

## Research Article

# Cultural Emperor Penguin Optimizer and Its Application for Face Recognition

Jie Yang and Hongyuan Gao 

*College of Information and Communication Engineering, Harbin Engineering University,  
No. 145 Nantong Street Nangang District, Harbin 150001, China*

Correspondence should be addressed to Hongyuan Gao; [gaohongyuan@hrbeu.edu.cn](mailto:gaohongyuan@hrbeu.edu.cn)

Received 11 June 2020; Revised 31 August 2020; Accepted 15 October 2020; Published 5 November 2020

Academic Editor: Pietro Bia

Copyright © 2020 Jie Yang and Hongyuan Gao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Face recognition is an important technology with practical application prospect. One of the most popular classifiers for face recognition is support vector machine (SVM). However, selection of penalty parameter and kernel parameter determines the performance of SVM, which is the major challenge for SVM to solve classification problems. In this paper, with a view to obtaining the optimal SVM model for face recognition, a new hybrid intelligent algorithm is proposed for multiparameter optimization problem of SVM, which is a fusion of cultural algorithm (CA) and emperor penguin optimizer (EPO), namely, cultural emperor penguin optimizer (CEPO). The key aim of CEPO is to enhance the exploitation capability of EPO with the help of cultural algorithm basic framework. The performance of CEPO is evaluated by six well-known benchmark test functions compared with eight state-of-the-art algorithms. To verify the performance of CEPO-SVM, particle swarm optimization-based SVM (PSO-SVM), genetic algorithm-based SVM (GA-SVM), CA-SVM, and EPO-SVM, moth-flame optimization-based SVM (MFO-SVM), grey wolf optimizer-based SVM (GWO-SVM), cultural firework algorithm-based SVM (CFA-SVM), and emperor penguin and social engineering optimizer-based SVM (EPSEO-SVM) are used for the comparison experiments. The experimental results confirm that the parameters optimized by CEPO are more instructive to make the classification performance of SVM better in terms of accuracy, convergence rate, stability, robustness, and run time.

## 1. Introduction

As an important branch of pattern recognition, face recognition has the widespread application demands in the field, such as intelligent monitoring, virtual reality, medicine examination, and human-computer interaction. The key step in face recognition is classifier design, namely, how to use the extracted features to classify the new face images. Recently, many classifiers have been presented, such as decision tree [1], neural network [2], k-nearest (KNN) [3], and support vector machine (SVM) [4], among which SVM is the most popular one.

Although SVM has the advantages of good generalization ability, small training set size limitation, and high noise stability, it sometimes leads to overfitting problems. To address this problem, some researchers have applied genetic

algorithm (GA) [5], particle swarm optimizer (PSO) [6], and simulated annealing (SA) [7] to parameter optimization of SVM. However, the implement of GA is complex although the search process has the characteristic of exclusivity, which makes the convergence speed slow. PSO and SA are successfully applied to solve some engineering optimization problems, but they have poor global search ability for parameters optimization. Therefore, it is worthy to design a new intelligent algorithm with superior performance for parameters optimization of SVM.

In the face of complex optimization problems which are difficult to deal with effectively by traditional methods, many new inspired algorithms have been proposed for solving problems by simulations of natural rules and social behaviours of biological population. Recently, some algorithms such as grey wolf optimizer (GWO) [8], moth-flame

optimization (MFO) [9], tunicate swarm algorithm (TSA) [10], and dice game optimizer (DGO) [11] are widely used. In 2018, Dhiman and Kumar proposed a novel bio-inspired algorithm named emperor penguin optimizer [12], which is inspired by the budding behavior of emperor penguin. However, the experimental results showed the convergence rate of EPO will decrease continuously in the later stage, which will affect the efficiency of optimization. Afterwards, Baliarsingh et al. proposed a hybrid algorithm using EPO with social engineering optimizer (SEO), namely, emperor penguin and social engineering optimizer (EPSEO) [13] to improve the performance of EPO.

Cultural algorithm (CA) is a bio-inspired intelligent algorithm proposed by Reynolds in 1994 [14], which is by simulating the cultural evolution of human society. CA is a double-layer evolutionary mechanism, which uses all kinds of knowledge accumulated in belief space to update continuously, so as to guide the evolution of population space and accelerate the optimization efficiency of the algorithm. Therefore, it can provide a space that allows other evolutionary algorithms to be embedded to cooperate and promote each other. Recently, some researchers have proposed PSO, bacterial foraging algorithm (BFA), and firework algorithm (FA) as a population space to ingrate with CA [15–17] and verified that the performance of the hybrid algorithms are significantly improved.

Based on the above observations, due to the fact that CA with single population does not make full use of the knowledge of belief space and the convergence speed of EPO will decrease continuously in the later stage, we propose a hybrid metaheuristic algorithm named cultural emperor penguin optimizer (CEPO) with the help of the embedded framework provided by CA. In CEPO, hybrid of two different evolution processes for locations is an effective mechanism to enhance the exploitation capability of EPO.

Specifically, CEPO are implemented on six well-known benchmark test functions compared with eight state-of-the-art optimization algorithms. Furthermore, CEPO is applied to multiparameter optimization problem of SVM for face recognition. In the process of feature, extraction feature representation vectors of face images are obtained by Gabor filter and Principal Component Analysis (PCA), which are as input to CEPO-SVM. PCA is used to make up Gabor wavelet transform disadvantages caused by increased dimensionality. Along with feature classification, penalty parameter and kernel parameter of SVM are optimized by CEPO to get the training model with high classification accuracy. To prove the performance of CEPO-SVM, eight state-of-the-art optimization algorithm-based SVM are used for the comparison experiments in terms of accuracy, convergence rate, stability, robustness, and run time.

The main contributions of this paper are as follows:

- (1) A new hybrid intelligent algorithm named CEPO inspired by EPO and the framework of CA is proposed
- (2) CEPO is applied to automatic learning of the network parameters of SVM to solve the classification problem for face recognition

- (3) We demonstrate that the performance of CEPO-SVM is superior on the basis of accuracy, convergence rate, stability, robustness, and run time, compared with eight state-of-the-art algorithm-based SVM
- (4) CEPO as a multidimensional search algorithm can not only obtain the optimal penalty parameter and kernel parameter of SVM but also can be employed in other classifiers and other real number optimization problems

The rest of the work is organized as follows. Section 2 describes feature extraction of face images including the Gabor face image representation and dimensionality reduction by Principal Component Analysis. Section 3 presents formulations of parameter optimization of SVM for classification as an optimization problem. The design of cultural emperor penguin optimizer for parameter optimization of SVM is depicted in Section 4. Section 5 describes the experimental setup. In section 6, the experiment and result analysis are presented. Section 7 provides the conclusion and future works.

## 2. The Feature Extraction for Face Images

At the preprocessing stage, each image is first converted into grayscale and then adjusted to the same size. Due to the superior robustness of the Gabor filter to changes in brightness and attitude of face images, the Gabor filter is used to capture more useful features from face images. Furthermore, to make up Gabor wavelet transform disadvantages caused by increased dimensionality, PCA is applied to dimensionality reduction. After the preprocessing stage, face images are divided into training sets and testing sets to obtain feature representation vectors, which are as the input to SVM. Figure 1 shows the block diagram of face recognition based on Gabor-PCA.

*2.1. Gabor Face Image Representation.* The two-dimensional Gabor filter can be defined as follows:

$$\begin{aligned} \Phi_{o,r}(\mathbf{z}) = & \frac{\|\mathbf{W}_{o,r}\|^2}{\sigma^2} \exp\left(-\frac{\|\mathbf{W}_{o,r}\|^2 \|\mathbf{z}\|^2}{2\sigma^2}\right) \\ & \cdot \left[ \exp(j\mathbf{W}_{o,r}\mathbf{z}) - \exp\left(-\frac{\sigma^2}{2}\right) \right], \end{aligned} \quad (1)$$

where  $\mathbf{z} = (\bar{x}, \bar{y})$  defines the pixel position in the spatial domain,  $o$  defines the orientation of the Gabor filter,  $r$  defines the scale of the Gabor filter.  $\sigma$  represents the radius of Gaussian function, which is used to limit the size of the Gabor filter, and  $\mathbf{W}_{o,r}$  is the wave vector of the filter at orientation  $o$  and scale  $r$  [18, 19]. In this paper, with five different scales,  $r \in \{0, \dots, 4\}$ , and eight different orientations,  $o \in \{0, \dots, 7\}$ , 40 different Gabor filters can be obtained. The real part and the magnitude of the 40 Gabor filters are shown in Figures 2 and 3, which exhibit desirable

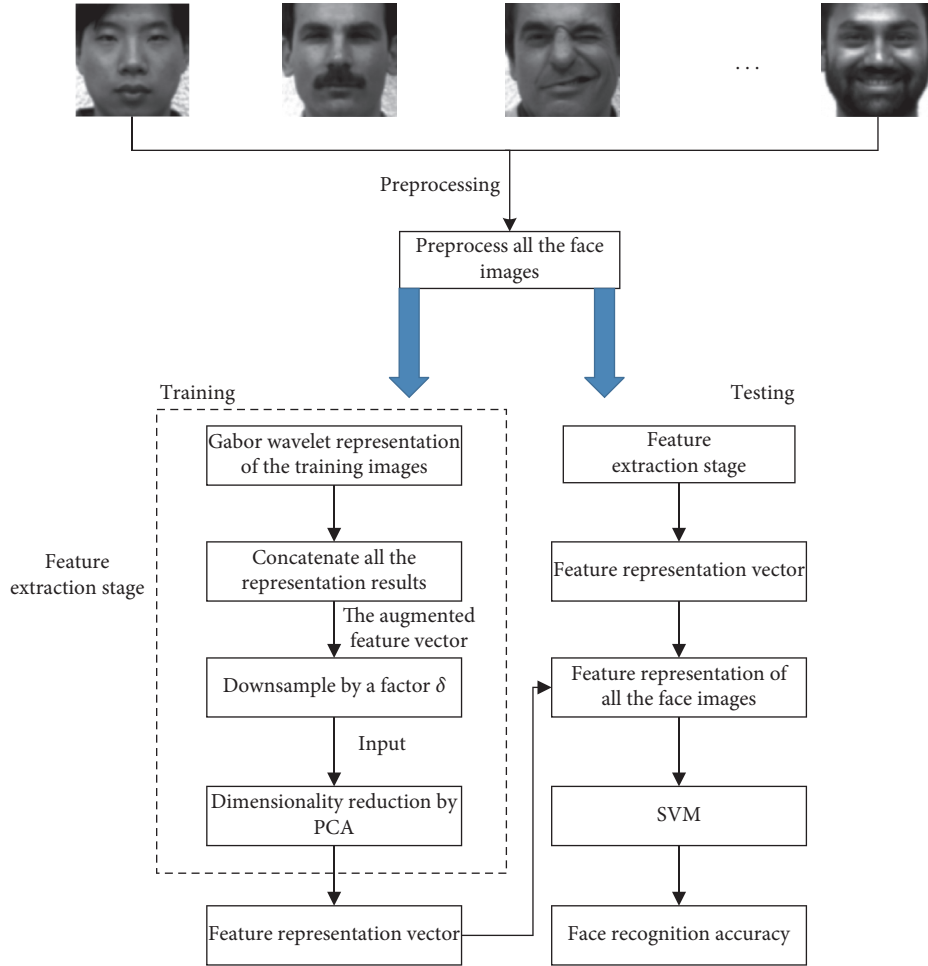


FIGURE 1: Block diagram of face recognition based on Gabor-PCA.

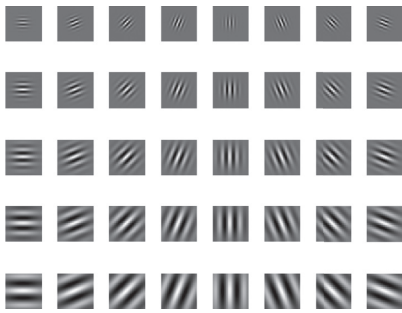


FIGURE 2: The real part of the 40 Gabor filters.

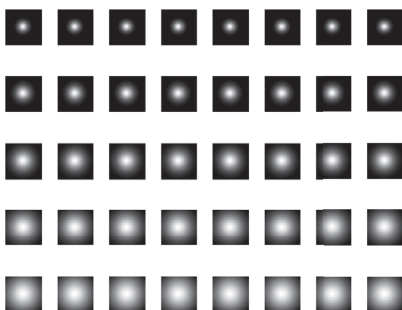


FIGURE 3: The magnitude of the 40 Gabor filters.

characteristics of spatial locality, spatial frequency, and orientation selectivity.

Let  $Y(\mathbf{z})$  represent the grey level distribution of an image, and the Gabor wavelet representation of an image can be given by

$$\mathbf{G}_{o,r}(\mathbf{z}) = Y(\mathbf{z}) * \varphi_{o,r}(\mathbf{z}), \quad (2)$$

where  $\mathbf{G}_{o,r}(\mathbf{z})$  defines the convolution result between image  $Y(\mathbf{z})$  and the Gabor filters  $\varphi_{o,r}(\mathbf{z})$  at different orientations and scales. In order to improve the computation speed, the convolution computation can be transformed by FFT [20]. To contain the different spatial localities, spatial frequencies, and orientation selectivities, all these representation results will be concatenated to derive an augmented feature vector  $\mathbf{Q}$ . In [19], before the concatenation, each  $\mathbf{G}_{o,r}(\mathbf{z})$  will be downsampled by a factor  $\delta$  at different scales and orientations. Therefore, the augmented Gabor feature vector  $\mathbf{Q}^{(\delta)}$  can be given by

$$\mathbf{Q}^{(\delta)} = \left( (\mathbf{G}_{0,0}^{(\delta)})^T (\mathbf{G}_{0,1}^{(\delta)})^T \dots (\mathbf{G}_{4,7}^{(\delta)})^T \right)^T, \quad (3)$$

where  $T$  denotes transpose operator,  $\delta$  is the downsample factor, and  $\mathbf{G}_{o,r}^{(\delta)}$  is the concatenated column vector.

### 2.2. Principle Component Analysis for Feature Optimization.

The augmented Gabor feature vector which is introduced in equation (3) has very high dimensionality:  $\mathbf{Q}^{(\delta)} \in \mathbf{R}^O$ , where  $O$  is the dimensionality of vector space. Obviously, the high dimensionality seriously affects the computing speed and recognition rate in the process of face recognition. In the method with PCA [21], we can find orthogonal basis for feature vector, sort dimensions in order of importance, and discard low significance dimensions. Let  $\sum \mathbf{Q}^{(\delta)} \in \mathbf{R}^{O \times O}$  be the covariance matrix of the augmented feature vector  $\mathbf{Q}^{(\delta)}$ :

$$\sum \mathbf{Q}^{(\delta)} = E\left\{[\mathbf{Q}^{(\delta)} - E(\mathbf{Q}^{(\delta)})][\mathbf{Q}^{(\delta)} - E(\mathbf{Q}^{(\delta)})]^T\right\}, \quad (4)$$

where  $E(\cdot)$  is the expectation operator. The covariance matrix  $\sum \mathbf{Q}^{(\delta)}$  can be transformed into the following form:

$$\sum \mathbf{Q}^{(\delta)} = \mathbf{\Omega} \mathbf{\Lambda} \mathbf{\Omega}^T, \quad (5)$$

$$\mathbf{\Omega} = [U_1, U_2, \dots, U_O], \quad (6)$$

$$\mathbf{\Lambda} = \text{diag}\{p_1, p_2, \dots, p_O\}, \quad (7)$$

where  $\mathbf{\Omega} \in \mathbf{R}^{O \times O}$  is an orthogonal eigenvector matrix and  $\mathbf{\Lambda} \in \mathbf{R}^{O \times O}$  is a diagonal eigenvalue matrix, and the diagonal elements are arranged in descending order ( $p_1 \geq p_2 \geq \dots \geq p_O$ ).

An important property of PCA is that it is the optimal signal reconstruction in the sense of minimum mean-square error by using a subset of principal components to represent the original signal [19]. Following this property, PCA can be applied to dimensionality reduction:

$$\mathbf{X}^{(\delta)} = \mathbf{S}^T \mathbf{Q}^{(\delta)}, \quad (8)$$

where  $\mathbf{S} = [U_1, U_2, \dots, U_J]$ ,  $J < O$ , and  $\mathbf{S} \in \mathbf{R}^{O \times J}$ . The lower dimensional vector  $\mathbf{X}^{(\delta)} \in \mathbf{R}^J$  captures the most expressive features of the original data  $\mathbf{Q}^{(\delta)}$ .

## 3. Formulations of Parameter Optimization of SVM for Classification

**3.1. Model of Support Vector Machine.** Support vector machine (SVM) [22] is a kind of binary classifier with strong learning ability and generalization ability. The key point of SVM is to find the optimal hyperplane for accurate classification of two classes and ensure that the classification gap is large enough. Figure 4 shows the optimal hyperplane of SVM, where  $H$  indicates the hyperplane and *margin* represents the gap between class  $H_1$  and class  $H_2$ .

Suppose the training data  $\mathbf{V}$  can be given by

$$\mathbf{V} = \{(a_h, b_h) | a_h \in \mathbf{R}^w, b_h \in \{-1, +1\}, h = 1, 2, \dots, n\}, \quad (9)$$

where  $a_h$  are the training samples,  $b_h$  are the labels in  $w$ -dimensional vector,  $n$  is the number of training data, and each variable must subject to the criteria  $a_h \in \mathbf{R}^w$  and  $b_h \in \{-1, +1\}$ . For linear data, the hyperplane  $g(\mathbf{a}) = 0$  that separates the given data can be determined:

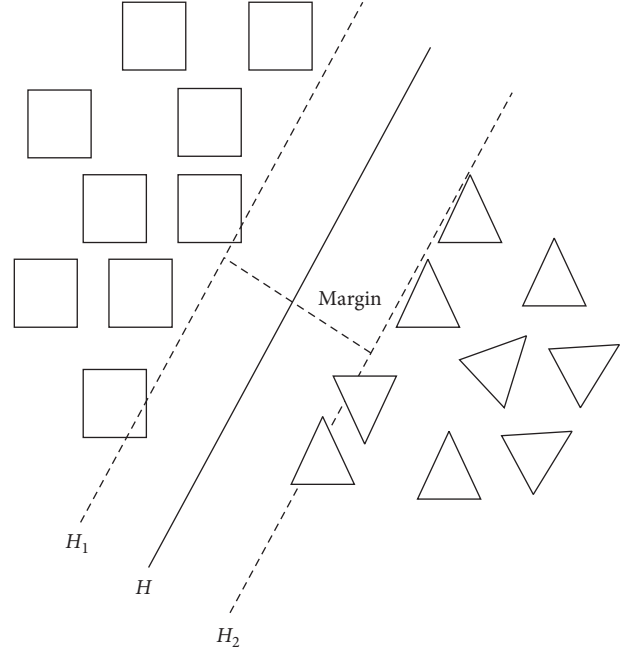


FIGURE 4: The schematic model of a linear SVM.

$$g(\mathbf{a}) = \boldsymbol{\omega}^T \mathbf{a} + c, \quad (10)$$

where  $\boldsymbol{\omega}$  is  $n$ -dimensional vector and  $c$  is a scalar.  $\boldsymbol{\omega}$  and  $c$  determine all the boundaries. SVM could be classified according to the problem types, when the hyperplane with minimum margin width  $\|\boldsymbol{\omega}^2\|/2$  subject to

$$b_h(\boldsymbol{\omega}^T \mathbf{a} + c) \geq 1. \quad (11)$$

For the data that cannot be separated linearly, the relaxing variables  $\xi_h$  are introduced to allow for optimal generalization and classification by controlling the size of them. Then, the optimal hyperplane can be obtained by the following optimization problem:

$$\Psi(\boldsymbol{\omega}, \boldsymbol{\xi}) = \frac{1}{2} \boldsymbol{\omega}^2 + C \sum_{h=1}^n \xi_h, \quad (12)$$

$$\text{subject to } b_h(\boldsymbol{\omega}^T \mathbf{a} + c) \geq 1 - \xi_h, \xi_h \geq 0, \quad (13)$$

where  $C$  denotes the penalty parameter. To satisfy the Karush-Kuhn-Tucker (KKT) conditions, the Lagrange multipliers  $\alpha_h$  is introduced. The abovementioned optimization problem is converted into the dual quadratic optimization problem:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{h=1}^n \sum_{k=1}^n b_h b_k \alpha_h \alpha_k (a_h \cdot a_k) - \sum_{k=1}^n \alpha_k, \quad (14)$$

$$\text{subject to } \sum_{h=1}^n b_h \alpha_h = 0, \quad 0 \leq \alpha_h \leq C. \quad (15)$$

Thus, the ultimate classification function can be obtained by solving the dual optimization problem:

$$g(\mathbf{a}) = \text{sgn} \left\{ \left( \sum_{h=1}^n \alpha_h b_h (a_h \cdot a_k) \right) + c \right\}, \quad (16)$$

where  $\text{sgn}$  represents symbolic function.

Kernel functions can be used to map the data in the low-dimensional input space to the high-dimensional space by nonlinearity. Therefore, the linear indivisibility of the input can be transformed into the linear separable problem by this mapping. Kernel function can be defined as follows:

$$K(a_h, b_k) = (\psi(a_h), \psi(b_k)). \quad (17)$$

Then, the optimal function can be given by

$$g(\mathbf{a}) = \text{sgn} \left\{ \left( \sum_{h=1}^n \alpha_h b_h K(a_h \cdot a_k) \right) + c \right\}. \quad (18)$$

Several types of kernel functions such as Linear Kernel, Polynomial Kernel, and Radial Basis Function (RBF) are widely used. However, RBF has advantages of realizing nonlinear mapping, less parameters, and less numerical difficulties. In this paper, RBF is selected for face recognition, which can be represented as follows:

$$K(\mathbf{a}, \mathbf{b}) = \exp(-\gamma \|\mathbf{a} - \mathbf{b}\|^2), \quad (19)$$

where  $\gamma$  denotes the kernel parameter.

**3.2. Objective Function of SVM.** With a view to obtaining training model with high classification accuracy, the penalty parameter  $C$  which is used to adjust confidence scale and the kernel parameter  $\gamma$  which determines the width of kernel and the range of data points could be optimized by an effective intelligent algorithm. Furthermore, the mean square error of parameters of SVM can be selected as the objective function of the intelligent algorithm:

$$\text{objective} = \frac{1}{n} \sum_{h=1}^n (b_h - \bar{b}_h), \quad (20)$$

where  $b_h$  is the output value of the corresponding parameter and  $\bar{b}_h$  is the actual value of the corresponding parameter.

## 4. Design of Cultural Emperor Penguin Optimizer for Parameter Optimization of SVM

**4.1. The Design of Cultural Emperor Penguin Optimizer.** EPO is a novel optimization algorithm presented by Dhiman and Kumar in 2018 [12], which is inspired by the budding behavior of emperor penguin. In this paper, a hybrid algorithm named CEPO is proposed to solve the real numbers optimization problems, which is with the help of cultural algorithm basic framework in Figure 5. The key idea of CEPO is to obtain problem-solving knowledge from the budding behavior of emperor penguin and make use of that knowledge to guide the evolution of emperor penguin population in return.

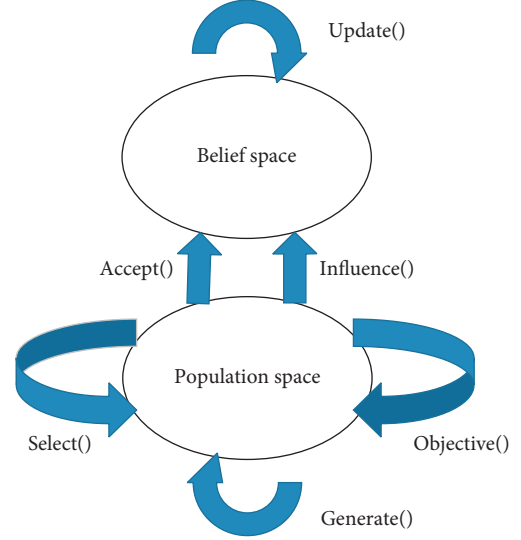


FIGURE 5: The cultural algorithm basic framework.

Suppose CEPO is designed for general minimum optimization problem:

$$\min f(\mathbf{x}_i), \quad (21)$$

where  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  denotes the position of the  $i$ th emperor penguin in the  $D$ -dimensional search space,  $x_j^{\min} < x_{ij} < x_j^{\max}$ , ( $j = 1, 2, \dots, D$ ).  $f()$  is the objective function, and  $f(\mathbf{x}_i)$  denotes the objective value of the position  $\mathbf{x}_i$ .  $x_j^{\min}$  and  $x_j^{\max}$  represent the lower and upper boundary of the position of emperor penguins in the  $j$ th dimension.

Belief space of emperor penguin population in the  $t$ th generation defined is given by  $\mathbf{s}^t$  and  $\mathbf{N}_j^t$ , where  $\mathbf{s}^t$  is situational knowledge component.  $\mathbf{N}_j^t$  is normative knowledge which represents the value space information for each parameter in the  $j$ th dimension and in the  $t$ th generation.  $\mathbf{N}_j^t$  denotes  $\mathbf{I}, L, U, \mathbf{I}_j^t = [l_j^t, u_j^t]$ , where  $\mathbf{I}_j^t$  represents the interval of normative knowledge in the  $j$ th dimension. The lower boundary  $l_j^t$  and the upper boundary  $u_j^t$  are initialized according to the value range of variables given by the problem.  $l_j^t$  represents the objective value of the lower boundary  $l_j^t$  of the  $j$ th parameter, and  $U_j^t$  represents the objective value of the upper boundary  $u_j^t$  of the  $j$ th parameter.

The acceptance function is used to select the emperor penguins who can directly influence the current belief space. In CEPO, the acceptance function selects the cultural individual in proportion top 20% from the current population space to update the belief space.

Situational knowledge  $\mathbf{s}^t$  can be updated by update function:

$$\mathbf{s}^{t+1} = \begin{cases} \mathbf{x}_{\text{best}}^{t+1} & \text{if } f(\mathbf{x}_{\text{best}}^{t+1}) < f(\mathbf{s}^t), \\ \mathbf{s}^t, & \text{else,} \end{cases} \quad (22)$$

where  $\mathbf{x}_{\text{best}}^{t+1}$  is the optimal position of emperor penguin population space in the  $(t + 1)$ th generation.

Assume that, for the  $q$ th cultural individual, a random variable  $\theta_q$  lies in the range of  $[0, 1]$  is produced. The  $q$ th cultural individual affects the lower boundary of normative knowledge in the  $j$ th dimension when  $\theta_q < 0.5$  is satisfied. Normative knowledge  $\mathbf{N}_j^t$  can be updated by update function:

$$\begin{aligned} l_j^{t+1} &= \begin{cases} x_{qj}^{t+1}, & \text{if } x_{qj}^{t+1} \leq l_j^t \text{ or } f(\mathbf{x}_q^{t+1}) < L_j^t, \\ l_j^t, & \text{else,} \end{cases} \\ L_j^{t+1} &= \begin{cases} f(\mathbf{x}_q^{t+1}), & \text{if } x_{qj}^{t+1} \leq l_j^t \text{ or } f(\mathbf{x}_q^{t+1}) < L_j^t, \\ L_j^t, & \text{else.} \end{cases} \end{aligned} \quad (23)$$

The  $q$ th cultural individual affects the upper boundary of normative knowledge in the  $j$ th dimension when  $\theta_q \geq 0.5$  is satisfied:

$$\begin{aligned} u_j^{t+1} &= \begin{cases} x_{qj}^{t+1}, & \text{if } x_{qj}^{t+1} \geq u_j^t \text{ or } f(\mathbf{x}_q^{t+1}) < U_j^t, \\ u_j^t, & \text{else,} \end{cases} \\ U_j^{t+1} &= \begin{cases} f(\mathbf{x}_q^{t+1}), & \text{if } x_{qj}^{t+1} \geq u_j^t \text{ or } f(\mathbf{x}_q^{t+1}) < U_j^t, \\ U_j^t, & \text{else.} \end{cases} \end{aligned} \quad (24)$$

Situational knowledge and normative knowledge can be used to guide emperor penguin population evolution by the influence function. In CEPO, a selection operator  $\beta$  is produced to select one of two ways to influence the of evolution emperor penguin population:

$$\beta = \frac{\text{Max}_{\text{iteration}} - t}{\text{Max}_{\text{iteration}}}, \quad (25)$$

where  $\text{Max}_{\text{iteration}}$  denotes the maximum number of iterations.

Assume that, for the  $i$ th emperor penguin, a random variable  $\lambda_i$  which lies in the range of  $[0, 1]$  is produced. The first way is to update the position of emperor penguin by changing the search size and direction of the variation with belief space, which will be implemented when satisfied  $\lambda_i \leq \beta$ . The position of emperor penguin in the  $j$ th dimension can be updated by

$$\mathbf{x}_{ij}^{t+1} = \begin{cases} x_{ij}^t + |\text{size}(\mathbf{I}_j^t) \cdot N(0, 1)|, & \text{if } x_{ij}^t < s_j^t, \\ x_{ij}^t - |\text{size}(\mathbf{I}_j^t) \cdot N(0, 1)|, & \text{if } x_{ij}^t > s_j^t, \\ x_{ij}^t + \eta \cdot \text{size}(\mathbf{I}_j^t) \cdot N(0, 1), & \text{else,} \end{cases} \quad (26)$$

where  $N(0, 1)$  is a random number subjecting to the standard normal distribution.  $\text{size}(\mathbf{I}_j^t)$  is the length of adjustable interval of the  $j$ th parameter in belief space in the  $t$ th generation.  $\eta$  is set to be in the range of  $[0.01, 0.6]$  by [14].

The other way is by a series of steps in EPO which are the huddle boundary generation, temperature profile around the huddle computing, the distance calculation between emperor penguins, and the position update of emperor penguins, which will be carried when satisfied  $\lambda_i > \beta$ . The specific steps can be represented as follows:

$$\begin{aligned} T' &= T - \frac{t - \text{Max}_{\text{iteration}}}{\text{Max}_{\text{iteration}}}, \\ T &= \begin{cases} 0, & \text{if } R \geq 0.5, \\ 1, & \text{if } R < 0.5, \end{cases} \end{aligned} \quad (27)$$

where  $T'$  represents the temperature profile around the huddle,  $T$  is the time for finding best optimal solution, and  $R$  is a random variable which lies in the range of  $[0, 1]$ .

$$\mathbf{D}_{\text{ep}}^t = S_{\text{ep}}(A^t) \cdot \mathbf{x}_{\text{best}}^t - B^t \cdot \mathbf{x}_i^t, \quad (28)$$

where  $\mathbf{D}_{\text{ep}}^t$  denotes the distance between the emperor penguin and the optimal solution,  $\mathbf{x}_{\text{best}}^t$  represents the current optimal solution found in emperor penguin population space in the  $t$ th generation,  $S_{\text{ep}}$  represents the social forces of the emperor penguin that is responsible for convergence towards the optimal solution,  $A^t$  and  $B^t$  are used to avoid the collision between neighboring emperor penguins, and  $B^t$  is a random variable which lies in the range of  $[0, 1]$ .  $A^t$  can be computed as follows:

$$\begin{aligned} A^t &= (M \times (T' + P_{\text{grid}}^t(\text{Accuracy})) \times \text{rand}()) - T', \\ P_{\text{grid}}^t(\text{Accuracy}) &= |\mathbf{x}_{\text{best}}^t - \mathbf{x}_i^t|, \end{aligned} \quad (29)$$

where  $M$  is the movement parameter which holds a gap between emperor penguins for collision avoidance and  $P_{\text{grid}}^t(\text{Accuracy})$  defines the absolute difference by comparing the difference between emperor penguins.  $S_{\text{ep}}(A^t)$  in equation (28) is computed as follows:

$$S_{\text{ep}}(A^t) = \sqrt{\varepsilon \cdot e^{-t/\rho} - e^{-t}}, \quad (30)$$

where  $e$  represents the base of natural logarithm.  $\varepsilon$  and  $\rho$  are two control parameters for better exploration and exploitation, which is in the range of  $[1.5, 2]$  and  $[2, 3]$ . Ultimately, the position of emperor penguin is updated as follows:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_{\text{best}}^t - A^t \cdot \mathbf{D}_{\text{ep}}^t. \quad (31)$$

The algorithm procedure of CEPO is sketched in Algorithm 1.

**4.2. CEPO for Parameter Optimization of SVM.** During the iterative searching process, each emperor penguin adjusts the optimal position by comparing the value of the objective function, and the optimal position  $\mathbf{x}_i$  is considered to be the parameter pair  $(C$  and  $\gamma)$  for repeated training of SVM and update of each emperor penguin. Figure 6 shows a schematic diagram of SVM based on CEPO.

## 5. The Experiments

**5.1. Face Databases.** In the following experiments, to test our proposed model, we select four face databases that are used commonly and internationally, i.e., YALE (<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>),

Algorithm: CEPO  
**Input:** the emperor penguin population  $\mathbf{x}_i (i = 1, 2, \dots, m)$   
**Output:** the optimal position of emperor penguin

- (1) **Procedure** CEPO
- (2) Initialize the size of population,  $\text{Max}_{\text{iteration}}$ , the parameters of EPO and CA
- (3) Initialize the situational knowledge and normative knowledge of the belief space
- (4) Initialize the population space of Eps
- (5) Compute the fitness of each EP
- (6) Arrange the fitness value of each EP
- (7) Initialize the belief space by acceptance proportion 20% and **AdjustCulture**
- (8) Initialize the belief space
- (9) **While** ( $t < \text{Max}_{\text{iteration}}$ ) **do**
- (10) Calculate  $T$  and  $T'$  using equations. (9) and (10)
- (11) **For**  $i \leftarrow 1$  to  $m$  **do**
- (12)  $\lambda_i \leftarrow \text{Rand}()$
- (13) **For**  $j \leftarrow 1$  to  $D$  **do**
- (14) **If** ( $\lambda_i \leq \beta$ ) using equation (7)
- (15) Update the position of EPs using equation (8)
- (16) **Else**
- (17) Compute  $A$  and  $B$  using equations (12) and (13)
- (18) Compute  $S_{\text{ep}}(A)$  using equation (14)
- (19) Compute the new position of EPs using equation (15)
- (20) Compute the fitness of current Eps
- (21) Update the better position of EP compared with the previous position
- (22) **End if**
- (23) **End for**
- (24) **End for**
- (25) Amend EPs which cross the boundary
- (26) Arrange the fitness value of each EP
- (27) Update the belief space by acceptance proportion top 20% and **AdjustCulture**
- (28)  $t \leftarrow t + 1$
- (29) **End while**
- (30) Return the optimal position
- (31) **End procedure**
- (32) **Procedure** AdjustCulture
- (33) Update situational knowledge using equation (2)
- (34) **For**  $q \leftarrow 1$  to  $m/5$  **do**
- (35)  $\theta_q \leftarrow \text{Rand}()$
- (36) **If** ( $\theta_q < 0.5$ )
- (37) **For**  $j \leftarrow 1$  to  $D$  **do**
- (38) Update the lower boundary of normative knowledge using equation (3) and (4)
- (39) **Else**
- (40) **For**  $j \leftarrow 1$  to  $D$  **do**
- (41) Update the upper boundary of normative knowledge using equation (5) and (6)
- (42) **End if**
- (43) **End for**
- (44) **End for**
- (45) **End procedure**

ALGORITHM 1: The algorithm procedure of CEPO.

ORL (<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>), UMIST (<http://images.ee.umist.ac.uk/danny/database.html>), and FERET (<http://www.frvt.org/>). Specifically, the YALE database contains 165 images of 15 individuals, including changes in illumination, expression, and posture. The ORL database consists of 400 photographs of 40 human subjects with different expressions and variations at different scales and orientations. The UMIST database includes 564 images of 20 human subjects with different angles and poses. The

FERET database consists of a total of 1400 images for 200 human subjects, corresponding to different poses, expressions, and illumination conditions. To fully verify the validity and rationality of the experiments, configuration of training and testing samples on four face databases is shown in Table 1.

*5.2. State-of-the-Art Algorithms for Comparison.* To validate the performance of CEPO, the eight state-of-the-art meta-heuristics are used for comparison. These are Moth-Flame



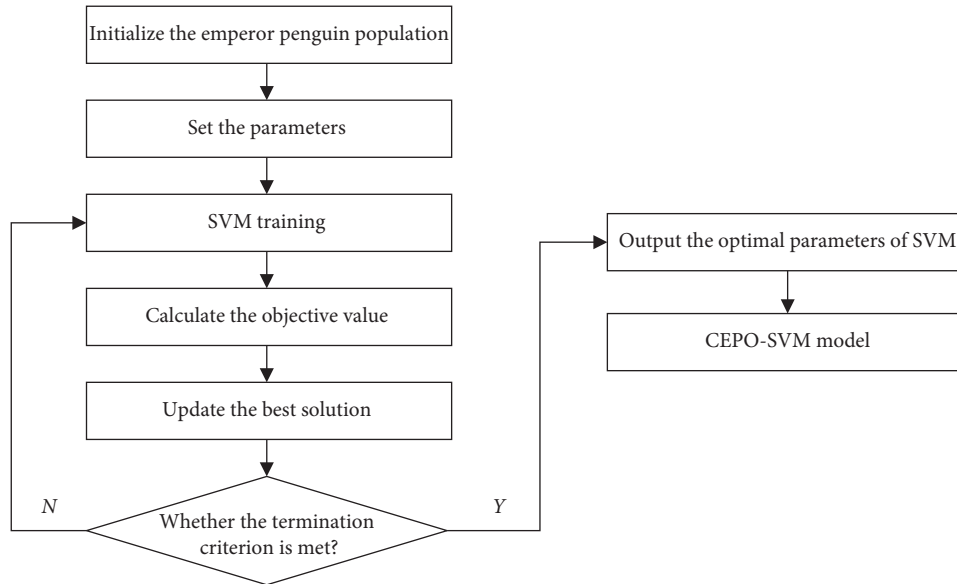


FIGURE 6: Block diagram of SVM based on CEPO.

TABLE 1: Configuration of training and test samples on four face databases.

Databases	Total numbers	Numbers of training samples	Numbers of testing samples
YALE	165	70 percent	the rest
ORL	400	70 percent	the rest
UMIST	564	70 percent	the rest
FERET	200	70 percent	the rest

Optimization (MFO) [9], Grey Wolf Optimizer (GWO) [8], Particle Swarm Optimization (PSO) [6], Genetic Algorithm (GA) [5], Cultural Algorithm (CA) [14], Emperor Penguin Optimizer (EPO) [12], Cultural Firework Algorithm (CFA) [17], and Emperor Penguin and Social Engineering Optimizer (EPSEO) [13].

**5.3. Experimental Setup.** The experimental environment is MATLAB 2018a with LIBSVM on the computer with Intel Core i5 processor and 16 GB of RAM. Table 2 Shows the parameter settings of the proposed CEPO and eight competitor intelligent algorithms, i.e., MFO, GWO, PSO, GA, CA, EPO, CFA, and EPSEO. And the parameter values of these algorithms are set to be recommended in their original paper.

## 6. Simulation Results and Discussion

**6.1. Performance of CEPO Comparison Experiment.** Six benchmark test functions are applied to verify the superiority and applicability of CEPO. Table 3 shows the details of six benchmark test functions including single-peak benchmark functions which are Sphere (F1) and Schwefel 2.22 (F2) and multipeak benchmark functions which are Rastrigin (F3), Griewank (F4), Ackley (F5), and Schaffer (F6). For each benchmark test function, each algorithm carries out thirty independent experiments. The mean and the standard deviation of the thirty independent results are shown in Table 4.

As we can see, the statistical results of CEPO for six benchmark test functions are obviously superior to the other eight comparison algorithms. Under the same optimization conditions, the two evaluation indexes of CEPO are both superior to several or even ten orders of magnitude of other algorithms, which proves that the CEPO has strong optimization performance. The optimal mean index shows that the CEPO maintains a high overall optimization level in thirty independent experiments and implies that CEPO has better single optimization accuracy. And the optimal standard deviation index proves that CEPO has strong optimization stability and robustness. Furthermore, under the same test constraints, CEPO attains the high optimization accuracy in the single-peak functions (F1-F2) and multipeak functions (F3-F6) optimization process, which indicates that the CEPO has strong local mining performance and good convergence accuracy, especially for the four multipeak functions (F3-F6) still maintain a high optimization objective value, which verifies its better global exploration performance and stronger local extreme value avoidance ability. If CEPO takes the target value optimization precision as the evaluation index, the preset precision threshold can be as high as  $1.0E-12$ . Especially, for function F1, F3, F6, the threshold value can even be relaxed to  $1.0E-25$  under the premise of ensuring the success rate of optimization, and the precision can effectively meet the error tolerance range of engineering problems, which further proves that CEPO has better application potential in engineering optimization problems.



TABLE 2: Parameter setting of nine algorithms.

Algorithms	Parameters	Values
Cultural emperor penguin optimizer (CEPO)	Size of population	80
	Control parameter $\varepsilon$	[1.5, 2]
	Control parameter $\rho$	[2, 3]
	Movement parameter $M$	2
	The constant $\eta$	0.06
	Maximum iteration	100
Moth-flame optimization (MFO) [9]	Size of population	80
	Convergence constant	[-1, -2]
	Logarithmic spiral	0.75
	Maximum iteration	100
Grey wolf optimizer (GWO) [8]	Size of population	80
	Control parameter	[0, 2]
	Maximum iteration	100
Particle swarm optimization (PSO) [6]	Size of population	80
	Inertia weight	0.75
	Cognitive and social coeff	1.8, 2
	Maximum iteration	100
Genetic algorithm (GA) [5]	Size of population	80
	Probability of crossover	0.9
	Probability of mutation	0.05
	Maximum iteration	100
Cultural algorithm (CA) [14]	Size of population	80
	The constant	0.06
	Maximum iteration	100
Emperor penguin optimizer (EPO) [12]	Size of population	80
	Control parameter	[1.5, 2]
	Control parameter	[2, 3]
	Movement parameter	2
	Maximum iteration	100
Cultural firework algorithm (CFA) [17]	Size of population	80
	Cost parameter	0.025, 0.2
	The constant	0.3
	Maximum iteration	100
Emperor penguin and social engineering optimizer (EPSEO) [13]	Size of population	80
	Rate of training	0.2
	Rate of spotting an attack	0.05
	Number of attacks	50
	Control parameter	[1.5, 2]
	Control parameter	[2, 3]
	Movement parameter	2
	Maximum iteration	100

### 6.2. Performance of CEPO-SVM Comparison Experiment.

To verify the superiority of the proposed CEPO-SVM classification model for face recognition, CEPO and eight competitor intelligent algorithms are used to iteratively optimize the penalty parameter  $C$  and kernel parameter  $\gamma$  of SVM to assess the performance of CEPO. The average recognition rates of thirty independent runs are shown in Table 5. As can be seen clearly from Table 5, CEPO-SVM achieves the highest accuracy among all nine models. In case of YALE face database, an accuracy of 98.1% is obtained by CEPO-SVM model. For ORL, UMIST, and FERET face database, CEPO-SVM gets an accuracy of 97.8%, 98.4%, and 98.5%, respectively. Furthermore, the convergence rate of CEPO-SVM, PSO-SVM, GA-SVM, EPO-SVM, CA-SVM, MFO-SVM, GWO-SVM, CFA-SVM, and EPSEO-SVM are examined and depicted in Figures 7(a)–7(d). From this

figure, it can be observed that the recognition rate successively increases from iteration 1 to iteration 100 on all four face databases. For YALE database, the accuracy rate has not improved after 14th, 22nd, 31st, 19th, 18th, 21st, 20th, 23rd, and 24th iterations using CEPO-SVM, PSO-SVM, GA-SVM, EPO-SVM, CA-SVM, MFO-SVM, GWO-SVM, CFA-SVM, and EPSEO-SVM, respectively. Similarly, for ORL database, no improvement in accuracy is noticed after 11th, 20th, 22nd, 19th, 15th, 20th, 18th, 21st, and 24th iterations using CEPO-SVM, PSO-SVM, GA-SVM, EPO-SVM, CA-SVM, MFO-SVM, GWO-SVM, CFA-SVM, and EPSEO-SVM, respectively. In case of UMIST database, CEPO-SVM, PSO-SVM, GA-SVM, EPO-SVM, CA-SVM, MFO-SVM, GWO-SVM, CFA-SVM, and EPSEO-SVM converge after 15th, 24th, 25th, 19th, 20th, 26th, 18th, 24th, and 21st iterations, respectively. For FERET database, no rise in accuracy is

TABLE 3: The details of six test benchmark functions.

No.	Function	Formulation	$D$	Search range
F1	Sphere	$\sum_{i=1}^D x_i^2()$	10	$[-100, 100]$
F2	Schwefel 2.22	$\sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i ()$	10	$[-10, 10]$
F3	Rastrigin	$\sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)()$	10	$[-5.12, 5.12]$
F4	Griewank	$\sum_{i=1}^D x_i^2/4000 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1()$	10	$[-600, 600]$
F5	Ackley	$-20 \exp(-0.2 \sqrt{1/n \sum_{i=1}^D x_i^2}) - \exp(1/n \sum_{i=1}^D \cos(2\pi x_i) + 20 + e)()$	10	$[-32, 32]$
F6	Schaffer	$0.5 + (\sin \sqrt{x_1^2 - x_2^2})^2 - 0.5/(1 + 0.001(x_1^2 + x_2^2))^2()$	2	$[-100, 100]$

TABLE 4: The results of benchmark test functions by nine algorithms.

Algorithms		Functions					
		F1	F2	F3	F4	F5	F6
MFO	Mean	5.28E-11	1.69E-06	7.21E-10	8.38E-11	2.67E-12	5.09E-13
	Std	4.19E-12	3.77E-06	4.93E-11	3.41E-11	7.19E-12	4.81E-12
GWO	Mean	7.03E-13	4.43E-05	8.48E-12	9.76E-09	3.83E-13	2.92E-14
	Std	6.92E-13	7.82E-05	3.12E-12	4.28E-09	9.06E-12	6.37E-14
PSO	Mean	6.36E-06	6.14E-04	5.56E-04	2.07E-05	2.98E-07	8.14E-06
	Std	4.23E-05	8.25E-03	1.79E-04	6.23E-05	5.24E-06	9.93E-07
GA	Mean	2.55E-07	2.47E-04	5.32E-05	4.95E-05	1.56E-08	3.38E-07
	Std	3.14E-07	8.05E-04	3.87E-05	1.69E-05	6.06E-08	2.72E-07
EPO	Mean	8.57E-06	9.14E-05	3.52E-07	6.89E-07	5.83E-09	4.02E-08
	Std	4.87E-06	2.43E-05	4.99E-07	4.27E-07	7.18E-09	5.39E-08
CA	Mean	4.13E-12	1.49E-05	4.41E-09	6.65E-12	2.33E-07	7.53E-11
	Std	3.78E-13	1.36E-05	2.23E-10	4.89E-13	2.68E-08	6.21E-09
CEPO	Mean	<b>1.28E-27</b>	<b>4.15E-14</b>	<b>6.54E-26</b>	<b>2.13E-21</b>	<b>2.39E-24</b>	<b>7.74E-29</b>
	Std	<b>4.63E-27</b>	<b>2.39E-14</b>	<b>7.16E-26</b>	<b>1.49E-21</b>	<b>4.76E-24</b>	<b>3.22E-29</b>
CFA	Mean	4.23E-23	3.49E-10	2.37E-19	8.01E-17	5.41E-20	1.19E-24
	Std	1.89E-22	8.83E-10	9.44E-19	2.41E-17	2.93E-19	4.32E-23
EPSEO	Mean	8.67E-24	7.25E-11	3.91E-20	6.61E-18	4.25E-21	8.81E-22
	Std	5.42E-24	9.41E-11	8.23E-20	3.93E-18	7.89E-21	6.67E-21

TABLE 5: Average recognition rates (%) obtained by nine models on four databases.

Average recognition rate (%)	Database			
	YALE	ORL	UMIST	FERET
MFO-SVM	96.6	96.8	98.0	97.7
GWO-SVM	97.1	97.0	96.5	98.2
PSO-SVM	96.9	96.5	97.1	97.2
GA-SVM	97.4	97.5	97.6	97.4
EPO-SVM	97.6	97.2	97.9	97.6
CA-SVM	97.8	97.4	97.7	97.4
CEPO-SVM	<b>98.1</b>	<b>97.8</b>	<b>98.4</b>	<b>98.5</b>
CFA-SVM	97.9	97.7	98.3	98.3
EPSEO-SVM	97.7	97.6	98.1	98.0

seen after 16th, 22nd, 24th, 19th, 18th, 20th, 18th, 21st, and 25th iterations using CEPO-SVM, PSO-SVM, GA-SVM, EPO-SVM, CA-SVM, MFO-SVM, GWO-SVM, CFA-SVM, and EPSEO-SVM, respectively. Obviously, CEPO-SVM converges firstly among nine proposed models, which verifies that EPO with the help of the embedded framework provided by CA could remarkably enhance convergence rate. Therefore, the parameters of optimization by CEPO are superior to make the classification performance of SVM better.

**6.3. Stability of CEPO-SVM Comparison Experiment.** For all nine models, the stability of the models is measured by the mean and standard deviation of penalty parameter  $C$  and kernel parameter  $\gamma$ . In this experiment, each model runs thirty experiments. Tables 6–9 show the comparison of mean and standard deviation of parameters among nine models on four face databases. As we can see from Tables 6–9, the mean of penalty parameter  $C$  in CEPO-SVM is less than that of other models on four face databases, which could reduce the possibility of overlearning of SVM. And CEPO-SVM provides the lowest standard deviation of penalty parameter  $C$  among all nine models on all four databases, which verifies the stability of CEPO-SVM is better. For all four databases, the mean and standard deviation of kernel parameter  $\gamma$  are close among the nine models. However, it can still be seen that CEPO-SVM has the minimum standard deviation of kernel parameter  $\gamma$ . Obviously, CEPO is more stable to optimize the parameters of SVM for face recognition.

**6.4. Robustness of CEPO-SVM Comparison Experiment.** To prove the robustness of CEPO-SVM, salt and pepper noise is added to each image of the original four databases. The noise databases generated are used for face recognition

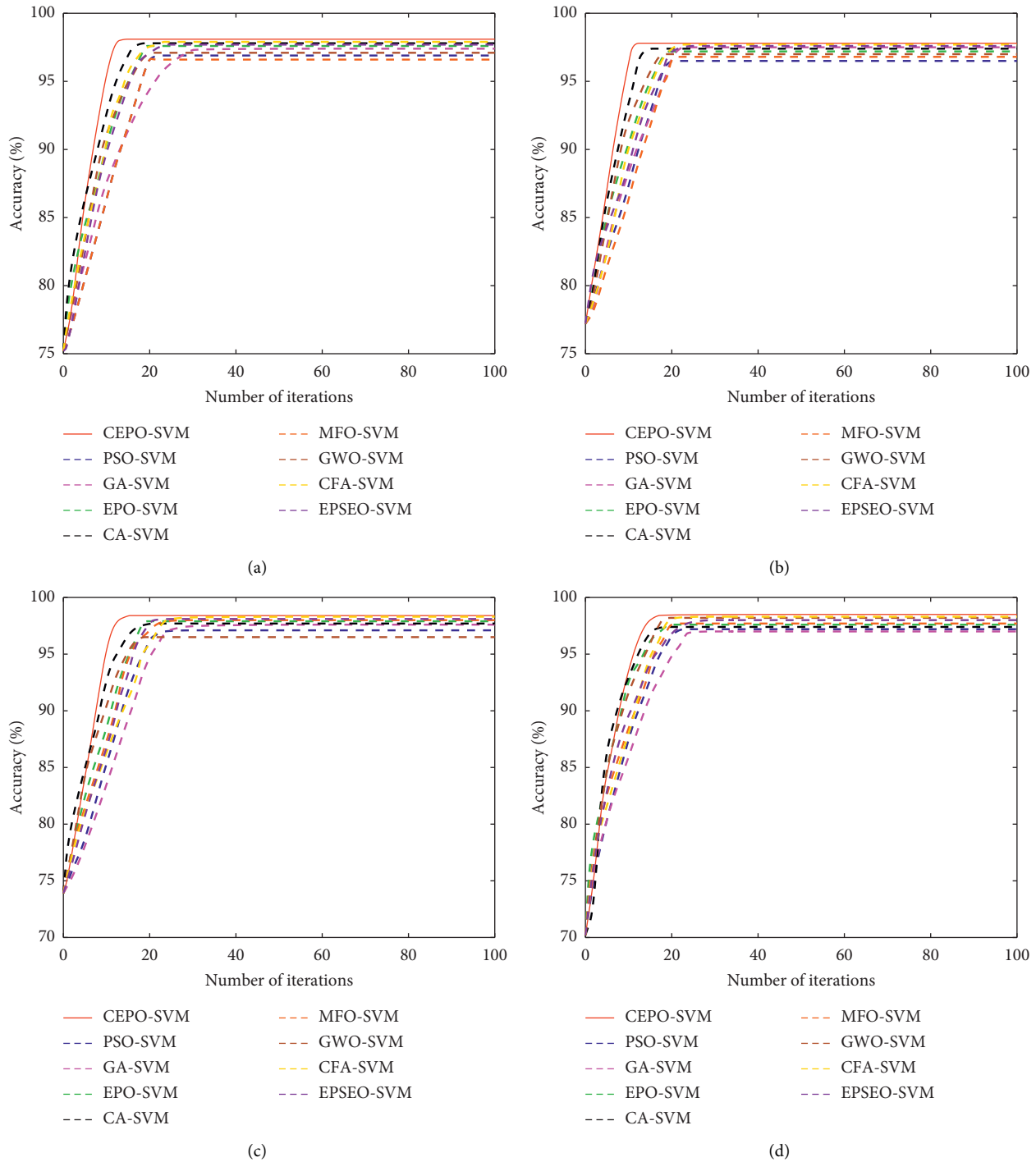


FIGURE 7: Number of iterations versus accuracy on four databases. (a) YALE database. (b) ORL database. (c) UMIST database. (d) FERET database.

by CEPO-SVM, PSO-SVM, GA-SVM, EPO-SVM, CA-SVM, MFO-SVM, GWO-SVM, CFA-SVM, and EPSEO-SVM. For each noise databases, we selected 70% of noise samples as the training samples and the rest 30% of noise samples as the testing samples. Salt and pepper noise is a random black or white pixels in an image. Four different levels of noise image on YALE database which are 0%, 5%, 10%, and 15% are shown in Figure 8. From Figure 8, we can

see that, as the noise level improves, the face image becomes more blurred. Figure 9 shows the detailed recognition rates among different models on four databases. As we can see from Figure 9, CEPO-SVM obtains the highest recognition rate at all four noise levels on among four databases in comparison with other eight models. Although the face image is seriously polluted by salt and pepper noise which is up to 15%, the recognition rate of CEPO-SVM can still attain

TABLE 6: Comparison of mean and standard deviation of parameter on YALE face database among nine optimization algorithms.

Algorithms	Penalty parameter C		Kernel parameter $\gamma$	
	Mean	Std	Mean	Std
MFO	18.3721	4.4921	0.0633	0.1109
GWO	16.8983	4.0748	0.0518	0.0978
PSO	25.2319	7.4295	0.1292	0.1753
GA	28.3684	8.7016	0.1449	0.2387
CA	19.1670	4.8603	0.0672	0.1283
EPO	22.6792	5.7065	0.0812	0.1456
CEPO	13.2441	2.4982	0.0100	0.0112
CFA	15.4039	3.0409	0.0349	0.0778
EPSEO	15.6382	2.7636	0.0426	0.0692

TABLE 7: Comparison of mean and standard deviation of parameter on ORL face database among nine optimization algorithms.

Algorithms	Penalty parameter C		Kernel parameter $\gamma$	
	Mean	Std	Mean	Std
MFO	18.3921	4.7817	0.0965	0.1143
GWO	19.0418	5.1526	0.0883	0.1277
PSO	26.1129	6.4990	0.1620	0.2563
GA	27.5823	8.0652	0.1953	0.2497
CA	19.9301	3.9721	0.0827	0.1683
EPO	22.2801	6.2076	0.1198	0.1762
CEPO	14.3873	2.0187	0.0121	0.0098
CFA	16.2265	4.1552	0.0573	0.0829
EPSEO	15.7192	3.5228	0.0525	0.0701

TABLE 8: Comparison of mean and standard deviation of parameter on UMIST face database among nine optimization algorithms.

Algorithms	Penalty parameter C		Kernel parameter $\gamma$	
	Mean	Std	Mean	Std
MFO	17.2782	4.2048	0.0804	0.1449
GWO	16.9556	3.7649	0.0773	0.1283
PSO	24.9031	7.1991	0.1439	0.1958
GA	25.3415	8.2315	0.1557	0.2261
CA	18.8737	4.0103	0.0762	0.1813
EPO	22.0217	5.9525	0.0973	0.1502
CEPO	12.9109	2.2081	0.0114	0.0101
CFA	15.2271	3.4594	0.0638	0.0822
EPSEO	15.9964	3.0182	0.0514	0.0769

TABLE 9: Comparison of mean and standard deviation of parameter on FERET face database among nine optimization algorithms.

Algorithms	Penalty parameter C		Kernel parameter $\gamma$	
	Mean	Std	Mean	Std
MFO	18.1928	4.1892	0.0619	0.1023
GWO	17.3209	3.9713	0.0557	0.0921
PSO	25.5827	6.9143	0.1401	0.1882
GA	26.6929	8.4939	0.1379	0.2273
CA	15.3462	5.0541	0.0924	0.1419
EPO	21.2679	5.6314	0.0846	0.1364
CEPO	11.1811	2.1192	0.0126	0.0087
CFA	15.0647	3.7672	0.0489	0.0748
EPSEO	14.4689	3.4295	0.0377	0.0661

82.5% on YALE database, 83.5% on ORL database, 81.1% on UMIST database, and 82.1% on FERET database, which far exceed that of other eight models. Besides, with the increase of noise level, the recognition rate of CEPO-SVM on four

databases decreases more slowly than that of other eight models. Obviously, the results show that CEPO-SVM can not only attain good noise-free face recognition accuracy but also achieve good accuracy under noise conditions, which

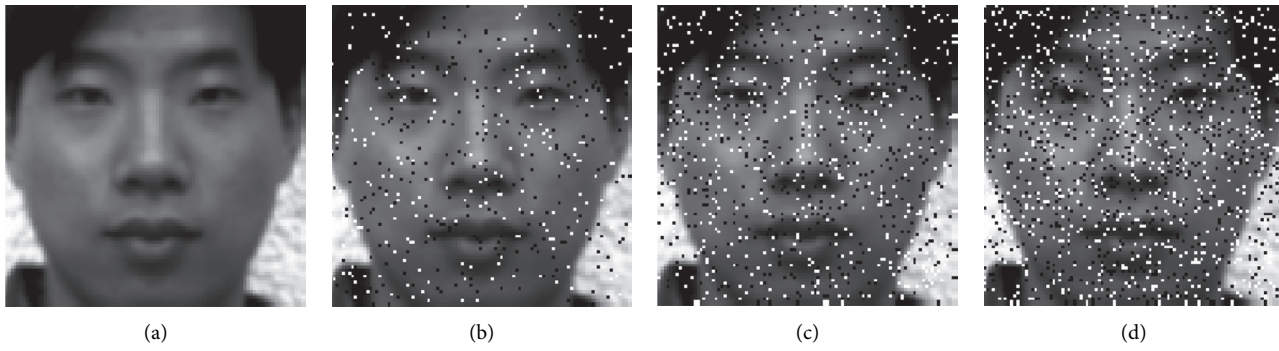


FIGURE 8: Four different levels of noise image on YALE database: (a) 0%. (b) 5%. (c) 10%. (d) 15%.

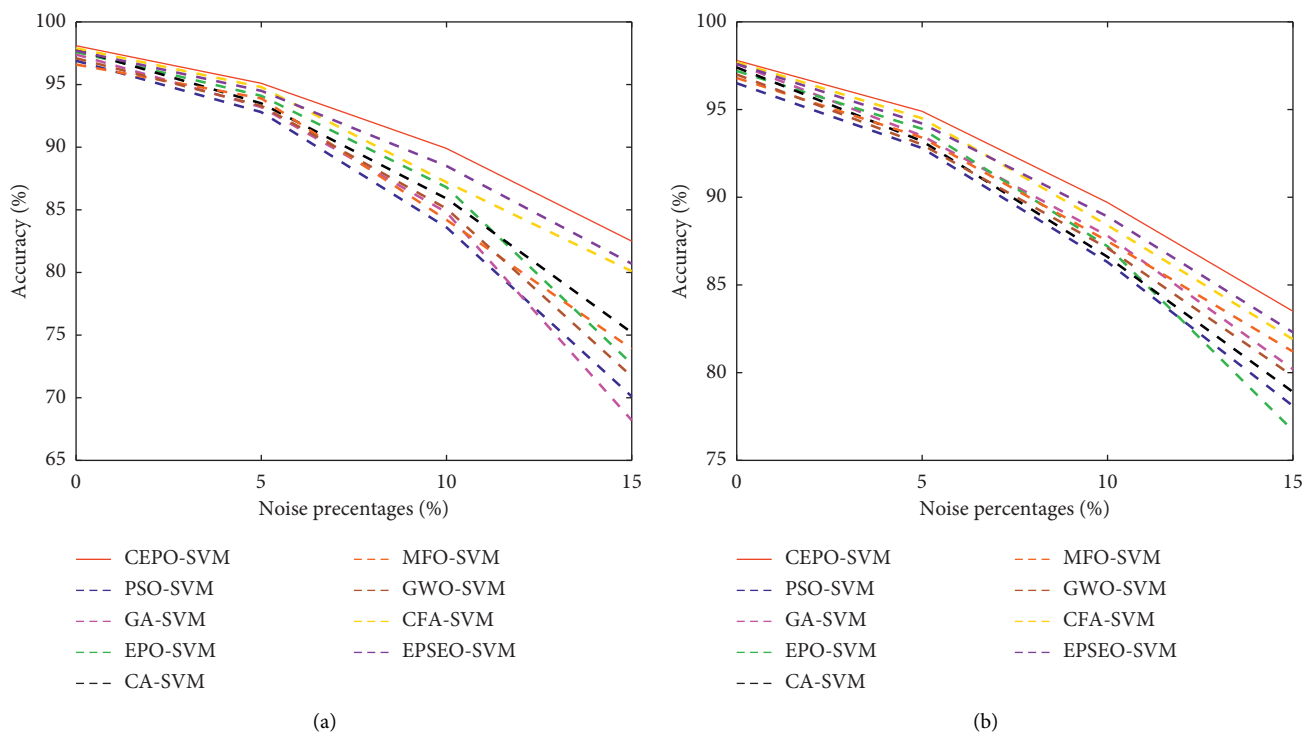


FIGURE 9: Continued.

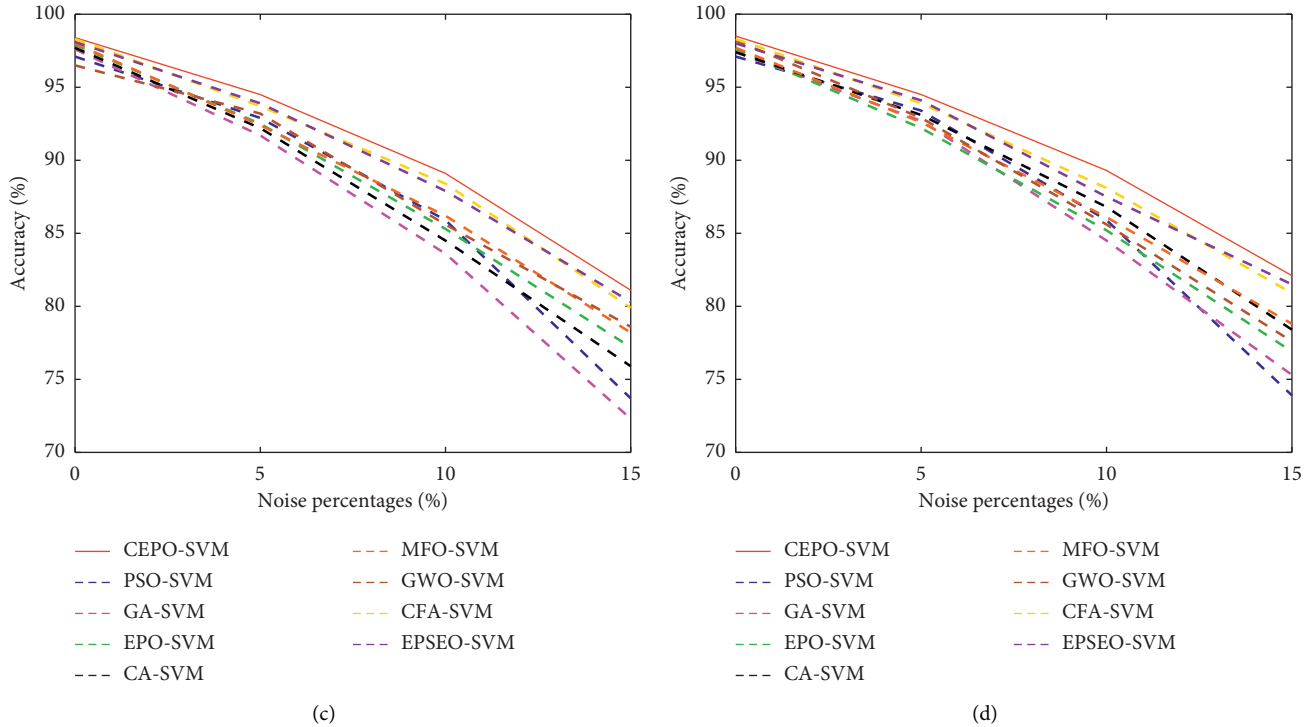


FIGURE 9: Noise percentages versus accuracy on four databases: (a) YALE database. (b) ORL database. (c) UMIST database. (d) FERET database.

TABLE 10: Average training time (s) obtained by nine models on four databases.

Training time (s)	Databases			
	YALE	ORL	UMIST	FERET
MFO-SVM	47.93	61.54	70.95	46.03
GWO-SVM	45.06	58.81	72.31	50.54
PSO-SVM	43.95	54.12	65.69	48.91
GA-SVM	69.71	82.73	98.03	79.42
EPO-SVM	40.68	47.49	68.55	42.87
CA-SVM	46.33	56.87	77.41	53.24
CEPO-SVM	59.76	79.23	95.76	67.34
CFA-SVM	78.25	81.96	112.87	84.59
EPSEO-SVM	82.43	83.15	107.43	86.77

verifies that robustness of CEPO-SVM is superior to the other eight models.

**6.5. Run Time Analysis.** The average training time of thirty independent runs by the proposed CEPO-SVM and eight competitor intelligent algorithm-based SVMs are shown in Table 10. As we can see from Table 10, in case of all four databases, the average training time by the proposed CEPO-SVM is more than that of six models with single algorithms, i.e., MFO-SVM, GWO-SVM, PSO-SVM, GA-SVM, EPO-SVM, and CA-SVM. This may be because hybrid of two algorithms increases the training time. Although CEPO-SVM takes more time than these six models with single algorithms, it can be considered better than these six models

because of its high classification performance. However, for all four databases, the average training time by the proposed CEPO-SVM is less than that of two models with hybrid algorithms, i.e., CFA-SVM and EPSEO-SVM. Obviously, CEPO-SVM has better performance than these two models with hybrid algorithms.

**6.6. Limitation Analysis.** Like other hybrid algorithms, although CEPO provides superior performance in face recognition, due to large amount of computation, the proposed CEPO will cost more time for iterations compared with single algorithms. Besides, when CEPO is applied to solve super-dimensional optimization problems, the performance may be reduced.

## 7. Conclusions

In this paper, with respective advantages of EPO and CA, we propose a hybrid metaheuristic algorithm named CEPO with the help of the embedded framework provided by CA to promote each other and overcome each other's self-defects. The proposed algorithm has been experimented on six benchmark test functions compared with other eight state-of-the-art algorithms. Furthermore, we apply this hybrid metaheuristic algorithm to automatic learning of the network parameters of SVM to solve the classification problem for face recognition. The proposed classifier model has been tested on four face databases that are used commonly and internationally by comparing with models with other eight state-of-the-art algorithms. The experimental results show that the proposed model has better performance in terms of accuracy, convergence rate, stability, robustness, and run time.

We provide future works in different directions. First, the proposed model has only been experimented on four face databases; however, other new face databases should be tested to extend the current work. Second, the proposed hybrid algorithm should be applied to other classifiers for face recognition. Lastly, the proposed hybrid algorithm should be applied to solve other optimization problems, such as digital filters design, cognitive radio spectrum allocation, and image segmentation.

## List of Definitions of Partial Parameters

$x_i$ :	The position of the $i$ th emperor penguin in the $D$ -dimensional search space
$s^t$ :	Situational knowledge component
$N_j^t$ :	Normative knowledge in the $j$ th dimension and in the $t$ th generation
$I_j^t$ :	The interval of normative knowledge in the $j$ th dimension
$l_j^t$ :	The lower boundary in the $j$ th dimension and in the $t$ th generation
$L_j^t$ :	The objective value of the lower boundary $l_j^t$ of the $j$ th parameter
$u_j^t$ :	The higher boundary in the $j$ th dimension and in the $t$ th generation
$U_j^t$ :	The objective value of the upper boundary $u_j^t$ of the $j$ th parameter
$D_{ep}^t$ :	The distance between the emperor penguin and the optimal solution
$M$ :	The movement parameter
$c$ :	The scalar
$C$ :	The penalty parameter
$\xi_h$ :	The relaxing variables
$a_h$ :	The training samples
$\alpha_h$ :	The Lagrange multipliers
$r$ :	The scale of Gabor filter
$o$ :	The orientation of Gabor filter
$Q^{(\delta)}$ :	The concatenated column vector
$\sum Q^{(\delta)}$ :	The covariance matrix of the augmented feature vector $Q^{(\delta)}$
$\Omega$ :	The orthogonal eigenvector matrix
$\Lambda$ :	The diagonal eigenvalue matrix

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant no. 61571149), Heilongjiang Province Key Laboratory Fund of High Accuracy Satellite Navigation and Marine Application Laboratory (Grant no. HKL-2020-Y01), and Initiation Fund for Postdoctoral Research in Heilongjiang Province (Grant no. LBH-Q19098).

## References

- [1] S. Jiang, H. Mao, Z. Ding, and Y. Fu, "Deep decision tree transfer boosting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 2, pp. 383–395, 2020.
- [2] Y. Xia and J. Wang, "A bi-projection neural network for solving constrained quadratic optimization problems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 214–224, 2016.
- [3] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2018.
- [4] A. Mathur and G. M. Foody, "Multiclass and binary SVM classification: implications for training and classification users," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 241–245, 2008.
- [5] J. Ruan, H. Jiang, X. Li, Y. Shi, F. T. S. Chan, and W. Rao, "A granular GA-SVM predictor for big data in agricultural cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6510–6521, 2019.
- [6] F. Melgani and Y. Bazi, "Classification of electrocardiogram signals with support vector machines and particle swarm optimization," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 5, pp. 667–677, 2008.
- [7] L. Wanling, W. Zhensheng, and S. Xiangjun, "Parameters optimization of support vector machine based on simulated annealing and improved QPSO," in *Proceedings of the 2012 International Conference on Industrial Control and Electronics Engineering*, pp. 1089–1091, Xi'an, China, August 2012.
- [8] S. Mirjalili, S. M. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [9] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-based Systems*, vol. 89, pp. 228–249, 2015.
- [10] S. Kaur, K. Lalit, A. L. S. Awasthi et al., "Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, Article ID 103541, 2020.
- [11] M. Dehghani, Z. Montazeri, O. P. Malik et al., "A new methodology called dice game optimizer for capacitor placement in distribution systems," *Electrical Engineering & Electromechanics*, vol. 1, pp. 61–64, 2020.



- [12] G. Dhiman and V. Kumar, "Emperor penguin optimizer: a bio-inspired algorithm for engineering problems," *Knowledge-Based Systems*, vol. 159, pp. 20–50, 2018.
- [13] S. Kumar Baliarsingh, W. Ding, S. Vipsita et al., "A memetic algorithm using emperor penguin and social engineering optimization for medical data classification," *Applied Soft Computing*, vol. 85, 2019.
- [14] Reynolds and G. Robert, "An introduction to cultural algorithms," in *Proceedings of the Third Annual Conference on Evolutionary Programming*, pp. 131–139, San Diego, CA, USA, January 1994.
- [15] C. Lin, C. Chen, and C. Lin, "A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 1, pp. 55–68, 2009.
- [16] H. Gao, C. Xu, and C. Li, "Quantum-inspired cultural bacterial foraging algorithm for direction finding of impulse noise," *International Journal of Innovative Computing and Applications*, vol. 6, no. 1, pp. 44–54, 2014.
- [17] H. Gao and M. Diao, "Cultural firework algorithm and its application for digital filters design," *International Journal of Modelling, Identification and Control*, vol. 14, no. 4, pp. 324–331, 2011.
- [18] C. Liu, "Gabor-based kernel PCA with fractional power polynomial models for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, pp. 572–581, 2004.
- [19] J. Ilonen, J.-K. Kämäräinen, and H. Kälviäinen, "Efficient computation of gabor features," Research Report 100, Lappeenranta University of Technology, Department of Information Technology, Lappeenranta, Finland, 2005.
- [20] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 11, no. 4, pp. 467–476, 2002.
- [21] M. D. Farrell and R. M. Mersereau, "On the impact of PCA dimension reduction for hyperspectral detection of difficult targets," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 192–195, 2005.
- [22] B. Fei and J. Jinbai Liu, "Binary tree of SVM: a new fast multiclass training and classification algorithm," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 696–704, 2006.