CIT. Journal of Computing and Information Technology, Vol. 33, No. 3, September 2025, 183–200
doi: 10.20532/cit.2025.1006020

Digit Recognition Method Based on Discrete Hopfield Neural Network Optimized by Artificial Fish Swarm Algorithm

Xin Su, Daiheng Xie and Yue Zhao

School of Business, Guilin University of Electronic Technology, Guilin, Guangxi, China

To solve the problems that the weights and thresholds of discrete Hopfield neural networks are easy to fall into local optima and have insufficient anti-noise ability in digit recognition, a digit recognition method based on discrete Hopfield neural network is proposed, which is optimized by fish swarm algorithm and called AFSA-HOP integration method. The parameters of the discrete Hopfield neural network are optimized by using AFSA's powerful global search ability, and the recognition accuracy of the Hopfield neural network is taken as the fitness function. This allows the Hopfield neural network to maintain a high associative success rate even under high noise-to-signal ratios. Computer simulation experiments show that while the recognition performance of the traditional Hopfield neural network significantly deteriorates when the noise intensity is 0.2, the AFSA-HOP method maintains a high recognition accuracy even at noise intensities of 0.4 and 0.5, demonstrating superior digital recognition performance. This method provides a robust new approach for digital recognition and could be further extended in future applications by integrating other optimization algorithms.

ACM CCS (2012) Classification: Computing methodologies → Machine learning → Machine learning approaches → Neural networks

Keywords: digit recognition, artificial fish swarm algorithm, Hopfield neural network

1. Introduction

In daily life, we often encounter recognition challenges involving noisy characters, such as the vehicle license plate character recognition problem in intelligent transportation systems [1][2]. Due to factors like exposure to wind, sunlight, sudden lighting changes, and differences in plate color, the accurate recognition of license plate characters becomes difficult [3][4]. In many existing models, the presence of additional noise in the input leads to a decrease in the recognition system's accuracy, often resulting in incorrect character predictions. Therefore, extracting complete character information from noisy information is a key point in the field of character recognition [5]. As an important component of the character recognition research, digit recognition holds significant application value not only in traffic management but also in fields such as postal services and commercial bill management [6][7]. Currently, numerous methods have been proposed in the character recognition domain, such as BP neural network recognition [4][8], classifier fusion recognition [9], and fuzzy recognition [10]. However, traditional digit recognition methods often struggle to accurately recognize digits when confronted with interference. The discrete Hopfield neural network mimics the memory mechanism of biological neural networks through a structure and learning method distinct from that of hierarchical neural networks [11], and has achieved relatively satisfactory results in digit recognition applications [12].

Although discrete Hopfield neural networks have been widely applied in the field of digit recognition, traditional Hopfield neural networks often struggle to reach a true steady state (with some pseudo-stable points present), and the network weights and thresholds are prone to falling into local optima during digit recognition. This leads to suboptimal recognition results, especially under high noise-to-signal ratios [13]. It is worth noting that this limitation does not exist in isolation, but is closely related to the dynamic characteristics of the network structure itself, that is, the design of the energy function makes the system possible to contain multiple local energy minima. In order to break through this bottleneck, scholars have explored a variety of optimization paths that combine meta heuristic algorithm with Hopfield network in recent years. Related research has shown that combining metaheuristic algorithms with discrete Hopfield neural networks can further enhance the associative memory capability of the network, significantly improving the application effectiveness and digit recognition accuracy of Hopfield network [13-15]. Common meta-heuristic algorithms, such as Genetic Algorithm (GA), Simulated Annealing Algorithm (SA), Tabu Search Algorithm (TS), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) [16][17], are distinguished by their global search capabilities, which set them apart from conventional algorithms. However, the existing heuristic optimization methods still face the challenge of algorithm adaptability. Among them, although genetic algorithm has coding flexibility, its selection pressure is easy to lead to premature loss of population diversity and lead to "premature convergence" [18]. The temperature control parameters of simulated annealing algorithm need to be carefully adjusted, and the search efficiency is restricted by the cooling scheme [15][19]. The particle position updating mechanism of particle swarm optimization in discrete space has natural limitations [20]. Ant colony algorithm is easy to fall into search stagnation because of pheromone update mechanism [21]. These defects weaken the ability of the algorithm to explore the weight space of Hopfield network to varying degrees.

As a result, researchers have been seeking more ideal optimization algorithms, and the Artificial Fish Swarm Algorithm (AFSA) has gradually emerged as a new highlight in the field of swarm intelligence optimization algorithms [22]. The advantages of this algorithm are obvious, and its design characteristics are highly consistent with the optimization requirements of discrete Hopfield networks. First, the bionic behavior mechanism gives the algorithm a strong ability to jump out of local minima and effectively avoid the interference of pseudo-stable points; Secondly, the parameter setting of the algorithm is relatively loose, and the sensitivity to the initial value is low, which reduces the complexity of parameter adjustment; Furthermore, the characteristics of parallel search based on multi-thread significantly improve the optimization efficiency, and it is especially suitable for dealing with discrete optimization problems of high weight matrix [22][23]. More importantly, the discrete iterative rules of artificial fish swarm algorithm can directly act on the weight coding space of Hopfield neural network, thus avoiding the quantization error problem of continuous algorithm. Therefore, this paper proposes a discrete Hopfield neural network digit recognition method optimized by the Artificial Fish Swarm Algorithm. By leveraging the strengths and powerful global search ability of AFSA, the associative memory steady state of the Hopfield network is optimized, enabling the recognition patterns to escape pseudo-stable points, thereby maintaining a higher associative success rate under high noise-to-signal ratios. Simulation experiments demonstrate the effectiveness of this method.

2. Algorithm Principles

2.1. Discrete Hopfield Neural Network

The Discrete Hopfield Neural Network (DHNN) is an important feedback-type network structure in the field of artificial neural networks. This network model has demonstrated significant application value in various fields such as optimization computation, pattern recognition, and intelligent control. As a typical recurrent neural network, DHNN effectively simulates the dynamic behavior of complex systems by introducing the concept of an energy function.

From the perspective of network topology, DHNN has three key characteristics: First, the network adopts a fully connected architecture, where each processing unit is connected to every other unit. Second, the connections in the network have bidirectional symmetry. Finally, there are no self-feedback connections between the processing units. These unique structural features enable the network to exhibit distinct dynamic properties and maintain a stable operating state under specific conditions.

2.1.1. Network Structure

DHNN is a single-layer, binary output feedback network, in which every neuron has the same function. The outputs of the neurons are connected to the inputs of other neurons, with no self-feedback at any node. As the input of the network, Layer 0 has no computing function, the 1st layer consists of neurons that perform the summation of the product of the input information and weight coefficients, and the resulting sum is processed through a nonlinear function, f, to produce the output information.

The input is processed by the threshold function f, which is the sigmoid function, primarily used to reduce the input to two extreme values. If the output information of the neuron is greater than the threshold θ, the neuron's output is set to 1; if it is less than the threshold θ, the neuron's output is set to −1. The threshold function is given by:

$$f(t) = (1 + e^{-t})^{-1} \tag{1}$$

2.1.2. Network Operation Mode

The Hopfield network operates based on its dynamics, where the process is the evolution of the neuron states. That is, starting from the initial state, the evolution follows the direction of "energy" (Lyapunov function) decreasing until it reaches a stable state. The stable state is considered the output of the network. In this paper, the serial (asynchronous) operation mode is primarily selected. In this mode, at any given time t, only a specific neuron i changes, while the states of the other neurons remain the same. The steps of operation are as follows.

1. Initialize the network.
2. Randomly select a neuron i from the network.
3. Compute the input $ui(t)$ of the neuron i.
4. Compute the output $vi(t + 1)$ of neuron i, while the outputs of the other neurons in the network remain unchanged.
5. Check if the network has reached a stable state. If it has reached a stable state or meets the given conditions, the process ends. Otherwise, go back to step 2 and continue.

The stable state of the network is defined as follows: if the network's state no longer changes after a certain time, the network is considered to be in a stable state.

$$v(t + \Delta t) = v(t) \quad \Delta t > 0 \tag{2}$$

2.1.3. Network Stability

From the structure of the DHNN, it is evident that it is a multi-input, thresholded binary nonlinear dynamic system. In dynamic systems, a stable equilibrium state refers to that the system's energy function continually decreasing during its motion, eventually reaching a minimum value. The sufficient condition for the stability of the Hopfield network is:

$$\begin{cases} w_{ij} = 0, & i = j \\ w_{ij} = w_{ji}, & i \neq j \end{cases} \tag{3}$$

When the weight coefficient matrix W is a symmetric matrix and the off-diagonal elements are zero, the network is stable. In this paper, the design method for the weight coefficient matrix utilizes the outer product method.

That is, for a given sample vector to be memorized $\{t^1, t^2, ..., t^N\}$, if the state of t^k is +1 or −1, the learning of its connection weights can be performed using the "outer product rule," i.e.,

$$W = \sum_{k=1}^{N} [t^k (t^k)^T - I] \tag{4}$$

The steps for designing a discrete Hopfield network using the outer product method are as follows.

Step 1. Calculate the weight coefficient matrix according to the formula above, based on the sample vectors to be memorized $\{t^1, t^2, ..., t^N\}$.

Step 2. Let the test sample $p_i(i = 1, 2, ..., n)$ be the initial output values of the network, , and set the number of iterations.

Step 3. The formula for the iterative calculation is

$$y_i(k+1) = f(\sum_{j=1}^{N} w_{ij} y_j) \tag{5}$$

Step 4. The iteration stops when the maximum number of iterations is reached or when the neuron output states remain unchanged. Otherwise, continue iterating in Step 3.

2.2. Artificial Fish Swarm Algorithm

2.2.1. Typical Behavioral Patterns of Artificial Fish Schools

The Artificial Fish Swarm Algorithm (AFSA) is a bio-inspired optimization algorithm based on swarm intelligence theory, first proposed by Dr. Xiao Lei Li in 2002 while studying swarm intelligence behavior. This algorithm simulates the collective foraging behavior of fish in nature, establishing a complete intelligent optimization framework. Compared to traditional optimization algorithms, the Artificial Fish Swarm Algorithm offers stronger robustness and global search capabilities, and is especially suitable for solving complex nonlinear optimization problems. Its unique self-organizing and self-adaptive characteristics give it significant advantages in fields such as function optimization, combinatorial optimization, and parameter identification.

The design inspiration for the Artificial Fish Swarm Algorithm stems from biologists' long-term observation and study of fish group behaviors. In natural aquatic environments, fish schools tend to gather in areas with the most abundant nutrients, revealing an important characteristic of swarm intelligence. The algorithm simulates three typical behavioral patterns of fish schools by establishing an individual model of artificial fish:

- Foraging Behavior

Foraging behavior imitates the way fish schools search for food. It is based on random searching, where the fish randomly selects a direction with more food and moves towards it. Let the position of the i-th artificial fish be X_i, and a new position X_j is randomly selected within the visual range of the i-th artificial fish:

$$X_j = X_i + Visual \times rand \tag{6}$$

Here, *rand* is a random number within the range [0, 1]. In the case of a maximization problem, if $Y_i < Y_j$ (similarly for a minimization problem), the fish will move one step in the direction of Y_j:

$$XNext = X_i + rand \times Step \times \frac{x_j - x_i}{D_{ij}} \tag{7}$$

D_{ij} represents the distance between the i-th and j-th artificial fish. If the above condition is not met, a new X_j is randomly selected, and the condition is checked again. If after try-number attempts the condition still does not hold, the fish will move one step in a random direction.

$$XNext = X_i + rand \times Visual \tag{8}$$

- Schooling Behavior

Let the position of the i-th artificial fish be X_i, and within its visual range, there are NF other artificial fish. The center of these neighbors is represented by X_c. If $Y_c / NF > \delta Y_i$ (where δ is the crowding coefficient), it indicates that the center position has enough food and is not too crowded. In this case, the i-th artificial fish will move one step towards the center position X_c.

$$XNext = X_i + rand \times Step \times \frac{x_c - x_i}{\|x_c - x_i\|} \tag{9}$$

- Following Behavior

Following behavior involves moving one step towards a neighbor that has the highest food concentration among all its neighbors. Let the position of the i-th artificial fish be X_i, and the position of the neighbor be X_m, if $Y_c / NF > \delta Y_i$ (indicating that X_m has enough food around it and is not too crowded), then the i-th artificial fish will move one step towards X_m.

$$XNext = X_i + rand \times Step \times \frac{x_m - x_i}{D_{ij}} \tag{10}$$

- Billboard

The billboard is used to record the best state of the artificial fish. After an artificial fish moves, its current state is compared with the state recorded on the billboard. If the current state is better than the one on the billboard, the billboard is updated with the current state. If it is not better, the state on the billboard remains unchanged. In this way, the state on the billboard will always reflect the best state of the fish swarm.

2.2.2. Parameter Configuration of Artificial Fish Swarm Algorithm

To ensure the repeatability and operability of the algorithm, the specific parameter configuration and description of the artificial fish swarm algorithm are shown in Table 1.

Table 1. Explanation of parameter configuration for artificial fish swarm algorithm.

Parameter Names	**Definition Description**	**Value Range (10×10 Search Space)**	**Example Value**
Visual Range (Visual)	The maximum distance that artificial fish can perceive from other fish when foraging or gathering in groups.	10%–20% of the search space range (1 to 2)	1.2
Step size (Step)	The minimum distance unit for artificial fish during movement.	1%–5% of the search space range (0.1 to 0.5)	0.1
Number of Fish (N)	The number of fish in artificial fish schools	30 to 50	50
Crowd Factor (δ)	Used to control the degree of crowding when schools of fish gather, avoiding excessive aggregation that leads to local optima.	0.1 to 0.5	0.2
Maximum Iterations (Tmax)	The maximum number of iterations for the algorithm to run.	100 to 300	100
Behavior Weights (w_1, w_2, w_3)	Used to balance the weights of Foraging behavior, Schooling behavior, Following behavior.	Normalize processing to satisfy $w_1 + w_2 + w_3 = 1$	$w_1 = 0.3$, $w_2 = 0.3$, $w_3 = 0.4$
Random Factor (Rand)	Used to introduce randomness and avoid the algorithm getting stuck in local optima.	Uniform distribution of random numbers	R~U(0,1)

3. AFSA-HOP Integrated Method Design

This paper proposes a Discrete Hopfield Neural Network integration method optimized by the Artificial Fish Swarm Algorithm (AFSA-HOP integration method). This method significantly improves the accuracy and robustness of digital recognition in high-noise environments by innovatively combining the global optimization capability of AFSA with the associative memory characteristics of the Hopfield Neural Network. In response to the issues of traditional Hopfield Neural Networks, which are prone to falling into local optima and are sensitive to noise interference during digital recognition, the AFSA-HOP method fully leverages the three intelligent behavior mechanisms of the Artificial Fish Swarm Algorithm-foraging, clustering, and chasing-to perform global optimization search on the weight matrix and threshold parameters of the Hopfield Neural Network. The recognition accuracy of the network is used as the fitness function. This optimization strategy not only effectively avoids the network from getting stuck in local optima, but also ensures that the network maintains stable recognition performance under high noise levels, overcoming the limitation of traditional methods, where recognition accuracy decreases when noise intensity exceeds 0.2. The implementation steps of the AFSA-HOP integration method are as follows:

Step 1. Data Preprocessing

Collect a dataset containing multiple digit samples and convert each digit sample into a binary matrix form. For example, for the digits 0–9, create corresponding 10×10 (or other appropriate size) matrices to represent each digit. The areas containing the digit are represented by 1, and the blank areas are represented by −1. Let the set of digit samples be $S = \{s_1, s_2, ..., s_n\}$, where s_i represents the i-th digit sample, and its corresponding binary digit matrix is M_i.

Add varying levels of noise (random noise) to the digit sample matrices to simulate the interference encountered in practical applications. The noise addition can be implemented using the following formula: $M_i' = M_i + N_i$ where M_i' is the matrix with added noise, and N_i is the noise matrix, with its elements following a certain probability distribution (e.g., normal distribution).

Step 2. Hopfield Neural Network Initialization

Based on the characteristics of the digit samples, determine the number of neurons, connection patterns, and other parameters for the Hopfield neural network. For example, if the input is a 10×10 digit matrix, the network must contain at least 100 neurons to represent a single digit image. Let the number of neurons be N, the connection weight matrix be $W \in \mathbb{R}^{N\times N}$, and the threshold vector be $\theta \in \mathbb{R}^N$.

Randomly initialize the connection weights and thresholds between the neurons, ensuring that the weight matrix satisfies conditions such as symmetry to guarantee network stability. Specifically, $W_{ij} = W_{ji}$, and $W_{ii} = 0 (i, j = 1, 2, ..., N)$.

Step 3. Artificial Fish Swarm Algorithm Optimization

- Define the Fitness Function

The recognition accuracy of the Hopfield neural network on noisy digit samples is used as the fitness function for the AFSA. Let the training set consist of m noisy digit samples. For the k-th weight and threshold combination (i.e., the k-th fish), its fitness function f_k can be defined as: $f_k = \frac{1}{m}\sum_{i=1}^{m} I(y_i^{(k)} == t_i)$ where, $y_i^{(k)}$ is the recognition result of the k-th fish (i.e., the network corresponding to the k-th combination of weights and thresholds) for the i-th sample, is the true label of the i-th sample, and I is the indicator function, which equals 1 if the condition inside the parentheses is true, and 0 otherwise. This means that the higher the fitness value, the better the network's recognition performance for that digit.

- Artificial Fish Swarm Algorithm Iterative Optimization

Foraging Behavior: For each fish in the swarm (i.e., each weight and threshold combination), a random search for a new position (new weight and threshold combination) is conducted within its neighborhood, and the fitness of the new position is computed. If the fitness of the new position is better than that of the current position, the fish moves to the new position; otherwise, it continues searching within the neighborhood. Let the current position of the k-th fish be $\omega_k = (\omega_{k1}, \omega_{k2}, ..., \omega_{kN})$ and $\theta_k = (\theta_{k1}, \theta_{k2}, ..., \theta_{kN})$, and a new position in its neighborhood be $\omega'_k = (\omega'_{k1}, \omega'_{k2}, ..., \omega'_{kN})$ and $\theta'_k = (\theta'_{k1}, \theta'_{k2}, ..., \theta'_{kN})$. If $f(\omega'_k, \theta'_k) > f(\omega_k, \theta_k)$, then update $\omega_k = \omega'_k$ and $\theta_k = \theta'_k$.

Foraging behavior: For each fish in the swarm (i.e., each combination of weights and thresholds), a new position (new combination of weights and thresholds) is randomly searched within its neighborhood, and the fitness of the new position is evaluated. If the fitness of the new position is better than the current position, the fish moves to the new position; otherwise, it continues to search within the neighborhood. Let the current position of the k-th fish be denoted as $\omega_k = (\omega_{k1}, \omega_{k2}, ..., \omega_{kN})$ and $\theta_k = (\theta_{k1}, \theta_{k2}, ..., \theta_{kN})$, and a new position within its neighborhood as $\omega'_k = (\omega'_{k1}, \omega'_{k2}, ..., \omega'_{kN})$ and $\theta'_k = (\theta'_{k1}, \theta'_{k2}, ..., \theta'_{kN})$. If $f(\omega'_k, \theta'_k) > f(\omega_k, \theta_k)$, then update $\omega_k = \omega'_k$ and $\theta_k = \theta'_k$.

Swarming behavior: Each fish moves a certain distance toward the center of the swarm to facilitate aggregation. The center of the swarm is determined based on the positions of all the fish. Let the center of the swarm be denoted as $\overline{\omega}$ and $\overline{\theta}$.The update formula for the k-th fish is given by: $\omega_k = \omega_k + r_1(\overline{\omega} - \omega_k)$, $\theta_k = \theta_k + r_1(\overline{\theta} - \theta_k)$, where r_1 is a random number within the range [0, 1].

Following behavior: Each fish moves a certain distance toward the position of the fish in the swarm with the highest fitness (i.e., the highest recognition accuracy), in order to learn from the optimal individual's behavior. Let the position of the fish with the highest fitness be denoted as ω_{best} and θ_{best}. The update formula for the k-th fish is given by: $\omega_k = \omega_k + r_2(\omega_{best} - \omega_k)$, $\theta_k = \theta_k + r_2(\theta_{best} - \theta_k)$, where r_2 is a random number within the range [0, 1].

Based on the results of the above three behaviors, the position (weights and thresholds) of each fish is updated, thereby enabling the swarm to search and optimize within the weight space.

- Convergence Condition

Set the convergence conditions. If the maximum number of iterations is reached, the fitness function value will no longer change significantly. Once the convergence condition is met, the iteration of the swarm algorithm is stopped, and the optimized weight values ω^* and θ^* are obtained.

Step 4. Hopfield Neural Network Training and Testing

The optimized weights are applied to the Hopfield neural network, using digit samples with added noise as the training set. During the training process, the network gradually learns the features and patterns of the digits by continuously adjusting the states of the neurons. Let the training set consist of p samples. For the j-th sample x_j, the output of the network is y_i. The neurons' states are adjusted by minimizing the loss function $L = \sum_{j=1}^{p} l(y_j, t_j)$, where l represents the loss function, and t_j is the true label of the sample.

Using digit samples with different noise intensities as the test set, these samples are input into the trained Hopfield neural network, and the network's output is observed. If the network can correctly recognize the input samples as the corresponding digits, it indicates that the network possesses good digit recognition ability. Let the test set consist of q samples. For the i-th sample x'_i, the network's output is y'_i. If $y'_i == t'_i$ (where t'_i is the true label of the sample), the recognition is correct. By calculating the ratio of correctly recognized samples to the total number of test samples, the network's recognition accuracy can be obtained.

Through the above steps, the AFSA-HOP integration method establishes a relatively complete digital recognition optimization framework. This method innovatively combines the optimization mechanism of AFSA with the memory characteristics of Hopfield Neural Network, proposing a new solution approach at the theoretical level. In the specific implementation process, the method constructs a multi-dimensional optimization search space for the network weights and threshold values through the three behavioral mechanisms of the AFSA (foraging, clustering, and chasing). In terms of algorithm design, recognition accuracy is used as the fitness function, directly linking the optimization objective with the actual recognition performance. In terms of the implementation process, the method first completes the preprocessing of the digital samples and noise simulation, then initializes the network structure, followed by parameter optimization search using the AFSA, and finally applies the optimization results to network training. This systematic optimization strategy provides a complete technical path for improving the accuracy and robustness of digital recognition.

4. Simulation Experiments

4.1. Digital Recognition Simulation Experiment Based on Traditional Hopfield Neural Network

4.1.1. Sample Construction and Testing

In digital recognition research, the Discrete Hopfield Neural Network has become an important research subject due to its unique associative memory capability. This study first constructs a standardized digital sample library containing 10 categories of digits, from 0 to 9, with each digit represented by a 10×10 pixel matrix. In the sample preprocessing phase, image normalization is applied to ensure uniform size, and an adaptive thresholding algorithm is used to achieve precise binarization. The digit stroke regions are set to 1, and the background is set to −1, forming a standardized network input format. By constructing a dataset to design a digital dot matrix, the dot matrix designs for number 4 and number 5 are shown in Figure 1 and Figure 2.

The dot matrix of number 4 is:

array_four = [−1 1 1 −1 −1 −1 −1 1 1 −1; −1 1 1 −1 −1 −1 −1 1 1 −1; −1 1 1 −1 −1 −1 −1 1 1 −1; −1 1 1 −1 −1 −1 −1 1 1 −1; −1 1 1 −1 −1 −1 −1 1 1 −1;−1 1 1 1 1 1 1 1 1 −1; −1 1 1 1 1 1 1 1 1 −1;−1 −1 −1 −1 −1 −1 −1 1 1 −1; −1 −1 −1 −1 −1 −1 −1 1 1 −1; −1 −1 −1 −1 −1 −1 −1 1 1 −1].

The dot matrix of number 5 is:

array_five = [−1 1 1 1 1 1 1 1 1 −1; −1 1 1 1 1 1 1 1 1 −1; −1 1 1 −1 −1 −1 −1 −1 −1 −1; −1 1 1 −1 −1 −1 −1 −1 −1 −1; −1 1 1 1 1 1 1 1 1 −1; −1 1 1 1 1 1 1 1 1 −1; −1 −1 −1 −1 −1 −1 −1 1 1 −1; −1 1 1 1 1 1 1 1 1 −1; −1 1 1 1 1 1 1 1 1 −1].

Network training employs the classical outer-product method to calculate the weight matrix, strictly ensuring the symmetry and stability of the network. During the training process, each digit sample is encoded as a stable state of the network, allowing the network to accurately memorize these digit patterns. To comprehensively evaluate the network's performance, the study designs an extended test set containing digits with different fonts and slight deformations to test the network's generalization ability. This multi-level training scheme ensures both

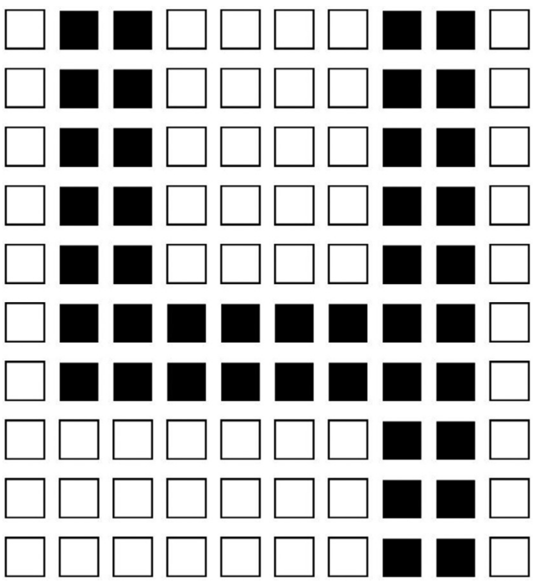

Figure 1. Dot Matrix of Number 4.

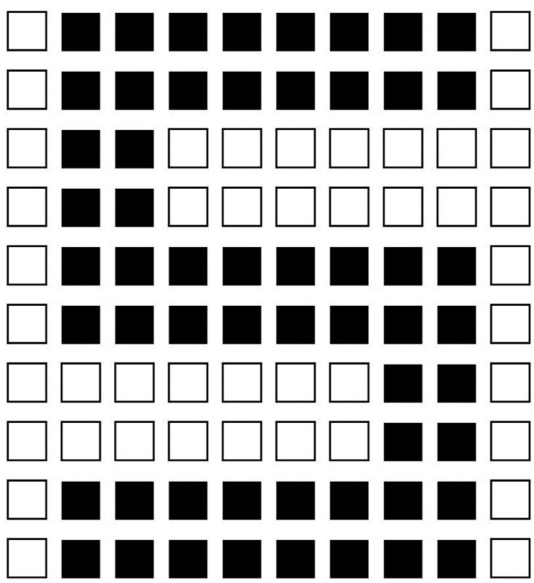

Figure 2. Dot Matrix of Number 5.

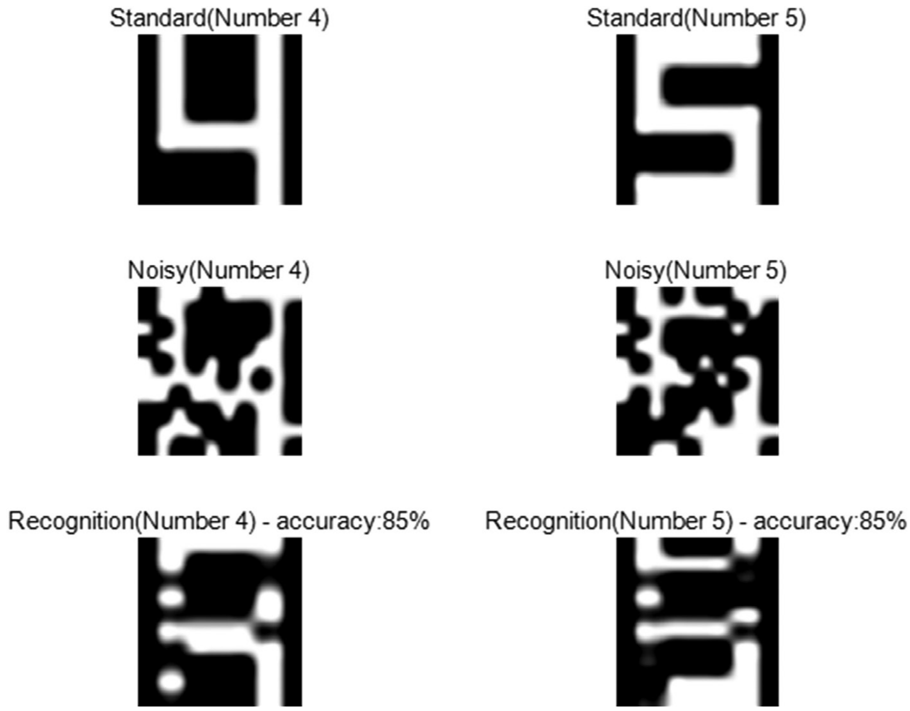

Figure 3. Digit Recognition Results at a Noise Intensity of 0.2 (Traditional Hopfield Neural Network).

the network's ability to recognize standard digits and its adaptability to digit variations.

In the robustness testing phase, a rigorous noise simulation scheme was designed. A probabilistic pixel-flipping method is used to set five noise gradients ranging from 0.1 to 0.5, with 100 test samples generated for each gradient. This progressive noise simulation realistically reflects various interference situations in practical applications, including device acquisition noise, transmission interference, lighting changes, and other factors. During testing, not only the final recognition results are recorded, but also the network's dynamic convergence process is closely monitored, including convergence steps, stable states, and other key indicators. Through this comprehensive testing approach, the performance characteristics of the network under different noise conditions can be thoroughly analyzed, providing reliable data for algorithm improvement. All test data are collected through a professional simulation platform and displayed using various visualization techniques to ensure the accuracy and interpretability of the research results.

4.1.2. Analysis of Experimental Results

Through statistical analysis of extensive experimental data, it was found that the network's recognition performance was optimal when the noise intensity was 0.1 (i.e., 10% of the digital dot matrix position values were altered). This indicates that under slight noise interference, the network is able to effectively utilize the learned digital features to accurately recognize and classify digits. However, as the noise intensity increases, the network's recognition performance declines significantly, as stronger noise disrupts the original features of the digital image, making it difficult for the network to accurately extract and match digit patterns.

Taking the digits 4 and 5 as examples, the recognition results at a noise intensity of 0.2 are shown in Figure 3. From Figure 3, it is evident that the Hopfield neural network struggles to recognize the digits accurately under these conditions (the accuracy is 85%). A closer examination of the 10×10 matrix structure reveals that there are a total of 200 possible pattern combinations. However, in the digit recognition task, the network is expected to accurately recall only 10 specific patterns (corresponding to the steady states of digits 0–9). This means the network must search through a complex pattern space to identify the pattern that best matches the training samples. As noise intensity increases, the difficulty of this task grows significantly, leading to a marked decrease in the network's recognition capability.

4.2. Digital Recognition Simulation Experiment Based on the AFSA-HOP Integrated Method

In the traditional Hopfield neural network digital recognition simulation experiment, when the noise intensity is greater than or equal to 0.2, the network's ability to accurately recognize digits decreases significantly. This phenomenon indicates that the traditional Hopfield neural network experiences poor recognition performance when subjected to noise interference of a certain intensity. To address this issue, this paper proposes the AFSA-HOP integrated method, which aims to achieve accurate digit recognition through associative memory optimization. The core of this method lies in using the AFSA to find the optimal individual. Specifically, it involves determining the optimal weights and thresholds for the network, thereby optimizing the performance of the Hopfield neural network and enhancing its noise resistance and recognition accuracy.

4.2.1. Convergence Analysis of the Artificial Fish Swarm Algorithm

To verify the effectiveness of the AFSA-HOP integrated method, this experiment uses MATLAB for simulation. Convergence analysis is conducted based on the best fitness iteration curves of the Artificial Fish Swarm Algorithm under different noise intensities.

Noise Intensity of 0.3: The maximum number of iterations is set to 100, and the obtained optimal fitness iteration curve is as shown in Figure 4 (taking number 4 and 5 as example). As can be seen from the figure, with the increase of iterations, the Artificial Fish Swarm Algorithm gradually converges, with the best fitness continuously improving. This indicates that the algorithm is continuously searching for a better combination of weights and thresholds, thereby enhancing the network's recognition performance.

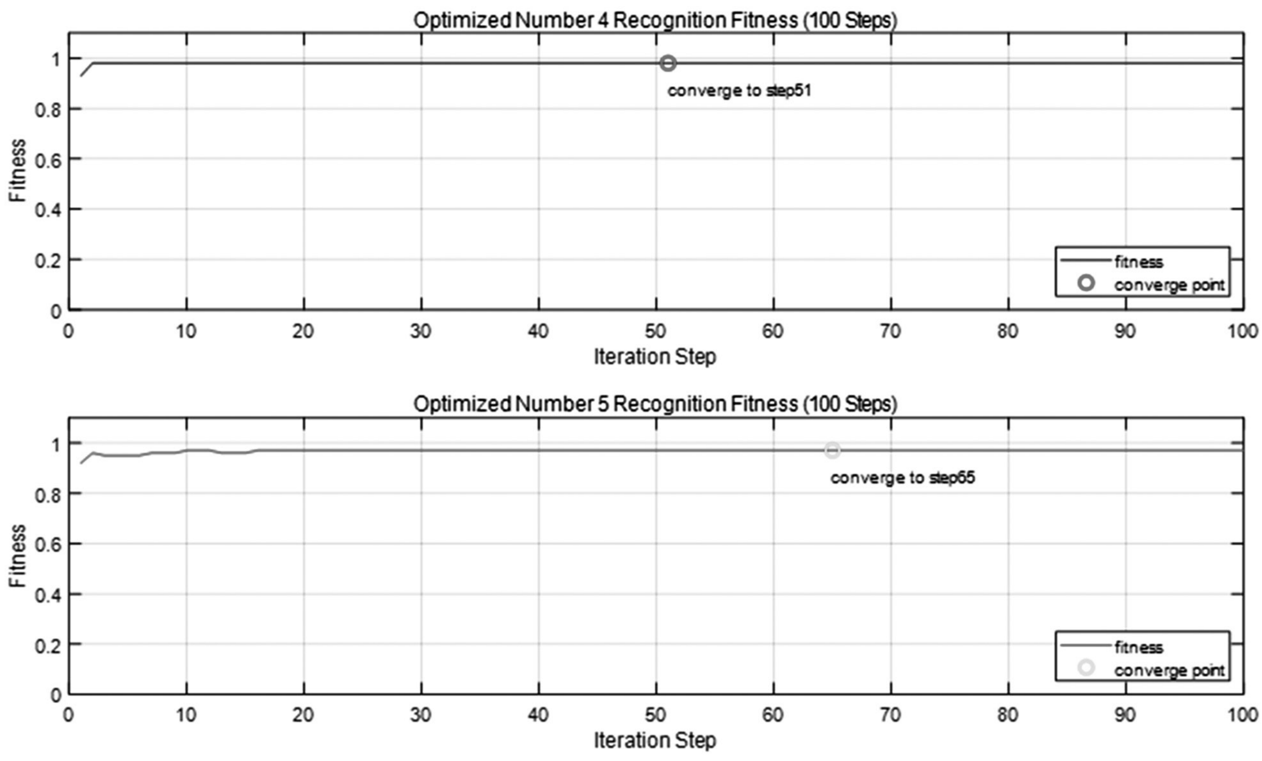

Figure 4. Optimal Fitness Iteration Curve at Noise Intensity of 0.3.

Noise Intensity of 0.4 and 0.5: The maximum number of iterations is set to 100, and the corresponding best fitness iteration curve is shown in Figure 5 and Figure 6. Similarly, the algorithm shows a gradual convergence trend during the iterations, further demonstrating that the Artificial Fish Swarm Algorithm is capable of effectively searching for the optimal solution even in a complex noise environment, providing strong support for accurate digit recognition.

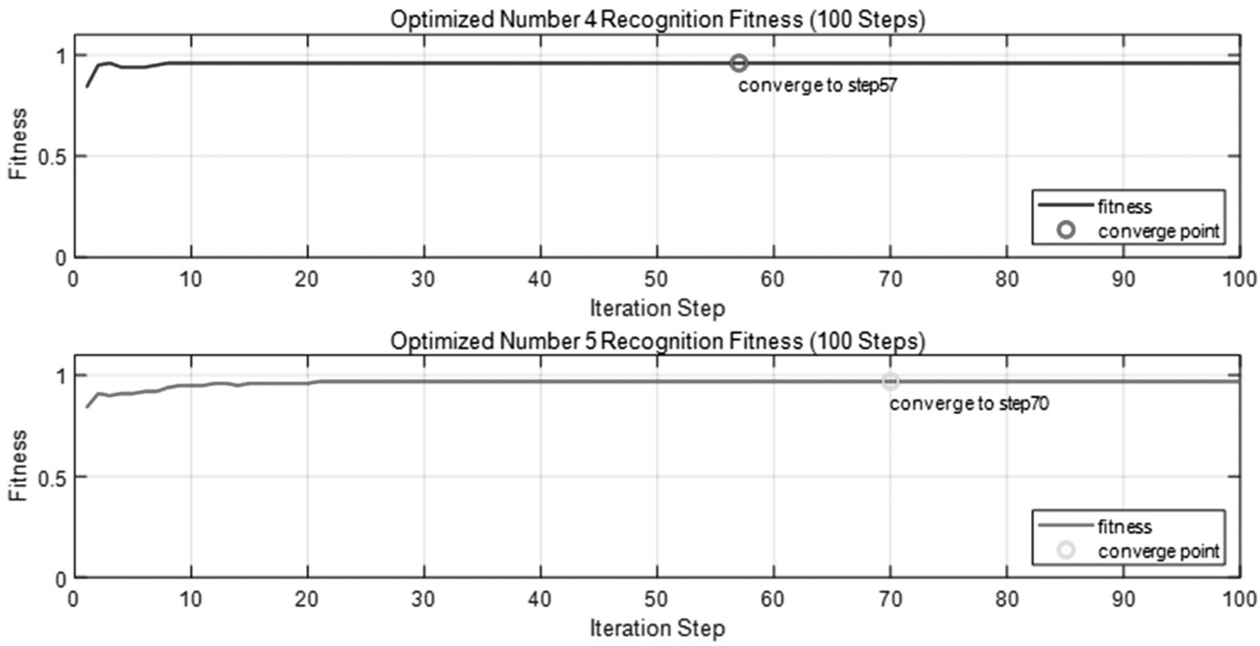

Figure 5. Optimal Fitness Iteration Curve at Noise Intensity of 0.4.

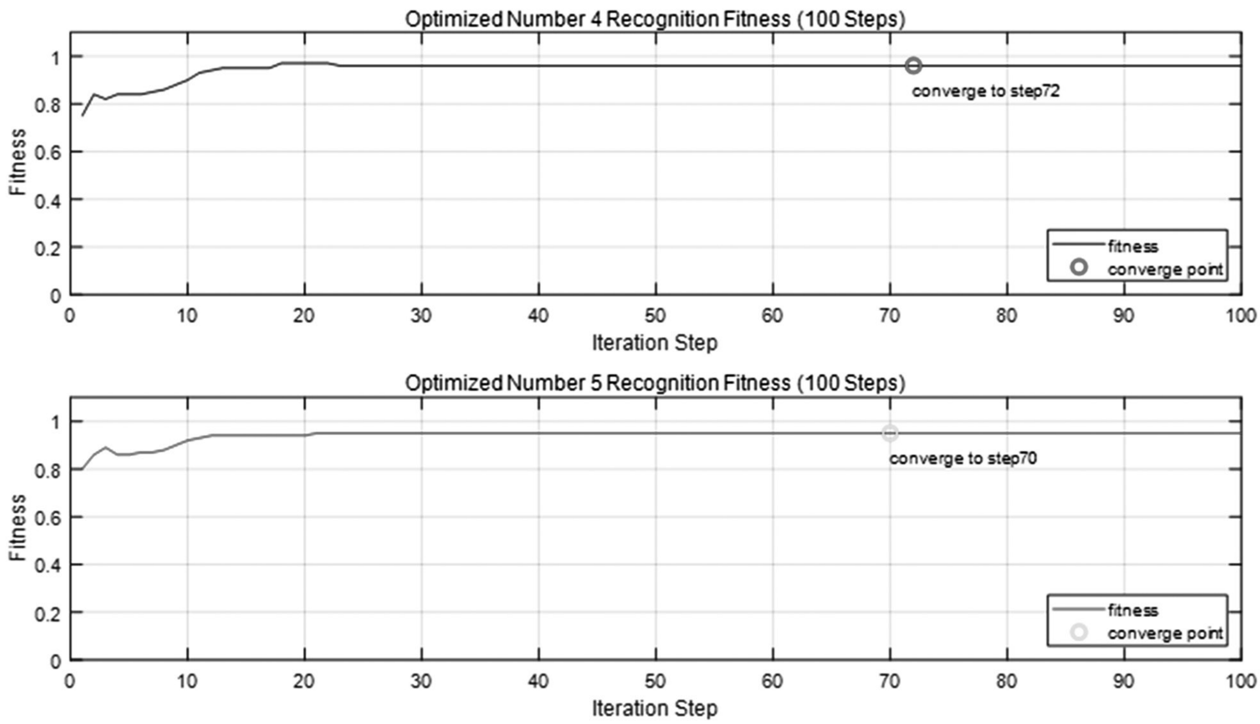

Figure 6. Optimal Fitness Iteration Curve at Noise Intensity of 0.5.

4.2.2. Analysis of Experimental Results

The optimized weight and threshold combinations are applied to the discrete Hopfield neural network. Through training and testing with the Hopfield neural network, the output results of the network are observed. The simulation experiments show that the Hopfield neural network optimized by the Artificial Fish Swarm Algorithm (AFSA-HOP integrated method) not only correctly recognizes digits at a noise intensity of 0.3 but also performs well in digit recognition at a noise intensity of 0.4 and 0.5. Taking digits 4 and 5 as examples, the digit recognition results at noise intensities of 0.3, 0.4 and 0.5 are shown in Figures 7, Figures 8 and Figures 9,with accuracy rates of 97.5%, 96.5%, and 96%, respectively.

The systematic experimental results clearly demonstrate that the AFSA-HOP integration method exhibits multiple performance advantages in digital recognition tasks. Compared to the traditional Hopfield Neural Network, this innovative method shows greater robustness and stability when dealing with noise interference of varying intensities. Specifically, in a test environment with a noise intensity of 0.2, the recognition performance of the traditional method significantly decreases, while the AFSA-HOP method still maintains good recognition performance. When the noise intensity increases to 0.4 and 0.5, the traditional method essentially fails, but the AFSA-HOP method continues to maintain excellent recognition performance. Experimental data indicate that the weight matrix of the Hopfield Neural Network, optimized by the Artificial Fish Swarm Algorithm, exhibits a more reasonable distribution, and the threshold parameters are configured in a more optimal state. These optimizations enable the network to more accurately identify the essential features of the digits when facing noise interference, effectively suppressing the negative impacts of noise. The Artificial Fish Swarm Algorithm, as an optimization tool, successfully finds the optimal combination of weight and threshold values for the Hopfield Neural Network, thus optimizing the network's performance.

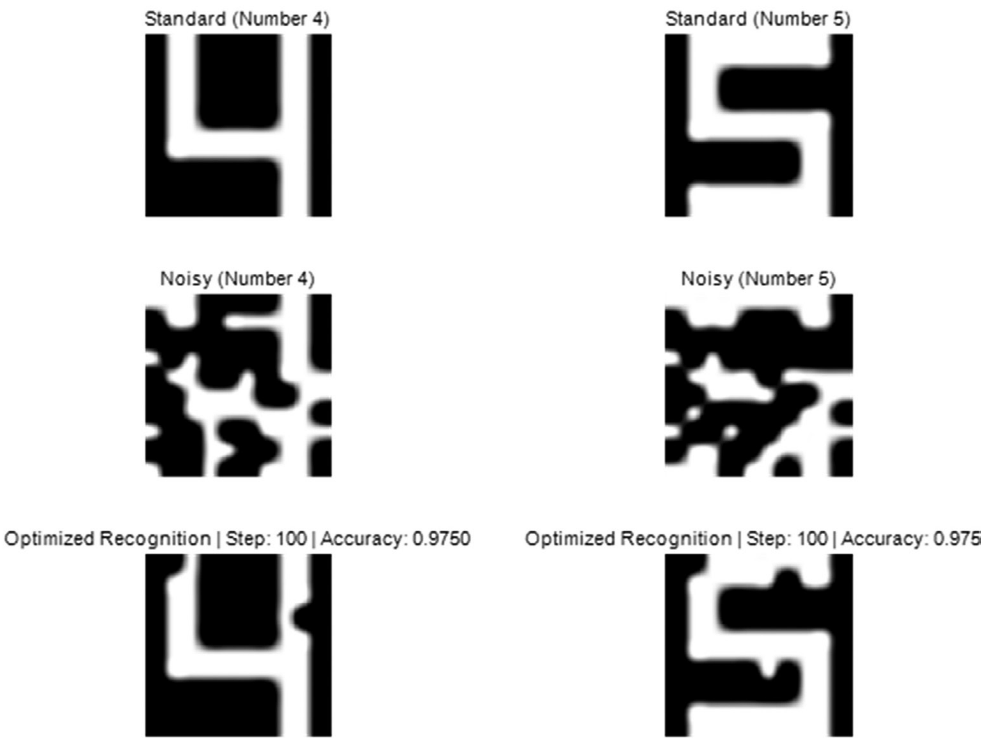

Figure 7. Digit Recognition Results at Noise Intensity of 0.3 (AFSA-HOP Integrated Method).

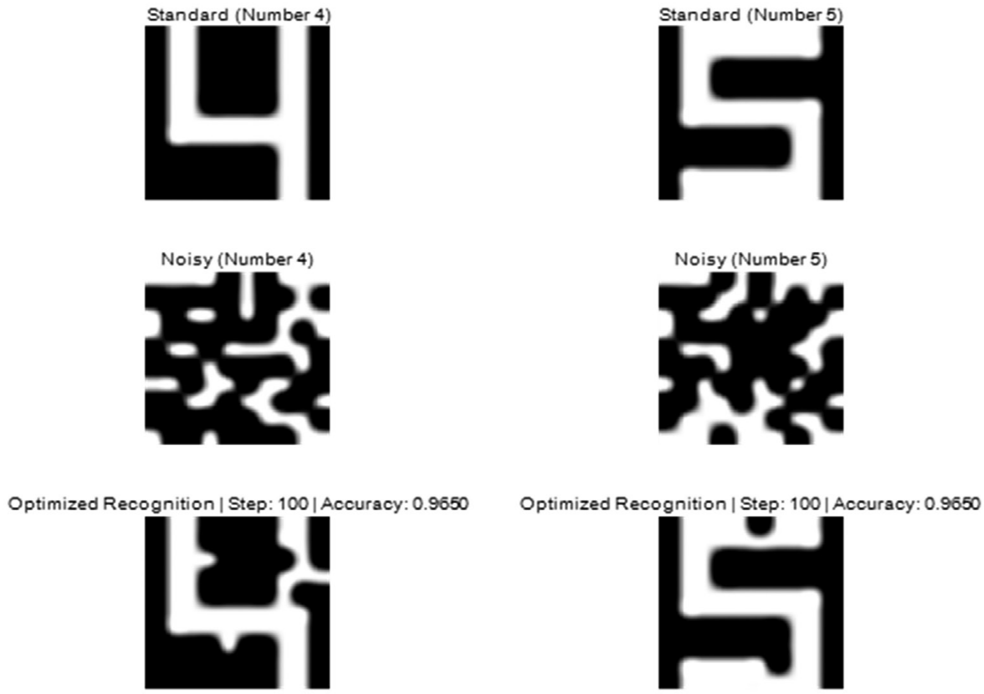

Figure 8. Digit Recognition Results at Noise Intensity of 0.4 (AFSA-HOP Integrated Method).

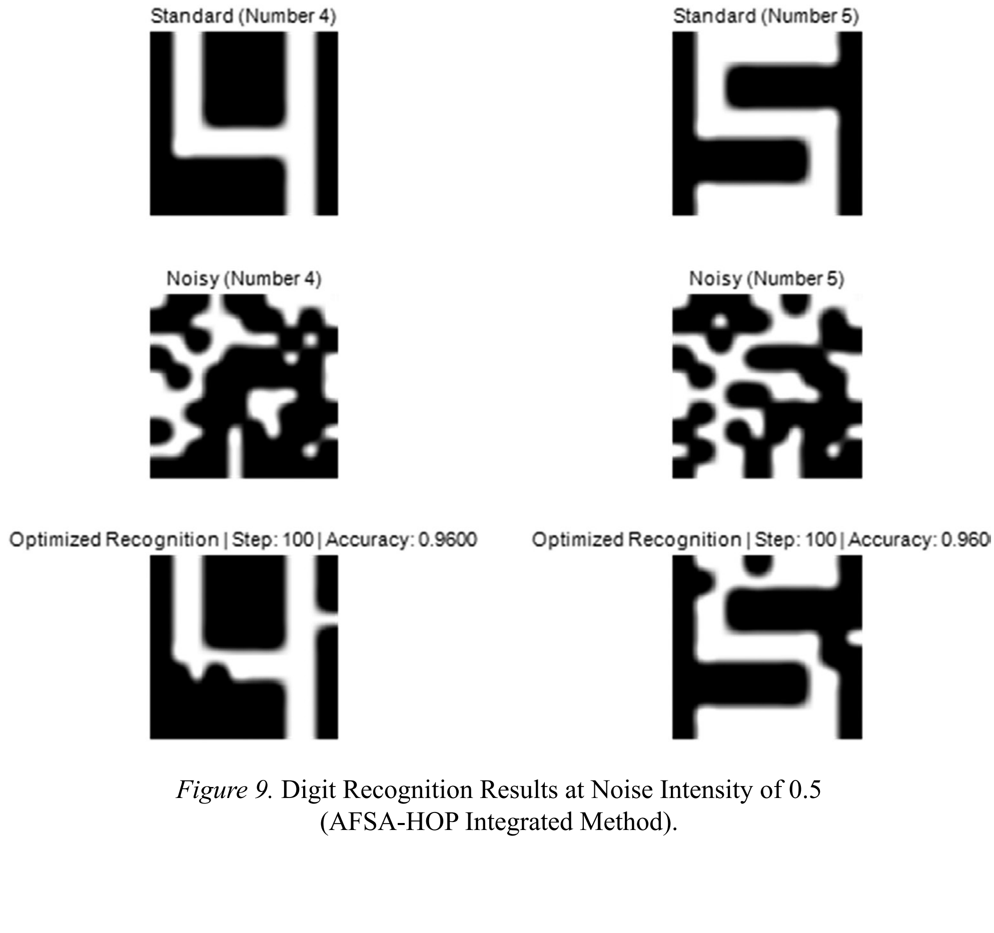

Figure 9. Digit Recognition Results at Noise Intensity of 0.5 (AFSA-HOP Integrated Method).

4.2.3. Statistical Significance Test

In the numerical recognition experiments of discrete Hopfield neural networks, analysis of variance (ANOVA) is a powerful statistical tool that can be used to compare whether there are significant differences in recognition accuracy under multiple sets of noise intensities. In this experiment, a combination of ANOVA and post hoc testing (Tukey HSD) was used to verify the reliability of the performance improvement of the optimized model.

- Research hypothesis

The purpose of analysis of variance (ANOVA) is to determine whether there is a significant difference in recognition accuracy under different levels of noise intensity. Therefore, this article proposes the following hypothesis:

1. Null hypothesis (H0): The mean of all groups is equal (i.e., noise intensity has no significant impact on recognition accuracy).
2. Alternative hypothesis (H1): At least one group has a mean that is different from the other groups (i.e., noise intensity has a significant impact on recognition accuracy).

- Single factor ANOVA

Through simulation experiments, the results of three experiments with noise levels of 0.2 (before fish swarm algorithm optimization), 0.3 (after fish swarm algorithm optimization), 0.4 (after fish swarm algorithm optimization), and 0.5 (after fish swarm algorithm optimization) were used as the experimental data for single factor ANOVA, as shown in Table 2.

Statistical tests were conducted using one-way ANOVA, and the relevant indicators for statistical tests in analysis of variance are shown in Table 3. From Table 3, it can be observed that, F = 419.515, Indicating significant differences between groups; The p-value is 0.000 ($p < 0.05$), therefore rejecting the null hypothesis (H_0), indicating that at least one group has a mean different from the other groups.

The results of the one-way ANOVA indicate that at least one group mean differs significantly from the others. However, this analysis does not reveal which specific group pairs exhibit significant differences. Therefore, a post hoc test is required, specifically Tukey's Honestly Significant Difference Test (Tukey HSD).

Step 1. Calculate the mean square error (MSE). From Table 3, the within-group mean square error is obtained as MSE = 0.229.

Step 2. Calculate the critical value for Tukey HSD.

$$HSD = q \times \sqrt{\frac{MSE}{2}} \tag{11}$$

Here, the Tukey critical value is determined by referring to the studentized range distribution table ($\alpha = 0.05$, number of groups k = 4, and within-group degrees of freedom df_within = 8), yielding $q \approx 4.04$. Based on this value, the HSD critical value is then calculated.

$$HSD = 4.04 \times \sqrt{\frac{0.229}{2}} = 1.367 \tag{12}$$

Table 2. Experimental Data for One-Way ANOVA.

Noise Intensity	Recognition Accuracy 1 (%)	Recognition Accuracy 2 (%)	Recognition Accuracy 3 (%)
0.2 (Before Optimization)	85	86	85
0.3 (After Optimization)	97.5	97	98
0.4 (After Optimization)	96.5	96	97
0.5 (After Optimization)	96	95.5	95.5

Table 3. Statistical Indicators for ANOVA Significance Testing.

Source of Variation	SS	df	MSE	F	P-value	F crit
Between Groups	288.417	3	96.139	419.515	0.000	4.066
Within Groups	1.833	8	0.229			

Step 3. Calculate the mean differences between each pair of groups. By comparing the mean differences among all group pairs, it can be determined whether they exceed the HSD critical value.

1. Noise 0.2 (before optimization) vs. Noise 0.3 (after optimization): $|85.3 - 97.5| = 12.2 > 1.367 \rightarrow$ Significant
2. Noise 0.2 (before optimization) vs. Noise 0.4 (after optimization): $|85.3 - 96.5| = 11.2 > 1.367 \rightarrow$ Significant
3. Noise 0.2 (before optimization) vs. Noise 0.5 (after optimization): $|85.3 - 95.7| = 10.4 > 1.367 \rightarrow$ Significant
4. Noise 0.3 (after optimization) vs. Noise 0.4 (after optimization): $|97.5 - 96.5| = 1 < 1.367 \rightarrow$ Not significant
5. Noise 0.3 (after optimization) vs. Noise 0.5 (after optimization): $|97.5 - 95.7| = 1.8 > 1.367 \rightarrow$ Significant
6. Noise 0.4 (after optimization) vs. Noise 0.5 (after optimization): $|96.5 - 95.7| = 0.8 < 1.367 \rightarrow$ Not significant

Based on these results, it can be concluded that the overall digit recognition performance after optimization (under noise levels 0.3, 0.4, and 0.5) is significantly better than that before optimization (under noise level 0.2). Within the optimized group, the recognition performance under noise level 0.3 differs significantly from that under 0.5, while no significant differences are observed between noise levels 0.3 and 0.4, or between 0.4 and 0.5. These statistical test results further demonstrate that the Artificial Fish Swarm Algorithm, as an optimization tool, has effectively improved the overall performance of the discrete Hopfield neural network and substantially enhanced the accuracy of digit recognition.

4.2.4. Comparison of Models Based on Different Optimization Algorithms

The simulation results show that the Artificial Fish Swarm Algorithm (AFSA) can consistently optimize the discrete Hopfield neural network, achieving a recognition accuracy of 96% under a noise level of 0.5. To further validate the superiority of the AFSA-HOP integration method, comparisons were made with other optimization strategies, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Through experiments, it was found that optimizing the model using genetic algorithm and particle swarm optimization algorithm cannot achieve digit recognition at a noise level of 0.5, but the accuracy of digit recognition is higher at a noise level of 0.3. When the noise level was set to 0.3, classification was performed on the same digit dataset using integrated models based on three optimization algorithms (including AFSA-HOP, GA-HOP, and PSO-HOP). The digit recognition performance of the AFSA-HOP method was compared to the other methods, and the experimental results are shown in Table 4.

As shown in Table 4, the discrete Hopfield neural network model optimized by the Artificial Fish Swarm Algorithm (AFSA) achieves the highest recognition accuracy in digit classification. Specifically, when the noise is 0.3, the AFSA-HOP model after 100 iterations is 3.3 percentage points higher than the GA-HOP model after 300 iterations, and 2.5 percentage points higher than the POS-HOP model after 150 iterations. This indicates that the proposed AFSA-HOP integration method can effectively reduce recognition errors under the same noise, and as the noise increases, the advantages of AFSA-HOP integration method will become more apparent.

Table 4. Comparison of Digit Recognition Performance for Different Models (Noise Level = 0.3).

Model	Mean Recognition Accuracy (%)
AFSA-HOP (100 iterations)	97.5
GA-HOP (300 iterations)	94.2
POS-HOP (150 iterations)	95.0

5. Conclusion

This paper presents a digital recognition method based on the Artificial Fish Swarm Algorithm optimized Discrete Hopfield Neural Network (AFSA-HOP integration method), which effectively addresses the issue of the Discrete Hopfield Network getting trapped in local optima during associative memory. The traditional Hopfield Neural Network fails to effectively recognize digits under a noise intensity of 0.2. However, by first optimizing the weight and threshold parameters of the Discrete Hopfield Network using the Artificial Fish Swarm Algorithm, and then training and testing the network, high signal-to-noise ratio digital recognition is achieved. In addition to correctly recognizing the digits under noise intensities of 0.2 and 0.3, the method still performs well under noise intensities of 0.4 and 0.5. Simulation results show that this method achieves better recognition performance, significantly outperforming the traditional Hopfield neural networks, GA optimized Hopfield neural networks and POS optimized Hopfield neural networks. This innovation is not only reflected in the algorithm design, where the Artificial Fish Swarm Algorithm is integrated with the Hopfield Network, but also in the construction of a complete digital recognition optimization framework, including systematic processes such as data preprocessing, network initialization, parameter optimization, and performance testing. Future research could further explore improvements to the performance of the Artificial Fish Swarm Algorithm or combine it with other optimization algorithms to further enhance the efficiency and recognition accuracy of the AFSA-HOP integration method. The AFSA-HOP method provides a new technical pathway for digital recognition in complex environments, and its core concept can be extended to other pattern recognition fields such as license plate recognition and invoice processing, offering significant theoretical value and broad application prospects.

Conflict of interest

The authors declare no conflict of interest.

Funding

This paper is supported by the National Natural Science Foundation of China under contract no. 72464008, 72364008, Guangxi Project of Philosophy and Social Science for Planning no. 24JYF027, Bagui Young Top-Notch Talent Project.

Data availability

Data used in this article are available upon request from the authors.

References

[1] S. Ibrahim et al., "Malaysian Vehicle Plate Number Identification Using Deep Learning", *AIP Conference Proceedings*, vol. 2582, no. 1, pp. 1–7, 2023.
http://dx.doi.org/10.1063/-5.0123958

[2] D. Yang and L. Yang, "A Deep Learning-based Framework for Vehicle License Plate Detection", *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 1, p. 1009, 2024.
http://dx.doi.org/10.14569/IJACSA.2024.01501100

[3] M. Arshid et al, "A Comparative Study on Detection and Recognition of Nonuniform License Plates", *Big Data and Cognitive Computing*, vol. 8, no. 11, p. 155, 2024.
http://-dx.doi.org/10.3390/bdcc8110155

[4] A. O. Tayo and A. D. Gideon, "Vehicle License Plate Recognition Using Edge Detection and Neural Network", *International Journal of Science and Research*, vol. 4, no. 4, pp. 3158–3161, 2015. https://www.ijsr.net/getabstract.php?paperid=SUB153793

[5] H. N. Sharada et al, "An Optimized Neural Network-based Character Recognition and Relation Finding for Mathematical Expression Images", *Multimedia Tools & Applications*, vol. 83, no. 19, pp. 57163–57185, 2024. http://dx.doi.org/10.1007/s11042-023-17725-4

[6] P. Gader et al., "A Fuzzy Logic System for the Detection and Recognition of Street Number Fields on Handwritten Postal Addresses", *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 1, pp. 83–95, 2025. http://dx.doi.org/10.1109/91.366562

[7] Y. Wang, "Intelligent Invoice Identification Technology Based on Zxing Technology", *Lecture Notes in Electrical Engineering*, vol. 791, pp. 87–93, 2022. http://dx.doi.org/-10.1007/978-981-16-4258-6_11

[8] J. Wang and W. Q. Yan, "BP-Neural Network for Plate Number Recognition", *International Journal of Digital Crime and Forensics*, vol. 8, no. 3, pp. 34–45, 2016. http://dx.doi.org/10.4018/IJDCF.2016070103

[9] S. Prabhanjan and R. Dinesh, "Handwritten Devanagari Numeral Recognition by Fusion of Classifiers", *International Journal of Signal Processing Image Processing & Pattern Recognition*, vol. 8, no. 1, pp. 47–54, 2015. http://dx.doi.org/10.4172/2324-9307.1000128

[10] J. Xu et al, "Adaptive Fuzzy Contribution Classification", *Journal of Electronic Imaging*, vol. 33, no. 6, p. 21, 2024. http://dx.doi.org/10.1117/1.JEI.33.6.063009

[11] S. Sampath and V. Srivastava, "On Stability and Associative Recall of Memories in Attractor Neural Networks", *PLoS ONE*, vol. 15, no. 9, p. e0238054, 2020. http://dx-.doi.org/10.1371/journal.pone.0238054

[12] T. Lan et al., "Recognition of Vehicle License Plate Based on Hopfield Artificial Neural Network", *Sensors and Materials*, vol. 33, no. 3, pp. 3983–3990, 2021. http://dx.doi.org/-10.18494/SAM.2021.3518

[13] S. Zhang et al, "Design and Implementation for Hopfield Associative Memory Neural Networks", *Journal of Shanghai Normal University*, vol. 45, no. 1, pp. 16–20, 2016. http://dx.doi.org/info:doi/10.3969/J.ISSN.1000-5137.2016.01.003

[14] T. Kuremoto et al., "The Dynamical Recollection of Interconnected Neural Networks Using Meta-heuristics", *Electronics and Communications in Japan*, vol. 95, no. 6, pp. 12–23, 2012. http://dx.doi.org/10.1002/ecj.11372

[15] J. Chen et al., "MTS-PRO2SAT: Hybrid Mutation Tabu Search Algorithm in Optimizing Probabilistic 2 Satisfiability in Discrete Hopfield Neural Network", *Mathematics*, vol. 12, no. 5, p. 721, 2024. http://dx.doi.org/10.3390/math12050721

[16] S. Y. Köse et al., "Capacity Improvement Using Simulation Optimization Approaches: A Case Study in the Thermotechnology Industry", *Engineering Optimization*, vol. 47, no. 2, pp. 149–164, 2015. http://dx.doi.org/10.1080/0305215X.2013.875166

[17] S. M. Almufti et al., "Overview of Metaheuristic Algorithms", *Polaris Global Journal of Scholarly Research and Trends*, vol. 2, no. 2, pp. 10–32, 2023. http://dx.doi.org/-10.58429/pgjsrt.v2n2a144

[18] J. Jiang et al., "Embedded Static Task Allocation and Scheduling Based on Simulated Annealing and Genetic Algorithm", *Journal of Computational Information Systems*, vol. 10, no. 4, pp. 1465–1472, 2014. http://dx.doi.org/10.12733/jcis9288

[19] H. Allaoua, "Optimal Adjusting of Simulated Annealing Parameters", *Military Technical Courier / Vojnotehnicki Glasnik*, vol. 72, no. 1, p. 80, 2024. http://dx.doi.org/10.5937/-vojtehg72-47242

[20] E. Ba and G. Yildizdan, "A Discrete Particle Swarm Algorithm with Symmetry Methods for Discrete Optimization Problems", *Konya Journal of Engineering Sciences / Konya Mühendislik Bilimleri Dergisi*, vol. 11, no. 3, p. 610, 2023. http://dx.doi.org/10.36306/-konjes.1199674

[21] C. Liu et al., "An Improved Heuristic Mechanism Ant Colony Optimization Algorithm for Solving Path Planning", *Knowledge-based Systems*, vol. 271, p. 110540, 2023. http://dx-.doi.org/10.1016/j.knosys.2023.110540

[22] F. Pourpanah et al., "A Review of Artificial Fish Swarm Algorithms: Recent Advances and Applications", *Artificial Intelligence Review*, vol. 56, no. 3, pp. 1867–1903, 2022. http://-dx.doi.org/10.1007/s10462-022-10214-4

[23] V. Kourepinis et al., "An Artificial Fish Swarm Optimization Algorithm for the Urban Transit Routing Problem", *Applied Soft Computing*, vol. 155, p. 111446, 2024. http://dx.doi.org/-10.1016/j.asoc.2024.111446

Received: May 2025
Revised: June 2025
Accepted: June 2025

Contact addresses:
Xin Su
School of Business
Guilin University of Electronic Technology
Guilin
Guangxi
China
e-mail: suxin0530@guet.edu.cn

Daiheng Xie
School of Business
Guilin University of Electronic Technology
Guilin
Guangxi
China
e-mail: 1163206394@qq.com

Yue Zhao*
School of Business
Guilin University of Electronic Technology
Guilin
Guangxi
China
e-mail: zhaoyue@guet.edu.cn
*Corresponding author

Xin Su obtained her PhD in Management Science and Engineering from Harbin Engineering University in 2019. She worked as a post-doctoral fellow in the Department of Engineering Physics at Tsinghua University from 2020 to 2022. Currently, she is an associate researcher and doctoral supervisor at Guilin University of Electronic Science and Technology. Her interests focus on machine learning, optimization algorithms and complex networks.

Daiheng Xie obtained his Bachelor's degree in Industrial Engineering from Guilin University of Electronic Science and Technology in 2022. He is currently a master's student majoring in Management Science and Engineering at the School of Business, Guilin University of Electronic Science and Technology. His interests focus on computer science, optimization algorithms and information technology.

Yue Zhao obtained her PhD degree in Forestry Economics and Management from Northeast Forestry University in 2018. She worked as a postdoctoral researcher at Tsinghua University from 2020 to 2022. Currently, she is an associate professor and master's supervisor at Guilin University of Electronic Science and Technology. Her interests focus on optimization algorithms, complex networks and data mining.