Intelligent Hyperparameter Optimization of Convolutional Neural Networks for Robust Multimodal Classification

Jing Zhao[1], Hui Xing[1], Qinwei Fan[1,2]
[1]School of Science, Xi'an Polytechnic University, Xi'an, Shaanxi, China.
[2]School of Mathematics and Information Science, Guangzhou University, Guangzhou, China

Abstract: In this study, a novel hyperparameter optimisation algorithm for Convolutional Neural Networks (CNNs), based on a Multi-Strategy Improved Whale Optimisation Algorithm (IWOA), is proposed to enhance performance across diverse tasks. Recognising the critical impact of hyperparameters on CNN efficacy, the algorithm is designed to autonomously identify optimal parameter settings. To improve population diversity and uniformity during initialisation, the Singer chaotic mapping strategy is employed. Additionally, a nonlinear dynamic speed regulation mechanism is introduced to refine the spiral update control parameters in WOA, thereby enhancing the optimisation process. To further address premature convergence and avoid local optima, Gaussian mutation is utilised, enabling the algorithm to achieve faster convergence toward the global optimum. The enhanced IWOA is integrated with CNNs and evaluated on multiple benchmark functions to validate its optimisation capability. Moreover, extensive image classification experiments on various datasets demonstrate the algorithm's effectiveness in improving CNN recall and accuracy while showcasing strong generalisation ability. The results highlight that the proposed approach significantly outperforms traditional methods in searching and optimising CNN hyperparameters, delivering superior performance and robustness across multiple tasks.

Keywords: Whale Optimisation Algorithm; Convolutional Neural Network; Hyperparameters Optimization; Image Classification

1. Introduction

Swarm intelligence optimization algorithms are inspired by collective behaviors observed in nature, where cooperation and communication among individuals enable effective problem-solving. These algorithms exhibit flexibility, self-organization, robustness, and distributed processing, making them powerful tools for addressing complex optimization problems [1] Over the past decade, their applications have expanded rapidly, accompanied by the continuous emergence of new variants [2]. Representative examples include the Black-winged Kite Optimization Algorithm [3], the Electric Eel Foraging Optimization Algorithm [4], the Honey Badger Optimization Algorithm [5], and the South American Raccoon Optimization Algorithm [6]. In 2016, Mirjalili et al. introduced the Whale Optimization Algorithm (WOA), inspired by the unique biological behaviors of humpback whales [7]. WOA models hunting strategies such as prey encirclement, bubble-net predation (including shrinking encirclement and spiral updating), and prey searching. The algorithm initializes the positions of whales in the population and iteratively updates them to converge toward the optimal solution [8]. By simulating behaviors such as floating, diving, and role differentiation (e.g., leader, follower, and random whales), WOA effectively balances global exploration with local exploitation. Moreover, during the iterative process, whales dynamically adjust their search strategies according to the changes in objective function values, thereby enhancing the ability to discover high-quality solutions [9].

Convolutional Neural Networks (CNNs) are among the most extensively studied models in the field of image recognition, achieving recognition accuracy that in some cases even surpasses human performance. Their flexibility arises from the combination of multiple layers and nonlinear activation functions, enabling users to adapt architectures to different tasks [10]. The history of CNNs can be traced back to 1962 [11], while the LeNet model marked the prototype that underwent iterative refinements and laid the

foundation of modern CNNs [12]. A major breakthrough occurred in 2012 with the introduction of AlexNet, which significantly advanced deep learning for computer vision. Since then, numerous architectures have been developed, including VGG [13], ResNet [14], and GoogLeNe t[15], each further enhancing CNN capabilities.

Despite these advances, increasing network depth often leads to reduced precision due to issues such as vanishing gradients and overfitting[16]. To address this challenge, researchers have sought to improve performance while maintaining the architectural structure. CNN efficiency and accuracy are influenced by a range of hyperparameters, including kernel size, number of channels, stride, and pooling window. As model complexity grows, the sensitivity of performance to hyperparameter settings amplifies, making hyperparameter optimization a critical factor in maximizing CNN performance [17]. Traditional tuning methods based on expert knowledge rely heavily on trial-and-error, which is inefficient and difficult to scale. Consequently, systematic approaches such as grid search, random search, Bayesian optimization, and gradient-based optimization techniques have been developed [18]. However, their computational complexity increases rapidly with the dimensionality of the parameter space, leading to significant efficiency bottlenecks when applied to high-dimensional optimization problems [19].

To address the aforementioned challenges, researchers have explored biomimetic intelligence-based optimization strategies, with notable examples including the artificial bee colony algorithm inspired by swarm intelligence [20], particle swarm optimization that mimics biological migration [21], and genetic algorithms derived from evolutionary mechanisms [22]. Nevertheless, most existing approaches primarily rely on structural modifications of network architectures or the fusion of multiple algorithms. Such composite optimization mechanisms inevitably increase system complexity, reduce algorithm interpretability, and raise the difficulty of practical applications. Therefore, developing hyperparameter optimization methods that preserve the native CNN architecture while ensuring simplicity and usability has become an urgent research priority.

In this study, we aim to optimize CNN hyperparameters using the Whale Optimization Algorithm (WOA), originally proposed by Mirjalili [23]. Owing to its simple structure and efficient computational mechanism, WOA has attracted considerable attention and has demonstrated competitive performance across a wide range of optimization problems [24]. Building upon this foundation, we further introduce an Improved Whale Optimization Algorithm (IWOA) to intelligently optimize CNN hyperparameters. Specifically, the proposed IWOA is applied to CNN models without altering their architectural layers, enabling automatic hyperparameter optimization across different datasets. Experimental results validate the effectiveness of the improved method in enhancing CNN performance [25]. The main contributions as follows:

(1) A High-Efficiency Deep Learning Framework Based on CNN: This study presents an efficient deep learning framework built on Convolutional Neural Networks (CNNs), employing a hierarchical feature extraction mechanism. Through local perception, weight sharing, and pooling operations, the framework achieves spatial dimensionality reduction and exhibits outstanding image feature abstraction capabilities. It is well-suited for image classification, object detection, and feature extraction tasks. Key modules, including convolutional layers, activation functions, pooling layers, and fully connected classifiers, are designed to significantly enhance robustness against geometric transformations (e.g., translation and rotation) while reducing parameter complexity.

(2) Improvements to the Whale Optimisation Algorithm (IWOA): To address the limitations of the standard Whale Optimisation Algorithm (WOA)—such as susceptibility to local optima and limited global search ability—this study proposes three key enhancements. First, the Singer chaotic map is adopted to initialise a uniformly distributed and diverse population, boosting search space exploration. Second, a nonlinear time-varying convergence factor is introduced to dynamically balance global exploration and local exploitation. Third, Gaussian mutation operations are applied during later iterations to perturb the optimal region and prevent premature convergence. These improvements collectively enhance global convergence accuracy and optimisation stability.

(3) IWOA-CNN: A Hybrid Approach and Experimental Validation: The proposed IWOA is

integrated with a CNN model to optimise its hyperparameters, forming the IWOA-CNN framework. Experimental results on multiple datasets demonstrate the framework's effectiveness in image classification, achieving superior accuracy, recall, and generalisation performance. These findings not only confirm the practical applicability of the approach but also validate the theoretical analysis, providing a reliable and efficient solution for CNN hyperparameter optimisation.

The rest of this study is structured as follows. The second section introduces the background work related to whale optimisation algorithms and convolutional neural networks. The third part introduces in detail the improved strategy of the whale optimisation algorithm and the design of the convolutional neural network model structure. Section IV includes experimental results, discussion, and analysis. Finally, the conclusions are briefly summarised in section V.

2. Relevant Technical Framework

In this part, we will retrospect the detailed steps of WOA and some details of CNNs.

2.1 The Fundamental Whale Optimization Algorithm

Whales are highly intelligent marine mammals with complex emotional expressions and unique foraging strategies. One of their most remarkable behaviors is bubble-net hunting, a cooperative feeding method that inspired Mirjalili and Lewis to propose the Whale Optimization Algorithm (WOA) in 2016 [26]. WOA is a novel swarm intelligence optimization algorithm that mimics the hunting strategies of humpback whales. Its core mechanism is based on biomimetic simulation of whale predation, and the algorithm mainly consists of three key phases: searching for prey, encircling prey, and bubble-net hunting [27]. In this framework, each whale's position in the search space represents a candidate solution, and the optimization process simulates whale hunting strategies through spiral position updates and shrinking encirclement operations on solution vectors. Over successive iterations, the population collectively explores the search space and gradually converges toward the optimal solution of the objective function [28].

Assume that there are N whales in the WOA. The spatial coordinates of the target prey in the solution space correspond to the global extremum of the optimization problem. At the initialization stage, the position of the i-th whale, also referred to as the leader whale, can be mathematically represented as: Assume that there are N whales in the WOA. The spatial coordinates of the target prey in the solution space correspond to the global extremum of the optimization problem. At the initialization stage, the position of the i-th whale, also referred to as the leader whale, can be mathematically represented as: $X_i = [x_{1i}, x_{2i}, x_{3i} ... x_{Di}], i = 1, 2, ..., N.$ Where D represents the dimensional space of the solution vector.

In the evolutionary mechanism of WOA, once a search agent identifies a candidate optimal position vector, individuals employ a probabilistic exploration strategy to conduct global random searches in the solution space, rather than solely updating their positions relative to the current best solution. Specifically, when the condition A≥1 is satisfied (see equation (2)), the search agent selects reference whales at random and performs position migration based on them. The coordinate values are then iteratively updated according to the predefined mathematical update rules, thereby guiding the swarm toward potential global optima. The specific formal expression is equation (1).

$$X_i^{t+1} = X_{rand}^t - A\left|C \cdot X_{rand}^t - X_i^t\right| \quad (1)$$

Where X_{rand}^t is the current population's randomly selected location, X_i^t and X_i^{t+1} are the current and updated positions of the search agent, and A and C are modulus vectors, as defined by equation (2) and (3) respectively.

$$\mathrm{A} = 2\mathrm{a} \cdot r - a \quad (2)$$

$$\mathrm{C} = 2 \cdot \mathrm{r} \quad (3)$$

Here, r is a random vector within the interval [0,1], typically referred to as the spiral update control parameter. During iterations, it decreases linearly from 2 to 0. The associated parameter a guides the agents to gradually converge toward the neighborhood of the optimal solution as the algorithm progresses. This parameter plays a critical role in regulating both the speed and the amplitude of the whales' spiral movements within the search space.

In the swarm intelligence simulation of whale hunting, when individuals enter the prey-encirclement stage, the system designates the current best solution vector as the global optimal position. At this stage, the population follows a phasesynchronization mechanism, and

the coordinates of each whale are iteratively updated according to the predefined differential equation (4). Through this process, all agents progressively converge toward the global optimum.

$$X_i^{t+1} = X_{best}^t - A\left|C \cdot X_{best}^t - X_i^t\right| \quad (4)$$

Where t is the present iteration and X_{best}^t is the position of the optimal solution representing the present best candidate solution. Furthermore, when the whale surrounds its prey, $A < 1$.

When humpback whales employ bubble-net feeding, the strategy can be described by two mechanisms.

(1) Shrink-and-wrap mechanism. In this mechanism, A is defined as a random value within the interval [−a, a], where a decreases linearly from 2 to 0 during iterations, as shown in equation (2). When $A \in (-1,1)$, the position update rule enables the search agent to relocate to any point within the region bounded by its current position and the best-known solution. As illustrated in Figure 1, this process reflects the geometric accessibility of the search space, capturing the possibility for agents to migrate from their initial coordinates (X, Y) to the target coordinates (X′, Y′).

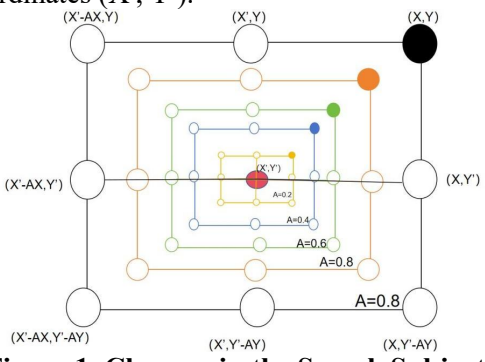


Figure 1. Changes in the Search Subject's Position Under the Shrinking and Surrounding Mechanism

(2) Spiral update position mechanism. This mechanism simulates the distinctive foraging strategy of humpback whales by computing the spatial distance between the current position (X, Y) and the target position (X′, Y′). The prey is then approached along a logarithmic spiral trajectory, as expressed in equation (5), which effectively models the spiral motion patterns observed in bubble-net hunting.

$$X_i^{t+1} = D' e^{bl} \cos(2\pi l) + X_{best}^t \quad (5)$$

Let $D = \left|X_{best}^t - X_i^t\right|$ denote the Euclidean distance between the i-th whale and the current best candidate solution, where l is a normalized random coefficient in the interval [−1, 1], and b is the adjustment parameter controlling the spiral shape. Based on these parameter definitions, a spiral approximation model can be constructed to characterize the hunting behavior of humpback whales. This model describes the motion of whale populations performing spiral position updates within a gradually shrinking circular region around the prey. Its core mechanism lies in simultaneously satisfying two constraints: contraction toward the prey and trajectory tracking along a logarithmic spiral. The resulting motion can be mathematically formulated as equation (6).

$$D = \begin{cases} X_{best}^t - A\left|C \cdot X_{best}^t - X_i^t\right|, P < 0.5 \\ D' e^{bl} \cos(2\pi l) + X_{best}^t, P \geq 0.5 \end{cases} \quad (6)$$

The parameter p is used as a probability distribution variable within the [0, 1] interval to control the switching threshold of the search agent location update strategy. As shown in Figure 2, in the visualization of the spiral position update mechanism, the horizontal axis corresponds to the adjustment factor l in formula (5), and the vertical axis maps the updated coordinate distribution in two-dimensional space. This dual parameter joint regulation mechanism achieves dynamic optimization of search paths through the synergistic effect of probability selection and geometric constraints.

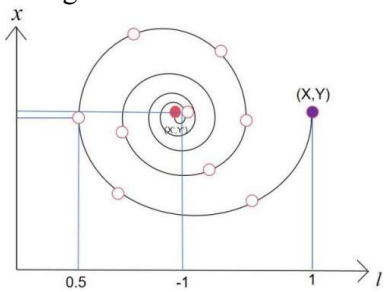


Figure 2. Spiral Update Search Agent Position

2.2 Architecture of CNNs

This section focuses on analyzing the representative architectures of deep CNNs. As an artificial intelligence system based on hierarchical feature learning, CNNs perform image object detection, recognition, and classification through parameterized filter banks, and are capable of advanced visual tasks such as pixel-level semantic segmentation [29]. Figure 3 illustrates the topology of a typical CNN, which consists of alternating convolutional modules and pooling layers for spatial downsampling. The network comprises two feature extraction stages, two spatial dimensionality reduction

operations, four progressively abstract feature map groups, and an end-to-end fully connected classifier. This hierarchical feature abstraction enables shallow layers to capture basic patterns such as edges and textures, while deeper layers extract higher-order semantic representations that are discriminative for complex visual recognition tasks.

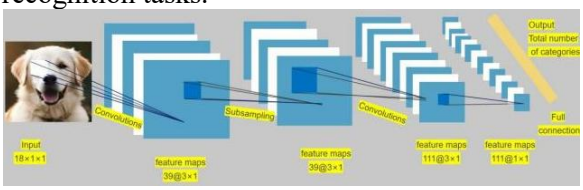


Figure 3. Architecture of the CNN Model

(1) Convolutional layer. The convolutional layer primarily performs feature extraction. It applies learnable filters to the input tensor via sliding convolution operations, enabling local feature detection and spatial information encoding. This process effectively captures spatial correlations in the input data through parameter sharing, allowing the network to hierarchically learn local patterns in images. Mathematically, given an input feature tensor X, convolution kernel parameters W, and a bias term b, the output feature map Y can be obtained by applying discrete convolution followed by a nonlinear transformation. Formally, this process can be expressed as the nonlinear transformation of the discrete convolution of X and W with the bias term b incorporated [30].

$$Y(i,j)=\sum_m\sum_n X(i+m,j+n)\times W(m,n)+b \tag{7}$$

In the above notation, $Y(i,j)$ denotes a pixel value in the output feature map from the convolutional layer, $X(i+m,j+n)$ denotes a pixel value in the input data, $W(m,n)$ represents the filter weight, b is the bias term; and i and j indicate the coordinates in the output feature map.

The convolution operation performs local feature extraction on the input tensor using a sliding filter, and its computation can be decomposed into three fundamental steps: first, an element-wise dot product is computed between the input patch and the convolution kernel within the sliding window; second, the results are summed and added to the bias term; finally, the corresponding activation value in the output feature map is generated. The learnable parameters of this layer, including the filter weights and bias vectors, are optimized via gradient-based updates during backpropagation, enabling the network to autonomously extract effective feature representations from the input data. As the core computational unit of CNNs, convolutional layers form a hierarchical feature abstraction structure through multi-layer stacking. Shallow layers capture basic patterns such as edges and textures, whereas deeper layers progressively integrate features to construct high-order semantic representations. This hierarchical feature abstraction allows the network to perform multi-level representation learning and in-depth analysis of visual information.

(2) Pooling layer. The pooling layer is a crucial module in CNN architectures, responsible for feature map dimensionality reduction and enhancement of model robustness through spatial downsampling. This mechanism reduces the number of model parameters while compressing the spatial dimensions of feature maps by aggregating information within local receptive fields. By introducing spatial invariance, the pooling layer improves the network's resilience to geometric variations in the input data, such as translation, scaling, and rotation. The core operation involves applying a sliding window over the feature map to extract regional features, followed by aggregation via operations such as maximum or average pooling. This dimensionality reduction strategy effectively mitigates the risk of overfitting while preserving essential feature representations.

There are various forms of implementation for pooling operations, among which the most widely used is the max pooling mechanism. The mathematical expression of this method can be defined as: given an input feature matrix X and a sliding window size S, the max pooling layer uses a local region feature extremum extraction mechanism to achieve spatial dimension compression. The calculation formula for its output matrix Y can be expressed as:

$$Y(i,j)=\max_{m,n} X(i+S\times m,j+S\times n) \tag{8}$$

where $Y(i,j)$ represents a pixel value in the feature map output by the pooling layer; $X(i+S\times m,j+S\times n)$ represents selecting the maximum value in the pooling window. $\max_{m,n}$ represents the pixel value inside the pooling area within the input data.

The pooling layer, as a core component of CNN architectures, performs data dimensionality reduction while preserving essential information

through feature aggregation. Specifically, the max pooling operation reduces dimensions by selecting the maximum value within each local receptive field, thereby emphasizing salient features and reducing data redundancy. Average pooling serves as a complementary strategy, performing smooth dimensionality reduction by computing the mean of features within a sliding window. These spatial downsampling operations not only reduce the size and complexity of feature maps, enhancing computational efficiency and model robustness, but also support improved generalization performance by promoting effective feature abstraction.

(3) Fully connected layer. The fully connected layer, also known as a dense or linear transformation layer, is a fundamental component in deep neural network architectures. It establishes connections between each neuron in the current layer and all nodes in the preceding layer through a fully interconnected weight matrix. This global connectivity enables powerful feature integration, allowing the layer to model complex nonlinear relationships among input features via spatial linear combinations of weighted parameters. Although this fully connected topology introduces substantial computational complexity, its parameterized feature fusion mechanism provides critical capabilities for high-level feature abstraction in deep learning models.

Suppose the output of the previous layer is vector X, the weight matrix of the fully connected layer is W. The output of the fully connected layer can be calculated by $Y(i,j)=\sigma(W\cdot X+b)$, where W is the weight matrix, each row of the matrix corresponds to the weight vector of a neuron; b is the bias term, which is used to regulate the activation value of each neuron; RELU is the activation function, usually a nonlinear function, such as Rectified Linear Unit function, Sigmoid function or Tanh function; represents matrix multiplication.

The fully connected layer performs an affine transformation of feature vectors by linearly combining the outputs of the preceding layer with the weight matrix, adding bias vectors, and applying nonlinear activation functions. This hierarchical processing enables the network to generate high-order abstract feature representations through cascaded nonlinear transformations, thereby enhancing the model's capacity for nonlinear feature modeling. In typical CNN architectures, the fully connected layer is positioned at the network's end, serving as the mapping from feature space to decision space. Its primary function is to transform the distributed feature representations extracted by convolutional and pooling modules into the dimensions of the target classification space. This design allows deep networks to learn parameterized mappings from high-dimensional feature embeddings to specific class predictions [31].

3. Proposed Method

This section introduces the convolutional neural network architecture designed for our multimodal classification performance evaluation framework. The hyperparameter optimization model based on the Improved Whale Optimization Algorithm (IWOA) is employed to effectively extract informative features from the input dataset and maximize the predictive performance. Figure 2 presents a schematic overview of the overall process. A regularized fully connected layer is incorporated into the CNN, and IWOA is applied for hyperparameter tuning during the initial training stage. The optimized network is subsequently used for model training and evaluation. The detailed procedure is described as follows:

3.1 Proposed Improvement Strategies

The canonical Whale Optimization Algorithm (WOA) exhibits limitations in solution accuracy, slow convergence rates, and susceptibility to premature convergence at local optima. To address these shortcomings, this study proposes several improvements to the standard WOA to enhance its position update mechanism and reduce the risk of stagnation in suboptimal regions. First, a Singer chaotic mapping system replaces the original random initialization, leveraging the ergodic properties of chaotic sequences to generate a more diverse initial population. Second, a novel nonlinear convergence factor is introduced in place of the original linear convergence factor, balancing global exploration and local exploitation during the search process. Finally, a Gaussian mutation strategy is applied to improve the overall search capability, facilitating the algorithm's ability to escape from local optima and achieve better global performance.

(1) Singer Chaotic Mapping. Chaotic behavior is an inherently unstable phenomenon that emerges spontaneously in deterministic systems, typically observed in nonlinear dynamics [32]. Due to the ergodic nature of chaos, a chaotic system can traverse all possible states within a given range without repetition. Consequently, optimization strategies leveraging chaotic variables can achieve superior performance compared with purely random or disordered search methods. Singer chaotic mapping is a widely used chaotic system characterized by rich and complex dynamic properties. It is derived from the logistic map and combines piecewise linear mapping with nonlinear functions, exhibiting simplicity in mathematical form, ergodicity, and randomness. The introduction of Singer chaotic mapping has provided a foundational framework for both theoretical studies and practical applications of chaos. This mapping was first proposed by the American mathematician J. Singer in 1994, and its general form is presented in equation (9).

$$x_{k+1} = \mu(7,86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4) \quad (9)$$

Among them, $\mu \in [0.9,1.08], x \in [0,1]$. The distribution diagram of Singer chaotic mapping after 200 iterations as shown in the Figure 4. It can be clearly seen that Singer mapping is evenly distributed between [0,1]. The figure shows the dynamic behavior of the Singer chaotic map, which contains two sub-graphs. The upper graph depicts the evolution process of the chaotic sequence with number of iterations, with the initial value $x_0 = 0.5$, showing non-periodic oscillation characteristics. The phase space graph in the lower graph presents a typical chaotic attract structure, and the distribution of the point set shows that the system has fractal characteristics, verifying the intrinsic randomness of the chaotic system. The abscissa X marks the value points from 0 to 0.9, which is used to observe the ergodicity of the state variable in the unit interval. Using this feature can make the search space more uniform, increase the uniformity and diversity of the whale population, and thus increase the overall situation search capability.

(2) Nonlinear Convergence Factor. In heuristic optimization algorithms, balancing global exploration and local exploitation is crucial, and WOA faces this core challenge as well [33]. In the original WOA, the whale population's behavior is controlled by the parameter $|A|$ (derived from the convergence factor a): when $|A| \geq 1$, whales perform global exploration; when $|A| < 1$, they switch to local exploitation. However, the original linear decreasing strategy for a has inherent limitations when addressing complex optimization problems.

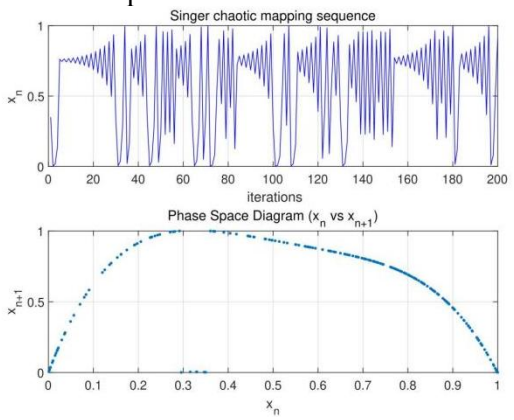


Figure 4. Singer Chaotic Map Iterative Sequence Diagram and Phase Space Diagram

First, it leads to insufficient early-stage exploration: high-intensity exploration is needed at the beginning to avoid premature convergence, but the linear decrease of a causes $|A|$ to enter the local exploitation stage too early, weakening the global search capability. Second, it results in inefficient late-stage exploitation: in the later stages, the algorithm must focus on local refinement, but the linear strategy decreases a gradually, delaying convergence to the ideal exploitation range and slowing solution quality improvement. Finally, it exhibits weak adaptability for high-dimensional, multi-modal problems, as linear changes cannot dynamically adjust the balance between exploration and exploitation according to the search process.

To overcome these limitations, we propose a nonlinear convergence factor to replace the original linear parameter (see equation (10)). This design reshapes the trajectory of a by introducing logarithmic functions: maintaining higher values in the early stage to extend global exploration, accelerating decay in the middle stage to facilitate a smooth transition from exploration to exploitation, and approaching zero in the later stage to enable deep local optimization. This nonlinear mechanism significantly enhances the algorithm's self-adaptability in complex solution spaces, ensuring sufficient early-stage exploration to

avoid premature convergence while improving late-stage exploitation accuracy, ultimately achieving a synergistic enhancement of global optimization efficiency and solution quality.

$$a = 2 - 2 \times (\frac{t}{Max\_iter})^2 \quad (10)$$

With in the algorithmic framework, Where Max_iter specifies the predefined maximum iteration threshold, whereas t recording the real-time iteration counter.

(3) Gaussian Mutation. Gaussian mutation is a probabilistic variation strategy based on the normal distribution, commonly employed in evolutionary algorithms. Its primary function is to fine-tune candidate solutions by introducing Gaussian perturbations, where the mean corresponds to the current solution and the standard deviation is adjustable, thereby emphasizing local refinement. Compared with uniform mutation, the high-probability region of the Gaussian distribution is concentrated near the mean, enabling the algorithm to perform small adjustments to explore the neighborhoods of high-quality solutions while still allowing low-probability larger jumps. This balance between exploitation and exploration makes Gaussian mutation particularly effective in continuous optimization problems, especially during the later stages of convergence, accelerating the approximation to the global optimum [34]. The position update operation for this strategy is defined in equation (11).

Among them, A is a dominate parameter used to regulate the amplitude of location update. D is the range between the target position and the present personal position. $N(0,\sigma^2)$ is a Gaussian variation term, which represents the value randomly sampled from a Gaussian scatter with a mean value of 0 and a variance of σ^2. By adjusting the σ value, the intensity of the variation can be controlled, thereby affecting the expansive roaming or focused refinement capabilities in the search. By introducing this method, WOA can introduce randomness when searching, thereby enhancing the potential of the algorithm to carry out a global search.

Now we call the algorithm that applies the above improved strategy IWOA, and provide a detailed explanation of its execution process through the following Table 1. The fitness curve of the improved algorithm is shown in Figure 5.

Table 1. Algorithm 1 IWOA

Algorithm 1 IWOA

1: Initialize the $X_i = (i = 1,2,...,N)$ by eq.(9)
2: Calculate the values of each whale
3: **while** t < maximumiteration **do**
4: **for** each whale **do**
5: Calculate values of a in eq.(10)
6: Generate random values of r, A, c, b and p
7: **if** p < 1=2 **then**
8: **if** $|A| \geq 1$ **then**
9: Select a random whale X_{rand}
10: Renew the position of the present whale by eq.(1)
11: **else**
12: Renew the location of the present whale by eq.(4)
13: **end if**
14: **end if**
15: Update the location of the present whale by eq.(5)
16: **end for**
17: Correct the location of the whales which goes beyond the search space
18: Convert the updated location by eq.(11)
19: Calculate the fitness value of whale before and after the conversion
20: Retain or instead of the unconverted location and its associated fitness value according to the caculation results
21: Update X_{best} if have the better solutions
22: $t = t + 1$
23: **end while**
24: Return X_{best}

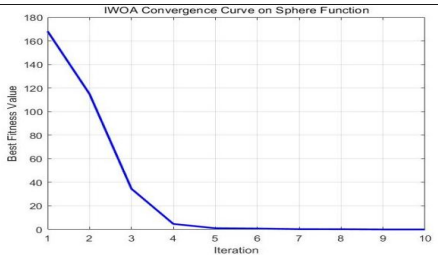


Figure 5. IWOA Fitness Curve

3.2 Network Architecture

This study constructs a deep CNN based on the classic LeNet-5 architecture [35]. The network comprises two convolutional layers for feature extraction, followed by a max pooling layer for dimensionality reduction, and concludes with three fully connected layers for classification, as illustrated in Figure 3 [36]. Network parameters are initialized using a random uniform distribution, and all convolutional and fully

connected layers employ ReLU activation to enhance the model's nonlinear representation capability. Model training is performed using the Adam optimizer, which integrates the benefits of momentum and adaptive learning rates. Key hyperparameters, including an initial learning rate of 1e-3 and an L2 weight decay coefficient of 1e-4, were determined through preliminary experimental validation. The Softmax function is used at the network's output layer to transform the final outputs into a probability distribution suitable for multi-class classification tasks.

During model initialization, the size and number of convolution kernels in the first convolutional layer (C_1), the pooling window size for the first pooling layer (P_2), the convolution kernel configuration for the second convolutional layer (C_2), and the parameter settings for the second pooling layer (P_3) are selected within predefined optimal ranges, with specific values provided in Table 2. The fully connected layer (FC_5) serves as the classifier output layer, with its neuron count strictly corresponding to the number of target categories. The structural parameters of this layer remain fixed and are excluded from subsequent hyperparameter optimization, based on the following considerations: 1) the output dimension is problem-dependent; 2) structural modifications may adversely affect classification performance; and 3) maintaining a fixed structure reduces the complexity of the optimization process.

Table 2. CNN Architecture and Hyperparameters Range

Layers	Hyperparameters range
C_1	Size of Kernels: 2×2, 3×3, 4×4
	Number of Kernels: [16, 64]
	Strides: 1×1
P_2	Size of Pooling: 2×2, 3×3, 4×4
	Strides: 1×1
C_3	Size of Kernels: 2×2, 3×3, 4×4
	Number of Kernels: [32, 128]
	Strides: 1×1
P_4	Size of Pooling: 2×2, 3×3, 4×4
	Strides: 1×1
FC_6	Manually set by number of categories

4. Experimental Result and Discussion

In this section, we evaluate the comprehensive performance advantages of the IWOA algorithm in optimizing CNNs through a series of multi-dimensional empirical analyses. The evaluation mainly focuses on two aspects: the improvement in convergence accuracy of IWOA on the CEC2017 benchmark functions, and the enhanced classification performance of IWOA-CNN on high-dimensional datasets. In the theoretical verification stage, the basic WOA, IWOA, and other representative intelligent optimization algorithms are employed to investigate their ability to locate global optima across multiple benchmark functions. In the application verification stage, five CNN-based optimization strategies are compared within the proposed multimodal classification performance evaluation framework, including CNN optimized with basic PSO (PSO-CNN), CNN optimized with basic GA (GA-CNN), CNN optimized with basic SCA (SCA-CNN), WOA-CNN, and CNN optimized with IWOA (IWOA-CNN).

The convergence behavior of IWOA on the Sphere function is illustrated in Fig.5. The fitness value decreases rapidly from 140 to 20 within 10 iterations, exhibiting an exponential decline that demonstrates efficient global search capability and fast convergence. In comparison, the original WOA achieves only a reduction from 140 to 60 in the initial iterations, indicating slower convergence and a higher likelihood of stagnating in local optima, which requires more iterations to reach comparable accuracy. Through its improved mechanisms, such as dynamic weighting and hybrid strategies, IWOA effectively balances global exploration and local exploitation, ultimately reaching a lower fitness value within 10 iterations. These results verify the superior convergence accuracy and stability of IWOA compared to the standard WOA.

4.1 Benchmark Functions Test

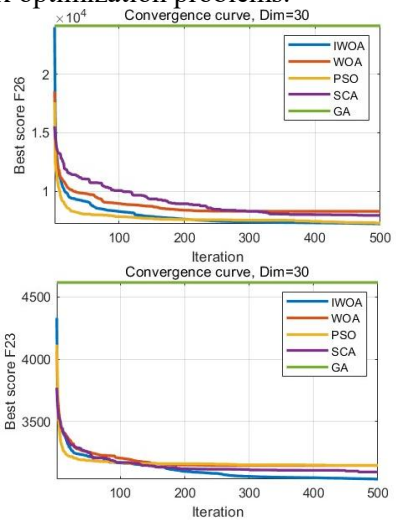
In this test, five intelligent optimization algorithms-PSO, GA, SCA, basic WOA, and IWOA are employed to search for the global optima of eight benchmark functions[37]. To ensure experimental fairness and eliminate confounding factors, all critical operational parameters were kept identical across comparative analyses. The experimental configuration is as follows: function dimension of 30, maximum iterations of 500, population size of 30, and 10 independent runs. Experiments were conducted on a personal computer with an Intel Core@ i9-12900H CPU (2.50 GHz), 16 GB of RAM, running a 64-bit operating system. The programming

environment used is MATLAB R2022b. To comprehensively evaluate algorithm performance, eight benchmark functions of varying complexity were selected from the CEC 2017 test suite.

In this experiment, we compared five different search agent algorithms. By analyzing the performance of each algorithm under certain parameter settings, we can compare their efficiency and accuracy in the search space [38].

To clearly illustrate the generalization capability of these algorithms, convergence curves are employed to characterize the optimization trajectory toward the global optimum. As shown in Figure 6, the trajectories of the five algorithms are compared across the benchmark functions. The IWOA algorithm demonstrates notable advantages, particularly on functions F_{10} and F_{22}. Compared with WOA, PSO, SCA, and GA, IWOA achieves solutions closer to the global optima for these functions, indicating faster convergence and superior iteration outcomes among the five algorithms.

The enhanced performance of IWOA can be attributed to the combined effects of Singer chaotic initialization and the nonlinear convergence factor, which significantly accelerate convergence toward the global optimum. Furthermore, the incorporation of the Gaussian mutation mechanism allows dynamic perturbation of the population, effectively preventing premature convergence to local extrema while maintaining population diversity. Experimental results confirm that IWOA exhibits superior performance in function optimization tasks, and its comprehensive optimization capability demonstrates its effectiveness as a robust method for solving complex optimization problems.



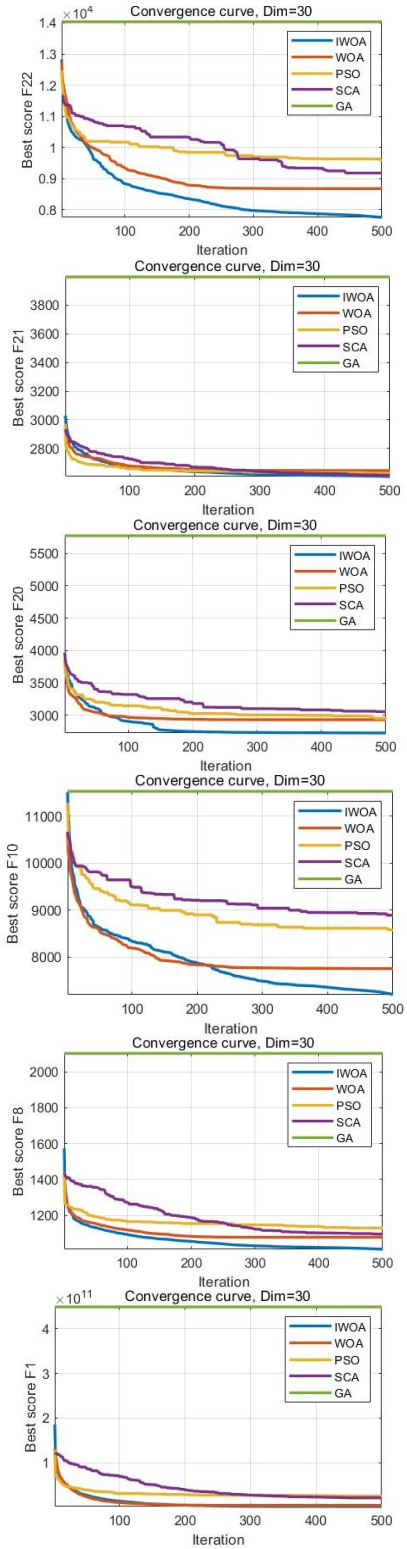


Figure 6. The Convergence Curves of Benchmark Function

4.2 Dataset Classification Experiments

Optimization Algorithm (IWOA) is employed to optimize both the number and dimensions of convolutional kernels, thereby enhancing classification performance. The proposed

IWOA-CNN model is rigorously benchmarked against alternative optimization methods across multiple datasets to evaluate its classification efficacy.

In the experimental design, all datasets were partitioned using a standardized 7:3 split for training and testing subsets. Under identical network configurations, with a population size of 25 and a maximum of 200 iterations, simulation experiments were conducted to obtain comparative classification accuracy metrics, facilitating detailed performance assessments. All datasets were sourced from the UCI Machine Learning Repository [39], with their characteristics summarized in Table 3.

To validate the practical utility of the IWOA algorithm, convergence behavior was analyzed based on accuracy benchmarks established in prior studies [40], with recognition accuracy and loss values showing stable convergence trends. Experimental outcomes were evaluated using two primary metrics: classification accuracy (covering both training and validation subsets) and confusion matrix analysis. Classification accuracy is quantified by the following equation (11):

$$Accuracy = \frac{TP}{TP + FN} \tag{11}$$

To validate the effectiveness of the IWOA-CNN algorithm, a comprehensive performance evaluation was conducted through comparative experiments. Experimental metrics, including True Positive (TP) and False Negative (FN) cases, were employed to quantify detection accuracy and error patterns. A systematic comparison framework was established to assess the advantages of IWOA-CNN by benchmarking it against four established optimization-based CNN approaches: SCA-CNN, GA-CNN, PSO-CNN, and WOA-CNN across multiple datasets. The experimental setup for cross-algorithm comparison is detailed in Table 3, while Table 4 reports quantitative classification accuracy measurements. This rigorous evaluation framework enables a multidimensional assessment of IWOA-CNN's operational efficiency, generalization capability, and performance stability relative to state-of-the-art counterparts.

Table 3. Classification Experiment Dataset

Datasets	Instance	Attributes	Categories
wdbc	569	30	2
vehicle	846	18	4
Cervical Cancer Behavior Risk	72	19	2
Ionosphere	351	34	2
Backup-large	307	35	19

Table 4. Datasrt classification Experimental Results

Datasets		IWOA-CNN	WOA-CNN	SCA-CNN	GA-CNN	PSO-CNN
wdbc	Train	1.0000	1.0000	1.0000	1.0000	1.0000
	Test	0.9883	0.9649	0.9883	0.9707	0.9707
vehicle	Train	0.9966	0.9763	0.9983	0.9847	1.0000
	Test	0.8189	0.7992	0.7716	0.7952	0.7598
CCBR	Train	1.0000	1.0000	1.0000	1.0000	1.0000
	Test	0.9091	0.8636	0.8181	0.8181	0.8636
Ionosphere	Train	0.9919	0.9918	0.9959	0.9918	0.9918
	Test	0.9528	0.9622	0.9622	0.9621	0.9339
Backup-large	Train	0.9907	0.9953	1.0000	0.9906	1.0000
	Test	0.8710	0.8602	0.9139	0.8817	0.8602

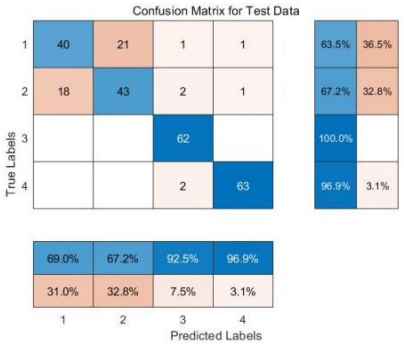
Based on the training and testing accuracy of five methods—IWOA-CNN, WOA-CNN, SCA-CNN, GA-CNN, and PSO-CNN—across multiple datasets, several conclusions can be drawn. IWOA-CNN consistently achieves superior test accuracy, particularly on the WDBC, Vehicle, and Cervical Cancer Behavior Risk datasets, demonstrating high classification performance. For instance, the test accuracy of IWOA-CNN on the Vehicle dataset (0.8189) surpasses that of WOA-CNN (0.7992), GA-CNN (0.7952), PSO-CNN (0.7598), and SCA-CNN (0.7716), highlighting its strong generalization capability in complex classification tasks.

While most methods achieve high training accuracy, a decrease in test accuracy is observed for some approaches, suggesting potential overfitting. IWOA-CNN exhibits excellent robustness on the Cervical Cancer Behavior Risk dataset. On the Ionosphere dataset, WOA-CNN attains slightly higher test accuracy (0.9622) compared to IWOA-CNN (0.9528), indicating marginally better generalization in this specific

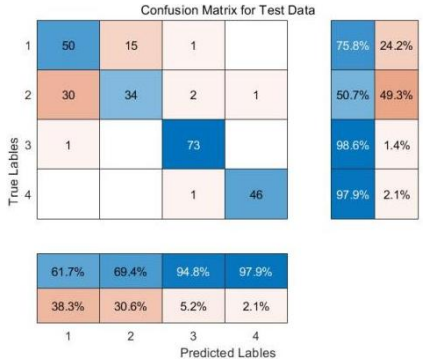
case. Overall, IWOA-CNN achieves optimal or near-optimal test accuracy across multiple datasets and demonstrates stable performance, confirming that its optimization strategy effectively enhances CNN training. In comparison, WOA-CNN, GA-CNN, and PSO-CNN perform well on certain datasets but generally exhibit slightly lower performance relative to IWOA-CNN.

Experimental results demonstrate that the IWOA-CNN model achieves excellent classification performance and generalization capability across most datasets, significantly enhancing CNN accuracy through intelligent hyperparameter optimization. The model exhibits strong robustness in handling complex data distributions, effectively mitigating the challenges posed by noise interference and overlapping inter-class features. The performance on the Vehicle dataset is particularly notable: as shown in the confusion matrix in Figure 7, the model achieves fine-grained classification with high precision, confirming a substantial improvement over the original CNN. Compared to the baseline CNN, IWOA-CNN significantly enhances feature discrimination, demonstrating the effectiveness of its optimization mechanism.

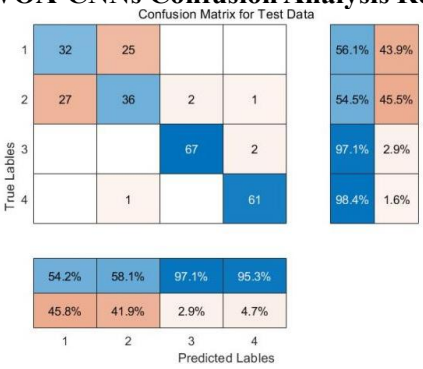
A key direction for future optimization is the integration of attention mechanisms. By dynamically weighting features to emphasize critical discriminative regions—such as adding a spatial attention module after the convolutional layer to suppress background interference, or implementing a channel attention mechanism to recalibrate feature map responses—the model can selectively reduce confusion between closely related categories. This approach is expected to further improve boundary recognition and fine-grained classification performance, enhancing the model's capability in complex visual tasks.



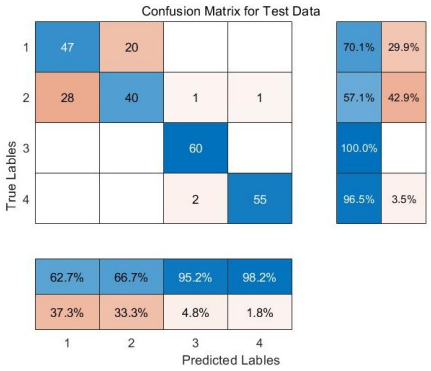
(a) Classification Matrix of the IWOA-Optimized CNN Model



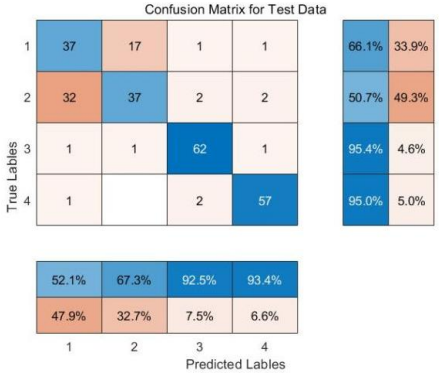
(b) WOA-CNNs Confusion Analysis Results



(c) Performance Matrix for the SCA-Enhanced CNN



(d) GA-CNNs Classification Accuracy Visualization



(e) PSO-driven CNN Models Error Distribution Matrix.

Figure 7. Confusion Matrix Outcomes for Various Network Architectures Evaluated on the Vehicle Dataset (Developed Using MATLAB)

5. Conclusions

This paper presents a high-efficiency deep learning framework based on Convolutional Neural Networks (CNNs) with robust feature extraction capabilities, improving performance in tasks such as image classification and object detection through hierarchical feature abstraction and reduced parameter complexity. To enhance optimisation performance, an improved Whale Optimisation Algorithm (IWOA) is proposed, employing a Singer chaotic map for population diversity, a nonlinear time-varying convergence factor for exploration-exploitation balance, and Gaussian mutation to prevent premature convergence. By integrating IWOA with CNN to form the IWOA-CNN framework, this study achieves optimised hyperparameters, resulting in superior classification accuracy, recall, and generalisation across various datasets, offering an efficient and reliable solution for deep learning optimisation.

It should be noted that the experimental scope was constrained by relatively limited sample sizes, and future investigations with more diverse datasets could yield stronger statistical validity. This work primarily demonstrates the significant potential of Heuristic and Meta heuristic algorithms in improving neural network capacity. Dawning research directions include developing more effective meta-heuristic enhancement strategies and exploring their broader Utilization in varied domains featuring selection regression analysis, predictive modeling, and path optimization tasks. The findings underscore the practical value of integrating evolutionary computation techniques with deep learning architectures.

6. Disclosure Statement

The authors declare that they have no conflicts of interest.

7. Data Availability Statement

Data will be made available on reasonable request.

References

[1] Chakraborty A, Kar A K. Swarm intelligence: A review of algorithms[J]. Nature-inspired Computing and Optimization: Theory and Applications, 2017: 475-494.

[2] Abdollahzadeh B, Khodadadi N, Barshandeh S, et al. Puma Optimizer (PO): a novel metaheuristic optimization algorithm and its application in machine learning[J]. Cluster Computing, 2024, 27(4): 5235-5283.

[3] Wang J, Wang W, Hu X, et al. Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems[J]. Artificial Intelligence Review, 2024, 57(4): 98.

[4] Zhao W G, Wang L Y, Zhang Z X, et al. Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications[J]. Expert Systems with Applications, 2024, 238: 122200.

[5] Passino K M. Biomimicry of bacterial foraging for distributed optimization and control[J]. IEEE Control Systems Magazine, 2002, 22(3): 52-67.

[6] Eusuff M, Lansey K, Pasha F. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization[J]. Engineering Optimization, 2006, 38(2): 129-154.

[7] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.

[8] Tanyildizi E, Cgali T. Whale Optimization Algorithms With Chaotic Mapping[J]. Frat Universitesi Muhendislik Bilimleri Dergisi, 2017, (1): 307-317.

[9] Asghari K, Masdari M, Gharehchopogh F S, et al. A chaotic and hybrid gray wolf-whale algorithm for solving continuous optimization problems[J]. Progress in Artificial Intelligence, 2021, 10(3): 349-374.

[10] Nam C. Convolutional neural network-based prediction of hardness in bulk metallic glasses with small data[J]. Journal of Non-Crystalline Solids, 2025, 654123451-123451.

[11] Hubel D H, Wiesel T N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex[J]. The Journal of Physiology, 1962, 160(1): 106.

[12] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.

[13] Ha I ,Kim H ,Park S , et al.Image retrieval using BIM and features from pretrained VGG network for indoor localization[J]. Building and Environment, 2018, 14023-31.

[14] Yang H, Wenjun K, Jinqiang L. ResNet Combined with Attention Mechanism for

Genomic Deletion Variant Prediction[J]. Automatic Control and Computer Sciences, 2024, 58(3): 252-264.
[15] Yao X, Wang X, Karaca Y, et al. Glomerulus classification via an improved GoogLeNet[J]. IEEE Access, 2020, 8: 176916-176923.
[16] Wang L, Yang Y, Min R, et al. Accelerating deep neural network training with inconsistent stochastic gradient descent[J]. Neural Networks, 2017, 93: 219-229.
[17] Raiaan M A K, Sakib S, Fahad N M, et al. A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks[J]. Decision Analytics Journal, 2024: 100470.
[18] Bergstra J, Bengio Y. Random search for hyper-parameter optimization[J]. The Journal of Machine Learning Research, 2012, 13(1): 281-305.
[19] Tuerxun W, Chang X, Hongyu G, et al. Fault diagnosis of wind turbines based on a support vector machine optimized by the sparrow search algorithm[J]. IEEE Access, 2021, 9: 69307-69315.
[20] Gorkemli B, Kaya E, Karaboga D, et al. A review on the versions of artificial bee colony algorithm for scheduling problems[J]. Journal of Combinatorial Optimization, 2025, 49(4): 1-46.
[21] Suriyan K, Nagarajan R. Particle swarm optimization in biomedical technologies: innovations, challenges, and opportunities[J]. Emerging Technologies for Health Literacy and Medical Practice, 2024: 220-238.
[22] Mirjalili S. Genetic algorithm[J]. Evolutionary Algorithms and Neural Networks: Theory and Applications, 2019: 43-55.
[23] Liu Z, Yeh W C. Simplified swarm optimisation for CNN hyperparameters: a sound classification approach[J]. International Journal of Web and Grid Services, 2024, 20(1): 93-113.
[24] Munsarif M, Sam'an M, Fahrezi A. Convolution neural network hyperparameter optimization using modified particle swarm optimization[J]. Bulletin of Electrical Engineering and Informatics, 2024, 13(2): 1268-1275.
[25] Aghabeigi F, Nazari S, Osati Eraghi N. An efficient facial emotion recognition using convolutional neural network with local sorting binary pattern and whale optimization algorithm[J]. International Journal of Data Science and Analytics, 2024: 1-16.
[26] Seyedali M, Andrew L. The Whale Optimization Algorithm[J]. Advances in Engineering Software, 2016 (1): 95-101.
[27] Luan F, Cai Z, Wu S, et al. Improved Whale Algorithm for Solving the Flexible Job Shop Scheduling Problem[J]. Mathematics, 2019, 7(5): 384-384.
[28] Hiba A, Mohammed A. Software fault prediction using Whale algorithm with genetics algorithm[J]. Software: Practice and Experience, 2020, 51(5): 1121-1146.
[29] Taye M M. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions[J]. Computation, 2023, 11(3): 52.
[30] Wang Z J, Turko R, Shaikh O, et al. CNN explainer: learning convolutional neural networks with interactive visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2020, 27(2): 1396-1406.
[31] Cheng N, Chen Y, Gao W, et al. An improved deep learning model: S-TextBLCNN for traditional Chinese medicine formula classification[J]. Frontiers in Genetics, 2021, 12: 807825.
[32] Ewees A A, El Aziz M A, Hassanien A E. Chaotic multi-verse optimizer-based feature selection[J]. Neural Computing and Applications, 2019, 31: 991-1006.
[33] Wei F , Shi X , Feng Y, et al. Improved Harris hawk algorithm based on multi-strategy synergy mechanism for global optimization [J]. Soft Computing, 2024, 28(21): 1-46.
[34] Qian P, Pu C, Liu L. Ultra-high-precision pneumatic force servo system based on a novel improved particle swarm optimization algorithm integrating Gaussian mutation and fuzzy theory[J]. ISA Transactions, 2024, 152: 453-466.
[35] Zhang J, Yu X, Lei X, et al. A novel deep LeNet-5 convolutional neural network model for image recognition[J]. Computer Science and Information Systems, 2022, 19(3): 1463-1480.
[36] Tulbure A A, Tulbure A A, Dulf E H. A review on modern defect detection models using DCNNs–Deep convolutional neural networks[J]. Journal of Advanced Research, 2022, 35: 33-48.

[37] Wu X, Li S, Jiang X, et al. Information acquisition optimizer: a new efficient algorithm for solving numerical and constrained engineering optimization problems[J]. The Journal of Supercomputing, 2024, 80(18): 25736-25791.

[38] Jamil M, Yang X S. A literature survey of benchmark functions for global optimisation problems[J]. International Journal of Mathematical Modelling and Numerical Optimisation, 2013, 4(2): 150-194.

[39] Tanveer M, Gautam C, Suganthan P N. Comprehensive evaluation of twin SVM based classifiers on UCI datasets[J]. Applied Soft Computing, 2019, 83: 105617.

[40] Wang C X, Shi T T, Han D N. Adaptive dimensional gaussian mutation of PSO-optimized convolutional neural network hyperparameters [J]. Applied Sciences, 2023, 13(7):4254.

