

A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition

Kamlesh Mistry, Li Zhang, *Member, IEEE*, Siew Chin Neoh, Chee Peng Lim, and Ben Fielding

Abstract—This paper proposes a facial expression recognition system using evolutionary particle swarm optimization (PSO)-based feature optimization. The system first employs modified local binary patterns, which conduct horizontal and vertical neighborhood pixel comparison, to generate a discriminative initial facial representation. Then, a PSO variant embedded with the concept of a micro genetic algorithm (mGA), called mGA-embedded PSO, is proposed to perform feature optimization. It incorporates a nonreplaceable memory, a small-population secondary swarm, a new velocity updating strategy, a subdimension-based in-depth local facial feature search, and a cooperation of local exploitation and global exploration search mechanism to mitigate the premature convergence problem of conventional PSO. Multiple classifiers are used for recognizing seven facial expressions. Based on a comprehensive study using within- and cross-domain images from the extended Cohn Kanade and MMI benchmark databases, respectively, the empirical results indicate that our proposed system outperforms other state-of-the-art PSO variants, conventional PSO, classical GA, and other related facial expression recognition models reported in the literature by a significant margin.

Index Terms—Ensemble classifier, facial expression recognition, feature selection, particle swarm optimization (PSO).

I. INTRODUCTION

FACIAL emotion recognition has opened up a new era for human–computer interaction, and has provided benefits to a wide range of computer vision applications, such as healthcare, surveillance, event detection, personalized learning, and robotics [1]–[7]. Robust emotion classification relies heavily on effective facial representation. However, it is still a challenging task for identifying significant discriminative facial features that could represent the characteristics of each

Manuscript received October 10, 2015; revised February 12, 2016; accepted March 30, 2016. Date of publication April 21, 2016; date of current version May 15, 2017. This work was supported in part by the European Union through the (Erasmus Mundus) Centre of Excellence for Learning, Innovation, Networking and Knowledge Project under Grant 2645, and in part by the Higher Education Innovation Fund. This paper was recommended by Associate Editor D. Wang.

K. Mistry, L. Zhang, and B. Fielding are with the Department of Computer Science and Digital Technologies, Faculty of Engineering and Environment, Northumbria University, Newcastle Upon Tyne NE1 8ST, U.K. (e-mail: kamlesh.mistry@northumbria.ac.uk; li.zhang@northumbria.ac.uk; ben.fielding@northumbria.ac.uk).

S. C. Neoh is with the Faculty of Engineering, Technology, and Built Environment, UCSI University, Kuala Lumpur 56000, Malaysia (e-mail: u_jane80@yahoo.co.uk).

C. P. Lim is with the Centre for Intelligent Systems Research, Deakin University, Geelong, VIC 3216, Australia (e-mail: chee.lim@deakin.edu.au).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCYB.2016.2549639

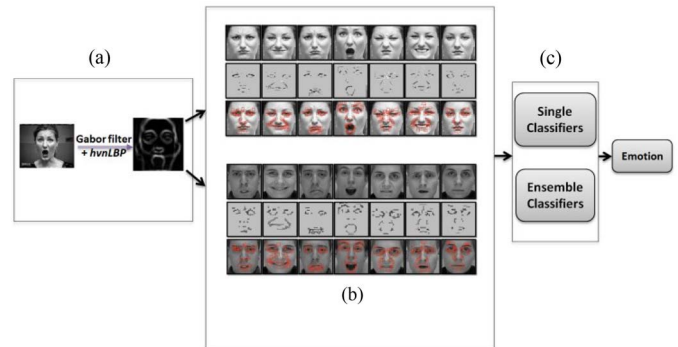


Fig. 1. System architecture. (a) Feature extraction. (b) Feature optimization using the proposed PSO variant. (c) Classification.

emotion because of the subtlety and variability of facial expressions.

This paper aims to deal with such challenges to produce effective and optimized discriminative facial representations to benefit real-time facial expression recognition. In comparison with other feature selection methods, evolutionary computational (EC) algorithms show powerful global search capabilities, and have been widely accepted as efficient techniques for feature selection [8]. Among different EC algorithms, the particle swarm optimization (PSO) algorithm is motivated by the flocking behaviors of birds, and has been extensively used for feature optimization with the benefits of a low-computational cost and a fast convergence speed. However, conventional PSO tends to converge prematurely and, therefore, be trapped in local optima [8]. As a result, in this paper, a PSO variant embedded with the concept of a micro genetic algorithm (mGA) is proposed. Known as mGA-embedded PSO, the proposed algorithm incorporates a nonreplaceable memory, a small-population secondary swarm, a new velocity updating strategy, a subdimension-based regional facial feature search strategy, and a cooperation of local exploitation and global exploration search strategy to overcome both premature convergence and local optimum problems encountered by conventional PSO.

The proposed facial emotion recognition system consists of three steps: 1) feature extraction; 2) feature optimization; and 3) emotion recognition. Fig. 1 illustrates the system architecture. First of all, we use modified local binary patterns (LBPs), i.e., horizontal and vertical neighborhood comparison LBP, to extract the initial facial representation. Then, the proposed mGA-embedded PSO algorithm

is used to identify the most discriminative and significant features for differentiating distinct facial expressions. Diverse classifiers (e.g., single and ensemble models) are applied to recognize seven emotions: 1) happiness; 2) sadness; 3) anger; 4) fear; 5) surprise; 6) disgust; and 7) neutral. The system is evaluated with two facial expression databases, i.e., the extended Cohn Kanade (CK+) [9] and MMI [10]. State-of-the-art PSO variants, conventional PSO, and classical genetic algorithm (GA) are used to compare with the proposed mGA-embedded PSO algorithm in feature optimization. The empirical results indicate that the proposed system outperforms state-of-the-art optimization methods and other related facial expression recognition research reported in the literature by a significant margin. The main contributions of this paper are summarized as follows.

- 1) A modified LBP operator that conducts horizontal and vertical neighborhood pixel comparison is proposed, in order to overcome the drawbacks of original LBP by retrieving the missing contrast information embedded in the neighborhood to generate the initial discriminative facial representation.
- 2) A novel mGA-embedded PSO algorithm is proposed for feature optimization, in order to mitigate the premature convergence and local optimum problems of conventional PSO. It provides great flexibility to allow the feature selection process to not only separate facial features into specific areas for in-depth local search but also combine facial features for overall global search.
- 3) The proposed algorithm includes a new velocity updating strategy by employing the personal average experience to generate the individual best, *pbest*, and Gaussian mutation to produce the global best, *gbest*, in order to increase swarm diversity.
- 4) The proposed algorithm also applies the diversity maintenance strategy of mGA to keep the original swarm in a nonreplaceable memory [11], which remains intact during the lifetime of the algorithm, in order to reduce the probability of premature convergence.
- 5) In order to speed up evolution for convergence, the small population size concept of mGA is used to generate a secondary swarm with five particles. The secondary swarm consists of the swarm leader and four follower particles from the nonreplaceable memory with the lowest or highest correlation with the leader to increase local exploitation and global exploration. These local and global search mechanisms work in a collaborative manner to guide the search toward global optima. A subdimension-based search strategy is also conducted, in order to identify optimal features for each facial region.
- 6) Our proposed system is evaluated with CK+ and MMI databases. It outperforms state-of-the-art LBP and PSO variants, and other facial expression recognition methods reported in the literature significantly.

This paper is organized as follows. We discuss the related work in Section II. Section III introduces the proposed LBP variant for feature extraction and the mGA-embedded

PSO algorithm for feature optimization. A comprehensive evaluation study is presented in Section IV. The conclusions and suggestions for future work are presented in Section V.

II. RELATED WORK

In this section, we discuss state-of-the-art research on texture extraction, PSO-based feature optimization and facial expression recognition.

A. Feature Extraction Techniques

A number of LBP variants are available to increase its robustness and discriminative power. As an example, dominant LBP (DLBP) is able to retrieve the most frequently occurred patterns of LBP to improve its texture descriptive capability. According to [12], uniform patterns in LBP can lead to a loss of information with respect to complex shapes despite their effectiveness in capturing fundamental patterns in an input image. Therefore, instead of purely using uniform patterns, DLBP calculates the occurrence frequencies of all the patterns extracted by LBP. These patterns are subsequently ranked based on the occurrence frequencies to enable the extraction of dominating patterns in texture images.

Completed LBP (CLBP) [13] employs three key components, i.e., CLBP-center, CLBP-sign, and CLBP-magnitude, to extract the image's local gray level and the sign and magnitude features of local difference, respectively. The final CLBP histogram is formed by fusing these three components. In comparison with LBP which only considers the sign component, CLBP takes the magnitude component and intensity of the central pixel into account for formulating the additional discriminative power. It produces superior texture classification accuracy than those from other state-of-the-art LBP algorithms. Center-symmetric LBP (CS-LBP) [14] aims to solve the lengthy histogram problem of LBP. In order to produce more compact binary patterns, CS-LBP purely employs the center-symmetric pairs of pixels for comparison. Therefore, compared with LBP, it enables a significant reduction in dimensionality while capturing better gradient information.

Local derivative pattern (LDP) [15] is a high-order local pattern descriptor, which encodes directional pattern features based on local derivative variations. In comparison with LBP (as a nondirectional first-order local pattern operator), LDP encodes more detailed discriminative information by calculating higher-order directional derivatives. It effectively extracts spatial relationships in a local region. LBP, on the other hand, only defines the relationships between the central point and its neighbors. In LDP, the first-order derivatives from four different directions, i.e., 0° , 45° , 90° , and 135° , are calculated. A set of 16 spatial relationship templates is defined for derivative direction comparisons with each template assigned a value of "0" or "1" based on whether it is a "monotonically increasing/decreasing" or a "turning point" pattern. The four first-order derivatives are then concatenated to form the second-order LDP. The n th-order LDP, therefore, encodes the $(n-1)$ th-order derivative direction variations. Higher-order LDP possesses superior capabilities in providing detailed discriminative features, but at the cost of an increasing level

of noise. Another novel texture descriptor, local phase quantization (LPQ) [16] deals with image blurring based on quantized phase of the discrete Fourier transform computed in local neighborhoods. The LPQ operator is tolerant to centrally symmetric blur including motion, out of focus, and atmospheric turbulence blur. It is developed based on the blur invariance characteristics of the Fourier phase spectrum. In LPQ, four Fourier coefficients are used to sample the phase component of the frequency at four discrete points for each individual pixel position. The resulting vector is then further processed by separating each value into the real and imaginary parts to generate an 8-D vector. Decorrelation is also conducted using a whitening transform to ensure statistical independence of the samples. A simple scalar quantizer is subsequently used to obtain the 8-bit binary code for each pixel position representing a blur insensitive, Fourier phase information of the pixel location. These codes are then converted into a histogram for image classification. Overall, LPQ is superior to LBP and Gabor filter bank-based methods in dealing with image blurring.

B. PSO Variants and Feature Selection Techniques

There are many PSO variants in the literature to overcome the local optimum problem of conventional PSO [17]. Mahmoodabadi *et al.* [18] proposed a PSO variant known as high exploration PSO (HEPSO). In HEPSO, PSO is integrated with a multicrossover mechanism of the GA and the food source finding operator of bee colony optimization for updating the particle velocity and position, respectively. Evaluated with well-known benchmark functions, HEPSO has shown superiority over other PSO variants. Li *et al.* [19] proposed another hybrid PSO algorithm with the integration of fuzzy reasoning and a weighted particle to guide the swarm. The weighted particle is used to adjust the search direction, whereas other parameters such as the attraction factor and inertia weight controlled by fuzzy reasoning are used to adjust local exploitation and global exploration to guide the search. The proposed model was tested with ten benchmark functions, and was further applied to nonlinear neural network (NN)-based modeling. Jordehi [20] proposed an enhanced leader PSO model known as ELPSO. ELPSO employs Gaussian, Cauchy, opposition-based, and differential evolution (DE)-based mutation to increase the diversity of the swarm leader.

PSO variants have also been extensively used for feature selection. Zhang *et al.* [21] extended the conventional bare bones PSO (BPSO) to feature selection problems with binary variables. Known as binary BPSO, a reinforced memory strategy is used to update pbest of each particle to retain swarm diversity, whereas a uniform combination technique is applied to increase local and global search capabilities of the algorithm. In binary BPSO, the influence of the uniform combination is strengthened as the occurrence of stagnated iterations of the algorithm increases. Wang *et al.* [22] proposed a parameter-free Gaussian bare-bones DE algorithm (GBDE). GBDE employs Gaussian distribution as the mutation strategy and a self-adaptive scheme for crossover probability

adjusting. GBDE has been further enhanced by integrating with DE/best/1 (another mutation strategy) to achieve a fast convergence rate. The enhanced model outperforms several DE variants and bare-bones algorithms. Chuang *et al.* [23] proposed chaotic binary PSO (CBPSO) for feature selection. It combines two chaotic maps, i.e., logistic and tent maps, with BPSO to determine the inertia weight, in order to overcome the local optima problem. The results indicate that CBPSO in combination with a tent map is able to produce the best performance.

Xue *et al.* [8] proposed two PSO-based multiobjective feature selection algorithms, i.e., nondominated sorting PSO (NSPSO) and crowding, mutation, and dominance PSO (CMDPSO), to generate a Pareto front of nondominated solutions. NSPSO integrates the concept of nondominated sorting with PSO, while CMDPSO embeds PSO with the strategies of crowding, mutation, and dominance. Both algorithms apply a crowding distance to the nondominated solutions for maintaining the selected gbest diversity for each particle. Specifically, CMDPSO employs an external archive to store the nondominated solutions and a binary tournament selection to generate gbest for each particle based on the crowding distance. It also uses the mutation operation to diversify the search. Evaluated with 12 datasets, CMDPSO outperforms NSPSO and other multiobjective algorithms, including nondominated sorting GA II (NSGAI).

C. Face and Facial Emotion Recognition

Krisshna *et al.* [24] developed a face recognition system with a method called threshold-based binary PSO feature selection (ThBPSO). ThBPSO conducts multiruns of conventional BPSO and stores gbest identified from each run. Then, a threshold is used to identify the importance of each dimension of the global best solutions. A feature is selected and considered as important if the total number of selections of this feature in the past runs is more than the predefined threshold. The system was tested with seven benchmark datasets, and showed superior performance over other state-of-the-art methods. Liu *et al.* [25] proposed a deep learning architecture, i.e., action units inspired deep networks (AUDNs), for learning facial expression features. AUDN consists of three sequential processes: 1) a convolutional layer and a max-pooling layer to learn the micro-action-pattern (MAP) representation; 2) feature grouping to integrate correlated MAPs to produce mid-level semantics; and 3) a multilayer learning process to construct subnetworks for higher-level representations.

Zavaschi *et al.* [26] proposed a novel facial expression recognition system with the integration of ensemble classifiers trained on both Gabor and LBP features. A set of 73 base support vector machine (SVM) classifiers was generated by varying parameter settings of Gabor filters and LBP. NSGAI was used to identify the most optimal ensemble structures whose fitness function focused on the minimization of both error rate and number of selected base classifiers in the ensemble. Diao *et al.* [27] proposed an adaptive ensemble reduction technique by applying the heuristic harmony search (HS) algorithm. HS identified an optimal ensemble size while preserving or increasing ensemble diversity and classification accuracy.

Zeng *et al.* [28] proposed a one-class classification system using KERNEL whitening and support vector data description to distinguish spontaneous emotional expressions from outlier nonemotional expressions. Meng and Bianchi-Berthouze [3] developed a multistage framework to explore continuous emotion recognition from naturalistic facial and vocal expressions where temporal relationships between consecutive levels of a given affective dimension were modeled using hidden Markov model (HMM). In terms of automatic multimodal emotion recognition, Zeng *et al.* [29] conducted spontaneous emotion detection from audio–visual modalities using AdaBoost multistream HMM. Soleymani *et al.* [30] performed continuous emotion recognition from electroencephalogram (EEG) signals and facial expressions. The power spectral density from EEG signals and facial landmarks were employed to represent multimodal emotional inputs. Diverse regression models such as recurrent NNs and continuous conditional random fields were used for emotion regression of the valence dimension.

Eleftheriadis *et al.* [31] proposed a discriminative shared Gaussian process latent variable model for multiview and view-invariant classification of facial expression. A discriminative manifold was derived based on learning of multiple views of a facial expression. Emotion classification was conducted using both expression manifold and view-invariant or multi-view information. Their work compared favorably with other related state-of-the-art developments. Happy and Routray [32] proposed a facial expression recognition system with the consideration of texture features of selected salient facial patches. Active facial patches associated with emotional expressions were initially extracted, which were then further analyzed to obtain discriminative salient facial features for distinguishing between each pair of emotion classes. A facial landmark detection technique to enable more accurate localization of facial patches with less computational costs was also proposed. The system employed the one-against-one classification method for emotion recognition.

III. PROPOSED FACIAL EXPRESSION RECOGNITION SYSTEM

A. Facial Feature Extraction Using the Proposed LBP

In this paper, in order to improve the discriminative abilities of LBP, we propose horizontal and vertical neighborhood pixel comparison LBP (hvnLBP). It is integrated with the Gabor filter for producing the discriminative facial representation.

There are four steps in the feature extraction process: 1) preprocessing for illumination changes and noise invariance; 2) face detection; 3) Gabor magnitude image generation; and 4) the proposed hvnLBP-based textural description. First of all, we apply histogram equalization and bilateral filter to compensate illumination variations and reduce noise in the input image, respectively. We then use a Haar-cascade face detector to detect faces. A 2-D Gabor filter is also applied to produce magnitude pictures. Finally, the proposed hvnLBP operator is used to generate the textural description of facial images.

As a well-known texture descriptor, LBP [33] employs a circular neighborhood for feature extraction. This original LBP operator performs a comparison purely between the central pixel and the eight surrounding neighborhood pixels, therefore likely to lose the contrast information among the neighborhood pixels. To solve this problem, we propose hvnLBP to capture missing contrast information among the neighborhood pixels. Instead of comparing with the central pixel as in original LBP, hvnLBP employs horizontal and vertical neighborhood pixels for direct comparison to produce the resulting textural descriptions. As an example, we employ $P = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$ to represent the eight neighborhood pixels in LBP, as shown in Fig. 2. In either vertical or horizontal comparison, the values of the vertical or horizontal neighboring pixels are compared with one another. A 1 is assigned to the pixel with the highest value and a 0 is assigned to the remaining pixels. This horizontal and vertical comparison process can be conducted in any order, i.e., horizontal comparison followed by vertical comparison, or vice versa. Moreover, in both vertical and horizontal comparisons, we do not include the center pixel for comparison. Referring to Fig. 2, as an example, for horizontal comparison, we first compare the pixel sets of $\{p_0, p_1, p_2\}$, $\{p_7, p_3\}$, and $\{p_6, p_5, p_4\}$. Subsequently, we conduct the vertical comparison with the pixel sets of $\{p_0, p_7, p_6\}$, $\{p_1, p_5\}$, and $\{p_2, p_3, p_4\}$. If a pixel has conflicting outputs in the horizontal and vertical comparisons (e.g., the highest value in the horizontal comparison but not in the vertical comparison, or vice versa), then the highest value (i.e., 1) is used as the final output, since the pixel is regarded as important, which contains valuable contrast information in the dimension that generates the highest value. The mathematical representation of this proposed hvnLBP_{*p,r*} operator is illustrated as follows:

$$\begin{aligned} \text{hvnLBP}_{p,r} = \{ & S(\max(l_0, l_1, l_2)), S(\max(l_7, l_3)), \\ & S(\max(l_6, l_5, l_4)), S(\max(l_0, l_7, l_6)), \\ & S(\max(l_1, l_5)), S(\max(l_2, l_3, l_4)) \} \end{aligned} \quad (1)$$

where p is the number of neighborhood pixels, and r is the radius. l_i represents the i th neighborhood of pixel l while S denotes the comparison operation, as follows:

$$S(\max(l_j, l_k, l_m)) = \begin{cases} 1 & \text{if maximum} \\ 0 & \text{if non_maximum} \end{cases} \quad (2)$$

where l_j , l_k , and l_m represent the neighborhood pixels in a row or column. Note that l_k is removed if it is the center pixel.

An example output of the proposed hvnLBP_{*p,r*} operator is provided in Fig. 2, where $p = 8$ and $r = 1$. In this paper, we use a window size of 75×75 pixels to represent a detected face image. Therefore, by applying the proposed hvnLBP operator, we obtain 25×25 (i.e., 625) subregions with the size of each subregion being 3×3 .

Overall, in comparison with the original LBP operator, the experimental results indicate that hvnLBP is more capable of capturing discriminative contrast information such as corners and edges among neighborhoods to inform subsequent PSO-based feature selection and facial expression analysis.

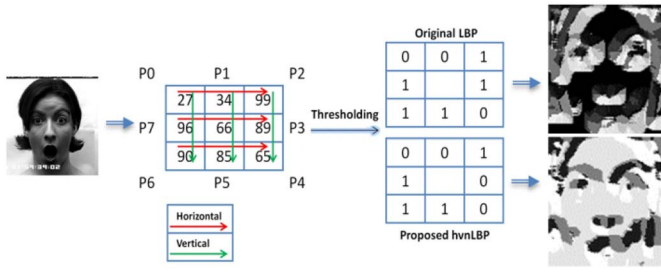


Fig. 2. Example output of the proposed hvnlBP operator in comparison with that of the original LBP.

B. Proposed PSO Algorithm for Feature Optimization

To identify the discriminative characteristics of each expression, we propose a PSO variant embedded with the concept of mGA for feature optimization, called the mGA-embedded PSO algorithm. This proposed PSO algorithm mitigates the premature convergence problem of conventional PSO, and shows superior capabilities of discriminative feature selection. The proposed mGA-embedded PSO algorithm employs personal average experience and Gaussian mutation for velocity updating. Furthermore, it integrates the diversity maintenance strategy of mGA to keep the original swarm in a nonreplaceable memory, which remains intact during the lifecycle of the algorithm to increase swarm diversity. Inherited from the concept of mGA, a secondary swarm with a small population size of five particles is employed. The swarm comprises a leader and four follower particles with the highest or lowest correlation to the leader from the nonreplaceable memory to increase local and global search capabilities and avoid premature convergence. Moreover, the algorithm separates facial features into specific areas for in-depth local subdimension-based search. Overall, the local exploitation and global exploration search strategies of the algorithm work cooperatively to lead the search process to the global optima. Algorithm 1 illustrates the pseudo code of the proposed mGA-embedded PSO algorithm, while Fig. 3 shows the flowchart of the algorithm.

1) *Update of pbest and gbest*: In conventional PSO, each solution is represented as a particle in the swarm. Particles move in the search space by following the swarm leader in order to find the optimal solutions. Each particle has a position in the search space represented as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, whereas it also has a velocity represented as $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, with D denoting the dimensionality of the search space. Each particle has a memory of its best experience whose position is represented as pbest. The swarm leader represents the best experience of the overall swarm, whose position is represented as gbest. The position, x_{id}^{t+1} , and velocity, v_{id}^{t+1} , of each particle are updated using the following equations [34]:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (3)$$

$$v_{id}^{t+1} = w * v_{id}^t + c_1 * r_1 * (p_{id} - x_{id}^t) + c_2 * r_2 * (p_{gd} - x_{id}^t) \quad (4)$$

where t and d indicate the t th iteration and d th dimension in the search space, respectively. An inertia weight, w , is used

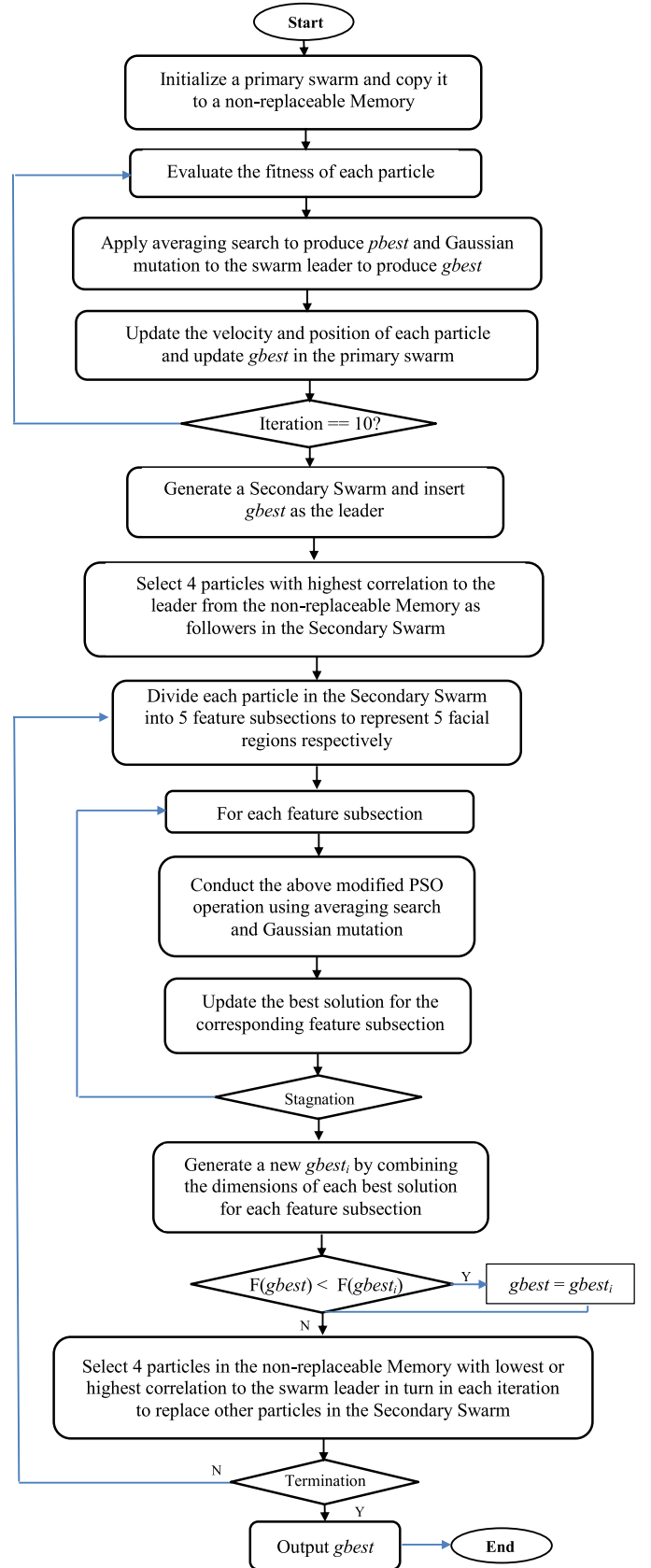


Fig. 3. Flowchart of the proposed PSO algorithm.

to embed iteration influence of the previous velocity. Note that r_1 and r_2 represent random values within the range of $[0, 1]$ whereas c_1 and c_2 are the acceleration constants.

Furthermore, p_{id} and p_{gd} indicate elements of pbest and gbest in the d th dimension. In this paper, we modify the velocity updating formula (4) by introducing the averaging search strategy for computing p_{id} and Gaussian mutation for computing p_{gd} . Specifically, the averaging search strategy takes the personal average experience into account, instead of the conventional personal best experience. The average experience is obtained by averaging the positions found from previous iterations of each individual particle for generating pbest. This enables the algorithm to better look into the search space in-between to increase local exploitation. Furthermore, instead of using the position of the global best experience directly, Gaussian distribution operation is applied to the swarm leader to generate gbest. This mutation technique enables the generation of offspring further away from its parent to increase global exploration. Therefore, the revised velocity updating strategy possesses more capability of sustaining search diversity. The updated formulas are provided as follows:

$$v_{id}^{t+1} = w * v_{id}^t + c_1 * r_1 * (p'_{id} - x_{id}^t) + c_2 * r_2 * (p'_{gd} - x_{id}^t) \quad (5)$$

$$p'_{id} = \frac{\sum X_{id}}{t} \quad (6)$$

$$p'_{gd} = p_{gd} + (x_{\max}^d - x_{\min}^d) \times \phi(o, h) \quad (7)$$

where p'_{id} and p'_{gd} represent the updated pbest and gbest in the d th dimension using personal average experience and Gaussian distribution, respectively, as defined in (6) and (7). Moreover, in (7), $\phi(o, h)$ indicates the Gaussian distribution and o represents the mean of the distribution with h as the standard deviation which decreases linearly during the execution. Note that x_{\max}^d and x_{\min}^d indicate the upper and lower bounds of the decision vector in the d th dimension, respectively, $d = 1, 2, \dots, D$.

As indicted in Algorithm 1, we first initialize the original swarm with 30 particles. The modified PSO operation with the proposed velocity updating formula is applied to the initial swarm. It iterates ten times at the beginning of the algorithm to find the best leader. We use a small number of iterations (i.e., 10) for this initial PSO search to accelerate convergence and allow benefits from subsequent search strategies to take place. This mainly aims to find the best balance between computational costs and performance. The following setting (obtained from experimental trials) is applied to this modified PSO operation, i.e., maximum velocity = 0.6, inertia weight = 0.78, population size = 30, acceleration constant $c_1 = c_2 = 1.2$, and maximum generations = 500. Moreover, (8) is used to define the fitness evaluation for each particle, C , which consists of two criteria, i.e., classification performance and the number of selected features. Since we apply the proposed PSO algorithm to each emotion category separately, in an attempt to identify the discriminative features for each distinct expression, the classification accuracy score in (8) indicates accuracy of each individual expression, rather than combined accuracy across all emotion categories. This helps avoid bias toward specific emotion categories during

optimization (see the related discussion in Section IV)

$$\text{fitness}(C) = w_a * \text{accuracy}_C + w_f * (\text{number_features}_C)^{-1} \quad (8)$$

where w_a and w_f are two predefined weights for classification accuracy and the number of selected features, respectively, with $w_a = 1 - w_f$. In addition, parameters w_a and w_f indicate the relative importance of classification performance and the number of selected features, respectively. In this paper, since the classification performance is considered to be more important than the number of selected features, w_a assumes a higher value than w_f , i.e., $w_a = 0.9$ and $w_f = 0.1$.

2) *Construction of Secondary Swarm Embedded With the Concept of mGA*: Besides the velocity updating mechanism, the proposed PSO algorithm integrates the concepts of mGA and a secondary swarm, as well as the cooperation of local exploitation and global exploration search strategies to balance between convergence speed and swarm diversity. In summary, the proposed algorithm employs the diversity maintenance strategy of mGA using a nonreplaceable memory. This nonreplaceable memory comprises the initialized swarm to sustain search diversity. Motivated by the small population size concept of mGA, a secondary swarm with five particles comprising the swarm leader and four follower particles from the nonreplaceable memory with the highest or lowest correlation with the leader is constructed to increase local exploitation and global exploration. A subdimension-based search in the secondary swarm is also conducted, in order to identify the discriminative regional facial features. Moreover, the local exploitation and global exploration search strategies of the secondary swarm work in a collaborative manner to avoid stagnation and overcome premature convergence. The details of these strategies are as follows.

mGA is a small-population GA with a reinitialization mechanism. It was initially proposed by Goldberg [35], whose theories suggested that a small population was sufficient enough to achieve convergence regardless of the chromosome length. mGA usually employs a population of 3–6 chromosomes and shows great capability of solving nonlinear optimization problems [36]. Instead of using the mutation operation as in classical GA, mGA employs a restart strategy to maintain genetic diversity in the population.

The mGA model is proven to be more capable of avoiding premature convergence and reaching the optimal search region than the classical GA [37]. Because of its impressive performance and fast convergence speed, mGA has been widely used to deal with single-objective and multiobjective optimization problems [38]. Furthermore, Coello and Pulido [11] proposed a multiobjective mGA with two memories, i.e., population memory and external memory. The population memory consists of replaceable and nonreplaceable aspects. The nonreplaceable fragment of the memory remains intact during the entire lifetime of the algorithm, in order to bring sufficient diversity to the algorithm, whereas the replaceable portion of the memory is used for conventional evolution where the solutions are kept updated in the subsequent evolutionary cycles. The multiobjective mGA shows efficient search

Algorithm 1 Pseudo-Code of mGA-Embedded PSO

```

1 Start
2 Initialize a primary swarm (e.g. 30 particles);
3 Copy the initialized swarm into a non-replaceable Memory;
4 For each particle in the primary swarm do //perform modified
   PSO operator
5 {
6   Evaluate each particle using the defined fitness function;
7   Compute the average fitness value of previous runs (if
   available) for each particle in the primary swarm;
8   Perform the proposed Averaging Search operation (Eqn. (6))
   to generate pbest for each individual particle;
9   Apply the Gaussian mutation operation to the swarm leader
   to produce gbest (Eqn. (7));
10  Update the velocity and position of each particle;
11  Update the best particle gbest in the primary swarm;
12  Until (iterations==10)
13 }
14 Generate a Secondary Swarm
15 {
16  Select the best particle gbest from primary swarm as the leader
   of the Secondary Swarm;
17  Select 4 particles that have the highest correlation with the
   leader from the non-replaceable Memory, which contains the
   original particle swarm, as the followers in the Secondary
   Swarm; //this is for local exploitation and a high correlation
   means very similar particles.
18 }
19 Divide each particle in the Secondary Swarm into five feature
   subsections with each subsection consisting of partial dimensions
   which indicates a specific facial region (e.g. eye, eyebrow, mouth
   etc);
20 For each feature subsection representing each facial region do
   // i.e. the corresponding partial dimensions of each particle in
   the Secondary Swarm
21 {
22  Apply operations of line 6-10;
23  Update the best solution for the corresponding feature sub-
   section;
24  Until (stagnation detected);
25 }
26 Replace the particles in the Secondary Swarm
27 {
28  Combine the dimensions of each best solution for each fea-
   ture subsection to replace the swarm leader in the Secondary
   Swarm if this newly generated combined leader has a better
   fitness value;
29  Select 4 particles in the non-replaceable Memory of the orig-
   inal swarm that have the lowest correlation with the above
   swarm leader to replace other particles in the Secondary
   Swarm; //this is for global exploration, and the lowest cor-
   relation means particles with high variations to the leader
30 }
31 While (Overall termination criteria are not achieved)
32 {
33  Repeat lines 19-25;
34  Repeat lines 26-30, but change from lowest correlation to
   highest correlation in a vice versa manner;
35 }
36 End
37 Return the most optimal solution;
38 End

```

diversity, and requires less computational cost compared with other algorithms such as Pareto archived evolutionary strategy.

This paper borrows the multiobjective mGA concept with the replaceable and nonreplaceable memories to update the

swarm leader (replaceable portion) and preserve diversity of the initialized swarm (nonreplaceable portion), respectively. After initializing the swarm with 30 randomly generated particles at the beginning of the algorithm (see Algorithm 1), this original swarm is stored in the nonreplaceable memory, which remains intact during the lifetime of the algorithm, in order to reward swarm diversity when stagnation occurs. To balance between swarm diversity and convergence speed, a secondary swarm embedded with the small population concept of mGA is constructed. It has a typical population size of five, and consists of a swarm leader and four follower particles from the nonreplaceable memory. As illustrated in Algorithm 1, the followers are chosen based on two types of correlation relationships with the leader: 1) the lowest and 2) the highest correlations. Particles with the lowest correlation provide higher variations in the swarm to enable global exploration whereas particles with the highest correlation bring more similarity in the swarm where local exploitation can be observed. Moreover, we define the correlation relationship between particles using (9) and (10) [39]. Since the extracted features using hvnLBP are in the binary format and can be converted into histogram easily, we use the histogram correlation comparison method, as shown in (9) and (10) [39], to identify particles with highest/lowest correlation to the leader

$$\text{corr}(H_1, H_2) = \frac{\sum_I (H_1(I) - H'_1)(H_2(I) - H'_2)}{\sqrt{\sum_I (H_1(I) - H'_1)^2 \sum_I (H_2(I) - H'_2)^2}} \quad (9)$$

where

$$H'_k = \frac{1}{N} \sum_I H_k(I) \quad (k = 1, 2) \quad (10)$$

where *corr* indicates the correlation relationship between two particles with H_1 and H_2 representing the histograms for the swarm leader and a follower particle, respectively. H'_k indicates the mean of the histogram for the k th particle ($k = 1, 2$), whereas N represents the number of histogram bins and I indicates the intensity range present in the histogram. Equation (9) produces an output in the range of [0, 1], with 0 and 1 representing the lowest and highest correlations, respectively.

As shown in Algorithm 1 and Fig. 3, first of all, after identifying the swarm leader by the previous modified PSO process, four follower particles from the nonreplaceable memory with the highest correlation with the leader are recruited to the secondary swarm. The aim of extracting the follower particles from the nonreplaceable memory, instead of using the particles from the main swarm, is to avoid diversity loss as the particles in the main swarm tend to be converged and become identical after ten iterations. Moreover, these follower particles with the highest correlation with the leader provide certain degree of position proximity in the secondary swarm, therefore enabling local exploitation of the search space. Subsequently, we divide each particle in the secondary swarm into five feature subsections, with each subsection representing each facial region to enable an in-depth local search to identify its discriminative features. This in-depth local optimal facial feature search is discussed in detail in Section III-B2a. This section-based local facial feature search

reveals a new swarm leader whose fitness value is compared with that of the previous leader, in order to elect a new leader for the next iteration.

After employing particles with the highest correlation with the leader as followers to conduct an in-depth local optimal facial feature search, the secondary swarm recruits a new set of four particles with the lowest correlation with the leader from the nonreplaceable memory to replace the existing follower particles. Since the new set of follower particles with the lowest correlation recruited from the original swarm inject high variation to the secondary swarm, it boosts the swarm diversity significantly to increase global exploration and avoid premature convergence. Subsequently, the newly updated diversified secondary swarm is also used to conduct a local facial feature search (see Section III-B2a) to identify a new swarm leader.

In this way, particles with the highest or lowest correlation with the swarm leader from the nonreplaceable memory are recruited alternately in the secondary swarm to increase local exploitation and global exploration. Moreover, when local exploitation in the subdimension search using particles with the highest correlation with the leader stagnates, our PSO algorithm employs follower particles with the lowest correlation with the leader from nonreplaceable memory to increase swarm diversity and drive the search out of local optimum trap. On the other hand, when global exploration in the subdimension search using particles with the lowest correlation with the leader fails to generate a fitter leader, it recruits follower particles with the highest correlation to the leader from nonreplaceable memory to avoid stagnation and enable local exploitation. Therefore, the local and global search mechanisms embedded in the secondary swarm work cooperatively to mitigate premature convergence and lead the search toward the global optima.

a) In-depth local optimal feature search: As discussed earlier, after particles with the highest or lowest correlation with the leader are recruited in the secondary swarm, we divide each particle in the secondary swarm into five feature sections with each section consisting of partial dimensions which indicates a specific facial region (e.g., eye, eyebrow, nose, mouth, and cheek). For each facial region, we apply the above modified PSO operation with the updated velocity updating formula defined in Section III-B to conduct an in-depth local search and to identify its optimal discriminative features. These optimal local solutions are then concatenated to generate a new swarm leader, which is used to replace the previous leader if it has a better fitness value.

The overall optimization process of our algorithm iterates until: 1) the number of evolution reaches 500 and 2) the fitness value does not show obvious improvement during the last 50 generations. The proposed PSO-based feature selection is conducted for each emotion category separately to identify discriminative features for each expression. The generated optimal feature subset of each expression by our PSO algorithm is shown in Fig. 4, with a detailed analysis provided in Section IV. Empirical results indicate that our algorithm outperforms other PSO variants and conventional methods significantly in terms of the search toward global optimum and discriminative feature selection.



Fig. 4. Selected optimal features and their distribution for each expression using the proposed mGA-embedded PSO algorithm (rows 1–3: CK+ images and rows 4–6: MMI images).

C. Emotion Recognition

In this paper, we conduct a study of seven-class facial emotion recognition using the features automatically generated by the mGA-embedded PSO. NN with backpropagation, a multiclass SVM [40], and ensemble classifiers are used for classification. The detailed setting of the classifiers is introduced, as follows. In this paper, the trial-and-error method is conducted to identify the optimal NN structure, whereas a grid-search method is applied to find the optimal parameters of the multiclass SVM with the RBF kernel. After several trials, the NN is equipped with one input layer with 25–40 nodes indicating the optimized features obtained from the proposed PSO algorithm, one hidden layer, and one output layer with seven nodes, respectively, representing seven expressions. For the grid search of optimal settings for the multiclass SVM with the RBF kernel, we use exponentially growing sequences and search the ranges of $[2^{-5} - 2^{15}]$, $[2^{-10} - 2^5]$, and $[2^{-8} - 2^{-1}]$, respectively, for a soft-margin constant, C , a kernel parameter, gamma (γ), and an epsilon (ϵ) in the loss function since the combination of these three parameters plays very important roles in affecting the SVM's performance. We also employ tenfold cross validation to identify the best combination of these parameters to avoid over-fitting. The identified optimal setting in the training stage is then applied to the subsequent experiments in the test stage.

Besides these single model classifiers, we also employ ensemble classifiers for expression recognition in order to improve accuracy. We use weighted majority voting for the construction of ensembles because of its impressive performance and suitability for undertaking small datasets (<1000) in this paper. We construct two ensembles with NN and multiclass SVM as the base model, respectively. Also the NN-based and SVM-based ensembles use three base

models, respectively. The optimal settings identified earlier for NN and SVM are applied for building each base model.

The ensemble classifiers are constructed using an AdaBoost process so that the performance of the three base models within each ensemble classifier is complementary to each other [5], [41]. The training process of each ensemble classifier focuses on misclassified instances. As an example, the weights of misclassified instances by the first base model are increased so that they are more likely to be selected for training the second base model. A similar case is also applied to the construction of the third base model, which employs the instances misclassified by the second base model for training. Therefore, each ensemble classifier is constructed with a number of base models that are complementary to each other [5], [41]. Weighted majority voting is applied to combine the outputs from the three base models to generate the final output for each ensemble. The empirical results indicate that the constructed ensembles outperform NN/SVM-based emotion recognition for both within and across database evaluations.

IV. EVALUATION

In this paper, both CK+ and MMI are employed for evaluation. A set of 250 images from CK+ is used for training while 175 images extracted from CK+ and MMI, respectively, are employed for testing.

A. Comparison of Feature Extraction Techniques

First of all, a series of experiments is conducted to compare the proposed hvnLBP operator with other state-of-the-art texture descriptors including CLBP, DLBP, CS-LBP, LDP, and LPQ. The Gabor filter is integrated with each texture descriptor algorithm for feature extraction. Low-level raw features extracted by each descriptor are directly used for emotion classification without any feature optimization. When ensemble classifiers are applied, all algorithms achieve the best performance. Table I shows the evaluation results of all descriptors integrated with ensembles with each ensemble trained with features extracted by each texture descriptor. Both second- and third-order LDPs are implemented. The results of the second-order LDP are presented in Table I, since it achieves the best performance.

The empirical results in Table I indicate that hvnLBP possesses more discriminative capabilities and outperforms all the selected state-of-the-art LBP variants, LPQ, and conventional LBP significantly for both within and cross database evaluations. When the SVM-based ensemble classifier is applied, all algorithms achieve the best accuracy rates, and hvnLBP outperforms LBP, DLBP, LPQ, CLBP, CS-LBP, and LDP by 12.93%, 12.65%, 12.33%, 11.06%, 9.24%, and 3.82%, respectively, for within-database evaluation and by 12.79%, 11.9%, 12.04%, 10.36%, 8.47%, and 2.18%, respectively, for cross-database evaluation.

Built upon the LBP methodology, DLBP and CLBP rely on the comparison between the center point and its neighbors but ignore the differences among neighborhood pixels themselves. Therefore, they show limitations in identifying different

TABLE I
COMPARISON BETWEEN THE hvnLBP OPERATOR AND
OTHER TEXTURE DESCRIPTORS

	CK+(train)/CK+(test)		CK+(train)/MMI(test)	
	Ensemble (NN) %	Ensemble (SVM) %	Ensemble (NN) %	Ensemble (SVM) %
LBP	75.00	77.67	62.98	64.34
DLBP	76.12	77.95	63.54	65.23
LPQ	76.00	78.27	63.90	65.09
CLBP	78.56	79.54	64.10	66.77
CS-LBP	80.19	81.36	65.70	68.66
LDP (2nd order)	85.92	86.78	71.05	74.95
<i>hvnLBP</i>	89.00	90.60	73.20	77.13

local structures embedded in the neighboring pixels. CS-LBP employs center-symmetric pixel pairs for comparison, in order to extract local discriminative information. However, it overlooks other local differences among horizontal and vertical pixels. An example that demonstrates the difference among the proposed hvnLBP operator, LBP, and CS-LBP is provided, as follows. Given two patterns (50, 80, 85, 70, 50, 45, 55, 53, center-60) and (100, 230, 240, 230, 100, 50, 120, 160, center-200), although the local structures of both patterns are different, LBP generates the same binary code, 01110000, for both patterns. CS-LBP produces 11111000 for both patterns too. However, hvnLBP is able to generate two distinctive binary codes for these patterns, i.e., 01110010 for the former and 01110011 for the latter, indicating the two different local structures.

LPQ with decorrelation is implemented in our experiment. LPQ shows great robustness to blurred images by employing local phase information calculated using a short-term Fourier transform for each pixel position. However, it has higher computational complexity, and is expensive for online applications in comparison with hvnLBP. In addition, the window size is one of the important parameters in LPQ. A smaller window is able to capture detailed texture information, but other unimportant patterns caused by illumination changes and noise factors are extracted as well. On the contrary, a larger window sometimes is not able to extract sufficient discriminative information, therefore decreasing the performances for sharp images [16].

Among all the comparable descriptors, the second-order LDP achieves the best accuracy rate, which extracts more detailed high-order local pattern information. However, the empirical results indicate that sometimes it also extracts over-detailed patterns which contain more noise in comparison with hvnLBP. Moreover, the second-order LDP also generates high-dimensional features with a high-computational cost, which makes it less suitable for real-time applications. Another limitation of using LDP is the requirement of identifying the optimal order of LDP that is suitable for a specific database although the third-order LDP outperformed all the other order LDPs in [15] for face recognition tasks.

In comparison with the abovementioned comparable methods, the proposed hvnLBP operator effectively extracts spatial

TABLE II
AVERAGE CLASSIFICATION PERFORMANCE USING THE SELECTED
OPTIMIZATION ALGORITHMS INTEGRATED WITH DIVERSE
CLASSIFIERS OVER 30 RUNS, RESPECTIVELY,
WITHIN DATABASE EVALUATION

Methods	No. of selected features	NN % (30 runs)	SVM % (30 runs)	Ensemble (NN) % (30 runs)	Ensemble (SVM) % (30 runs)
GA	150-220	74.06	77.73	78.03	79.8
PSO	150-200	75.46	78.10	83.5	85.3
ThBPSO	80-120	85.86	85.16	89.3	92.6
HEPSO	70-100	86.50	87.16	93.80	94.4
ELPSO	65-80	87.26	88.43	94.86	95.3
MFOPSO	70-110	87.90	89.16	95.60	97.3
BBPSO	50-100	88.55	89.88	96.33	97.4
Prop. PSO	25-40	94.33	95.50	100	100

relationships in a local region by conducting multiple direct horizontal and vertical neighborhood comparisons with an efficient computational cost. From the empirical study, it shows superior capabilities of preserving distinctiveness and differentiating different local structures embedded in the neighboring pixels for low contrast images.

B. Comparison of Feature Selection Techniques

To evaluate the proposed mGA-embedded PSO algorithm for feature selection, we have implemented state-of-the-art methods for comparison, i.e., ELPSO [20], a PSO variant for multimodal function optimization (MFOPSO) [42], binary BPSO (BBPSO) [21], ThBPSO [24], HEPHO [18], conventional PSO, and classical GA. The features extracted by hvnLBP are further processed by each feature optimization algorithm for dimensionality reduction. NN, SVM, and NN-based and SVM-based ensembles are applied to recognize seven emotions using automatically generated features based on each feature optimization technique.

Owing to the stochastic property of the proposed PSO algorithm and other compared methods, we have performed 30 runs for each method integrated with each classifier for within and cross database evaluations, respectively. First of all, we conduct the within-database evaluation by applying 250 and 175 images from CK+ for training and testing, respectively. Table II shows the average classification performance of 30 runs for all optimization algorithms in combination with diverse classifiers. The best results are obtained for each feature selection model when SVM-based ensemble is applied. The proposed mGA-embedded PSO algorithm achieves an average accuracy rate of 100% for seven emotions, and outperforms seven other algorithms by 2.6% (BBPSO), 2.7% (MFOPSO), 4.7% (ELPSO), 5.6% (HEPSO), 7.4% (ThBPSO), 14.7% (PSO), and 20.2% (GA), respectively. Moreover, our algorithm extracts a comparatively smaller set of features (25–40) with more efficient computational costs.

We have also conducted the cross-database evaluation with a training set of 250 images from CK+ and a test set of 175 images from MMI. Table III summarizes the average accuracy rates for all the selected models integrated with different classifiers over 30 runs for the cross-database evaluation. The best performances are yielded by the SVM-based ensemble for all feature selection methods. The proposed PSO

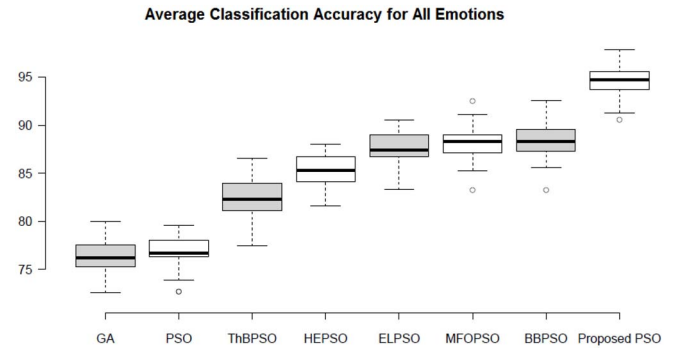


Fig. 5. Boxplot diagram for the distribution of average recognition results for each optimization algorithm + SVM-based ensemble over 30 runs for cross-database evaluation.

algorithm extracts the smallest number of features, achieves an average accuracy rate of 94.66% for seven emotions, and outperforms seven other methods by 6.35% (BBPSO), 6.57% (MFOPSO), 7.21% (ELPSO), 9.49% (HEPSO), 12.28% (ThBPSO), 17.89% (PSO), and 18.35% (GA), respectively. In Fig. 5, the boxplot diagrams clearly demonstrate the distribution of the classification results over 30 runs of all the feature selection methods in combination with the SVM-based ensemble for the cross-database evaluation.

As can be seen in Fig. 5, the results of all 30 runs of the proposed PSO algorithm outperform those of all other state-of-the-art PSO variants, conventional PSO, and classical GA by a significant margin. For example, all the results of 30 runs of our algorithm except for one outlier (with the lower whisker at 91.29%) are higher than the maximum results of all the following methods, i.e., 91.14% for MFOPSO, 90.57% for ELPSO, 88% for HEPHO, 86.57% for ThBPSO, 79.57% for PSO, and 80% for classic GA. Furthermore, at least 75% of the results of our algorithm (with the first quartile of 93.71%) are higher than the maximum result, i.e., 92.57% from BBPSO. Among all the selected state-of-the-art PSO variants, BBPSO, MFOPSO, and ELPSO achieve comparatively better performances than HEPHO and ThBPSO, i.e., with at least 25% of the results of these three PSO variants higher than the maximum result (88%) of HEPHO and at least 75% of the results of these three PSO variants higher than the maximum result (86.57%) of ThBPSO. In comparison with these three best PSO variants, i.e., BBPSO, MFOPSO, and ELPSO, the median value of our algorithm (94.71%) is higher than the median scores of BBPSO (88.29%), MFOPSO (88.29%), and ELPSO (87.43%) by 6.42%, 6.42%, and 7.28%, respectively. Besides outperforming these three best PSO variants, all the results of our algorithm are within a smaller variation range of [91.29%, 97.86%], as compared with those from BBPSO having a larger variation of [85.57%, 92.57%]. Moreover, the lowest result of our PSO algorithm (i.e., the lower whisker at 91.29%) outperforms the maximum results of HEPHO (88%), ThBPSO (86.57%), classical GA (80%), and PSO (79.57%) by 3.29%, 4.72%, 11.29%, and 11.72%, respectively.

Furthermore, the average classification results of each expression over the 30 runs for each optimization method with the SVM-based ensemble classifier for the cross-database evaluation are depicted in Fig. 6(a). Fig. 6(b)–(h) indicates the

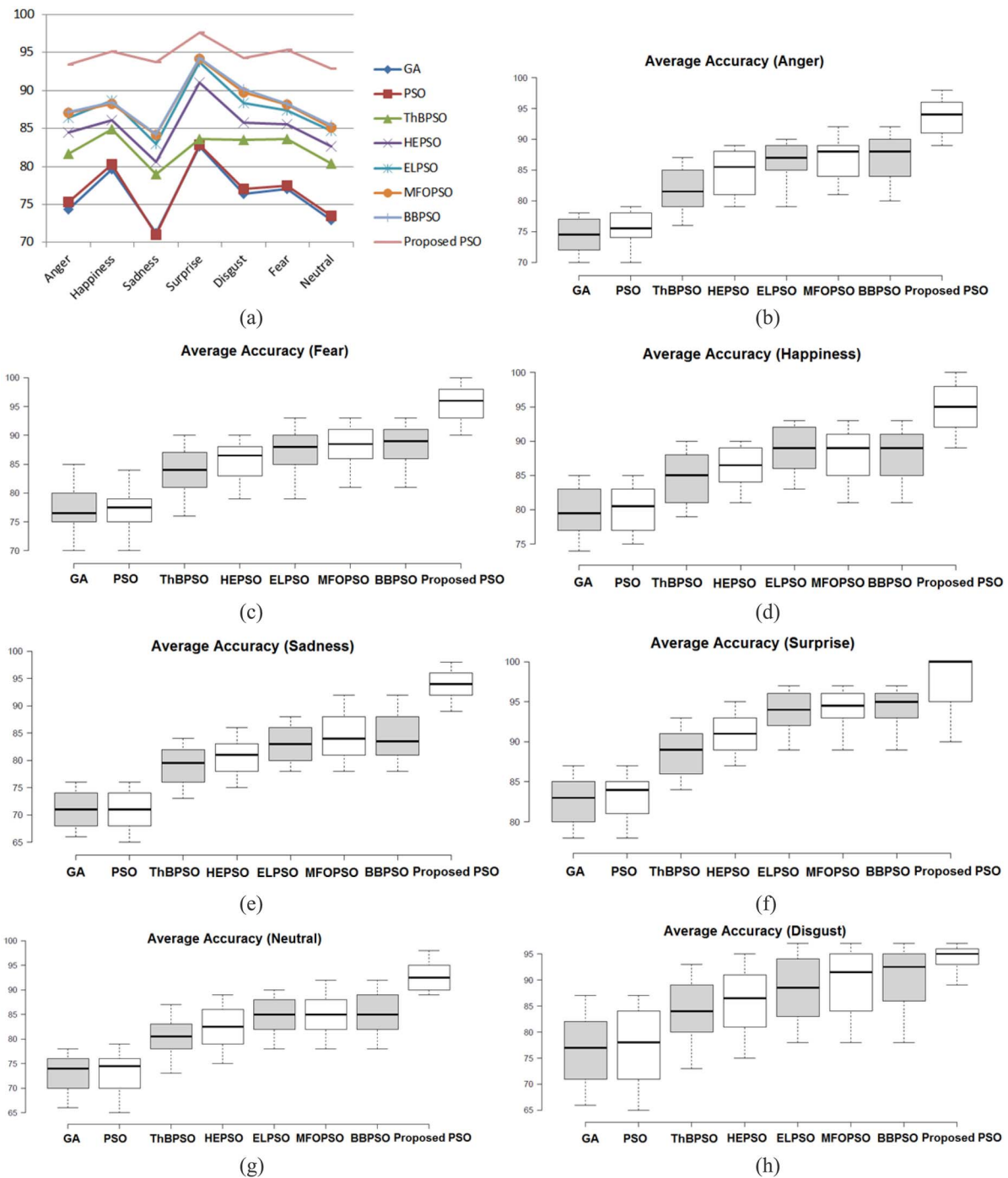


Fig. 6. (a) Overall comparison of our system with other methods. (b)–(h) Boxplot diagrams for the distribution of classification results for each emotion category for each optimization algorithm + SVM-based ensemble over 30 runs for cross-database evaluation.

detailed boxplot diagrams for the distribution of the detailed classification results over 30 runs for each emotion category. As indicated in Fig. 6(a)–(h), the proposed PSO algorithm achieves superior performance and outperforms all the other compared methods for each emotion significantly. With respect to the fear and sadness emotion categories, 75% of the classification results of our model are higher than the maximum results of all seven methods, whereas at least 50% of the results of our algorithm are also higher than the maximum results of all other methods for the anger, happiness, surprise, and neutral emotion classes. Meanwhile, for the disgust emotion,

the results of our algorithm over 30 runs indicate the overall smallest variation of [89%, 97%], as compared with other larger variations of the other results, e.g., [78%, 97%] for BBPSO, MFOP SO, and ELPSO, respectively. The proposed diversity maintenance strategies of our PSO algorithm contribute to its superior performance over other state-of-the-art and conventional methods.

An analysis pertaining to the theoretical contribution of the proposed algorithm is as follows. We compare our PSO algorithm with the three advanced PSO variants, i.e., BBPSO, MFOP SO, and ELPSO, theoretically.

TABLE III
AVERAGE CLASSIFICATION PERFORMANCE USING THE SELECTED OPTIMIZATION ALGORITHMS INTEGRATED WITH DIVERSE CLASSIFIERS OVER 30 RUNS, RESPECTIVELY, FOR CROSS-DATABASE EVALUATION

Methods	No. of selected features	NN % (30 runs)	SVM % (30 runs)	Ensemble (NN) % (30 runs)	Ensemble (SVM) % (30 runs)
GA	150-220	70.13	71.23	75.11	76.31
PSO	150-200	70.90	72.50	76.05	76.77
ThBPSO	80-120	76.75	78.56	81.30	82.38
HEPSO	70-100	78.55	79.76	83.50	85.17
ELPSO	65-80	78.84	80.33	86.00	87.45
MFOPSO	70-110	80.85	81.76	87.60	88.09
BBPSO	50-100	81.07	81.88	87.80	88.31
Prop. PSO	25-40	88.97	90.70	92.90	94.66

BBPSO [21] employs a reinforced memory strategy for updating pbest for each particle and a uniform combination technique to replace subdimensions of each particle using a random number with the corresponding elements of a randomly selected $pbest_k$ from a set of stored pbests to avoid stagnation. It increases the execution of a uniform combination with respect to increased stagnant iterations. However, since the uniform combination operation is only applied to the subelements of swarm particles and simulates the effects of crossover and mutation operations of the GA, the generated offspring could be significantly similar (i.e., with a high correlation) to the parent particles. Therefore, their search strategy focuses more on local exploitation. In contrast, our PSO variant applies follower particles which have the highest or lowest correlation with the leader to diversify the search and increase both local and global search capabilities, in an attempt to avoid stagnation. Therefore, it shows a superior performance than that of BBPSO.

MFOPSO [42] divides the original swarm into several subswarms to increase search diversity. It is capable of dealing with multimodal function optimization. However, when the search fails to generate fitter leaders in the subswarms, MFOPSO does not include any diversity maintenance or jump-out mutation strategy to diversify the search in the subswarms, in order to avoid premature convergence.

The same explanation applies to ELPSO [20]. It employs Gaussian, Cauchy, opposition-based, and DE-based mutation strategies to increase the exploration capability of the swarm leader. However, ELPSO only attempts to improve the leader when stagnation occurs, and no improvement strategy is applied to the follower particles to retain population diversity. In comparison with MFOPSO and ELPSO, our PSO algorithm utilizes the diversity maintenance mechanism of mGA and keeps a nonreplaceable memory to maintain swarm diversity. It not only applies Gaussian mutation to the swarm leader to enable long jumps in the primary swarm but also employs particles with the highest or lowest correlation with the swarm leader from the nonreplaceable memory to retain population diversity and increase local exploitation and global exploration. Most importantly, these local and global search strategies of the secondary swarm work collaboratively to lead the search toward the global optimum. Therefore, it outperforms MFOPSO and ELPSO significantly in terms of

TABLE IV
COMPARISON WITH RELATED RESEARCH ON THE CK+ DATABASE

Method	Classes	Dyna mic	Measure	Recog. rate (%)	
Shojaeilan gari <i>et al.</i> [43]	Histogram of Dominant Phase Congruency (HDPC) + SVM	6	Y	Leave-one-subject-out	95.44
Liu <i>et al.</i> [25]	Action Units inspired Deep Networks	7	N	10-fold	93.70
Xiao <i>et al.</i> [44]	Multiple manifolds	6	N	25% for testing	96.57
Neoh <i>et al.</i> [41]	Overlap LGBP + direct similarity + weighted majority vote	7	N	42.8% for testing	96.80
Neoh <i>et al.</i> [41]	Overlap LGBP + Pareto + weighted majority vote	7	N	42.8% for testing	97.40
This research	Modified LBP + proposed mGA-embedded PSO + ensemble (SVM)	7	N	Average of 30 runs with 46.6% for testing for each run	100

TABLE V
COMPARISON WITH RELATED DEVELOPMENTS ON THE MMI DATABASE

Method	Classes	Dyna mic	Measure	Recog. rate (%)	
Fang <i>et al.</i> [45]	Group-wise registration + parametric space + fuzzy-rough based nearest neighbour	6	Y	10-fold	75.96 (trained with MMI)
Fan and Tjahjadi [46]	Histogram of gradients + optical flow + SVM	7	Y	10-fold	58.7 (trained with CK+)
Liu <i>et al.</i> [25]	Action Units inspired Deep Networks	7	N	10-fold	72.2 (trained with CK+)
This research	Modified LBP + proposed mGA-embedded PSO + ensemble (SVM)	7	N	Average of 30 runs with 70% for testing for each run	94.66 (trained with CK+)

achieving the global optimum and enabling discriminative feature selection.

A comparison between our proposed PSO algorithm and other recent state-of-the-art facial expression recognition methods has been conducted. Tables IV and V show the comparison among different methods using the CK+ and MMI databases, respectively. As shown in Table IV, for the evaluation using CK+, which proposed both direct similarity and Pareto-based optimization for facial feature selection, Neoh *et al.* [41] achieve the best performance. The Pareto-based feature selection emphasizes both intraclass and interclass variations and achieves the highest accuracy rate. However, although related strategies are adopted in their fitness functions to prevent information loss, inspection of their results indicate that the algorithms produce a comparatively small subset of 13–39 features and, sometimes, could overlook certain important features pertaining to certain emotion categories (e.g., widened eyes for surprise, mouth stretch for fear, etc.) in comparison with our proposed algorithm. As illustrated in Fig. 4, the feature subregions extracted by our PSO algorithm indicate the most significant texture distortions around the eyes, eyebrows, and the mouth associated with each distinct expression. The key facial muscular actions defined in facial action coding system (FACS) [47] associated with each expression can be clearly seen in the optimized features revealed by

our algorithm. E.g., for anger, significant features indicating brow lower, eyelid, and lip tightener are produced by our PSO algorithm, whereas the subregions indicating the significance of lip corner puller and cheek raising are revealed for the happy expression. Feature distribution pertaining to sadness clearly indicates the implication of the inner brow raiser and lip corner depressor whereas eyebrow raiser, widened eyes, and mouth open are demonstrated in the selected subregions for surprise, etc. Overall, the features identified by our PSO algorithm represent the characteristics of each emotion significantly and map closely to the action units given in FACS.

We also conduct the cross-database evaluation to further assess the scalability of the proposed PSO algorithm using the MMI database. Table V shows a comparison with other related methods. Fang *et al.* [45] employed MMI for both training and testing, whereas other methods including this paper used CK+ for training and MMI for testing. Results indicate our algorithm shows great scalability and extracts the most discriminative features of each expression for the cross-domain evaluation. It outperforms all related methods by a significant margin of approximately 20%–37%.

V. CONCLUSION

In this paper, we have proposed a facial expression recognition system with hvnLBP based feature extraction, mGA-embedded PSO-based feature optimization and diverse classifier based expression recognition. The proposed hvnLBP operator performs horizontal and vertical neighborhood pixel comparison to retrieve the initial discriminative facial features. It outperforms state-of-the-art LBP variants, LPQ, and conventional LBP significantly for texture classification. Moreover, a new PSO algorithm, i.e., mGA-embedded PSO, has been proposed to mitigate the premature convergence problem of conventional PSO in terms of feature optimization. The mGA-embedded PSO algorithm incorporates personal average experience and Gaussian mutation for velocity updating as well as employs the diversity maintenance strategy of mGA by keeping the original swarm in a nonreplaceable memory, which remains intact during the lifecycle of the algorithm to increase swarm diversity. Furthermore, it also maintains a secondary swarm with a small population size of five to host the swarm leader and four follower particles with the highest/lowest correlation with the leader from the nonreplaceable memory to increase local and global search capabilities. The algorithm subsequently separates facial features into specific areas for in-depth local subdimension based search. Overall, the local exploitation and global exploration search mechanisms of the algorithm work cooperatively to guide the search toward the global optimal solutions. The empirical results indicate that our PSO algorithm outperforms other state-of-the-art PSO variants and conventional PSO and GA for optimal feature selection significantly. Integrated with the SVM-based ensemble, our algorithm achieves the best average accuracy of 100% over 30 runs for the within (CK+) database evaluation and 94.66% accuracy for the cross-domain (MMI) evaluation. On an average of 30 runs, it outperforms seven optimization algorithms by 2.6% (BBPSO), 2.7% (MFOPSO), 4.7% (ELPSO),

5.6% (HEPSO), 7.4% (ThBPSO), 14.7% (PSO), and 20.2% (GA), respectively, for the within-domain evaluation using CK+, and by 6.35% (BBPSO), 6.57% (MFOPSO), 7.21% (ELPSO), 9.49% (HEPSO), 12.28% (ThBPSO), 17.89% (PSO), and 18.35% (GA), respectively, for the cross-domain evaluation using MMI. The empirical results also indicate that our proposed PSO algorithm outperforms other related facial expression recognition methods reported in the literature by a significant margin.

We have identified the following directions for further improvements. Diverse search strategies such as the firefly algorithm and cuckoo search can be explored for search diversity of the overall swarm and for subdimension exploration. Multiobjective evolutionary algorithms can also be explored to further equip the current algorithm to deal with real-world challenging optimization problems containing multiple criteria. Motivated by Zavaschi *et al.* [26] and Diao *et al.* [27], ensemble construction using base models trained on diverse features provided by LBP variants and LPQ will be explored to further improve performance. We also aim to integrate the proposed PSO algorithm into a humanoid robot to enable it to deal with challenging real-world spontaneous human behavior interpretation and robot interaction tasks.

REFERENCES

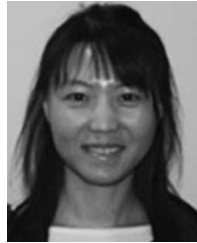
- [1] S. Wang, J. Sha, H. Y. Wu, and Y. Fu, "Hierarchical facial expression animation by motion capture data," in *Proc. IEEE Int. Conf. Multimedia Expo*, Chengdu, China, 2014, pp. 1–6.
- [2] L. Zhong, Q. Liu, P. Yang, J. Huang, and D. N. Metaxas, "Learning multiscale active facial patches for expression analysis," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1499–1510, Aug. 2015.
- [3] H. Meng and N. Bianchi-Berthouze, "Affective state level recognition in naturalistic facial and vocal expressions," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 315–328, Mar. 2014.
- [4] Y. Zhang, L. Zhang, S. C. Neoh, K. Mistry, and M. A. Hossain, "Intelligent affect regression for bodily expressions using hybrid particle swarm optimization and adaptive ensembles," *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8678–8697, 2015.
- [5] Y. Zhang, L. Zhang, and M. A. Hossain, "Adaptive 3D facial action intensity estimation and emotion recognition," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1446–1464, 2015.
- [6] L. Zhang, K. Mistry, M. Jiang, S. C. Neoh, and M. A. Hossain, "Adaptive facial point detection and emotion recognition for a humanoid robot," *Comput. Vis. Image Understand.*, vol. 140, pp. 93–114, Nov. 2015.
- [7] L. Zhang and J. Barnden, "Affect sensing using linguistic, semantic and cognitive cues in multi-threaded improvisational dialogue," *Cogn. Comput.*, vol. 4, no. 4, pp. 436–459, 2012.
- [8] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, 2013.
- [9] P. Lucey *et al.*, "The extended Cohn–Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. 3rd Int. Workshop CVPR Human Commun. Behav. Anal.*, San Francisco, CA, USA, 2010, pp. 94–101.
- [10] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, 2005, pp. 317–321.
- [11] C. A. C. Coello and G. T. Pulido, "A micro-genetic algorithm for multiobjective optimization," in *Evolutionary Multi-Criterion Optimization* (LNCS 1993). Heidelberg, Germany: Springer, 2001, pp. 126–140.
- [12] S. Liao, M. W. K. Law, and A. C. S. Chung, "Dominant local binary patterns for texture classification," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1107–1118, May 2009.
- [13] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, Jun. 2010.

- [14] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, 2009.
- [15] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 533–544, Feb. 2010.
- [16] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *Image and Signal Processing (LNCS 5099)*. Heidelberg, Germany: Springer, 2008, pp. 236–243.
- [17] M. Campos, R. A. Krohling, and I. Enriquez, "Bare bones particle swarm optimization with scale matrix adaptation," *IEEE Trans. Cybern.*, vol. 44, no. 9, pp. 1567–1578, Sep. 2014.
- [18] M. J. Mahmoodabadi, Z. S. Mottaghi, and A. Bagheri, "HEPSO: High exploration particle swarm optimization," *Inf. Sci.*, vol. 273, pp. 101–111, Jul. 2014.
- [19] N.-J. Li, W.-J. Wang, and C.-C. J. Hsu, "Hybrid particle swarm optimization incorporating fuzzy reasoning and weighted particle," *Neurocomputing*, vol. 167, pp. 488–501, Nov. 2015.
- [20] A. R. Jordehi, "Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimisation problems," *Appl. Soft Comput.*, vol. 26, pp. 401–417, Jan. 2015.
- [21] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, "Feature selection algorithm based on bare bones particle swarm optimization," *Neurocomputing*, vol. 148, pp. 150–157, Jan. 2015.
- [22] H. Wang, S. Rahnamayan, H. Sun, and M. G. H. Omran, "Gaussian bare-bones differential evolution," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 634–647, Apr. 2013.
- [23] L.-Y. Chuang, C.-H. Yang, and J.-C. Li, "Chaotic maps based on binary particle swarm optimization for feature selection," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 239–248, 2011.
- [24] N. L. A. Krishna, V. K. Deepak, K. Manikantan, and S. Ramachandran, "Face recognition using transform domain feature extraction and PSO-based feature selection," *Appl. Soft Comput.*, vol. 22, pp. 141–161, Sep. 2014.
- [25] M. Liu, S. Li, S. Shan, and X. Chen, "AU-inspired deep networks for facial expression feature learning," *Neurocomputing*, vol. 159, pp. 126–136, Jul. 2015.
- [26] T. H. H. Zavaschi, A. S. Britto, Jr., L. E. S. Oliveira, and A. L. Koerich, "Fusion of feature sets and classifiers for facial expression recognition," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 646–655, 2013.
- [27] R. Diao, F. Chao, T. Peng, N. Snooke, and G. Shen, "Feature selection inspired classifier ensemble reduction," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1259–1268, Aug. 2014.
- [28] Z. Zeng *et al.*, "One-class classification for spontaneous facial expression analysis," in *Proc. 7th Int. Conf. Autom. Face Gesture Recognit.*, Southampton, U.K., 2006, pp. 281–286.
- [29] Z. Zeng *et al.*, "Audio-visual spontaneous emotion recognition," in *Artificial Intelligence for Human Computing (LNCS 4451)*. Heidelberg, Germany: Springer, 2007, pp. 72–90.
- [30] M. Soleymani, S. Asghari-Esfeden, Y. Fu, and M. Pantic, "Analysis of EEG signals and facial expressions for continuous emotion detection," *IEEE Trans. Affect. Comput.*, vol. 7, no. 1, pp. 17–28, Jan./Mar. 2015.
- [31] S. Eleftheriadis, O. Rudovic, and M. Pantic, "Discriminative shared Gaussian processes for multiview and view-invariant facial expression recognition," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 189–204, Jan. 2015.
- [32] S. L. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches," *IEEE Trans. Affect. Comput.*, vol. 6, no. 1, pp. 1–12, Jan./Mar. 2015.
- [33] T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on featured distribution," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996.
- [34] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4. Perth, WA, Australia, 1995, pp. 1942–1948.
- [35] D. E. Goldberg, "Sizing populations for serial and parallel genetic algorithms," in *Proc. 3rd Int. Conf. Genet. Algorithms*, 1989, pp. 70–79.
- [36] C. A. C. Coello and G. T. Pulido, "Multiobjective structural optimization using a microgenetic algorithm," *Struct. Multidisciplinary Optim.*, vol. 30, no. 5, pp. 388–403, 2005.
- [37] K. Krishnakumar, "Micro-genetic algorithms for stationary and non-stationary function optimization," in *Proc. SPIE.*, vol. 1196. 1990, pp. 289–296.
- [38] S. F. Ali and A. Ramaswamy, "Optimal fuzzy logic control for MDOF structural systems using evolutionary algorithms," *Eng. Appl. Artif. Intell.*, vol. 22, no. 3, pp. 407–419, 2009.
- [39] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, 6th ed. Ames, IA, USA: Iowa State Univ. Press, 1967.
- [40] K. Cramer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, Mar. 2001.
- [41] S. C. Neoh *et al.*, "Intelligent facial emotion recognition using a layered encoding cascade optimization model," *Appl. Soft Comput.*, vol. 34, pp. 72–93, Sep. 2015.
- [42] W.-D. Chang, "A modified particle swarm optimization with multiple subpopulations for multimodal function optimization problems," *Appl. Soft Comput.*, vol. 33, pp. 170–182, Aug. 2015.
- [43] S. Shojaeilangari, W.-Y. Yau, and E.-K. Teoh, "A novel phase congruency based descriptor for dynamic facial expression analysis," *Pattern Recognit. Lett.*, vol. 49, pp. 55–61, Nov. 2014.
- [44] R. Xiao, Q. Zhao, D. Zhang, and P. Shi, "Facial expression recognition on multiple manifolds," *Pattern Recognit.*, vol. 44, no. 1, pp. 107–116, 2011.
- [45] H. Fang *et al.*, "Facial expression recognition in dynamic sequences: An integrated approach," *Pattern Recognit.*, vol. 47, no. 3, pp. 1271–1281, 2014.
- [46] X. Fan and T. Tjahjadi, "A spatial-temporal framework based on histogram of gradients and optical flow for facial expression recognition in video sequences," *Pattern Recognit.*, vol. 48, no. 11, pp. 3407–3416, 2015.
- [47] P. Ekman, W. V. Friesen, and J. C. Hager, *Facial Action Coding System, the Manual*, Res. Nexus Divis. Netw. Inf. Res. Corp., Salt Lake City, UT, USA, 2002.



Kamlesh Mistry received the B.Sc. degree in computer science from Pune University, Pune, India, and the M.Sc. degree in computer games programming with Teesside University, Middlesbrough, U.K. He is currently pursuing the Ph.D. degree with Northumbria University, Newcastle upon Tyne, U.K.

His current research interests include computer vision, machine learning, and intelligent robotics.



Li Zhang (M'14) received the Ph.D. degree from the University of Birmingham, Birmingham, U.K.

She is currently a Senior Lecturer of Computer Science with Northumbria University, Newcastle upon Tyne, U.K., and is also serving as an Honorary Research Fellow with the University of Birmingham. She holds expertise in artificial intelligence, machine learning, intelligent robotics, and deep learning.

Dr. Zhang has served as an Associate Editor for *Decision Support Systems*.



Siew Chin Neoh received the B.Tech. degree in quality control and instrumentation and the Ph.D. degree in evolutionary computation from the Universiti Sains Malaysia, Gelugor, Malaysia, in 2003 and 2009, respectively.

She is currently an Associate Professor with UCSI University, Kuala Lumpur, Malaysia. Her current research interests include pattern recognition, optimization and control, evolutionary computing, and production planning.



Chee Peng Lim received the B.Eng. degree in electrical engineering from the University of Technology Malaysia, Johor Bahru, Malaysia, in 1992, and the M.Sc. degree in control systems and the Ph.D. degree from the University of Sheffield, Sheffield, U.K., in 1993 and 1997, respectively.

He is currently an Associate Professor with Deakin University, Geelong, VIC, Australia, and has published over 250 technical papers in books, international journals, and conference proceedings. His current research interests include soft computing,

pattern recognition, medical prognosis and diagnosis, fault detection and diagnosis, and condition monitoring.

Ben Fielding is currently pursuing the Ph.D. degree with Northumbria University, Newcastle upon Tyne, U.K.

His current research interests include computer vision, optimization, and deep learning.