



# Facial expression recognition sensing the complexity of testing samples

Tianyuan Chang<sup>1</sup> · Huihui Li<sup>1</sup> · Guihua Wen<sup>2</sup> · Yang Hu<sup>1</sup> · Jiajiong Ma<sup>1</sup>

Published online: 14 June 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Facial expression recognition has always been a challenging issue due to the inconsistencies in the complexity of samples and variability of between expression categories. Many facial expression recognition methods train a classification model and then use this model to identify all test samples, without considering the complexity of each test sample. They are inconsistent with human cognition laws such as the principle of simplicity, so that they are easily under-learned and then are difficult to identify test samples correctly. Hence, this paper proposed a new facial expression recognition method sensing the complexity of test samples, which can nicely solve the problem of the inconsistent distribution of samples complexity. It firstly divided the training data into the hard subset and the easy subset for classification according to the complexity of samples for expression recognition. Subsequently, these two subsets are applied to train two classifiers. Instead of using the same classifier to predict all test samples, our method assigned each test sample to the corresponding classifier based on the complexity of the test sample. The experimental results demonstrated the effectiveness of the proposed method and obtained a significant improvements of the recognition performance on benchmark datasets.

**Keywords** Facial expression recognition · Sample complexity · Convolutional neural network · Gestalt principle

## 1 Introduction

FACIAL expression recognition (FER) has a wide range of research prospects in human computer interaction and affective computing, including polygraph detection, intelligent security, entertainment, Internet education, and intelligent medical treatment [1]. As we all known, facial expression is a

major way of expressing human emotions. Hence, the main task in determining emotion is how to automatically, reliably, and efficiently recognize the information conveyed by facial expressions. In FER research, Ekman and Friesen first proposed the Facial Action Coding System (FACS) [2]. The six basic categories of expressions (surprise, sadness, disgust, anger, happiness, and fear) are defined in FACS, and are commonly used as the basic expression labels. Generally, an application system includes the face detection, the facial expression features extraction, the feature selection, and the method for facial expression recognition [3]. Many work focus on the improvement of feature extraction methods and facial expression recognition methods as two key techniques. They ignored the relevance of several basic expression categories and did not consider the complexity of samples. As described by [4], it is difficult to definitively partition the expression feature space. This is because there are certain overlap among subspaces of some expression categories. Some expressions, such as happiness and surprise, belong to the highly recognizable categories, which are easily distinguished by facial features. However, there are other expressions, such as fear and sadness. They are very similar in some situations, making it difficult to distinguish them effectively. In addition, facial images are easily influenced by ethnicity, age, gender, hair, and other

✉ Huihui Li  
29777562@qq.com

Tianyuan Chang  
tianyuan\_chang@163.com

Guihua Wen  
crghwen@scut.edu.cn

Yang Hu  
superhy199148@hotmail.com

Jiajiong Ma  
mullma@outlook.com

<sup>1</sup> South China University of Technology, Guangzhou 510006, China

<sup>2</sup> Guangdong Research Center of Artificial Intelligence and Traditional Chinese Medicine Engineering, South China University of Technology, Guangzhou 510006, China

uncontrolled factors, resulting in different facial feature distributions and facial feature complexity for classification. On the other hand, the labelled facial expression images are generally hard to be collected, there may be larger inconsistency between the training samples and test samples. In such situation, no matter how suitable the feature extraction method of training samples is in expression recognition, the prediction accuracy of test samples is difficult to be guaranteed. An analogy of this is that we cannot ask an excellent student who has only grasped primary mathematics knowledge to be tested with knowledge of higher mathematics in a university. It is obvious that the complexity of the unequal knowledge needs to be distinguished for learning.

However, current facial expression recognition methods train a classification model and then use this model to identify all test samples, without considering the complexity of each test sample. They are inconsistent with human cognition laws in the real world [5]. This is because samples from the same emotional category may have different degrees of complexity in identifying their emotional categories. Some are easy to be identified, while others are hard to be classified. Secondly, the classifier is trained by mixing the hard samples with the easy samples. The classification boundary of the obtained classification model needs to take into account the majority of training samples. If the hard samples are in the majority, the easy samples will be misclassified. If most of the samples are easy to be classified, the classification boundary tends to classify the easy samples, which makes it difficult to classify the hard samples. Finally, to classify hard samples, the required model is definitely complex, involving many structural factors and super-parameters. In order to find the best model, there must be large training samples, but the number of training samples is generally very small. Therefore, the trained model is under-learned, lacks generalization ability, and may be difficult to identify samples correctly. Actually, Human beings change their methods dynamically based on the complexity of the current test samples, instead of identifying all test samples with the same method. Human thinking follows the principle of simplicity (Gestalt principle) [6]. Simple samples only need simple methods to recognize their emotion categories, while complex samples needs complex methods. Now, there are some methods consider the complexity of the whole dataset [7] and the complexity of the local neighborhood [8]. They did not distinguish the complexity of the sample to be identified.

Hence, this paper proposed a complexity perception classification (CPC) method for facial expression recognition, which is based on the simplicity principle [5]. It firstly divided the training dataset into two parts: the hard subset and the easy subset for classification. The hard subset is composed of samples which are difficult to be classified correctly. The easy subset is composed of samples which are easy to be classified correctly. The division between two parts was performed by evaluating the complexity of samples for expression

recognition. The next step was to separately train two classifiers using these two subsets. Instead of using the same classifier to predict the facial expression of all test samples, our method assigned the test sample to the corresponding classifier to perform facial expression recognition based on the complexity of each testing sample. Our main contributions can be summarized as follows: (1) A simple method is proposed to measure the complexity of the samples. (2) A complexity perception classification (CPC) method was proposed, which not only improved the recognition accuracy of easily recognizable facial expression categories, but also alleviated the problem of some misclassified expression categories. (3) A simple CNN is designed for our method. Furthermore, detailed experiments were designed and conducted. The experimental results demonstrated advantages of CPC over the compared facial expression recognition approaches.

The rest of this paper is organized as follows. In the section II, we briefly introduce some related work. Our proposed method is described in the section III. Section IV presents the experimental results and analysis. Section V summarizes the paper finally.

## 2 Related work

The CPC method proposed in this study is new in the field of emotion recognition. The related literatures are illustrated as follows.

### A. Feature extraction methods

Extracting the discriminative representation from facial expression images is a crucial step that impacts the recognition performance. In the appearance of feature-based methods [9], the typical texture-based methods have been used to extract expression features, including Gabor filters, local binary patterns (LBP) [10], local Gabor binary patterns (LGBP) [11], histograms of oriented gradients (HOG) [12], and scale-invariant feature transform (SIFT) [13]. A hybrid texture features method is also applied to solve discrepancies in different cultures and ethnicities in FER and then improve the recognition performance by adopting the random forest classifier [14]. In addition, appearance features for the recognition of facial expressions by dividing the whole face region into several specific local regions can achieve the better recognition accuracy [15–17]. Khan et al. employ a singular value decomposition (SVD) based co-clustering to search for the most salient regions of facial features which improve a high discriminating ability among all expressions [18]. On the other hand, in the geometry of feature-based methods [19, 20], each expression is decomposed into several facial action units (AUs) through geometric relationships such as landmarks and muscle motions. Inspired by the psychological theory,

Liu et al. introduce an AU-inspired Deep Networks (AUDN) to learn features especially for facial expression recognition which use computational representation MAP to capture the local appearance variations caused by facial expression [21]. These different ways have greatly improved the recognition performance and promoted the application ability of FER.

As the deep convolutional neural network (DCNN) [22–24] have been successfully applied to the computer vision, they have also been applied to FER. For example, Li et al. apply the deep neural network for FER and provide a good performance on CK+ dataset [25]. Yu et al. propose a simple CNN architecture and employ data perturbation and voting method to increase the recognition performance of CNN considerably [26]. Tang et al. replace the softmax layer with linear SVM layer and minimized a standard hinge loss instead of minimization of cross-entropy when use the CNN framework, which has achieved the winner of FER-2013 challenge [27]. Also, [28, 29] employ the model based on transfer features from pre-trained deep CNN, while the ensemble algorithms based on deep learning network [30, 31] have significantly improved recognition performance of FER. In this direction, many research works have made structural improvements of deep convolutional neural networks, taking into account the specific characteristics of facial expression data. As described in [32], Ding et al. present FaceNet2ExpNet which incorporate face domain knowledge to regularize the training of an expression recognition network and construct a new distribution function to capture improved high-level expression semantics. Moreover, the hybrid approach is proposed by combining SIFT and CNN to improve the recognition performance which instruct CNN framework to discriminative learn by adding traditional machine learning knowledge [33]. Another novel deep neural network was proposed to perform multi-view facial expression recognition [34], whose input is the scale invariant feature transforms (SIFT) features. There is also a facial expression recognition method, which combines specific image preprocessing methods and convolution neural network [35]. It extracts only expression-specific features from a face image, and ranks samples for training model. A more powerful method is called deep peak–neutral difference [36]. The difference is defined between deep representations of the fully expressive (peak) and neutral facial expression frames, where unsupervised clustering and semi-supervised methods are applied to automatically obtain the peak and neutral frames. The above methods perform expression recognition using 2D static images. However, their performance is vulnerable to illumination and head posture. Because facial expressions result from facial muscle movement, leading to different facial deformations that can be accurately captured in geometric channels [37, 38]. It indicates that more information of deformation can be obtained from 3D and 4D images. For example, the conditional random forest is applied to capture transition patterns among low-level expressions [39]. When

testing a video frame, pairs are created between the current frame and previous frames, and predictions for each previous frame are applied to draw trees from pairwise conditional random forests (PCRF). The pairwise outputs of PCRF are averaged over time to produce robust estimates. Another complex approach uses a set of radial curves to represent the face using Riemann-based shape analysis tools, and then classify the facial expressions using LDA and HMM [40, 41]. There are also methods for facial expression recognition using 4D face images. For example, multi-kernel learning is applied to combine different channels of facial expression recognition using 4D face images to obtain the final expression label [42, 43]. Deep learning emphasizes the modeling of dynamic shape information of facial expression motion using 4D face images [42–46], where the neural network uses a number of generated geometric images.

Considering the above different strategies, it is essential that exerting the advantages of deep neural networks while combining the face domain knowledge and specific characteristics of data for FER. However, these methods aim to extract the better features from facial images, much different from the idea of our CPC method.

#### B. *Dynamic classifier selection methods*

CPC differs from the current dynamic classifier selection methods. Current dynamic classifier selection methods can be categorized into four types. The first type depends on the classification accuracy of the local neighborhood of the test sample. For example, the overall local accuracy selects the optimal classifier in terms of the accuracy of the classifier in the local neighborhood [47]. Another method is the local class accuracy (LCA) that uses posteriori information to calculate the performance of the base classifier for particular classes [48]. Xiao et al. proposed a dynamic classifier ensemble model. It utilizes the idea of LCA, but the prior probability of each class is used to deal with the imbalanced data [49]. The difference between these methods is that the local information is used in different ways, but they are both based on the local neighborhood of the test sample. The second type, called decision template method, is also based on the local neighborhood, but the local neighborhood is defined in the decision space rather than in the feature space. For example, the k-nearest output profile method first defines the local neighborhood of the test sample in the decision space, and then uses a method to select the classifiers that correctly classified test samples in the neighborhood in order to form an ensemble by voting [50]. Although decision template methods are defined in the decision space, they are still built on the local neighborhood of the test samples. The third type focuses on the selection of candidate classifiers. This can be implemented by selecting training subsets for each candidate classifier [51]. For example, the particle swarm method directly selects a

training set for each candidate classifier using the evolutionary algorithm [52]. The final type uses the local neighborhood features as the training samples for machine learning [53], where features include meta-features of the test samples, the classification accuracy of the neighborhood samples, and the posterior probability of classes of the classified test samples.

These methods do not consider the complexity of the test sample. Our method considers the complexity of the test samples, while it does not calculate the neighborhood of the test sample, so that our method could be global.

### C. Ensemble learning for facial expression recognition

Ensemble learning has been used in facial expression recognition. For example, the fusion of video and audio is applied to recognize emotions [54]. The combination of facial expression data and voice data is utilized to recognize emotions [55]. Another approach combines thermal infrared images and visible light images, using both feature fusion and decision fusion, where a Bayesian network and support vector machine are used [56]. Geometric features and regional LBP features are fused with self-coding, and then a self-organizing mapping network is used to perform expression recognition [57]. By dividing the face image into several regions, the features of each region are extracted such as by LBP and Gabor [58], and then the evidence theory is applied to fuse them [59]. Some methods also use SIFT and deep convolution neural networks to extract features, and then use neural networks to fuse these features [60]. Wen et al. fused multiple convolutional neural network models by predicting the probability of each expression class for the test sample [61]. They also proposed an integrated convolutional echo state networks and a hybrid ensemble learning approach for facial expression classification [62, 63].

The CPC method differs from these ensemble learning methods for emotion recognition. CPC dynamically selects a classifier for the test sample based on the complexity of the test sample. It does not belong to ensemble learning.

### D. Data complexity

Data complexity has been successfully applied in the classification field [64–67], which is appropriate for high dimensionality and small scale datasets. They considered the methods to measure the complicity of the whole samples, instead of the complicity of the single sample, such as Fisher's Discrimination Ratio, and volume of Overlapping Region [64]. Souto et al. [67] compute the measures characterizing the complexity of gene expression datasets for cancer diagnosis, where the statistics of data geometry, topology and shape of the classification boundary are measured. However,

it does not measure the complicity of the single sample. Gui et al. [68] present Curriculum Learning that employ a novel learning technique for deep learning methods. Curriculum Learning rearranges the training samples in terms of the sample complexity levels based on a predetermined curriculum, and then begins with training from easy to hard samples. The method is novel. However, it trained the same model to classify all testing samples without considering the complexity of each test sample. Different from the curriculum learning, our proposed method trained two models and then select the model which is more suitable to classify the testing sample in terms of sample complexity. In addition the measure used in our method to define the sample complexity is also different.

### E. Comparison results of current methods

It can be concluded from literatures mentioned above that the CPC proposed in this study differs from currently available methods. It is the first work considering the complexity of the test samples. The many related methods are compared as follows in more details.

Compared with methods considering the complexity of the local neighborhood [8], our method considers the complexity of the test samples, and there was no need to calculate the neighborhood of the test samples. Since the complexity of the test sample is measured by a classifier trained throughout the whole training samples, it was global in nature to obtain the global optimal performance.

On the other hand, some methods consider the complexity of the entire dataset [7, 64, 67]. They are helpful to perform the model selection, but do not solve the problem of uneven distribution of samples complexity. They are difficult to simultaneously classify both hard samples and easy samples well. In order to classify all test samples, the required model is definitely complex, involving in many structural factors and super-parameters, leading to the trained model being under-learned and difficult to identify test samples correctly. Curriculum Learning is a novel deep learning method [68]. It rearranges the training samples in terms of the sample complexity levels and then begins with training from easy to hard samples. However, this method still trained the same model and then used this model to classify all test samples without considering the complexity of each test sample. It belongs to model selection, but could select the better model. Unfortunately, it still does not solve the problem of uneven distribution of samples complexity.

By contrast, our method can nicely solve the problem of the uneven distribution of samples complexity and then obtain the better performance. This is because our method contains two classifier for expression recognition. One is simple for classifying the easy test samples, whereas the other is complex for

classifying hard test samples. As the simple classifier involves in fewer structural factors and super-parameters, leading to the trained model to be more optimal for classifying the easy test samples. As to the complex classifier, its classification boundary needs to take into account the majority of training samples. When the easy training samples are removed from the whole training data, the classification boundary tends to correctly classify the hard samples.

### 3 Proposed method

Learning from the simplicity principle and the discrepancies among different individual samples for facial expression recognition, a complexity perception classification (CPC) method is proposed for FER. Its framework is presented in Fig. 1, which is composed of preprocessing, feature extraction, training stage, and testing stage. The preprocessing simply performs ZCA Whiten and Global Relative Normalization. The remaining stages will be explained in more details.

#### F. Feature Extraction by CNN

After the preprocessing, the feature extraction method should be applied to extract features for input images. Currently most neural network methods are improved from the classic CNN frameworks, such as ResNet [69] and DenseNet [24]. They not only alleviated the problems that deep networks are prone to gradient disappearance in back-propagation, but also remarkably improved the performance of image classification. Motivated by these methods and considering that small-scale datasets cannot be directly applied to train the above deep neural networks, we designed a simple CNN framework for facial feature extraction as shown in Fig. 2. It contains multiple convolutional layers, modified residual network blocks, and fully connected layers. In order to extract high level facial features, we selected the output of the second

fully connected layer as the feature representation, whose dimension is 1024.

In RestNets, the output of the feature mapping of the residual block consists of a non-linearly transformed composite function  $H(x)$  and an identity function  $x$ , which are combined as in Eq. 1:

$$F(x) = H(x) + x \tag{1}$$

This combination may hinder the flow of information through deep networks [24]. In order to improve the flow of information between layers, we improved the combination mode of the residual block. Motivated by DenseNet, we no longer summated the two inputs, but concatenated the two feature mappings. The output function of the feature mapping is shown in Eq. 2:

$$F(x) = [H(x), x] \tag{2}$$

Figure 3 shows the structure of the traditional residual block and our modified residual block.

#### B. Evaluating the Complexity of samples

After feature extraction, each image sample was represented as a vector. In this way, the training samples can be represented as  $X = [x_1, x_2, \dots, x_n]$  where  $x_i \in R_d, i = 1, 2, \dots, n, x_i$  is the vector of the  $i^{\text{th}}$  sample and  $d$  is the dimension of the vector. In order to evaluate the complexity of samples, a new method was proposed based on the commonsense: if a sample can be correctly predicted by many different classifiers, the sample can be considered as simple. It can be easily categorized. This is consistent with human intuition and cognitive law. Therefore, our proposed evaluation method is reasonable and simple.

In order to implement this new method, the training samples are randomly divided into  $k$  folds. Considering the generalization ability of the base classifier, we first chose a fold of samples as the training set and the remaining  $(k-1)$  folds of samples as the testing set. It resulted in  $k$  base classifiers

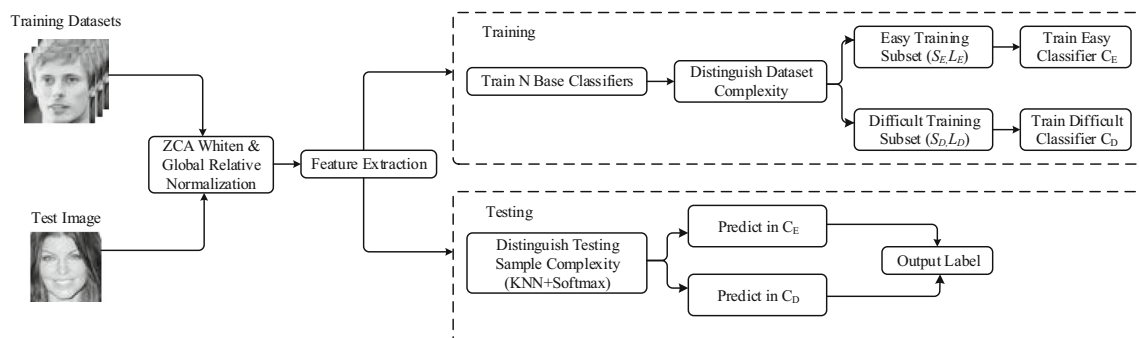


Fig. 1 Framework of the complexity perception classification (CPC) method

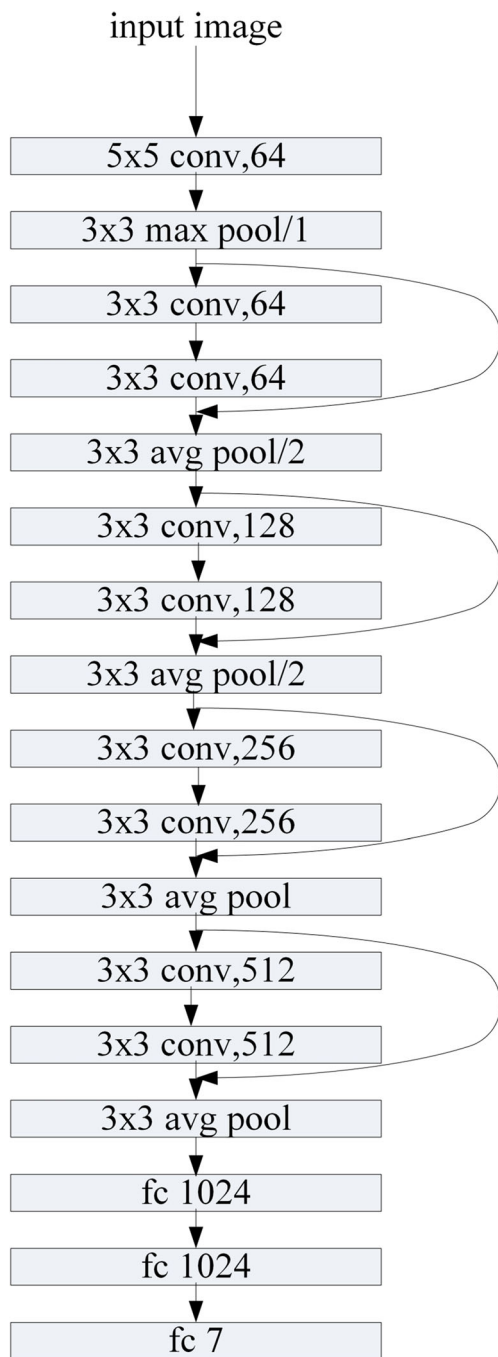


Fig. 2 Framework of the designed CNN

through training on different training sets. And then, the above process was repeated  $m$  times to obtain  $N = km$  trained base classifiers. In this way, each training sample was predicted by these base classifiers and then the correct number of correct prediction was counted, leading to the simplicity of the simplex $_i$  defined as follows:

$$R(x_i) = \frac{N(x_i)}{N} \tag{3}$$

$$N(x_i) = \sum_{j=1}^N c_j(x_i) \tag{4}$$

$$c_j(x) = \begin{cases} 1, & x \text{ is recognized correctly by the } j\text{-th classifier;} \\ 0, & \text{Else.} \end{cases} \tag{5}$$

where  $N(x_i)$  is the number of correct prediction for  $x_i$  by these base classifiers, and  $c_j(x_i)$  is determined by whether  $x$  is recognized correctly by the  $j^{\text{th}}$  classifier.

In addition, we set a parameter named the easy threshold  $\theta$  as the boundary to distinguish the easily classifiable samples from the difficultly classifiable samples.

$$S_E = \{x_i | x_i \in S \wedge R(x_i) \geq \theta\} \tag{6}$$

$$S_D = \{x_i | x_i \in S \wedge R(x_i) < \theta\} \tag{7}$$

According to Eq. 6, when  $R(x_i) \geq \theta$ , the training sample  $x_i$  will be put into the easy classification sample subspace ( $S_E, L_E$ ), where  $S_E$  is the easy classification dataset and  $L_E$  is labels of the easy classification dataset. By contrast, Eq. 7 will be applied to create a difficult classification sample subspace ( $S_D, L_D$ ) where  $S_D$  is the difficult classification dataset and  $L_D$  is its labels.

C. Sample Complexity Discriminator

In order to achieve discriminant learning in different subsets, we trained an easy sample classifier  $C_E$  in the easy classification sample subspace ( $S_E, L_E$ ), and trained a difficult sample classifier  $C_D$  in the difficult classification sample subspace ( $S_D, L_D$ ). They are defined as follows, where  $\xi$  is the specified classification method, such as Softmax, linear SVM, and Random forest.

$$S = S_E \cup S_D, S_E \cap S_D = \emptyset \tag{8}$$

$$C_E = \xi(S_E, L_E) \tag{9}$$

$$C_D = \xi(S_D, L_D) \tag{10}$$

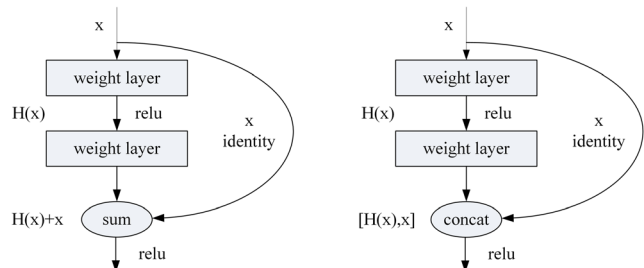


Fig. 3 Left: Traditional residual block. Right: Modified residual block

**Algorithm 1** Complexity perception classification (CPC) for facial expression recognition.

**Input:** Training samples  $\{x_i\}$ , testing sample  $X$ , classifier  $\xi$ , Complexity discriminator  $\psi$ , easy threshold  $\theta$ .  
**Output:** The predictive expression label  $\omega$  for testing sample  $X$ .

**Training:**

- 1: setup parameters  $k, m$ , and  $p$
- 2: **for**  $i = 1$  to  $m$  **do**
- 3:   partition the input training samples into  $k$  folds.
- 4:   **for**  $j = 1$  to  $k$  **do**
- 5:     apply the  $j^{\text{th}}$  fold as the temporal training samples to train  $\xi$  so as to obtain a base classifier.
- 6:   **end for**
- 7: **end for**
- 8: Apply  $N=km$  base classifiers to compute the easy degree  $R(x_i)$  for each input training sample.
- 9: **for** each training sample  $x_i$  **do**
- 10:   **if**  $R(x_i) \geq \theta$
- 11:     Put the sample  $x_i$  into  $S_E$
- 12:   **else**
- 13:     Put the sample  $x_i$  into  $S_D$
- 14:   **end for**
- 15:  $C_E = \xi(S_E, L_E)$
- 16:  $C_D = \xi(S_D, L_D)$
- 17:  $C_s = \psi(S_E \cup S_D, \{+, -\})$

**Testing:**

- 18: Find  $p$  nearest neighbors for  $X$  to train LR so as to obtain  $C_s$
- 19:  $y \leftarrow C_s(X)$
- 20: **if**  $y \in \{+\}$ ,  $\omega \leftarrow C_E(X)$
- 21: **if**  $y \in \{-\}$ ,  $\omega \leftarrow C_D(X)$

In this way, they can learn different feature distributions in two different sample subspaces, and then applied to nicely recognize the facial expression of the testing sample with the corresponding complicity. This need to clearly determine whether the test sample belonged to the easy classification sample subspace or difficult classification sample subspace. Therefore a sample complexity discriminator was proposed that was defined by Eq. 11 where  $\{+\}$  is the label of the easy classification dataset,  $\{-\}$  is the label of the difficult classification dataset.

$$C_s = \psi(S_E \cup S_D, \{+, -\}) \tag{11}$$

The sample complexity discriminator model  $C_s$  was based on a dynamic hybrid model composed of K-Nearest Neighbor (KNN) and Logistic Regression (LR). For each testing sample, it firstly employed KNN to find the  $p$  nearest neighbors from the training samples and used these neighbors with complexity labels to dynamically train a local Logistic Regression classifier, still denoted as LR. This local classifier

is utilized to determine the simplicity of the testing sample with Eq. 12 and 13.

$$LR_{\beta}(x) = \frac{1}{1 + e^{-\beta^T x}} \tag{12}$$

$$C_s(x_i) = \begin{cases} + & LR_{\beta}(x_i) > 0.5 \\ - & \text{Else} \end{cases} \tag{13}$$

where  $LR_{\beta}(x)$  is the probability output of LR,  $\beta^T$  is the parameter of LR and  $C_s(x_i)$  is the complexity label of the testing sample  $x_i$ . If  $x_i$  is predicted as  $\{+\}$  by the complexity discriminator  $C_s$ , the easy sample classifier  $C_E$  will be applied to recognize its facial expression. By contrast, if the label was  $\{-\}$ , the difficult sample classifier  $C_D$  will be applied.

In summary, based on the framework and the technical details discussed above, the whole method for recognizing the facial expression is summarized as in Algorithm 1.

## 4 Experiment

Our proposed algorithm was mainly applied to solve the problem of the static facial expression recognition for the single image, hence we evaluated the performance of the proposed complexity perception classification (CPC) algorithm by do experiments on static benchmark datasets: Fer2013 [70], JAFFE [71] and CK+ [72] datasets.

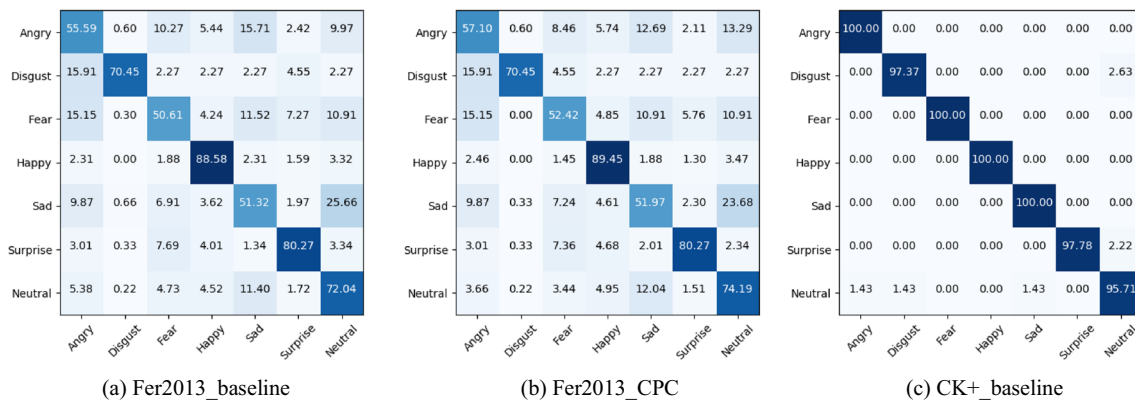
Experiments were performed in three aspects. The different methods for feature extraction were evaluated. The performance of CPC with different types of feature extraction methods was evaluated. Finally, the results of relevant state-of-art methods were compared with that of CPC so as to justify the effectiveness of CPC.

### D. Datasets

The Fer2013 dataset is a facial expression recognition challenge dataset that ICML 2013 launched on Kaggle. The dataset contains 28,709 training images, 3589 validation images, and 3589 test images. The image size is  $48 \times 48$  and facial expression is classified into seven types (0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral).

The CK+ dataset is an extension of the CK dataset. It contains 327 labeled facial videos. Each is labeled into one of the seven categories used in the Fer2013 dataset. We extracted four frames from each sequence in the CK+ dataset, which contains a total of 1308 facial expressions.

The JAFFE database consists of 213 images from 10 Japanese female subjects, with pixel resolution  $256 \times 256$ .



**Fig. 4** Confusion matrices for facial expressions on three datasets. The baseline refers to the case that CNN is taken for feature extraction and Softmax is the classifier. CPC also takes CNN for feature extraction and Softmax as the base classifier

Same as the above databases, each group consists of seven different types of facial expression.

### E. Experimental Settings

#### 1) Pre-Processing

Before the feature extraction phase in the CNN, we standardized the size of all images to  $48 \times 48$  pixels. This is because in three data for experiments, Fer2013 is largest, while the other two data sets are very small. The image size of the sample in Fer2013 dataset is  $48 \times 48$ . In order to use deep convolution neural network to extract image features, a large number of training samples is required. A small number of training samples will lead to the over-fitting, and thus the recognition performance on the test samples is low. For example, we ever used CK+ and Jaffe as training samples to train deep convolution neural network to extract image features, the serious over-fitting occurred, resulting in the much bad recognition performance on the testing samples. Therefore, similar to the method [33], we use Fer2013 to pre-train the deep

convolution neural network, and then convert the size of samples of CK+ and Jaffe into the size of samples in Fer2013, so as to train and test CK+ and Jaffe respectively on the pre-trained model from Fer2013.

In order to further improve the generalization ability and the recognition performance of our model, we randomly perturbed each training sample with additional transforms for data augmentation. The transformations included horizontal flip, randomly shifting horizontally and vertically, randomly rotating with an angle between  $(-30, 30)$ , zooming at the four corners in the range between  $(0.8, 1.2)$ . After that we preprocessed the dataset with ZCA whitening and global relative normalization, which can effectively remove the redundant information of the input image and reduce the correlation between adjacent pixels in the image.

#### 2) CNN for Feature Extraction

In the pre-training phase of CNN feature extraction as shown as in Fig. 1, we used Fer2013 training dataset to train the CNN networks. The initial learning rate was set to 0.05,

**Table 1** Comparison of the recognition accuracies (%) using different feature extraction methods and classifiers on fer2013

Test samples	Easy		Difficult		All	
	$\xi_A$	$\xi_E$	$\xi_A$	$\xi_D$	$\xi_A$	CPC
CNN + Softmax	70.50±0.06	<b>72.74±0.02</b>	62.15±0.14	<b>63.42±0.01</b>	70.34±0.01	<b>71.67±0.01</b>
CNN + LinearSVM	70.36±0.12	<b>71.54±0.01</b>	66.62±0.21	<b>66.90±0.16</b>	70.04±0.02	<b>71.18±0.01</b>
CNN + RandomForest	70.62±0.02	<b>71.76±0.04</b>	60.04±0.54	<b>65.67±0.33</b>	69.83±0.03	<b>70.45±0.02</b>
LGBP+Softmax	43.88±1.41	<b>49.12±0.87</b>	27.58±2.33	<b>40.11±1.74</b>	44.52±0.21	<b>46.35±0.35</b>
LGBP+LinearSVM	44.58±2.48	<b>47.15±0.06</b>	18.32±2.60	<b>25.49±1.31</b>	43.60±0.30	<b>44.72±0.18</b>
LGBP+ RandomForest	39.42±2.19	<b>41.50±2.42</b>	60.21±1.12	<b>65.83±1.34</b>	39.04±1.46	<b>41.04±0.89</b>
HOG+Softmax	52.13±2.44	51.49±1.29	21.26±2.75	<b>24.90±3.11</b>	47.28±0.73	<b>48.22±0.66</b>
HOG+LinearSVM	51.47±0.85	<b>52.96±0.68</b>	25.26±4.31	<b>32.39±2.42</b>	48.23±0.69	<b>49.74±0.41</b>
HOG+ RandomForest	47.55±3.34	<b>48.10±1.50</b>	36.48±3.87	<b>40.52±3.15</b>	47.46±0.87	<b>49.56±0.62</b>



**Table 2** Comparison of the recognition accuracies (%) using different feature extraction methods and classifiers on ck+

Test samples	Easy		Difficult		All	
	$\xi_A$	$\xi_E$	$\xi_A$	$\xi_D$	$\xi_A$	CPC
CNN + Softmax	98.26±0.02	<b>98.72±0.01</b>	99.93±0.00	<b>99.98±0.00</b>	98.58±0.00	<b>99.38±0.00</b>
CNN + LinearSVM	97.88±0.04	<b>98.32±0.02</b>	99.90±0.00	<b>99.96±0.00</b>	97.93±0.01	<b>98.57±0.01</b>
CNN + RandomForest	98.26±0.01	<b>97.69±0.01</b>	80.32±0.53	<b>96.89±0.42</b>	96.87±0.05	<b>98.61±0.03</b>
LGBP+Softmax	97.80±0.01	<b>97.82±0.01</b>	93.02±0.03	<b>93.60±0.02</b>	94.70±0.02	<b>95.25±0.01</b>
LGBP+LinearSVM	99.98±0.00	<b>99.98±0.00</b>	92.38±0.05	<b>93.29±0.04</b>	93.66±0.15	<b>95.18±0.03</b>
LGBP+ RandomForest	68.02±1.37	<b>69.03±0.86</b>	41.21±2.18	<b>58.35±1.18</b>	60.19±0.28	<b>65.04±0.16</b>
HOG+Softmax	96.98±0.03	96.94±0.02	84.79±0.22	<b>85.96±0.19</b>	85.98±0.02	<b>87.51±0.02</b>
HOG+LinearSVM	97.61±0.01	<b>99.92±0.00</b>	89.55±0.13	<b>94.32±0.06</b>	91.74±0.10	<b>94.88±0.03</b>
HOG+ RandomForest	99.89±0.00	<b>99.96±0.00</b>	82.03±0.02	<b>84.54±0.01</b>	83.82±0.04	<b>86.01±0.01</b>

while the decay of the learning rate was 1e-6. The mini batch size was 128, the momentum was set to 0.5, and the dropout was set to 0.5. The activation function used in the convolutional layers was the rectified linear unit (ReLU) activation function. The stochastic gradient descent (SGD) was used as the optimization algorithm. In addition, the last fully connected layer used Softmax as a multi-class activation function.

In order to use CNN model for feature extraction on CK+ and JAFFE datasets, it is better for us to build a larger training database for these two datasets. Hence we firstly combined Fer2013 train dataset, validation dataset, with test dataset to form a new dataset. Then we adopted the dlib face detector to filter out noise samples from this new dataset and then used it as the training dataset to pre-train our CNN model which designed in Fig. 2. Here we used 5-fold cross validation strategy to ensure the stability of the experimental results. We also adopted the same pre-processing strategies and network learning parameters to train this CNN network. Finally, the pre-

trained networks were them fine-tuned respectively on CK+ and JAFFE dataset with following parameters: the Adam optimization algorithm, batch size 32, and the dropout 0.8 to prevent the over-fitting.

### 3) Traditional Feature Extraction

We employed the typical texture-based LGBP [11] and HOG [12] methods to extract facial features as the contrast experiments to our designed CNN. For LGBP feature extraction, we convoluted the original image by a total of 32 filters at 8 scales  $\mu \{7, 9, 11 \dots 17\}$  in 4 directions  $\theta(k\pi/4, k \in \{0 \dots 3\})$ , and applied the uniform LBP to an image split into  $4 \times 4$  grid local regions to extract facial features. In addition, we used PCA algorithm to reduce the dimension to 200 as the final facial features of LGBP.

For HOG feature extraction, we resized the original image and divided the original image into  $6 \times 6 = 36$  cells. Each cell is of  $8 \times 8$  pixel size. The gradient direction was divided

**Table 3** Comparison of the recognition accuracies (%) using different feature extraction methods and classifiers on JAFFE

Test samples	Easy		Difficult		All	
	$\xi_A$	$\xi_E$	$\xi_A$	$\xi_D$	$\xi_A$	CPC
CNN + Softmax	94.85±0.01	<b>97.09±0.01</b>	89.73±0.01	<b>93.21±0.02</b>	97.75±0.01	<b>98.97±0.01</b>
CNN + LinearSVM	84.65±0.01	<b>88.42±0.00</b>	94.12±0.00	<b>96.08±0.00</b>	96.31±0.01	<b>97.69±0.01</b>
CNN + RandomForest	85.63±0.05	<b>92.64±0.01</b>	71.50±0.13	<b>79.22±0.02</b>	93.20±0.02	<b>95.48±0.01</b>
LGBP+Softmax	99.92±0.00	<b>100.00±0.00</b>	75.38±0.02	<b>76.55±0.01</b>	75.69±0.01	<b>76.56±0.00</b>
LGBP+LinearSVM	100.00±0.00	<b>100.00±0.00</b>	80.06±0.01	<b>80.55±0.00</b>	81.22±0.00	<b>81.82±0.01</b>
LGBP+ RandomForest	77.69±0.15	<b>83.55±0.06</b>	69.72±0.39	<b>71.68±0.02</b>	70.77±0.08	<b>73.30±0.02</b>
HOG+Softmax	–	–	61.79±0.21	<b>63.91±0.03</b>	61.79±0.21	<b>63.91±0.03</b>
HOG+LinearSVM	–	–	73.70±0.01	<b>74.67±0.01</b>	73.70±0.01	<b>74.67±0.01</b>
HOG+ RandomForest	–	–	57.48±0.38	<b>60.66±0.03</b>	57.48±0.38	<b>60.66±0.03</b>

**Table 4** Comparison of the recognition accuracies (%) using different neural networks for feature extraction on Fer2013

Test samples	Easy		Difficult		All	
	$\xi_A$	$\xi_E$	$\xi_A$	$\xi_D$	$\xi_A$	CPC
VGG19	70.35±0.01	<b>71.12±0.02</b>	45.50±1.26	<b>45.57±1.44</b>	69.78±0.04	<b>70.53±0.03</b>
ResNet_basic_32	67.44±0.01	<b>68.82±0.03</b>	41.90±2.24	<b>50.41±2.46</b>	66.47±0.02	<b>68.12±0.05</b>
ResNet_basic_56	66.31±0.06	<b>68.58±0.02</b>	57.12±1.44	<b>59.71±0.85</b>	65.33±0.02	<b>67.89±0.03</b>
ResNet_v2_32	69.78±0.01	<b>70.84±0.01</b>	44.30±2.69	<b>50.19±2.19</b>	69.50±0.03	<b>70.67±0.02</b>
ResNet_v2_56	68.99±0.00	<b>69.69±0.00</b>	40.50±3.35	<b>60.54±2.36</b>	68.78±0.01	<b>69.95±0.02</b>
Wide_ResNet_28_10	68.15±0.00	<b>69.45±0.04</b>	49.78±0.15	<b>51.22±0.24</b>	65.62±0.06	<b>66.90±0.03</b>

averagely into nine regions and the original image was divided into blocks of size  $5 \times 5 = 25$ . Finally, HOG created features for each facial image.

#### F. Baseline Classification vs. CPC

We firstly observed the impact of CPC on the recognition accuracy of different expression categories using experiments. Figure 4 illustrated the confusion matrices of testing accuracies (%) on different datasets, where five-fold cross validation was employed to ensure the stability of results. The effectiveness of CPC was verified, where CNN was taken for feature extraction.

From Fig. 4 (a), we can clearly see that the recognition rates of happiness in Fer2013 were significantly higher than that of other expressions, illustrating that it belongs to the easily distinguishable category. Meanwhile, the recognition rates of fear were the most difficult to be distinguished. From Fig. 4 (b), it can be observed that the effectiveness of CPC is obvious. The recognition rate of happiness is increased by 0.87%, the recognition rate of fear similarly is increased by 1.81%. The error rate due to mistaking fear for sadness is decreased by 0.61%. Similarly, the recognition rates of anger, sadness, and neutral are also increased. The experimental results illustrated that CPC improved the recognition rates of most categories while it did not decrease the recognition rates of other categories.

As to CK+, it can be seen from Fig. 4 (c) and (d) that the recognition accuracy of the disgust class is increased from 97.37% to 100% and the surprise class is increased from



**Fig. 5** Testing images that both Softmax and CPC misclassified

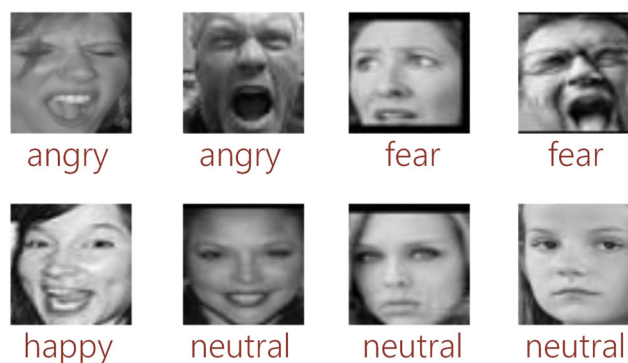
97.78% to 100%, even if the baseline classifier had performed well in these two cases. On the other hand, it also increased the recognition accuracy of natural class by 3.5%.

As to JAFFE, we can see from Fig. 4 (e) and (f) that the recognition accuracies of surprise, happiness and disgust were significantly improved, while the error rate due to mistaking happiness for neutral was decreased by 13.13%. The other categories kept unchanged as they had reached accuracies with 100%.

These exciting recognition results demonstrated that the proposed algorithm not only improved the recognition accuracies of easily distinguishable classes, but also alleviated the easy misclassification rates of difficultly distinguishable classes. Particularly, the accuracies of any expression class on all datasets was not discovered to be decreased when CPC was used. These illustrated that our proposed algorithm did not raise the recognition accuracies of certain classes at the cost of sacrificing the accuracies of other classes. It is meaningful for practical application in facial expression recognition.

#### G. Performance of the CPC Algorithm

In order to prove that CPC was effective in most cases, more experiments were conducted in the cases that different feature extraction methods and the base classifiers were used. Table 1, 2, and 3 showed the recognition accuracies of different feature extraction methods with different base classifiers



**Fig. 6** Testing images which Softmax misclassified but CPC correctly recognized

**Table 5** Recognition rate (%) on CK+ dataset for state-of-the-art methods

Methods	Recognition rate (%)
Ali et al. [73]	93.20
Hui et al. [32]	98.60
Chongliang et al. [74]	98.54
S L Happy et al. [75]	94.09
Yaxin et al. [1]	94.87
Haibin et al. [76]	96.60
Asit et al. [77]	98.30
Mundher et al. [33]	99.10
Evangelos et al. [78]	96.02
Proposed Approach	<b>99.38</b>

on three datasets, including the recognition accuracies of CPC. Table 4 showed the recognition performance of different neural networks feature extraction methods and CPC. In all experiments, we repeated the five-fold cross validation on different data sets to calculate the mean and variance, aiming to verify the performance and robustness of the proposed CPC algorithm. In Table 1,  $71.67 \pm 0.01$  indicates that the average accuracy of CPC is 71.67% and the variance is 0.01.

In tables,  $\xi_A$  was the base classifier trained on the whole training dataset,  $\xi_E$  was the base classifier trained on easy classification samples, and  $\xi_D$  was the base classifier trained on difficult classification samples, where  $\xi$  was the base classifier taken from Softmax, LinearSVM and RandomForest respectively. In Table 1, the value located at the cross point between 3th line and 2th column indicates the classification accuracy that  $\xi_A$  performed for the easy testing samples, where  $\xi$  is Softmax and the feature extraction method is CNN. Similarly, the value located at the cross point between 3th line and 3th column indicates the classification accuracy that  $\xi_E$  performed for the easy testing samples, where  $\xi$  is Softmax and the feature extraction method is CNN. The meaning of the other items in Table are similar.

**Table 6** Recognition rate (%) on JAFFE dataset for state-of-the-art methods

Methods	Recognition rate (%)
S L Happy et al. [75]	91.8
Xiaoguang et al. [79]	87.73
Asit et al. [77]	94.9
Mengyi et al. [80]	96.24
André et al. [71]	82.10
Arfan et al. [14]	92.38
Waleed et al. [81]	94.30
Proposed Approach	<b>98.97</b>

It can be observed from Table 1 on Fer2013 that  $\xi_D$  outperformed  $\xi_A$  for classifying difficult classification samples in any case that different feature extraction methods and base classifiers were combined. It is generally regarded that a larger training data is more useful for a classifier. However, the comparison results here illustrated that the simple samples added to the training data did actually degraded the performance of the classifier. On the other hand, the recognition rate of  $\xi_E$  was better than that of  $\xi_A$  on most cases, which was increased by 1.25% on the average. These results showed that the testing samples can be recognized better when the classifier was trained on the samples with the corresponding complexity of the test samples, instead of trained on the whole training dataset. As to the whole testing samples, our proposed method obtained the better performance than the compared methods. CNN + Softmax obtained the best baseline performance, from which our proposed method was still improved by 1.33%. It can be also seen that CNN was much better than traditional feature extraction methods.

In order to visually explain why our method is better, we provide some visualization results. We randomly selected 150 face images from Fer2013 test samples. The classification accuracy of Softmax is 73.33%, and that of CPC is 80.67%, where CNN is taken to extract features. Obviously, there are misclassified samples for both classifiers, shown as Fig. 5, indicating that the complexity and diversity of human emotional expression. It may be hard for human being to recognize them correctly. Simultaneously, we select some images which Softmax misclassified but CPC correctly recognized. As shown in Fig. 6, it can be observed that CPC can improve the recognition rate of categories such as angry, fear and neutral expressions.

On CK+, the experiments were conducted based on the five-fold cross validation. It can be seen from Table 2 that  $\xi_D$  significantly outperformed  $\xi_A$  for classifying difficult classification samples at most cases that different feature extraction methods and base classifiers were combined, with the improvement by 5.56% on the average. It can be observed that most methods performed well on this data set. However, the accuracy of our proposed method is still higher than that of the

**Table 7** Recognition rate (%) on Fer2013 dataset for state-of-the-art methods

Methods	Recognition rate (%)
Unsupervised [70]	69.26
Ali et al. [73]	66.40
RBM(Tang) [27]	71.16
Subnet Ensemble [26]	65.03
DNNRL [82]	70.60
FC3072 [83]	70.58
Mundher et al. [33]	<b>73.40</b>
Proposed Approach	71.67

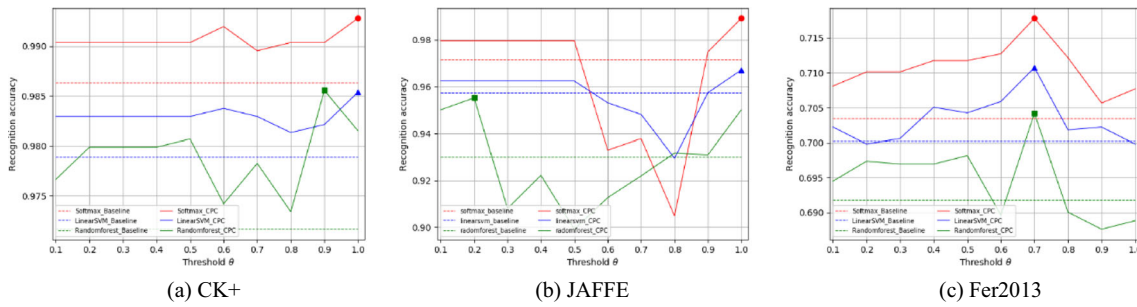


Fig. 7 The recognition accuracies of our method and compared methods with CNN as the feature extraction method

compared methods by 1.76% on the average for classifying the entire test samples. Particularly, when CNN and Softmax were used, our method obtained the accuracy up to 99.38%, which exceeded the other state-of-the-art approaches.

classifying the whole test samples. Particularly, when CNN and Softmax were used, our method obtained the accuracy up to 99.38%, which exceeded the other state-of-the-art approaches. Compared with traditional feature extraction methods, CNN is still more excellent to our method.

Table 3 showed the experimental results on JAFFE based on the five-fold cross validation strategy. It was also found that  $\xi_D$  outperformed  $\xi_A$  in most cases for difficult samples. Similarly, the recognition rate of  $\xi_E$  was better than  $\xi_A$  on most methods, except the case that the feature extraction of HOG was used. As this data is smaller, easy test samples cannot be available from the whole test samples when HOG was used as the feature extraction method. The symbol – in Table 3 denotes the cases. However, in the whole training samples, the simple training samples can be still selected. Therefore, our proposed method had still obtained better results than the

compared methods. Particularly, when CNN and Softmax were used, our method obtained the accuracy up to 98.97%, which exceeded the other state-of-the-art approaches. Compared with traditional feature extraction methods, CNN is still more excellent to our method.

It can be observed from Tables 1, 2, and 3 that the recognition accuracies of  $\xi_E$  was higher than that of  $\xi_A$  by 1.6% on the average and the recognition accuracies of  $\xi_D$  was higher than that of  $\xi_A$  by 4.3% on the average. This means that a sample can

be more accurately classified when its complexity is considered. It provided the strong reasons for that our proposed method performed better than the compared methods.

In order to prove the effectiveness of CPC on some new deep learning methods, some experiments are conducted to compare these deep learning methods with our method. The experimental data are Fer2013, and five-fold cross validation is used. The experimental results are shown in Table 5. It can be observed that our method still performs better than the compared ones while the variance of the 5-fold cross validation experiments is very small. For example, compared with

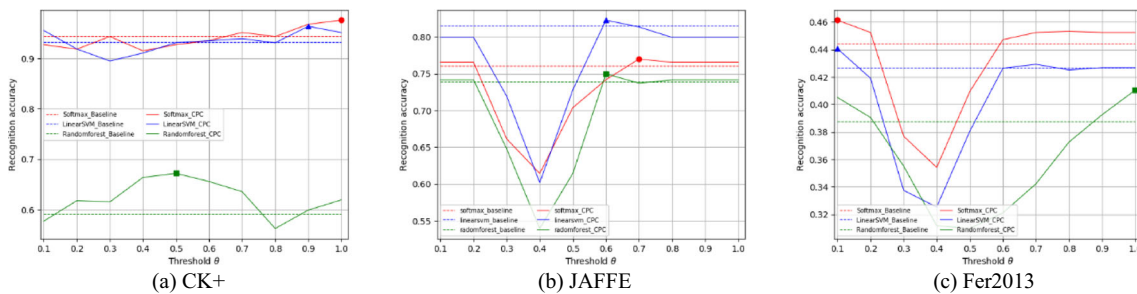


Fig. 8 The recognition accuracies of our method and compared methods with LGBP as the feature extraction method

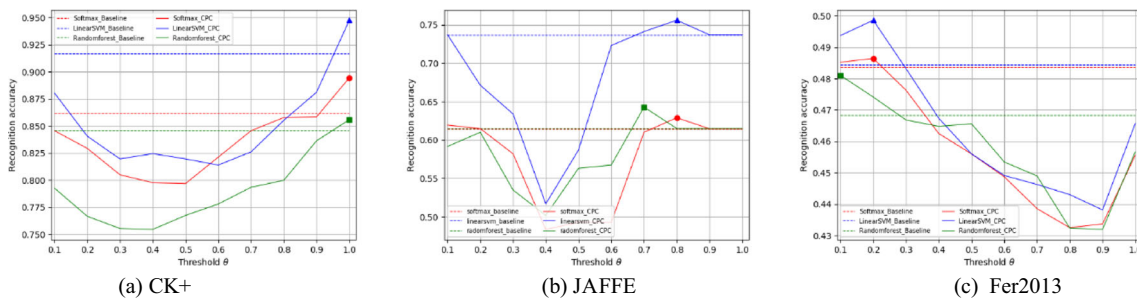


Fig. 9 The recognition accuracies of our method and compared methods with HOG as the feature extraction method

ResNet\_v2\_32 deep convolution neural network with the best performance, the classification accuracy of our method increases by 1.17%, while the variance decreases, indicating that our method is more stable.

#### H. Comparison of State-of-the-art Methods

As there are many state-of-the-art methods for FER, it is better to make the comparison between them and our proposed method. They performed the same experiments on these benchmark datasets. Tables 5, 6, and 7 respectively show the recognition accuracies of the different FER methods as well as our proposed methods on CK+, JAFFE and Fer2013 datasets. As can be seen in Table 5, the recognition accuracy of our proposed method outperforms all other compared methods in terms of the mean recognition accuracy on the CK+ dataset. Similar results can also be observed from Table 6 that the recognition accuracy of our proposed method is better than all other existing approaches on the JAFFE dataset. As shown in Table 7 on Fer2013 dataset, our proposed method obtained the second best result, just worse than the result obtained in [33]. The latter method developed the better feature extraction method. It can be expected that our method can use this new feature extraction method to further improve the performance.

##### I. Parameter Analysis

In our approach, there is an ease threshold  $\theta$  parameter. Experiments are conducted to investigate its importance. In order to determine the parameter value that CPC obtains the best performance of FER, we employed five-fold cross validation on three datasets. Figures 7, 8, and 9 respectively showed the recognition accuracy of our method versus different values of the ease threshold  $\theta$  and different feature extraction methods, where the different baseline classifiers are considered.

Figure 7 showed that the performance of CPC algorithm with the increment of the ease threshold  $\theta$  by using CNN feature extraction framework, where the different baseline classifiers are considered. Our proposed algorithm performed better than the baseline at the most values of the ease threshold  $\theta$ , which illustrated that our method is robust and the parameter can be selected easily. Compared to LinearSVM and RadomForest classifiers, Softmax performed the best on three datasets. In addition, the relationship between recognition accuracies and values of  $\theta$  was similar for the linear SVM and Softmax classifiers.

Figures 8 and 9 showed the recognition rates versus the parameter of ease threshold  $\theta$  by using LGBP and HOG feature extraction methods respectively. Our proposed algorithm only performed better than the base method with a few ease thresholds, illustrating that it was hard to find an appropriate parameter of ease threshold  $\theta$  when LGBP and HOG feature extraction methods are applied. It also further illustrated that our

method heavily depends on the used feature extraction method. Fortunately, the more efficient the feature extraction method, the better our proposed method.

## 5 Conclusion

In order to improve the performance of FER, this paper proposed a complexity perception classification (CPC) method based on the simplicity principle. We also evaluated the performance of our proposed algorithm on three datasets by using three feature extraction frameworks. It was observed from the experimental results that our algorithm performed best on all datasets. It was also observed that the performance of our method using CNN was generally much better than that using traditional feature extraction methods. Thirdly, our proposed method not only improved the recognition accuracy of easily distinguishable classes, but also that of difficult distinguishable classes. Namely, it did not raise the recognition accuracy of certain classes at the cost of sacrificing the accuracy of other classes.

As our proposed method was general, it can be expected that our method can use the better feature extraction methods to further improve the recognition performance. In the future, most strong feature extraction methods will be considered [33]. For example, the energy function of image reconstruction consists of both the energy of image gradient in low density region and the image total variation in high density region [84]. This idea can be applied to design the better loss function for the convolutional neural network so as to extract the better features for the facial images. On the other hand, there are some good methods used in sensor networks, such as method detecting holes boundaries [85] and the method using the heat diffusion equation for the navigation [86]. They provide good inspiration to us. Supposed that we regard samples as sensors in sensor networks, we can construct sample networks. Subsequently, the methods used in sensor networks can be adapted to our sample networks. Finally, our method only classified the complexity of the samples into the simple category and the difficult category. In the future work, more categories of the complexity of the samples will be applied to further improve the performance of our method.

## 6 Acknowledgements

This study was supported by China National Science Foundation (Grant Nos. 60,973,083 and 61,273,363), Science and Technology Planning Project of Guangdong Province (Grant Nos. 2014A010103009 and 2015A020217002), and Guangzhou Science and Technology Planning Project (Grant No. 201604020179, 201,803,010,088).

## References

- Sun Y, Wen G (2017) Cognitive facial expression recognition with constrained dimensionality reduction. *Neurocomputing* 230(2016): 397–408
- Friesen WV, Ekman P (1983) EMFACS-7: Emotional Facial Action Coding System
- Siddiqi MH (2018) Accurate and robust facial expression recognition system using real-time YouTube-based datasets[J]. *Appl Intell* 48(9):2912–2929
- Lopes AT, de Aguiar E, De Souza AF, Oliveira-Santos T (2017) Facial expression recognition with convolutional neural networks: coping with few data and the training sample order. *Pattern Recogn* 61:610–628
- Wen G, Wei J, Wang J, Zhou T, Chen L (2013) Cognitive gravitation model for classification on small noisy data. *Neurocomputing* 118:245–252
- Baruchello G (2015) A classification of classic, gestalt psychology and the tropes of rhetoric. *New idea Psychol* 26:10–24
- Smith MR, Martinez T, Giraud-Carrier C (2014) An instance level analysis of data complexity. *Mach Learn* 95:7225–7256
- Brun AL, Britto AS Jr, Oliveira LS, Enembreck F, Sabourin R (2018) A framework for dynamic classifier selection oriented by the classification problem difficulty[J]. *Pattern Recogn* 76:175–190
- Wang Z, Ruan Q, An G (2016) Facial expression recognition using sparse local fisher discriminant analysis. *Neurocomputing* 174:756–766
- Savran A, Cao H, Nenkova A, Verma R (2015) Temporal Bayesian Fusion for Affect Sensing: Combining Video, Audio, and Lexical Modalities. *IEEE Trans. Cybern*
- Zhang W, Shan S, Gao W, Chen X, Zhang H (2005) Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A novel non-statistical model for face representation and recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*
- Dahmane M, Meunier J (2011) Emotion recognition using dynamic grid-based HoG features. in *2011 IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, FG 2011*
- Berretti S, Ben Amor B, Daoudi M, Del Bimbo A (2011) 3D facial expression recognition using SIFT descriptors of automatically detected keypoints. *Vis Comput*
- Jaffar MA (2017) Facial expression recognition using hybrid texture features based ensemble classifier. *Int J Adv Comput Sci Appl* 8(6):449–453
- Ghimire D, Jeong S, Lee J, Park SH (2017) Facial expression recognition based on local region specific features and support vector machines. *Multimed Tools Appl* 76(6):7803–7821
- Lajevardi SM, Hussain ZM (2012) Automatic facial expression recognition: feature extraction and selection. *Signal, Image Video Proc* 6(1):159–169
- Shan C, Gritti T (2008) Learning discriminative LBP-histogram bins for facial expression recognition. *Proc Br Mach Vis Conf*: 27.1–27.10
- Khan S, Chen L, Yan H (2017) Co-clustering to reveal salient facial features for expression recognition. *IEEE Trans Affect Comput* 3045(c):1–14
- Pantic M, Patras I (2006) Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Trans Syst Man, Cybern Part B Cybern* 36(2):433–449
- Tong Y, Liao W, Ji Q (2007) Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans Pattern Anal Mach Intell* 29(10):1683–1699
- Liu M, Li S, Shan S, Chen X (2015) AU-inspired deep networks for facial expression feature learning. *Neurocomputing* 159(1):126–136
- Ranzato M, Susskind J, Mnih V, Hinton G (2011) On deep generative models with applications to recognition. *Cvpr* 2011:2857–2864
- Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F (2009) ImageNet: a large-scale hierarchical image database. *Cvpr*:248–255
- Huang G, Liu Z, Maaten LVD, Weinberger KQ (2017) Densely connected convolutional networks. *2017 IEEE Conf Comput Vis Pattern Recognit*:2261–2269
- Li J, Lam EY (2015) Facial expression recognition using deep neural networks. *Imaging Syst Tech (IST), 2015 IEEE Int Conf*:1–6
- Yu Z, Zhang C (2015) Image based static facial expression recognition with multiple deep network learning. *Proc 2015 ACM Int Conf Multimodal Interact - ICMI '15*:435–442
- Tang Y (2013) Deep learning using linear support vector machines. *Comput Therm Sci*
- Xu M, Cheng W, Zhao Q, Ma L, Xu F (2015) Facial expression recognition based on transfer learning from deep convolutional networks. *2015 11th Int Conf Nat Comput*:702–708
- Ng H-W, Nguyen VD, Vonikakis V, Winkler S (2015) Deep learning for emotion recognition on small datasets using transfer learning. *Proc 2015 ACM Int Conf Multimodal Interact - ICMI '15*:443–449
- Li D, Wen G (2017) MRMR-based ensemble pruning for facial expression recognition. *Multimed Tools Appl*
- Li D, Wen G, Hou Z, Huan E, Hu Y, Li H (2018) RTCRelief-F: an effective clustering and ordering-based ensemble pruning algorithm for facial expression recognition. *Knowl Inf Syst*:1–32
- Ding H, Zhou SK, Chellappa R (2017) “FaceNet2ExpNet: Regularizing a Deep Face Recognition Net for Expression Recognition,” *Proc. - 12th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2017 - 1st Int. Work. Adapt. Shot Learn. Gesture Underst. Prod. ASL4GUP 2017, Biometrics Wild, Bwild 2017, Heteroge*, pp. 118–126
- Al-Shabi M, Cheah WP, Connie T (2016) Facial Expression Recognition Using a Hybrid CNN– SIFT Aggregator. *Int Work Multi-disciplinary Trends Artif Intell*
- Zhang T, Zheng W, Cui Z, Zong Y, Yan J (2016) A deep neural network-driven feature learning method for multi-view facial expression recognition [J]. *IEEE Trans Multimed* 18(12):2528–2536
- Lopes AT, Aguiar ED, Souza AF, Oliveira-Santos T (2017) Facial expression recognition with convolutional neural networks: coping with few data and the training sample order [J]. *Pattern Recogn* 61:610–628
- Chen J, Xu R, Liu L (2018) Deep peak-neutral difference feature for facial expression recognition[J]. *Multimed Tools Appl*. <https://doi.org/10.1007/s11042-018-5909-5>
- Fang T, Zhao X, Ocegueda O, Shah SK, Kakadiaris IA (2011) 3D facial expression recognition: a perspective on promises and challenges[C]. *IEEE Int Conf Autom Face Gesture Recog* 28:603–610
- Zhen Q, Huang D, Wang Y, Chen L (2016) Muscular movement model-based automatic 3D/4D facial expression recognition[J]. *IEEE Trans Multimed* 18(7):1438–1450
- Dapogny A, Bailly K, Dubuisson S (2017) Dynamic pose-robust facial expression recognition by multi-view pairwise conditional random forests [J]. *IEEE Trans on Affect Comput* 99:1–14
- Drira H, Ben Amor B, Daoudi M, Srivastava A, Berretti S (2012) 3D dynamic expression recognition based on a novel deformation vector field and random Forest[C]. *IEEE Int Conf Patt Recog*:1104–1107
- Ben Amor B, Drira H, Berretti S, Daoudi M, Srivastava A (2017) 4D facial expression recognition by learning geometric deformations[J]. *IEEE Trans Cybernet* 44(12):2443–2457
- Yao Y, Huang D, Yang X, Wang Y, Chen L (2018) Texture and Geometry Scattering Representation based Facial Expression Recognition in 2D+3D Videos [J], *ACM Transactions on Multimedia Computing and Applications*
- Joan B, Stephane M (2013) Invariant scattering Nonvolution networks[J]. *IEEE Trans Pattern Anal Mach Intell* 35(8):1872–1886
- Yang X, Huang D, Wang Y, Chen L (2015) Automatic 3D Facial Expression Recognition using Geometric Scattering Representation[C]. *IEEE International Conference on Automatic Face and Gesture Recognition*
- Liu Y, Zeng J, Shan S, Zheng Z (2018) Multi-channel pose-aware convolution neural networks for multi-view facial expression

- recognition[C], 13th IEEE International Conference on Automatic Face & Gesture Recognition
46. Li W, Huang D, Li H, Wang Y (2018) Automatic 4D Facial Expression Recognition using Dynamic Geometrical Image Network[C], 13th IEEE International Conference on Automatic Face & Gesture Recognition
  47. Mendialdua I, Martínez-Otzeta JM, Rodríguez-Rodríguez I, Ruiz-Vázquez T, Sierra B (2015) Dynamic selection of the best base classifier in one versus one[J]. *Knowl-Based Syst* 85:298–310
  48. Didaci L, Giacinto G, Roli F, Marcialis GL (2005) A study on the performances of dynamic classifier selection based on local accuracy estimation[J]. *Pattern Recogn* 38(11):2188–2191
  49. Xiao J, Xie L, He C, Jiang X (2012) Dynamic classifier ensemble model for customer classification with imbalanced class distribution[J]. *Expert Syst Appl* 39:3668–3675
  50. Cavalin PR, Sabourin R, Suen CY (2012) Logid: an adaptive framework combining local and global incremental learning for dynamic selection of ensembles of HMMs[J]. *Pattern Recogn* 45(9):3544–3556
  51. Szeppanek G, Bischl B, Weihs C (2009) On the combination of locally optimal pairwise classifiers [J]. *Eng Appl Artif Intell* 22:79–85
  52. de Souza BF, de Carvalho A, Calvo R, Ishii RP (2006) Multiclass svm model selection using particle swarm optimization[C]. Sixth Int Conf Hybrid Intel Syst, IEEE:31
  53. Rafael MO (2015) Cruz, Robert Sabourin, George D.C. Cavalcanti, Tsang Ing Ren, META-DES: a dynamic ensemble selection framework using META-learning [J]. *Pattern Recogn* 48:1925–1935
  54. Xu C, Du PF, Feng ZY, Meng ZP, Cao TY, Dong CC (2013) Multimodal emotion recognition fusing video and audio [J]. *Appl Math Inform Sci* 7(2):455–462
  55. Wang Y, Yang X, Zou J (2013) Research of emotion recognition based on speech and facial expression[J]. *Inst Adv Eng Sci* 11(1):83–90
  56. Wang SF, He S, Wu Y, He MH, Ji Q (2014) Fusion of visible and thermal images for facial expression recognition [J]. *Front Comput Sci* 8(2):232–242
  57. Majumder A, Behera L, Subramanian VK (2018) Automatic facial expression recognition system using deep network-based data fusion [J]. *IEEE Trans Cybernet* 48(1):103–114
  58. Sun YC, Yu J (2017) Facial expression recognition by fusing Gabor and local binary pattern features [J]. *Multimed Model* 10133:209–220
  59. Wang WC, Chang FL, Liu YL, Wu XJ (2017) Expression recognition method based on evidence theory and local texture [J]. *Multimed Tools Appl* 76(5):7365–7379
  60. Sun B, Li LD, Zhou GY et al (2016) Facial expression recognition in the wild based on multimodal texture features [J]. *J Electron Imaging* 25(6)
  61. Wen GH, Hou Z, Li HH, Li DY, Jiang LJ, Xun EY (2017) Ensemble of deep neural networks with probability-based fusion for facial expression recognition [J]. *Cogn Comput* 9(5):597–610
  62. Wen GH, Li HH, Li DY (2015) An ensemble convolutional echo state networks for facial expression recognition [C]. 2015 International Conference on Affective Computing and Intelligent Interaction (ACII), Xian, China 873–878
  63. Li D, Wen G, Hou Z, Huan E, Hu Y, Li H (2018) RTCRelief-F: An effective clustering and ordering-based ensemble pruning algorithm for facial expression recognition[J]. *Knowl Inf Syst*:1–32
  64. Sun M, Liu K, Hong Q (2017) An ECOC approach for microarray data classification based on minimizing feature related complexities. 10th Int Symp Comput Intell Des 3:300–303
  65. Pujol O, Radeva P, Vitrià J (2006) Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes. *IEEE Trans Pattern Anal Mach Intell* 28(6):1007–1012
  66. Mansilla EB, Ho TK (2004) On classifier domains of competence. *Proc - Int Conf Pattern Recognit* 1:136–139
  67. de Souto MCP, Lorena AC, Spolaor N, Costa IG (2010) Complexity measures of supervised classifications tasks: a case study for cancer gene expression data. *Int Jt Conf Neural Networks*:1–7
  68. Gui L, Baltrusaitis T, Morency L-P (2017) Curriculum learning for facial expression recognition. 12th IEEE Int Conf Autom Face Gesture Recognit:505–511
  69. Wu S, Zhong S, Liu Y (2017) Deep residual learning for image steganalysis. *Multimed Tools Appl*:1–17
  70. Goodfellow IJ et al (2015) Challenges in representation learning: a report on three machine learning contests. *Neural Netw* 64:59–63
  71. Lyons MJ (1999) Automatic classification of single facial images. *IEEE Trans Pattern Anal Mach Intell* 21(12):1357–1362
  72. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The extended cohn-kande dataset (CK+): a complete facial expression dataset for action unit and emotions pecified expression. *Cvprw*:94–101
  73. Mollahosseini A, Chan D, Mahoor MH (2016) Going deeper in facial expression recognition using deep neural networks. *IEEE Winter Conf Appl Comput Vis* 1:–10
  74. Wu C, Wang S (2015) Multi-instance hidden Markov model for facial expression recognition. *Int Conf Autom Face Gesture Recog*
  75. Happy SL, Routray A (2015) Automatic facial expression recognition using features of salient facial patches. *IEEE Trans Affect Comput*
  76. Yan H (2018) Collaborative discriminative multi-metric learning for facial expression recognition in video. *Pattern Recogn* 75: 1339–1351
  77. Barman A, Dutta P (2017) Facial expression recognition using distance and shape signature features. *Pattern Recogn Lett* 0:1–8
  78. Sariyanidi E, Gunes H, Cavallaro A (2017) “Learning Bases of Activity for Facial Expression Recognition,” *IEEE Trans. Image Process*
  79. Chen X, Yang X, Wang M, Zou J (2017) Convolution neural network for automatic facial expression recognition. *Proc IEEE Int Conf Appl Syst Innov Appl Syst Innov Mod Technol ICASI 2017* 814–817
  80. Liu M, Shan S, Wang R, Chen X (2014) Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition
  81. Deaney W, Venter I, Ghaziasgar M, Dodds R (2017) A comparison of facial feature representation methods for automatic facial expression recognition. *Proc South African Inst Comput Sci Inf Technol* 10:1–10
  82. Guo Y, Tao D, Yu J, Hao X, Li Y, Tao D (2016) Deep Neural Networks with Relativity Learning for facial expression recognition,” in 2016 IEEE International Conference on Multimedia and Expo Workshop, ICMEW
  83. Kim B-K, Roh J, Dong S-Y, Lee S-Y (2016) Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *J Multimodal User Interfaces*
  84. Wei W, Yang X-L, Zhou B et al (2012) Combined energy minimization for image reconstruction from few views. *Math Probl Eng*
  85. Wei W, Yang X-L, Shen P-Y et al (2012) Holes detection in anisotropic Sensemets: topological methods. *Int J Distribut Sensor Netw*
  86. Wei W, Qi Y (2011) Information potential fields navigation wireless adoc sensor networks. *Sensors* 11(5):4794–4807



**Tianyuan Chang** is currently a master candidate in the College of Computer Science and Engineering, South China University of Technology. His main research interests include image recognition, facial expression recognition and deep learning.



**Yang Hu** received the MA.Eng. degree from Kunming University of Science and Technology in 2016, where he is currently pursuing the Ph.D. degree in South China University of Technology, China. His research interests include neural network and biomedical information processing.



**Huihui Li** received the M.S. degree in South China University of Technology. She is currently working towards the Ph. D degree from the Department of Computer Science and Technology of South China University of Technology. Her research area includes facial expression recognition, artificial intelligence and Machine Learning in Traditional Chinese Medicine.



**Jiajiong Ma** is currently a master candidate in the College of Computer Science and Engineering, South China University of Technology. His main research interests include object detection, image classification and deep learning.



**Guihua Wen** is now professor, doctoral supervisor at the school of computer science and technology of South China University of Technology. His research area includes Cognitive affective computing, Machine Learning and data mining.