0.4952 across ten iterations. The smoother trajectory reflects stronger exploration capacity and resilience against local minima, though with less aggressive convergence dynamics than GWO.

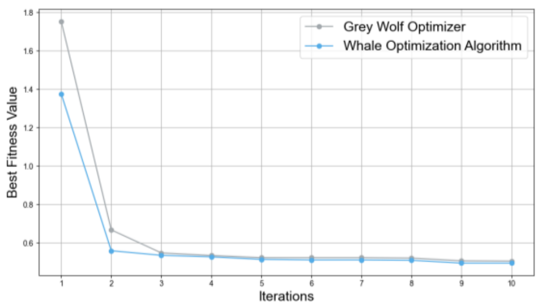

Figure 4. Comparing the best value found in each metaheuristic optimization iteration.

4.2. Best results found

Figures 5 and 6 provide compelling evidence of the comparative performance of the proposed CNN model when optimized by two metaheuristic algorithms: the Grey Wolf Optimizer (GWO) and the Whale Optimization Algorithm (WOA). Both figures are derived from a rigorous five-fold cross-validation procedure, ensuring that the reported outcomes reflect robust and generalizable trends rather than fold-specific variations. In these experiments, the optimal hyperparameters governing both the model architecture and the learning process were identified and applied through the respective optimization strategies.

Validation Accuracy (Figure 5): The trajectory of validation accuracy across 30 epochs clearly demonstrates the superiority of GWO over WOA. The accuracy curve associated with GWO exhibits a smoother and more stable ascent, culminating in values approaching 0.86. This reflects the algorithm's strong ability to exploit solutions and refine hyperparameters effectively. In contrast, WOA maintains comparatively lower accuracy levels throughout training, suggesting slower adaptation and weaker convergence dynamics.

Figure 6 illustrates the validation loss curves of the models optimized using two different optimizers, demonstrating the clear superiority of the GWO optimized model over the WOA-optimized model. the advantage of GWO. Across all epochs, GWO consistently yields lower loss values, indicating more efficient convergence and reduced susceptibility to overfitting. WOA, however, displays higher and more fluctuating loss values, reflecting less stability in guiding the learning process and weaker control over error minimization.

Figures 5 and 6 confirm that the proposed cost function effectively guided the optimization process. GWO achieved faster and more stable convergence, whereas WOA exhibited slower and less consistent improvements. These findings underscore the robustness of the optimization framework and highlight its potential to advance EEG-based emotion recognition by simultaneously enhancing accuracy and reducing error.

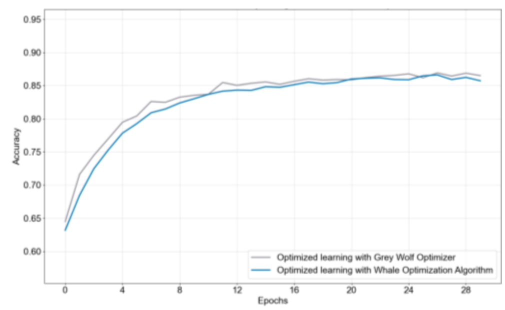

Figure 5. Comparison of overall validation accuracy between models optimized by metaheuristic algorithm.

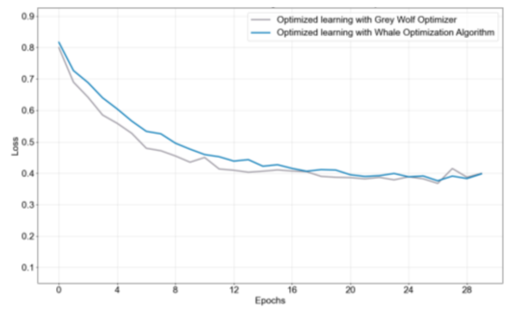

Figure 6. Comparison of validation loss between models optimized by metaheuristic algorithm.

Figures 7 and 8 illustrate the confusion matrices of the models optimized using GWO and WOA. The reported values include mean accuracies with standard deviations, computed through five-fold cross-validation. In comparison, the model optimized with the Grey Wolf Optimizer (GWO) demonstrated a more balanced performance, reflected in its lower standard deviation of prediction accuracy and achieved over 84% accuracy consistently across all classes (positive, negative, neutral). This indicates a more stable and reliable classification performance than the model optimized with WOA.

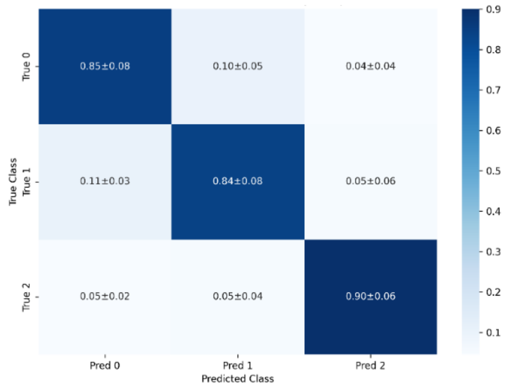

Figure 7. Confusion matrix and standard deviation for the method introduced with GWO.

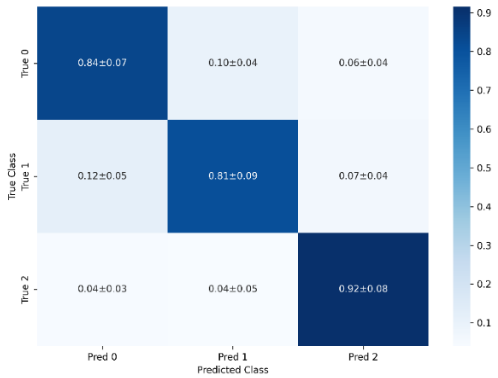

Figure 8. Confusion matrix and standard deviation for the method introduced with WOA.

5. Conclusion

This study demonstrates that metaheuristic optimization, guided by a comprehensive cost function, can significantly enhance the performance of Convolutional Neural Networks (CNNs) in EEG-based emotion recognition.

To ensure a fair and valid comparison, we evaluated the performance of our optimized model under conditions strictly equivalent to previous works. Specifically, we employed the KFoldGroupByTrial cross-validation method with 1-second EEG windows on the three-class SEED EEG dataset, a methodology and dataset consistent with the comparison presented in reference [12]. As observed in Table 5, Convolutional Neural Network (CNN)-based models have garnered particular attention in the field of emotion recognition due to their effectiveness. Our findings, when compared to other baseline models, indicate a highly competitive opportunity for optimizing the performance of prior deep learning models. It is noteworthy that, similar to CCNN, the authors of [12] introduced revised versions in 2024 tailored for the SEED EEG dataset.

Table 5. Comparison of the accuracy of our developed model with the latest development in TorchEEG Toobox.

DL Models	ACC%	Year	ref	
EEGNet	63/49%	2018	[19]	CNN-based
FBCNet	74/28	2021	[20]	CNN-based
TSCeption	74/95%	2022	[21]	CNN-based
CCNN	83%	2018	[8]	CNN-based
CCNN + GWO	**86%**	**-**	**Ours**	**CNN-based**
CCNN + WOA	**86%**	**-**	**Ours**	**CNN-based**
DGCNN	78/83%	2018	[22]	GCN-based
LGGNet	75/23%	2021	[23]	GCN-based
Transformer	57/86%	2017	[24]	transformer-based
ViT	57/38%	2020	[25]	transformer-based
SimpleVit	58/69%	2022	[26]	transformer-based
ArjunViT	53/53%	2021	[27]	transformer-based
SST-EmotionNet	73.83%	2020	[28]	SST based attention

By refining critical hyperparameters, classification accuracy improved from 83% in the baseline CCNN (TorchEEG) to 86% in the optimized models.

As shown in Table 6 and Figure 9, the optimized architectures achieved higher accuracy while reducing model complexity and computational resource requirements.

Table 6. Comparing the overall accuracy results of the previous model with our research.

Introduced model	CCNN (TorchEEG) [12]	CCNN + GWO	CCNN + WOA
Acc (%)	83%	86%	86%
Act Conv	RELU	GELU	Sigmoid
Act FC	SiLU	PRELU	PRELU
Conv1	64	128	32
Conv2	128	96	48
Conv3	256	96	32
Conv4	64	48	32
Fc1	1024	196	128
Lr Adam	0.0001	0.00053	0.00053
Weight decay	0.0001	9.99e-06	9.99e-06
Batch size	64	128	64

The number of parameters, FLOPs, and memory requirements were computed using the ptflops library in PyTorch for inputs of size $9\times9\times4$ with batch size 1. The results confirm that the optimized models are more efficient than the baseline. Compared to the baseline, both GWO- and WOA-optimized CCNNs employed more efficient activation functions, streamlined convolutional and fully connected layers, and adjusted learning parameters to balance accuracy with efficiency.

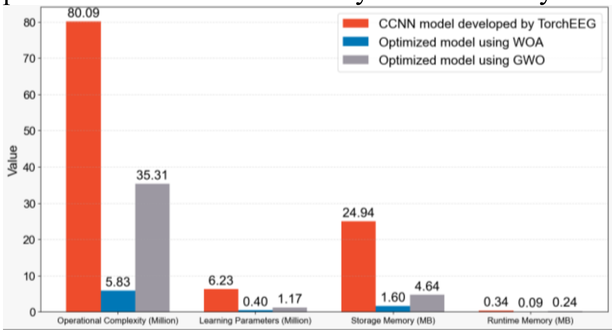

Figure 9. Review of sources and model complexity.

These changes lowered the number of trainable parameters, reduced operational overhead, minimized memory requirements, and decreased inference load, thereby enabling deployment in resource-constrained environments.

Among the metaheuristic strategies, the Grey Wolf Optimizer (GWO) consistently demonstrated superior convergence speed, stability, and balanced classification outcomes, confirming its suitability for tasks where reliability and efficiency are critical. While the Whale Optimization Algorithm (WOA) contributed meaningful exploration ability, its performance was less stable in this context. According to the "No Free Lunch" theorem, these findings emphasize that algorithm choice should reflect the dimensionality and complexity of the task.

Overall, this research highlights a holistic engineering approach that integrates accuracy, stability, and computational efficiency. By demonstrating the effectiveness of GWO in optimizing CNNs for EEG emotion recognition, the study contributes both theoretical insight and practical relevance, paving the way for scalable and reliable affective computing systems.

6. Limitations and suggestions

Although the results are promising, challenges persist. Model transferability to new datasets: An optimized model may not achieve similar performance when applied to other datasets. It is therefore recommended to design a composite cost function that simultaneously considers multiple datasets.

Information leakage in validation: The use of the KFoldGroupByTrial module in TorchEEG increases the risk of information leakage across sessions. Future designs should adopt more rigorous and reliable validation methods to ensure the integrity of results.

Approximate nature of search: Because of their stochastic and constraint-driven processes, metaheuristic algorithms typically generate near-optimal rather than exact outcomes. In scenarios where adequate computational resources are available, adopting more exact algorithms within reduced search spaces—leveraging prior knowledge—becomes a more advisable strategy.

References

[1] K. Kamble and J. Sengupta, "A comprehensive survey on emotion recognition based on electroencephalograph (EEG) signals," *Multimedia Tools and Applications*, vol. 82, no. 18, pp. 27269–27304, 2023.

[2] D. W. Prabowo, H. A. Nugroho, N. A. Setiawan, and J. Debayle, "A systematic literature review of emotion recognition using EEG signals," *Cognitive Systems Research*, vol. 82, p. 1452, 2023.

[3] M. Jafari, et al., "Emotion recognition in EEG signals using deep learning methods: A review," *Computers in Biology and Medicine*, vol. 165, p. 107450, 2023.

[4] S. K. Khare, V. Blanes-Vidal, E. S. Nadimi, and U. R. Acharya, "Emotion recognition and artificial intelligence: A systematic review (2014–2023) and research recommendations," *Information Fusion*, vol. 102, p. 102019, 2024.

[5] M. Kaveh and M. S. Mesgari, "Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review," *Neural Processing Letters*, vol. 55, pp. 4519–4622, 2022.

[6] Z. Gao, Y. Li, Y. Yang, X. Wang, N. Dong, and H.-D. Chiang, "A GPSO-optimized convolutional neural networks for EEG-based emotion recognition," *Neurocomputing*, vol. 380, pp. 225–235, 2020.

[7] M. R. Falahzadeh, F. Farokhi, A. Harimi, et al., "Deep convolutional neural network and gray wolf optimization algorithm for speech emotion recognition," *Circuits Systems and Signal Processing*, vol. 42, pp. 449–492, 2023.

[8] Y. Yang, Q. Wu, Y. Fu, and X. Chen, "Continuous convolutional neural network with 3D input for EEG-based emotion recognition," in *ICONIP* , vol. 11307, pp. 433–443, 2018.

[9] S. Hwang, K. Hong, G. Son, and H. Byun, "Learning CNN features from DE features for EEG-based emotion recognition," *Pattern Analysis and Applications*, vol. 23, pp. 1323–1335, 2019.

[10] M. Abd Elaziz, A. Dahou, L. Abualigah, L. Yu, M. Alshinwan, A. M. Khasawneh, and S. Lu, "Advanced metaheuristic optimization techniques in applications of deep neural networks: A review," *Neural Computing and Applications*, vol. 33, pp. 14079–14099, 2021.

[11] A. K. Mishra, S. K. Singh, and A. K. Sahoo, "Evolution of convolutional neural network (CNN):

Compute vs memory bandwidth for edge AI," *arXiv:2311.12816,* 2023.

[12] Z. Zhang, S. Zhong, and Y. Liu, "TorchEEGEMO: A deep learning toolbox towards EEG-based emotion recognition," *Expert Systems with Applications*, vol. 249, 123550, 2024.

[13] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[15] S. B. Lee, H. H. Chang, and C. H. Chuang, "Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition," in *ACII*, Xi'an, China, pp. 1–6, 2015.

[16] R. Salgotra, P. Sharma, S. Raju, and A. H. Gandomi, "A Contemporary Systematic Review on Meta-heuristic Optimization Algorithms with Their MATLAB and Python Code Reference," *Archives of Computational Methods in Engineering*, vol. 31, pp. 1749–1822, 2024.

[17] S. Mirjalili, "Grey Wolf Optimizer (GWO)," https://seyedalimirjalili.com/gwo. Access Date: Dec. 2025.

[18] S. Mirjalili, "Whale Optimization Algorithm (WOA)," https://seyedalimirjalili.com/woa. Access Date: Dec. 2025.

[19] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 15, no. 5, 056013, 2018.

[20] R. Mane, E. Chew, K. Chua, et al., "FBCNet: A multi-view convolutional neural network for brain-computer interface," *arXiv:2104.01233,* 2021.

[21] Y. Ding, N. Robinson, S. Zhang, et al., "TSception: Capturing temporal dynamics and spatial asymmetry from EEG for emotion recognition," *arXiv:2104.02935,* 2021.

[22] W. Li, W. Huan, B. Hou, Y. Tian, Z. Zhang, and A. Song, "Can emotion be transferred?—A review on transfer learning for EEG-based emotion recognition," *IEEE Transactions on Cognitive and Developmental Systems,* vol. 14, no. 3, pp. 833–846, 2021.

[23] Y. Ding, N. Robinson, Q. Zeng, and C. Guan, "LGGNet: Learning from local global-graph representations for brain-computer interface," *arXiv:2105.02786,* 2021.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al., "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929,* 2020.

[26] L. Beyer, X. Zhai, and A. Kolesnikov, "Better Plain ViT Baselines for ImageNet-1k," *arXiv:2205.01580,* 2022.

[27] A. Arjun, A. S. Rajpoot, and M. R. Panicker, "Introducing attention mechanism for EEG signals: Emotion recognition with vision transformers," in *EMBC*, pp. 5723–5726, 2021.

[28] M. Esmaeiili and K. Kiani, "Enhancing Emotion Classification via EEG Signal Frame Selection," *Journal of Artificial Intelligence and Data Mining*, vol. 12, no. 1, pp. 83–93, 2024.

بهینه‌سازی فراابتکاری هایپرپارامترهای مدل یادگیری عمیق و شبکه عصبی کانولوشنی برای طبقه‌بندی احساسات مبتنی بر سیگنال‌های الکتروانسفالوگرام (EEG)

یاشار چهارده چریکی قیصری و هادی گرایلو*

دانشکده مهندسی برق، دانشگاه صنعتی شاهرود، شاهرود، ایران .

ارسال ۲۰۲۵/۱۲/۲۹؛ بازنگری ۲۰۲۶/۰۳/۱۴؛ پذیرش ۲۰۲۶/۰۵/۰۲

چکیده:

بازشناسی احساسات انسانی مبتنی بر سیگنال‌های الکتروانسفالوگرام همچنان یک چالش مهم در علوم اعصاب محاسباتی و هوش مصنوعی محسوب می‌شود. شبکه‌های عصبی کانولوشنی به‌دلیل توانایی‌های قوی خود در استخراج ویژگی و یادگیری بازنمایی، به‌طور گسترده برای مقابله با این چالش به کار گرفته شده‌اند. با این حال، تنظیم تجربی ابرپارامترها اغلب ناکارآمد و مستعد راه‌حل‌های غیربهینه است. برای غلبه بر این محدودیت، بهینه‌ساز گرگ خاکستری و الگوریتم بهینه‌سازی نهنگ—دو روش فراابتکاری مبتنی بر هوش جمعی—برای بهبود معماری و ابرپارامترهای یادگیری یک شبکه عصبی کانولوشنی پایه به کار گرفته شدند. عملکرد این مدل‌ها بر روی مجموعه‌داده SEED EEG برای بازشناسی احساسات ارزیابی شد. نتایج آزمایش‌ها نشان داد که مدل بهینه‌سازی‌شده به صحت ۸۶٪ دست یافت، در حالی که این مقدار برای مدل پایه ۸۳٪ بود، و ماتریس درهم‌ریختگی کاهش خطاهای طبقه‌بندی و بهبود بازشناسی حالات هیجانی را تأیید کرد. فراتر از صحت، فرآیند بهینه‌سازی تعداد پارامترهای قابل آموزش و بار محاسباتی را نیز کاهش داد و در نتیجه کارایی را افزایش داد. این یافته‌ها نشان می‌دهند که روش‌های مبتنی بر هوش جمعی قابلیت‌های جست‌وجو و بهره‌برداری برتری فراهم می‌کنند و امکان تنظیم نظام‌مند ابرپارامترها و ارائه یک رویکرد مهندسی متعادل را فراهم می‌سازند که عملکرد پیش‌بینی را با کارایی منابع ترکیب می‌کند

کلمات کلیدی: سیگنال‌های مغزی، بازشناسی احساسات، پردازش سیگنال EEG، الگوریتم فراابتکاری.