Facial expression recognition in the wild based on multimodal texture features

Bo Sun
Liandong Li
Guoyan Zhou
Jun He
Facial expression recognition in the wild based on multimodal texture features

Bo Sun, Liandong Li, Guoyan Zhou, and Jun He*
Beijing Normal University, College of Information Science and Technology, No. 19, XinJieKouWai Street, Beijing 100875, China

Abstract. Facial expression recognition in the wild is a very challenging task. We describe our work in static and continuous facial expression recognition in the wild. We evaluate the recognition results of gray deep features and color deep features, and explore the fusion of multimodal texture features. For the continuous facial expression recognition, we design two temporal–spatial dense scale-invariant feature transform (SIFT) features and combine multimodal features to recognize expression from image sequences. For the static facial expression recognition based on video frames, we extract dense SIFT and some deep convolutional neural network (CNN) features, including our proposed CNN architecture. We train linear support vector machine and partial least squares classifiers for those kinds of features on the static facial expression in the wild (SFEW) and acted facial expression in the wild (AFEW) dataset, and we propose a fusion network to combine all the extracted features at decision level. The final achievement we gained is 56.32% on the SFEW testing set and 50.67% on the AFEW validation set, which are much better than the baseline recognition rates of 35.96% and 36.08%.

Keywords: facial expression recognition; texture features; in the wild; deep learning; feature fusion.

1 Introduction

With the development of artificial intelligence and affective computing, facial expression recognition has shown prospects in human–computer interfaces, online education, entertainment, intelligent environments, and so on. In past years, much research has been done on the data collected in strictly controlled laboratory settings with frontal faces, perfect illumination, and posed expressions. As the application environment turns into a real world scenario, those methods using the monomial feature such as local binary patterns (LBP)1 or bag of visual words2 cannot achieve promising results. In addition, unlike the lab-controlled dataset, human heads in a real environment can be in any position of an image with all sorts of angles and poses. So, for most automatic facial expression recognition methods, the first step is to locate and extract the position of a face in the whole scene. The traditional way of this progress is always to combine the Viola–Jones face detector and the Haar-cascade eye detector.3 Recently, some methods, such as mixture of parts (MoPs)4 and supervised descent method,5 have robust face detection results in various head rotations.

To explore facial expression recognition in the real world, we do experiments on three public datasets: acted facial expression in the wild (AFEW), static facial expression in the wild (SFEW), and facial expression recognition (FER). The AFEW database6 consists of short video clips extracted from popular Hollywood movies. Each clip contains a film actor who has been labeled into one of the seven basic facial expression categories, namely Anger, Disgust, Fear, Happiness, Neutral, Sadness, and Surprise. The AFEW set has 711 training videos, 371 validation videos, and 539 test videos. We only know the labels of the training and validation sets, specific numbers of which are shown in Table 1. The SFEW database7 is almost the same as that of the AFEW set, except that it consists of static frames of the movies. Both of the datasets are very challenging for traditional facial expression recognition methods due to the complicated scenes of films, which can be seen from the uncompromising baseline recognition rate of 36.08% and 35.96%. The SFEW set consists of 891,427, and 372 RGB color images for training, validation, and testing, respectively. Samples of expression data are shown in Fig. 1. The FER-2013 dataset8 is a facial expression dataset created using the Google image search application programming interface to search for images of faces that match a set of 184 emotion-related keywords such as “blissful” and “enraged.” It has 28,709 gray images for training and 7178 images for validation. On the FER dataset, the human accuracy was 68 ± 5%.

In our proposed method, openly available tools such as MoPS4 and Intraface5 are used for face detection and alignment. For facial expression, we employ the descriptors of LBP,1 local phase quantization (LPQ),9 histogram of oriented gradients (HOG),10 and dense scale-invariant feature transform (SIFT).2 We also design a deep convolutional neural network (CNN)11 for feature learning and compare the recognition results between gray data and color data. Then, we propose a fusion network for classification, which is a decision-level fusion method for improving the result. Our fusion network fuses different features and gains a promising recognition performance. We also compare the result of it with that of other state-of-the-art fusion methods.

The rest of this paper is organized as follows: In Sec. 2, we review the related works. The facial image extraction progress is shown in Sec. 3. Section 4 details the deep features and handcrafted features we explored. Section 5 gives
Section 6 gives the experiments we have done, in which the feature components and the recognition results on three datasets are available. Then, the final conclusion is given in Sec. 7.

2 Related Works

There are many researches focusing on recognizing facial expression. Ekman and Friesen\cite{12} defined facial action coding system actions for manual facial expression analysis. Zhao and Pietikainen\cite{1} proposed a volume local texture feature LBP-TOP and achieved remarkable facial expression recognition results in a laboratory. Kahou et al.\cite{13} used convolutional neural network and deep belief network and got the top performance in the EmotiW 2013 Challenge. Liu et al.\cite{14} used Grassmannian Manifold to get facial expression features, then they combined Riemannian Manifold and deep convolutional neural network in Ref. 15. Yao et al.\cite{16} combined the CNN model with facial action unit aware features and got the state-of-the-art result for facial expression recognition in videos. Kim et al.\cite{17} explored several CNN architectures and data preprocessing methods. Yu and Zhang\cite{18} used a data disturb method to enhance data. Liu et al.\cite{19} proposed a boosted deep belief network for facial expression recognition and got promising results on some laboratory recorded datasets. Ng et al.\cite{20} explored transfer learning for deep models including VGG and AlexNet.\cite{21}

Since no feature descriptor can handle the problem of facial expression recognition in the wild alone, the fusion method can be used to combine multimodal features. Sikka et al.\cite{22} explored the fusion way of general multiple kernel learning (GMKL) and multi-label multiple kernel learning. Chen et al.\cite{23} used the SimpleMKL method to combine visual and acoustic features. Kim et al.\cite{17} proposed a committee machine method to combine 108 CNN models in. Kahou et al.\cite{24} proposed a voting matrix and used random search to tune the fusion weight parameters. They used the multilayer perceptron in Ref. 25 to combine neural networks at the feature level. Gönen and Alpaydın\cite{26} reviewed quite a few kinds of multiple kernel methods for the common pattern recognition problem. Bucak et al.\cite{27} reviewed the state-of-the-art for multiple kernel learning (MKL), with the focus on the applications of object recognition.

3 Face Extraction

We follow the face extraction and tracking method of Sikka et al.\cite{2} and Dhall et al.\cite{28} For the continuous facial expression recognition, the mixture of tree structured part model (MoPS)\cite{4} face detector is used to detect the position of a face in the first frame of a video. Then, the IntraFace toolkit used the supervised descent method\cite{5} to track 49 facial landmarks of the rest of the frames in a parameterized appearance model. All frames of the AFEW dataset are aligned to a base face through affine transformation and cut to $128 \times 128$ pixels.

For the static facial expression recognition, the MoPS and OpenCV\cite{29} detectors are used for SFEW and FER, respectively. Facial landmarks generated by MoPS are used to align faces for handcrafted features extraction. For deep CNN features that are robust to the poses of faces, only coarse face alignment is performed, by keeping the center of facial landmark points or bounding boxes at the middle of images. All face images are resized to $48 \times 48$ pixels for deep feature learning. For handcrafted features, the image size is set to $128 \times 128$. As illumination and brightness changes appeared frequently in the SFEW dataset, we evaluate the min–max normalization as image preprocessing method.
4 Multimodal Texture Features

4.1 Feature Learning

The deep CNN\textsuperscript{11} is a popular type of model in the community of computer vision. We deploy two kinds of CNN architectures, the AlexNet and regions CNN (RCNN). The AlexNet\textsuperscript{30} is a nine-layers deep model designed for object recognition of ImageNet dataset, using rectified linear unit as activation function. The AlexNet model has five convolutional layers and three fully connection layers. It introduces data enlarge strategy, local normalization, and dropout method to avoid over-fitting. The RCNN\textsuperscript{31} is a type of deep learning architecture that combines object detection with object recognition. This model can detect the object in a scene and then use the CNN feature for classification. These two models are all pretrained on the ImageNet dataset.

Based on the AlexNet, we design a deep CNN architecture for facial expression recognition. The whole architecture of our model is shown in Fig. 2. First, the facial images are cropped from four corners and the center and flipped to 10 patches of $40 \times 40$. Then, the first convolutional layer filters the $40 \times 40$ input patch with 64 kernels of size $5 \times 5$. The second convolutional layer takes as input the response-normalized and max-pooled output of the first convolutional layer and filters it with 64 kernels of size $3 \times 3 \times 64$. The third, fourth, and fifth convolutional layers are connected to one another without any intervening pooling or normalization layers. The third convolutional layer has 128 kernels of size $3 \times 3 \times 64$ connected to the (normalized, pooled) outputs of the second convolutional layer. The fourth and fifth convolutional layers both have 128 kernels of size $3 \times 3 \times 128$. The fully connected (FC) layers have 1024 neurons each. The rectified linear unit activations are applied to the output of every convolutional or fully connected layer. For validation of the training progress, the softmax regression is used as the output layer. For feature extraction, we use the last FC layer as the output. In our experiments, we visualize the activation values of the first convolutional layers of the AlexNet and our proposed CNN, which are shown in Fig. 3. We can see that some feature maps of the AlexNet are not activated in the task of expression recognition. This is reasonable since the AlexNet is trained on the ImageNet dataset, which makes its feature contain more information than human facial expression.

4.2 Handcrafted Features

For images of SFEW dataset, we extract LBP, dense SIFT, and deep CNN features. For video clips of AFEW dataset, we extract volume features such as LBP-TOP, LPQ-TOP and pooling the dense SIFT, HOG and DCNN features through the image sequences of a video. In addition, we also design two temporal–spatial features: SIFT-TOP and SIFT-LBP. The pipeline of extracting these handcrafted features is as follows: on the face images extracted from a video, alignment through facial landmark points and spatial pyramid

![Fig. 2 Deep CNN architecture for feature learning.](image)

![Fig. 3 Learnt features of the first convolutional layers. The left one (a) belongs to the AlexNet while the right one (b) belongs to our proposed CNN.](image)
4.2.1 Image descriptors
The LBP descriptor is an efficient representation of facial image texture, and has been successfully applied to facial expression recognition. It can be represented as follows:

$$d = \sum_{p} \sum_{i=1}^{k} 2^{i-1} I(O_p, N_i).$$ (1)

In Eq. (1), $I(O, N)$ means the Boolean comparison between a pixel $O_p$ and its neighboring pixels $N$ which has a total number of $K$. The binary labels form a local binary pattern $d$ over the whole $p$ pixels of an image.

The LPQ descriptor is calculated based on computing short-term Fourier transform on local image window. The descriptor utilizes phase information computed locally in a window for every image position. The phases of the four low-frequency coefficients are decorrelated and uniformly quantized in an eight-dimensional space.

The HOG descriptor is implemented by dividing the image window into small spatial regions, each region accumulating a local one-dimensional histogram of gradient directions or edge orientations over the pixels of the region. The combined histogram entries form the representation.

The dense SIFT feature is to perform SIFT descriptor on a dense grid of locations at a fixed scale and orientation. The SIFT descriptor associates to the grid a signature that identifies its appearance compactly and robustly. The dense SIFT feature characterizing appearance information is often used for categorization task.

4.2.2 Feature encoding and pyramid matching
For LBP and LPQ descriptor, histograms of all binary code-words are formed to encode the final image features. Take note that only the statistics of 59 uniform local binary patterns are considered. For dense SIFT descriptor, the bag of words model has shown remarkable performance on facial expression recognition. First, we extract multiscale dense SIFT descriptors from 100 randomly picked image samples. Then, 800 clustering centers are constructed using approximate K-means clustering algorithm. The number 800 is chosen throughout the experiments. Then, the whole data sets’ dense SIFT descriptors are encoded using the locality-constrained linear coding (LLC), which can guarantee the sparsity and locality of the coded words.

In our experiments, we tried spatial pyramid matching for the handcrafted descriptors. Experimental results show that spatial pyramid matching can add recognition accuracy by providing more spatial information to the final features. The number of layers of LBP, LPQ, and dense SIFT are 4, 4, and 5, respectively.

4.3 Temporal–Spatial Representation
For continuous facial expression recognition, the image feature has to be extended to temporal–spatial area. After getting the image features of all image frames of a video clip, max pooling is usually used to aggregate all frame features into one video feature. Though this is still decent performance, it actually loses much detailed temporal information of a video. Based on deep analysis on our experiments, we add temporal information through extracting LBP descriptors on the $XT$ and $YT$ planes (in which $T$ stands for the time domain) of a video, and combine it with the dense-SIFT feature of $XY$ plane (i.e., the image space) (SIFT-LBP), shown as Fig. 5. LBP descriptors of $XT$ and $YT$ frames are encoded to $XT$ histogram and $YT$ histogram, after spatial pyramid matching. Bag of multiscale dense SIFT feature is extracted from every $XY$ frame following the pipeline described in Sec. 4.2.2. We also explore how to directly extract dense SIFT feature on the three orthogonal planes of $XY$, $XT$, and $YT$ (SIFT-TOP). Our experiment shows that the new temporal–spatial descriptor, namely SIFT-LBP, has better performance. We also explore how to use a deep learnt feature for temporal–spatial representation, which is accomplished by taking the maximum pooling value of the CNN feature vectors over all frames. Unfortunately, the recognition result is uncompromising on the AFEW dataset.

---

Fig. 4 Pipeline of handcrafted features extraction. The dashed box means that the temporal–spatial representation is only used for AFEW dataset.

![Fig. 4](image-url)

Fig. 5 Our proposed SIFT-LBP temporal–spatial representation for video.
5 Fusion Classification

5.1 Classifiers

5.1.1 Support vector machine

The features we extract are all linearly separable under ideal conditions. So, we use linear support vector machine (SVM) as basic classifiers. Given a training set of \( L \) data points \((x_i, y_i), i = 1, \ldots, L, x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}\), the support vector classifier solves the following unconstrained optimization problem:\(^{35}\)

\[
\min_{\theta} \frac{1}{2} \theta^T \theta + C \sum_i \xi_i(\theta; \theta_t, y_i),
\]

where \( C \) is the penalty parameter and \( \xi_i(\theta; \theta_t, y_i) = \max (1 - y_i \theta^T x_i, 0)^2 \) is the loss function. For testing data \( x \), SVM predicts it as positive if \( \theta^T x > 0 \), and negative otherwise. Here, we use the SVM decision value \( D_{\text{SVM}} = \theta^T x \) as the input for the next fusion process. As SVM is a binary classifier, we follow one-versus-rest strategy, which classifies the data points between one category and the rest one at a time.

5.1.2 Partial least squares regression

Partial least squares (PLS) regression is a statistical method that bears some relation to principal components regression; instead of finding hyperplanes of minimum variance between the response and independent variables, it finds a linear model by projecting the variables and the observable variables to a new space. According to Ref. 14 given a feature set \( X \in \mathbb{R}^m \) with label \( Y \), the PLS classifier decomposes these variables into:

\[
X = U_x V_x^T + r_x,
\]

\[
Y = U_y V_y^T + r_y,
\]

where \( U_x \) and \( U_y \) contain the extracted score vectors, \( V_x \) and \( V_y \) are orthogonal loading matrices, and \( r_x \) and \( r_y \) are residuals. PLS tries to find the optimal weights \( w_x \) and \( w_y \) to get the maximum covariance such that

\[
|\text{cov}(u_x, u_y)|^2 = \max_{|u|=|v|=1} |\text{cov}(Xw_x + Yw_y)|^2.
\]

Then, we can get the regression coefficients \( \beta \) as

\[
\beta = X^T U_x (U_x^T X X^T U_x)^{-1} U_x^T Y.
\]

The PLS decision value can be estimated by \( D_{\text{PLS}} = X\beta \). Like in Sec. 5.1, we follow one-versus-rest strategy for the multiclass classification.

5.2 Fusion Network

As different features have different discriminative abilities on specific emotions,\(^{36}\) we propose a fusion network as shown in Fig. 6 to combine the results of each classifier.

Given \( m \) features and \( n \) classes, the SVM or PLS classifiers generate \( m \times n \) decision values, which can be denoted as \( a_{(j,k)} = \theta_{(j,k)}^T x_j, j = 1, \ldots, m, k = 1, \ldots, n \). Then, they are used as the input for the fusion network. For input \( a \), we use a hypothesis function \( h_w(a) \)

\[ P(y=1|x_1, x_2, \ldots) \]

\[ P(y=2|x_1, x_2, \ldots) \]

\[ P(y=3|x_1, x_2, \ldots) \]

\[ \vdots \]

\[ \text{Max - Win} \]

\[ \text{Estimated Probabilities} \]

\[ h_w(a) = \begin{bmatrix} p(y^{(i)} = 1|a^{(i)}; W) \\ p(y^{(i)} = 2|a^{(i)}; W) \\ \vdots \\ p(y^{(i)} = n|a^{(i)}; W) \end{bmatrix} = \frac{1}{\sum_{k=1}^n e^{\sum_{j=1}^m w_j^{(k)} a_j^{(i)}}} \begin{bmatrix} e^{\sum_{j=1}^m w_j^{(1)} a_j^{(i)}} \\ e^{\sum_{j=1}^m w_j^{(2)} a_j^{(i)}} \\ \vdots \\ e^{\sum_{j=1}^m w_j^{(n)} a_j^{(i)}} \end{bmatrix} \]

\begin{align*}
J(W) &= -\frac{1}{L} \left[ \sum_{i=1}^L \sum_{k=1}^n m \{ y^{(i)} = k \} \log \frac{e^{\sum_{j=1}^m w_j^{(k)} a_j^{(i)}}}{\sum_{k=1}^n e^{\sum_{j=1}^m w_j^{(k)} a_j^{(i)}}} \right] \\
&\quad + \frac{1}{2} \sum_{j=1}^m \sum_{k=1}^n W_j^{2k},
\end{align*}

\[ \nabla W, J(W) = -\frac{1}{L} \sum_{i=1}^L \left[ a^{(i)} \{ y^{(i)} = k \} - p(y^{(i)} = k|a^{(i)}; W) \right] + \lambda W_k, \]

\[ \lambda > 0 \]

\[ \lambda \text{ is the regularization parameter.} \]
where \( L \) is the number of training examples, \( \lambda \) is the L2-norm parameter, \( 1\{\cdot\} \) is the indicator function, which means \( 1\{\text{a true statement}\} = 1 \), and \( 1\{\text{a false statement}\} = 0 \).

In experiments, we try to fuse the decision values of SVM and PLS classifiers. We find that this kind of fusion network performs better than the SVM-only fusion network.

### 6 Experiments

#### 6.1 Deep Feature Learning of Color and Gray Images

For deep feature learning, we employ the Caffe\textsuperscript{37} implementation, which is commonly used in several recent works. To pretrain the CNN model according to our proposed architecture, we use expression images from the FER dataset. The base learning rate is set to 0.005, which will be divided by 10 after every 10,000 iterations. In each iteration, 256 samples are used for stochastic gradient optimization. After 200 epoch’s training, our proposed CNN gets 67.82% on the FER validation set. Then, we fine-tune the model on the SFEW set. The base learning rate is changed to 0.001. After 300 epoch’s fine-tuning, the validation accuracy is converged. The experiment results are shown in Table 2. We can see that the RGB color CNN model with min-max normalization can achieve slightly better recognition result.

#### 6.2 Results of Monomial Feature

We extract the features listed in Sec. 4 and apply the SVM and PLS classifiers. Results are shown in Tables 3 and 4. On the SFEW dataset, through comparison experiments, we extract the last pooling layer’s activation value as the feature of AlexNet and RCNN. For our proposed CNN, the last fully connected layer’s output is extracted. We can see that using the SVM and PLS classifier can further improve the recognition result of the CNN model. On the AFEW dataset, as each frame produces a CNN, a dense SIFT and a HOG feature vector, information from all frames of a video are combined using pooling strategy, which is accomplished by taking the maximum or mean value of all feature vectors over all frames. By experiment, max pooling has better results for dense SIFT and HOG. The SVM classifiers all use linear kernels. Classification models are trained on training set and parameters are tuned on validation set through fivefold cross validation in the range from \( 2^{-10} \) to \( 2^{10} \). Results show that our proposed CNN feature and SIFT-LBP feature performs well on the SFEW and AFEW dataset, respectively.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Comparison results of proposed CNN model, on color and gray image data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel</td>
<td>Preprocessing</td>
</tr>
<tr>
<td>Gray</td>
<td>Raw</td>
</tr>
<tr>
<td>RGB</td>
<td>Raw</td>
</tr>
<tr>
<td>Gray</td>
<td>Min-max norm</td>
</tr>
<tr>
<td>RGB</td>
<td>Min-max norm</td>
</tr>
</tbody>
</table>

#### 6.3 Fusion Results of Multimodal Features

For AFEW and SFEW datasets, we use four-Layer SPM for LBP and LPQ features. Each image is partitioned into \( 2^l \times 2^l \) segments at multiple scales \( l \in \{1, 2, 4, 8\} \). For example, the dimension of SPM-LBP TOP is 15,045. Too much SPM layers mean larger dimension and it would be harder to be optimized for classification. While as dense SIFT uses LLC coding, five-layer SPM can achieve the best performance.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Recognition accuracies on SFEW. ( C ) is the cost parameter of SVM, ( n ) is the PLS dimension. ( P ) represents the activation value of last pooling layer while FC means the activation value of last FC layer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>SVM Accuracy (%)</td>
</tr>
<tr>
<td>Baseline</td>
<td>35.96</td>
</tr>
<tr>
<td>LPQ</td>
<td>28.08</td>
</tr>
<tr>
<td>PHOG</td>
<td>31.79</td>
</tr>
<tr>
<td>Dense SIFT</td>
<td>43.33</td>
</tr>
<tr>
<td>AlexNet (P)</td>
<td>37.00</td>
</tr>
<tr>
<td>AlexNet (FC)</td>
<td>32.32</td>
</tr>
<tr>
<td>RCNN (P)</td>
<td>43.09</td>
</tr>
<tr>
<td>RCNN (FC)</td>
<td>32.32</td>
</tr>
<tr>
<td>Proposed CNN (P)</td>
<td>48.24</td>
</tr>
<tr>
<td>Proposed CNN (FC)</td>
<td>51.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Recognition accuracies on AFEW.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>SVM Accuracy (%)</td>
</tr>
<tr>
<td>Baseline (LBPTOP)</td>
<td>36.08</td>
</tr>
<tr>
<td>HOG</td>
<td>34.23</td>
</tr>
<tr>
<td>SPM-LBPTOP</td>
<td>43.67</td>
</tr>
<tr>
<td>LPQ TOP</td>
<td>42.59</td>
</tr>
<tr>
<td>SPM-LPQ TOP</td>
<td>44.47</td>
</tr>
<tr>
<td>SPM-dense SIFT</td>
<td>46.09</td>
</tr>
<tr>
<td>SIFT TOP</td>
<td>46.45</td>
</tr>
<tr>
<td>SIFT-LBP</td>
<td>49.33</td>
</tr>
<tr>
<td>AlexNet</td>
<td>35.69</td>
</tr>
</tbody>
</table>
network on the validation set. The L2-norm parameter $\lambda$ is chosen through a cross validation on the validation set. Fusion results are shown in Tables 5 and 6. Results show that our proposed method is better both on the validation set and testing set. We compare the fusion network with GMKL, 38 SimpleMKL, 39 and three other researcher’s work 17,18,20 on the SFEW set. We can see that our fusion network outperforms other methods on the validation set. As the test labels of the AFEW and SFEW datasets are not publicly opened, we do not get final test results for all our methods. Despite that we can see that our proposed fusion network performs well and robust through cross validation. Note that some features perform better when classified by PLS, so the fusion network combining PLS and SVM together can achieve better results than using only SVM.

7 Conclusions and Future Work

In this paper, we design some texture features for automatic human facial expression recognition in the real world. For each feature, we train individual SVM and PLS classifiers that have different discriminative ability for facial expressions classification. We propose a fusion network to utilize these feature characteristics. The method is evaluated on the AFEW and SFEW datasets and gains very promising achievement. In the future, we will try to deduce more kinds of temporal–spatial representation methods to further improve the continuous facial expression recognition result and investigate the use of component analysis methods to decrease the feature dimensions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61501035 and KJXZJC2016042), the Fundamental Research Funds for the Central Universities of China (2014KJJC1A15), and the National Education Science Twelfth Five-Year Plan Key Issues of the Ministry of Education (DCA140229).

References

7. A. Dhall et al., “Static facial expression analysis in tough conditions: data, evaluation protocol and benchmark,” in IEEE Int. Conf. on


Bo Sun received his BSc degree in computer science from Beihang University, China, and his MSc and PhD degrees from Beijing Normal University, China. He is currently a professor in the Department of Computer Science and Technology, Beijing Normal University. His research interests include pattern recognition, natural language processing, and information systems. He is a member of ACM and a senior member of China Society of Image and Graphics.

Liandong Li received his BSc degree in computer science and technology from Beijing Normal University, 2011. He is currently working toward the PhD in computer application technology at Beijing Normal University. His research interests include machine learning, computer vision, and emotion analysis.

Guoyan Zhou received his BSc degree in computer science and technology from Beijing Normal University, 2013. Currently, she is working toward the MSc degree in computer application technology at Beijing Normal University. Her research interests include machine learning and computer vision.

Jun He received his BSc degree in optical engineering and her PhD in physical electronics from Beijing Institute of Technology, Beijing, China, in 1998 and 2003, respectively. Since 2003, she has been with the College of Information Science and Technology, Beijing Normal University, Beijing, China. She was elected as a lecturer and an associate professor in 2003 and 2010, respectively. Her research interests include image processing application, and pattern recognition.