Machine Vision and Applications (2022) 33:40
https://doi.org/10.1007/s00138-022-01288-9

ORIGINAL PAPER

FERGCN: facial expression recognition based on graph convolution network

Lei Liao[1] · Yu Zhu[1,2] · Bingbing Zheng[1] · Xiaoben Jiang[1] · Jiajun Lin[1]

Received: 16 August 2021 / Revised: 20 December 2021 / Accepted: 14 February 2022 / Published online: 22 March 2022

Abstract

Due to the problems of occlusion, pose change, illumination change, and image blur in the wild facial expression dataset, it is a challenging computer vision problem to recognize facial expressions in a complex environment. To solve this problem, this paper proposes a deep neural network called facial expression recognition based on graph convolution network (FERGCN), which can effectively extract expression information from the face in a complex environment. The proposed FERGCN includes three essential parts. First, a feature extraction module is designed to obtain the global feature vectors from convolutional neural networks branch with triplet attention and the local feature vectors from key point-guided attention branch. Then, the proposed graph convolutional network uses the correlation between global features and local features to enhance the expression information of the non-occluded part, based on the topology graph of key points. Furthermore, the graph-matching module uses the similarity between images to enhance the network's ability to distinguish different expressions. Results on public datasets show that our FERGCN can effectively recognize facial expressions in real environment, with RAF-DB of 88.23%, SFEW of 56.15% and AffectNet of 62.03%.

Keywords Expression recognition · Graph convolutional network · Deep learning · In-the-wild data

1 Introduction

Facial expression is one of the most important signals for people to exchange emotional information [1]. Automatic facial expression recognition (FER) is widely utilized in many fields, such as social robots, medical treatment, intelligent driving, and public safety. Therefore, many researchers focus on the methods of FER [2–4].

In recent years, with the success of deep learning technology in various fields [5–7], more and more researchers use convolutional neural network for FER. Since deep learning needs a lot of data for training, scientists collect and process a large number of facial expression datasets. These datasets are divided into lab-controlled datasets and in-the-wild datasets. The popular lab-controlled datasets include CK+ [8], MMI [9], and OULU-CASIA [10]. These data are obtained in controlled laboratory. They are all positive, non-occluded and the illumination is constant. Therefore, these datasets can obtain better recognition results using the convolutional neural network [11, 12]. While in-the-wild datasets are obtained from the real environment. The commonly used include RAF-DB [13], SFEW [14], and AffectNet [15]. In-the-wild datasets can better reflect the real complex situation, such as extreme face pose, large area occlusions, and illumination changes, which are shown in Fig. 1. These problems have caused the recognition accuracy of in-the-wild datasets much lower than lab-controlled datasets. Therefore, in-the-wild FER technology is facing great challenges.

To improve the recognition accuracy of in-the-wild datasets, this paper proposes a FERGCN network based on graph convolution. The proposed network is divided into three parts: feature extraction module, graph convolutional network (GCN), and graph-matching module.

For the feature extraction module, many studies have shown that facial emotion changes are related to specific areas of the face (such as eyes, mouth, and cheek) [16–18]. Therefore, the proposed feature extraction module consists of two branches: key point-guided attention branch and CNN branch, and then the obtained 2 types of feature maps are fused to get 1 global and 18 local feature representations.

✉ Yu Zhu
zhuyu@ecust.edu.cn

1 School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

2 Shanghai Engineering Research Center of Internet of Things for Respiratory Medicine, Shanghai 200032, China

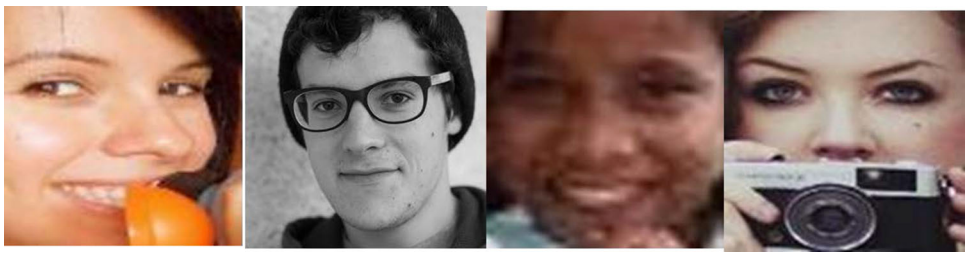


Fig. 1 Interference factors in wild facial expression datasets. From left to right are the side face, grayscale, low pixel, and occlusion

In CNN branch, we use triplet attention [19] to optimize the feature map.

When human beings recognize occluded facial expressions, they usually utilize local and global information to recognize facial expressions together. Therefore, in the GCN, we take the 18 local feature vectors as the nodes of the topology graph and propose a novel graph convolution layer to learn the expression information of the non-occluded parts. This layer can promote the information transfer of semantic features and suppress that of meaningless and noisy features. Finally, the learned nodes contain both semantic information and related information.

There are tiny differences between facial expressions, especially some negative expressions among them have high similarity (for example, disgust and sadness, fear and surprise), so we need more representative features to distinguish among these similar expressions. We adopt the strategy of hard sample triple loss [20] to obtain the positive samples with the largest distance of the same class and the negative samples with the smallest distance of different classes for anchor samples and then perform two groups of graph matching, respectively. In the graph-matching module, first, the corresponding relationship between two graphs is learned by graph matching, and the learned corresponding relationship is regarded as an adjacency matrix to transfer messages. Finally, the similarity between the two graphs is used to calculate the verification loss.

The main contributions of this paper are as follows: (1) we propose FERGCN for recognize facial expressions in the wild. Triplet attention and graph-matching modules are introduced to the field of expression recognition for the first time. (2) For the first time, a graph convolution network is introduced from the feature graph level to learn facial expression information. The experimental results show the effectiveness of the module. (3) Our network achieved competitive results on RAF-DB, AffectNet, SFEW, and occlusion test subset of RAF-DB.

2 Related work

2.1 Facial expression recognition

FER has always been an important research topic. Most traditional methods utilize hand-made features or shallow learning, for example, local binary pattern (LBP) [21], LBP on three orthogonal planes (LBP-TOP) [22], nonnegative matrix factorization (NMF) [23], and sparse learning [24]. The development of deep learning methods mainly focuses on data and models.

In recent years, great progress has been made in FER based on deep learning [25–27]. Considering the change of posture, Zhang et al. [28] designed a model that can generate any posture and any expression using a generative adversarial network [29]. Through this model, the data are enhanced to improve the accuracy of expression recognition. To obtain expression information better, Yang et al. [30] proposed a generative adversarial network to generate neutral faces from faces with any expression and then obtained expression information from the middle layer of the generator. To reduce the influence of subject appearance on expression recognition, Cai et al. [31] introduced the generation adversarial network to generate the average face image and made the generated average face consistent with the expression of the original face through supervised learning. Liu et al. [32] designed a Point Adversarial Self Mining (PASM) model, and then they utilized a point adversary self-mined network to enhance data and teacher–student pattern to train recognition networks. Jiang et al. [33] applied Gabor convolutional network [34] to the field of expression recognition, and obtained an efficient and fast model.

To recognize occluded facial expressions, Li et al. [16] proposed a convolutional neural network with an attention mechanism. The network divides the feature map into 24 blocks and emphasizes the information of non-occluded parts by learning the weights of each block. There are some images with poor quality in the wild datasets, which will affect the training of the model. To suppress the uncertainty in the dataset, Wang et al. [4] introduced the learned image quality coefficient into the loss function and re-labeled the low-quality images and labels. Chen et al. [11] trained the model using the labeling consistency of soft tags and similar images for suppressing the label inconsistency in large-scale datasets. These methods have achieved good results, but due to various interference factors in the wild datasets, expression recognition is still faced with great challenges.

2.2 Graph convolutional network

The structure of the graph is irregular, it has no translation invariance. Traditional CNN and RNN encounter great challenges in processing data with graph structure, so Bruna et al. [35] proposed graph convolution network for data with graph structure. Defferrard et al. [36] used Chebyshev polynomial function to enhance the spatial locality of GCN and reduce the computational complexity of GCN. Kipf et al. [37] used the effective layer wise propagation rule to further optimize graph convolution and formed the current graph convolution structure. The GCN, like CNN, is a feature extractor, but its object is graph data. GCN ingeniously designs a method to extract features from graph data, so that we can use these features to do the task of node classification, graph classification, link prediction, and graph embedding. Recently, GCN has received increasing attention in the field of computer vision. Yan et al. [38] applied GCN to the video field using spatial temporary graph revolutionary networks, which achieved good results in skeleton based action recognition. To apply GCN to regression task, Zhao et al. [39] proposed semantic graph progressive networks and verified it on 3D human pose region task. Wang et al. [40] used GCN to learn the information of human body topology, which greatly improved the accuracy of person re-identification under occlusion. The structure between key points in facial expressions is just a graph structure, so we can design a graph convolution module for expression recognition.

3 Methodology

In this section, we first introduce our overall network framework and then introduce the structure of the three modules and their corresponding loss functions.

3.1 Overview

As shown in Fig. 2, the proposed FERGCN framework includes three parts, a feature extraction module, GCN, and graph-matching module. Specifically, the input size is set as 224×224, while the feature map and the key point heat map can be obtained, respectively, through the CNN branch and key point branch. The landmark-guided attention branch predicts 68 landmarks according to face recognition technology. We design a mechanism to obtain 18 key points associated with the expression, and the key points can be further used to generate the attention heat maps. Meanwhile, the CNN branch utilizes ResNet18 [41] to obtain feature maps for the input face. To improve the recognition accuracy, we add the triplet attention [19] at the end of ResNet18. The output features of the CNN branch are multiplied and pooled with 18 heat maps to obtain 18 local feature vectors, which are used as the semantic information of key points.

We apply the Z-pool operation to the output features of CNN branches to obtain a global feature vector. The Z-pool operation is composed of global max pooling and global average pooling. In addition, inspired by [17], we divide the output features of the CNN branches into pieces to learn context information in face. Since the mouth of the lower part of the face is a whole, we divide the features into three parts without overlapping.

Then, according to the location information of key points, the local feature vectors are regarded as one-by-one points to form a topological graph. Based on the topological graph, all the feature vectors are operated by GCN to obtain the optimized feature vectors. Finally, the output of the GCN is matched to obtain the similarity between the two images, and the relationship between the face images is used for supervised learning.

3.2 Feature extraction module

3.2.1 CNN branch

As shown in Fig. 2, the feature extraction module is composed of the key point-guided attention branch and CNN branch. In the CNN branch, we utilize ResNet18 without the average pooling layer and fully connected layer as the backbone network to extract the global feature map from the given image. We set the stride of conv4_1 to be 1, which is conducive to obtain a larger feature map for richer local information.

In addition, we also use the triplet attention [19] network to process feature maps to obtain more expression information. The structure of the triplet attention network is shown in Fig. 3. The network has three branches: C–W branch, C–H branch, and channel attention branch. The C–W branch is to obtain the interaction information between channel C and spatial dimension W, and the C–H branch is to obtain the interaction information between channel C and spatial dimension H.

In the C–W branch, first, the dimension of the feature map is transposed to $H \times C \times W$, followed by a Z-pool operation on the H dimension to obtain the tensor with the size of $2 \times C \times W$. Then the attention matrix of $1 \times C \times W$ is obtained using the convolution layer and sigmoid. Finally, the attention weight is multiplied by the input feature map according to the corresponding dimension.

The C–H branch and channel attention branch are similar to the C–W branch. C–H branch is processed in W dimension, while channel attention branch is processed in C dimension. We calculate the average value of the output feature map of the three branches to get the final feature map.

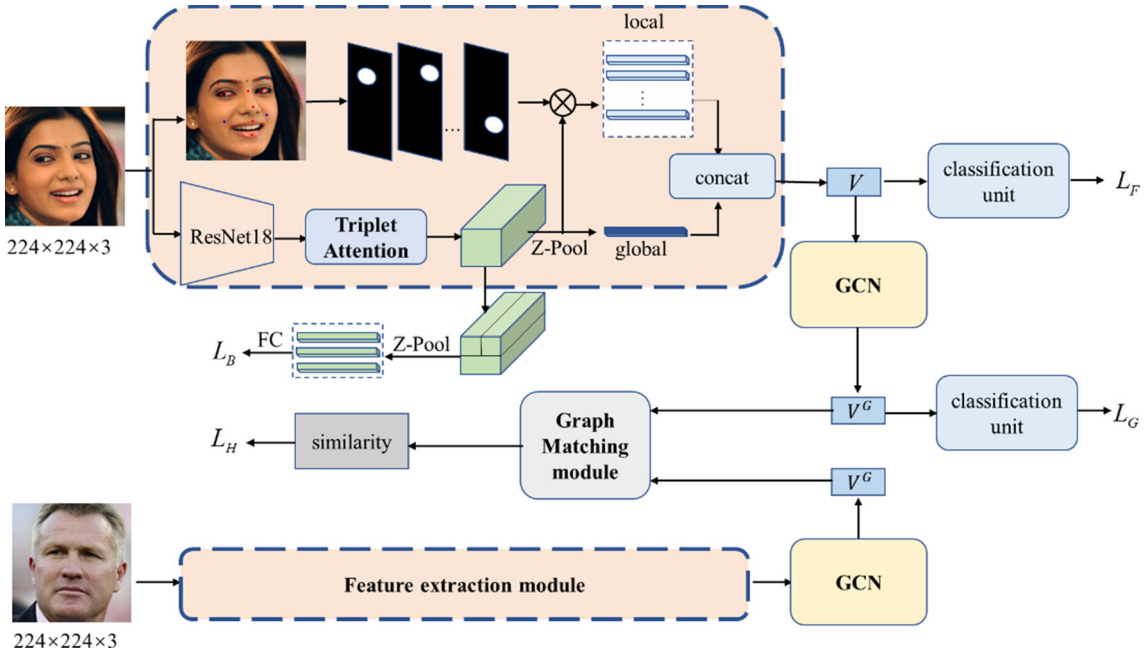


Fig. 2 The FERGCN neural network framework. FERGCN includes feature extraction module, GCN, and graph-matching module. ⊗ represents the multiplication of corresponding elements

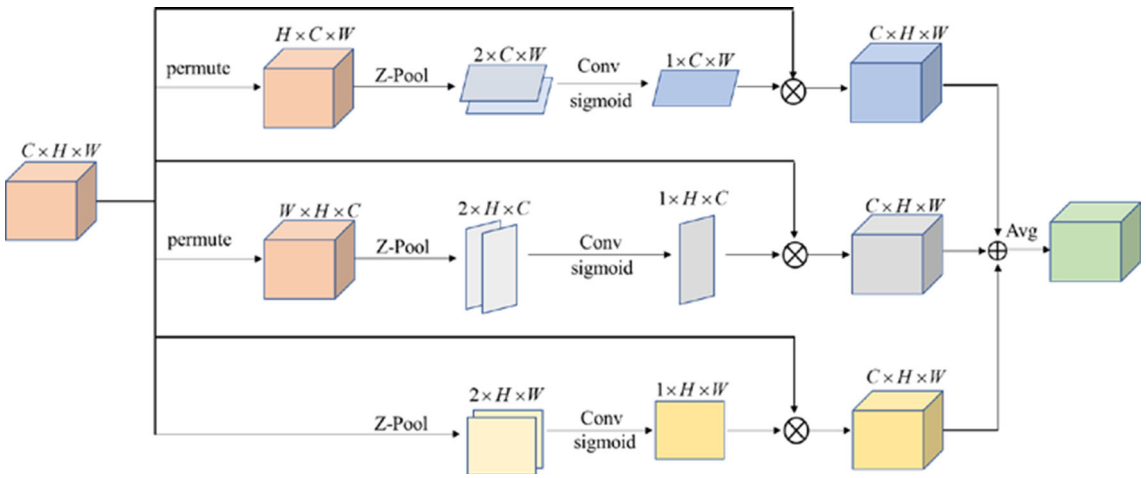


Fig. 3 Triplet attention network structure

The global feature map F is obtained from the input image X after ResNet18 and triplet attention. The formula is as follows:

$$F = T(f(x)), \tag{1}$$

where $f(\cdot)$ represents the adjusted ResNet18 and $T(\cdot)$ represents triplet attention.

3.2.2 Key point-guided attention branch

In the key point-guided attention branch, the SAN [42] method is utilized to detect 68 face landmarks from the input face image and the corresponding confidence level. Then, 16 key points (the red points in Fig. 4c) are selected from 68 face landmarks to represent eyebrows, eyes, mouth, and nose.

The indexes of these 16 points are 19, 26, 38, 45, 21, 24, 37, 46, 41, 48, 49, 55, 52, 58, 28 and 31. Since the cheek part also has rich expression information, we propose two extra key points representing the cheek which are calculated from the other neighbor landmarks, shown as blue points in Fig. 4c. First, we use landmarks with indexes of 3, 4, and 32 to form the triangle region of the left cheek, and landmarks with indexes of 36, 14, and 15 to form the triangle region of the right cheek, and take the center of gravity of these two triangle regions as the key points of the two cheeks.

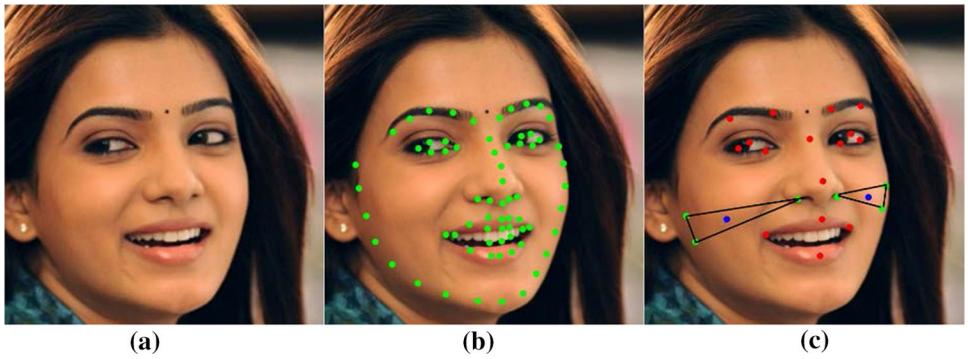


Fig. 4 Face key point acquisition process. **a** is the original face image, **b** is 68 landmarks, **c** is the key point of the face, in which the blue point is calculated

The confidence of the two key points is the average of the confidence of the three vertices in the corresponding triangle region.

As shown in Fig. 4c, the 18 key points and their corresponding confidence levels are obtained. Then, the 18 key points are taken as the center to generate 18 Gauss distribution attention heat maps $A_i (i = 1, 2 \ldots 18)$. Finally, a set of local feature vectors is obtained by the following formula:

$$v_{\text{local}}^{i} = g(F \otimes A_i), \tag{2}$$

where $\otimes$ represents multiplication of corresponding elements and $g(\cdot)$ represents global average pooling.

The global average pooling operation is carried out on F to obtain the vector v_{global} with global information. Finally, the set of vectors output by the feature extraction module is represented by V and $V = (v_{\text{local}}^{1}, v_{\text{local}}^{2}, \ldots, v_{\text{local}}^{18}, v_{\text{global}})$.

We utilize the global feature vector v_{global} to calculate the triple loss of hard samples [16] for better distinguishing similar expressions. Specifically, for each target image a, we select the farthest positive sample p and the nearest negative sample n to calculate the triple loss, the formula is as follows:

$$L_{\text{triple}} = \max(d(a, p) - d(a, n) + \gamma, 0), \tag{3}$$

where $d(\cdot)$ is the distance between two eigenvectors, and γ is a super parameter and is set to 0.3.

Facial expression is presented by multiple parts of the face, single local information cannot represent the whole expression, so we design a classification module for the feature vector group V, which can integrate the local feature information. Our proposed classification unit is shown in Fig. 5. Since each local position of the face in the wild datasets may have interfered to varying degrees, we multiply the confidence α_i of each key point by the corresponding feature vector v_{local}^{i} to suppress the possible interference factors. Then, the features of key points are fused by average pooling. Finally, the fused features are processed in the fully connected layer to obtain the local classification vector $v_{\text{class,fuse}}$.

The global feature vector v_{global} is directly processed by the fully connected layer to get the global classification vector $v_{\text{class,global}}$. The loss function of the feature extraction module is as follows:

$$\begin{aligned} L_F = {} & k \times L_{\text{class}}(v_{\text{class,fuse}}) \\ & + L_{\text{class}}(v_{\text{class,global}}) + L_{\text{triple}}(v_{\text{global}}), \end{aligned} \tag{4}$$

where k is a hyperparameter. We have 18 key points, so let $k = 18$, $L_{\text{class}}(\cdot)$ denotes the cross-entropy loss function.

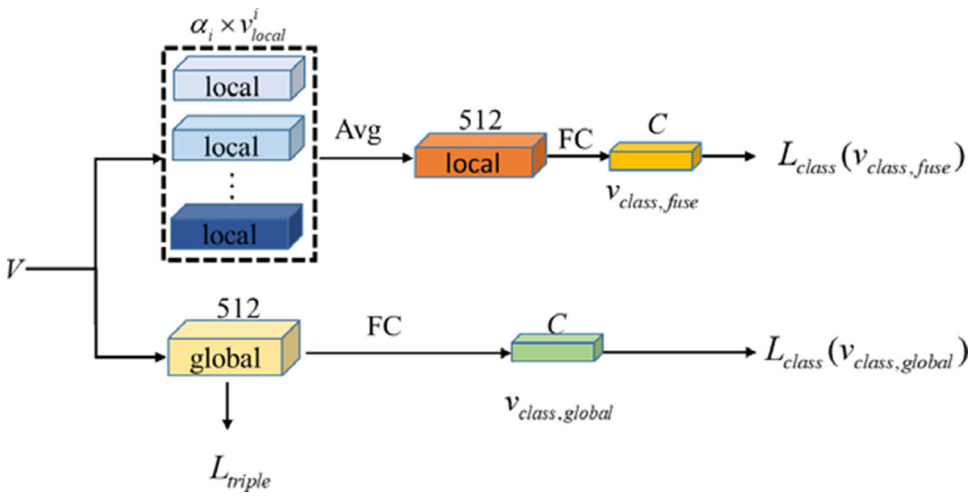
To enhance the robustness of the network and learn face context information, we divide the output features of CNN branches into three parts as shown in Fig. 2. Then, the three features are processed by Z-pool and fully connected layer to get the classification feature vector $v_{\text{class,block}}^{i} (i = 1, 2, 3)$. The loss L_{B} of this part is as follows:

$$L_{\text{B}} = \sum_{i=1}^{3} L_{\text{class}}(v_{\text{class,block}}^{i}). \tag{5}$$

3.2.3 GCN

Although we get the feature information of the key points in the feature extraction module, it is still a great challenge to recognize facial expressions in the real environment with occlusion and side faces. Studies have shown that human beings can effectively utilize local regions and whole faces to perceive the semantics of incomplete faces [43]. Therefore, we propose a graph convolution neural network. Here, we take the local feature vector output from the first module as the node and use the relationship between the key parts of the face and the relationship between the whole and the local to obtain deeper semantic information.

Under in-the-wild facial expression datasets, many faces have interference factors such as occlusion, side face, and light shadow. The 18 key points we obtained may also be affected by these interference factors. To suppress the interfering factors and emphasize the undisturbed local information, we proposed the Graph Convolutional Neural Network as shown in Fig. 6. Our proposed graph convolutional network uses the relationship between the whole



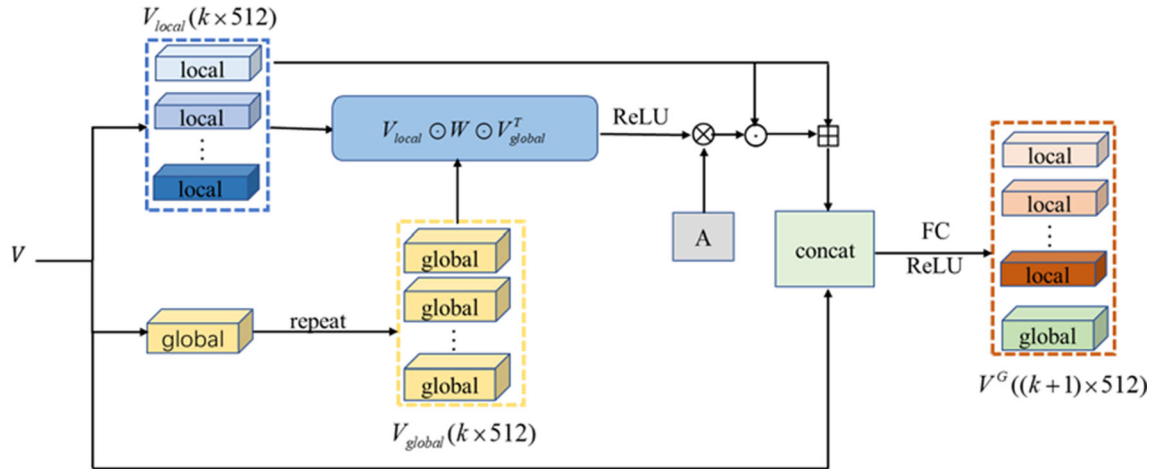


Fig. 5 The proposed classification unit. C is the number of expression classes

Fig. 6 The architecture of the plot module. $\odot$ denotes matrix multiplication, $\otimes$ denotes multiplication of corresponding elements, and A is the designed adjacency matrix

and the part to obtain the local information that needs to be emphasized.

As shown in Fig. 6, the feature vector group V is divided into two parts, the local feature vectors form $V_{\text{local}} \in R^{18\times512}$, and the global feature vectors are copied to get $V_{\text{global}} \in R^{18\times512}$. To obtain the enhanced feature information $V_d \in R^{18\times512}$, we propose the following formula:

$$V_{\text{d}} = \{[\text{Re}LU(V_{\text{local}} \odot W \odot V^T_{\text{global}})] \otimes A\} \odot V_{\text{local}}, \quad (6)$$

where $\odot$ denotes matrix multiplication,$\otimes$ denotes multiplication of corresponding elements, and W is a learnable parameter matrix of 512×512. A is the adjacency matrix of 18 key points. We design the topological map as shown in Fig. 7 according to the face structure for obtaining this adjacency matrix. The adjacency matrix is derived from Eq. 7, $A[i, j]$ represents the element values in the adjacency matrix, and V_i represents the key points in Fig. 7:

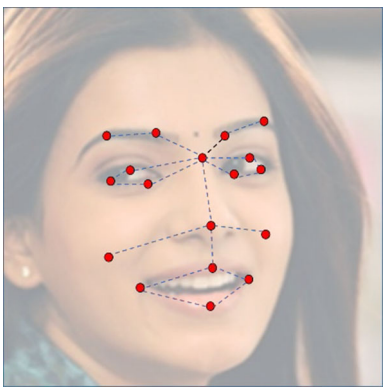


Fig. 7 The topological map between the key points

$$A[i, j] = \begin{cases} 1 & (V_i, V_j) \text{ is the edge in Figure 7} \\ 0 & (V_i, V_j) \text{ is not the edge in Figure 7} \end{cases}. \quad (7)$$

The output characteristic V^{G} of final graph convolution module is given by the following formula:

$$V^{G} = \mathrm{ReLU}\{f_1[\mathrm{concat}(V_{\mathrm{local}} + V_d,\ v_{\mathrm{global}})]\}, \tag{8}$$

where $f_1(\cdot)$ denotes the fully connected layer, and concat$(\cdot, \cdot)$ represents matrix splicing, which is to integrate the optimized local information with the global information.

The final V^{G} is similar to the V of the first module. V^{G} is processed by the classification unit to get $v^{\mathrm{G}}_{\mathrm{class,fuse}}$ and $v^{G}_{\mathrm{class,global}}$. Like formula (4), the loss function of GCN is as follows:

$$\begin{aligned} L_{\mathrm{G}} = k \times L_{\mathrm{class}}(v^{\mathrm{G}}_{\mathrm{class,fuse}}) \\ + L_{\mathrm{class}}(v^{\mathrm{G}}_{\mathrm{class,global}}) + L_{\mathrm{triple}}(v^{\mathrm{G}}_{\mathrm{global}}). \end{aligned} \tag{9}$$

3.2.4 Graph-matching module

To obtain higher order expression information and enhance the discrimination between similar expressions, we apply the graph-matching method for supervised learning of the feature vectors of the second module. The general graph-matching method [44, 45] is direct matching between corresponding points, but this method is very sensitive to outliers, which is not suitable for in-the-wild datasets with a lot of interference. Here, we utilize the Cross-Graph Embedded-Alignment Layer (CGEA) method in paper [40] to optimize the result of graph matching, and finally get the similarity S between the two images, the formula is as follows:

$$(V^{\mathrm{H}}_1,\ V^{\mathrm{H}}_2) = F_{\mathrm{H}}(V^{\mathrm{G}}_1,\ V^{\mathrm{G}}_2), \tag{10}$$

$$S^{\mathrm{H}}_{1,2} = \sigma[f_2(-|V^{\mathrm{H}}_1 - V^{\mathrm{H}}_2|)], \tag{11}$$

where F_{H} denotes CGEA method, f_2 is the fully connected layer. The loss function of the module is shown in the following formula 10:

$$L_{\mathrm{H}} = y \times \log S^{\mathrm{H}}_{1,2} + (1 - y) \times \log(1 - S^{\mathrm{H}}_{1,2}), \tag{12}$$

where y means ground truth. If the two pictures have the same expression, then $y = 1$, otherwise $y = 0$.

The training strategy of the module is: according to the real tag and V^{G} of the second module, find out the positive sample x^{+} farthest from the target image x, and the negative sample x^{-} nearest to the target image x. Finally, the verification losses of $(x,\ \hat{x}^{+})$ and $(x,\ \hat{x}^{-})$ are, respectively, calculated using Eqs. (10), (11), and (12).

3.2.5 Train and inference

In the training phase, the total loss function is as follows:

$$L = L_{\mathrm{F}} + L_{\mathrm{G}} + L_{\mathrm{H}} + L_{\mathrm{B}}. \tag{13}$$

We train our network framework by minimizing L. It should be noted that the third module in the first 20 training cycles does not participate in training. Our whole training process is shown in the following algorithm.

Algorithm 1: Training FERGCN

Input: Input image set $\{x_i, y_i\}$, $1 \le i \le N$; Iteration number iter.

Output: Network parameter w.

1 Preprocess the picture and initialize the network parameter w.
2 for ind = 1, ..., iter do:
3 for i = 1, ..., N do:
4 Feature extraction:
CNN branch extracting image features: $F = T(f(x_i))$;
F is combined with the output of the attention guidance branch to obtain V;
Calculate loss L_F using V according to formula (4);
Divide F into blocks to calculate the classification loss L_B according to formula (5).
5 GCN:
V is input into GCN to obtain V^G;
Calculate loss L_G using V^G according to formula (9).
6 if ind >= 20:
Graph Matching module:
V^G is input into Graph Matching module to obtain similarity between x_i and the positive/negative sample;
Calculate loss L_H using similarity according to formula (12).
7 Loss:
Update parameter w by minimizing total loss L, and $L = L_F + L_G + L_H + L_B$.
8 return w.

In the inference stage, the third module does not participate in the reasoning and takes the average value of $v^{G}_{class,fuse}$ and $v^{G}_{class,global}$ as the final classification basis.

4 Experiments

4.1 Datasets

RAF-DB [13] is a real-world database that contains 29,672 highly diverse facial images downloaded from the Internet. The image size of RAF-DB is resized to 100 × 100 pixels. In our experiment, we only employed six basic expressions (neutral, happiness, surprise, sadness, anger, disgust, fear) and neutral expressions, including 12,271 images as the training dataset and 3068 images as the testing dataset.

SFEW [14] is created by selecting static frames from the AFEW database [46]. It has six basic expressions and neutral expressions. The image size of this dataset is 143 × 181. The dataset contains 958 training images, 436 validation images, and 372 test images. In our experiment, we only use training images and validation images.

AffectNet [15] contains more than one million images from the Internet. It is the largest facial expression database at present. In AffectNet, 450,000 pieces are manually labeled. Like RAF-DB, we choose six basic expressions and neutral expressions for experiments, including 283,901 training images and 3500 validation images.

Occlusion-RAF-DB and Pose-RAF-DB [47] are occlusion test subsets extracted from RAF-DB. The Occlusion-RAF-DB dataset contains 735 occluded facial images. Pose-RAF-DB has 2 kinds of pictures, in which 1248 pictures are with side face angle greater than 30 degrees and 558 pictures are with side face angle greater than 45 degrees. These two datasets are not used for training, only for testing.

4.2 Implementation details

Image preprocessing Before the formal experiment, we adjusted the size of all the pictures to 224 × 224 pixels. Due to the serious imbalance between classes in these datasets, we utilize some online data enhancement methods to balance the datasets. These data enhancement methods include rotated by random degrees between − 10° and 10°, randomly

Table 1 Comparison to the state-of-the-art results

Method	RAF-DB	AffectNet
gACNN [16]	85.07	58.78
RAN [47]	86.90	–
SCN [4]	87.03	–
LDL-ALSG [11]	85.53	59.35
OADN [17]	87.16	61.89
Our	**88.23**	**62.03**

The best recognition results are shown in bold

horizontally flipped with 50% probability, and random erasing.

Training details The training of network is completed on a 1080Ti GPU with 12 GB memory. We use the PyTorch [48] framework to experiment. During the training stage, the batch size is set to 64, and we train for 80 cycles. The initial learning rate is 3.5e−4 and decaying to its 0.1 at 40 and 60 epochs. Our network is optimized by Adam [49]; Adam is deployed as the optimizer with betas of (0.9, 0.999).

4.3 Comparison to the state-of-the-art

We compare the proposed method with the latest face expression recognition methods on RAF-DB and AffectNet datasets. gACNN [16], RAN [47], and OADN [17] all propose their attention mechanisms for occlusion in face images, combining local information with global information. LDL-ALSG [11] utilizes the correlation between images and introduces soft tags to supervise the training of the model. OADN [17] introduces a landmark-guided attention branch to guide the network to learn the information of non-occluded areas.

The experimental results are shown in Table 1. For RAF-DB and AffectNet datasets, our results are better than other models. Our method achieves 88.23% accuracy in RAF-DB, which is 1.07% higher than OADN [17]. In AffectNet, we achieve 62.03% accuracy, which is 0.14% higher than OADN [17]. The results show that our method can effectively extract expression information from face images.

We draw the confusion matrix on RAF-DB, as shown in Fig. 8. This method has an outstanding performance in the recognition of happiness and sadness. Fear is similar to surprise, and people usually restrain their disgust to others, so our FERGCN cannot recognize fear and disgust well.

We also test our method on SFEW, the number of pictures in SFEW is small, so we first do pre-training in RAF-DB, and then use SFEW for training and testing. The experimental results are shown in Table 2. ICID [50] use the intra-category common feature representation (IC) channel and the Inter-category distinction feature representation (ID) channel for

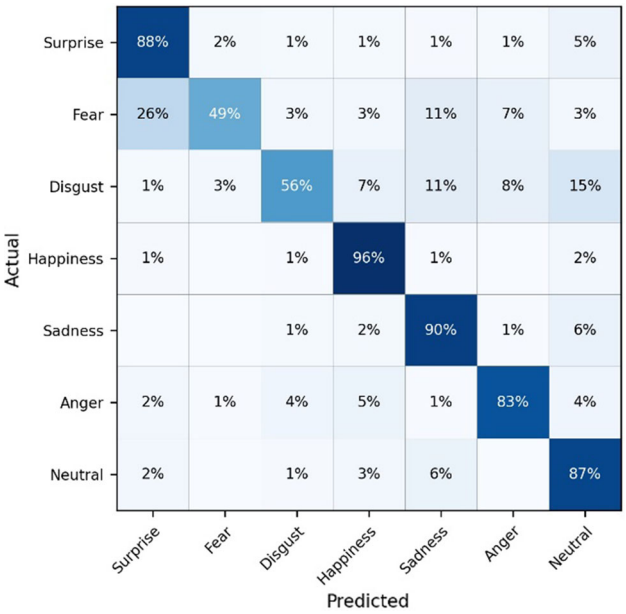


Fig. 8 Confusion matrix in RAF-DB

Table 2 Comparison to the state-of-the-art results in SFEW

Method	Accuracy
[28]	26.58
ICID [50]	51.2
LBF-NN [51]	49.31
RAN [47] (ResNet18)	54.19
Our (ResNet18)	**56.15**

The best recognition results are shown in Bold

facial expression recognition. LBF-NN [51] uses pixel difference features, ensembles of decision trees, and shallow neural network for facial expression recognition. We achieve 56.15% accuracy, which is 1.96% higher than RAN [47].

4.4 Performance evaluation on occlusion datasets

To evaluate the robustness of our proposed method to occlusion and pose change, we test the performance of our proposed method on Occlusion-RAF-DB and Pose-RAF-DB. Following RAN [47], we first train our network on RAF-DB and then test on the Occlusion-RAF-DB and Pose-RAF-DB dataset. In the same way, we compare it with the latest method. RAN [47] proposed a regional attention network, and they also designed a regional bias loss function to emphasize regional information. SCN [4] proposed image quality coefficient and label correction for the uncertainty of datasets. The results of the test are shown in Table 3, as we can see, our proposed framework significantly outperforms the

Table 3 Accuracy on Occlusion-RAF-DB and Pose-RAF-DB dataset

Model	Occ	Pose > 30	Pose > 45
RAN [47]	82.72	86.74	85.20
SCN [4]	82.18	86.45	**87.1**
Our	**83.40**	**87.89**	86.74

The best recognition results are shown in bold

Table 4 Ablation test results on RAF-DB

GCN	Triplet attention	Graph match	Accuracy
×	√	√	87.26
√	×	√	87.08
√	√	×	87.31
√	√	√	88.23

RAN [47] by 0.68%, 1.15%, and 1.54% in terms of accuracy on the three datasets. Comparing with SCN [4], our proposed method gains advantages in Occlusion-RAF-DB and Pose-RAF-DB (pose > 30). The experimental result emphasizes the effectiveness of our designed face key point graph and graph convolution.

4.5 Ablation experiment

To verify the effect of each module on expression recognition, we design ablation experiments to investigate GCN, triplet attention, and Graph Match on RAF-DB. For the ablation experiment of the GCN, we delete the GCN module and directly connect the first feature extraction module with the graph-matching module. The final ablation results are shown in Table 4. The experimental results show that after removing the GCN, our expression recognition accuracy is reduced by 0.97%, which shows that our GCN can well learn the information of the undisturbed part of the image, which is very important for in-the-wild datasets. For the triplet attention module, its effect on expression recognition is 1.15%, which indicates that triplet attention can pay attention to the information related to expression in space and channel. If without the graph match module, the accuracy of expression recognition will decrease by 0.92%, which indicates that the graph match module can guide our recognition network to distinguish similar expressions better.

4.6 Visualization

We visually compare our method with SCN [4] on RAF-DB, and the experimental results are shown in Fig. 9. Experimental results show that our method can recognize facial expressions better in the occlusion environment. However, our network cannot recognize the expressions of the last two pictures in Fig. 9, due to they are not only inapparent, but also seriously occluded.

5 Conclusion

This paper proposes a FERGCN deep neural network for recognizing facial expression in wild datasets. The proposed network consists of three modules: feature extraction module, GCN, and graph-matching module. In the feature extraction module, we use the key point information and triplet attention to guide the network to learn the local and global features of the face. In the GCN, we refine expression information to suppress the influence of complex environment. In the graph-matching module, we enhance the recognition ability of the network by reducing the similarity between classes. In addition, we also adopt hard sample triple loss to optimize our network. A large number of experimental results on FER datasets show that the proposed network performs well. On AffectNet, RAF-DB, and SFEW datasets, our method achieves 62.03%, 88.23%, and 56.15% recognition accuracy. Besides, it also achieves good results on the occlusion test subset of RAF-DB.

Fig. 9 Comparison of the SCN method and our method on RAF-DB

Acknowledgements The authors greatly appreciate the financial supports of Natural Science Foundation of Shanghai under Grant 19ZR1413400, National Natural Science Foundation of China under Grant 82170110, Science and Technology Commission of Shanghai Municipality under Grant 20DZ2254400.

References

1. Tian, Y.-I., Kanade, T., Cohn, J.F.: Recognizing action units for facial expression analysis. IEEE Trans. Pattern Anal. Mach. Intell. **23**(2), 97–115 (2001)
2. Shojaeilangari, S., Yau, W.-Y., Teoh, E.-K.: Pose-invariant descriptor for facial emotion recognition. Mach. Vis. Appl. **27**(7), 1063–1070 (2016)
3. Peng, Y., Yin, H.: Facial expression analysis and expression-invariant face recognition by manifold-based synthesis. Mach. Vis. Appl. **29**(2), 263–284 (2018)
4. Wang, K., Peng, X., Yang, J., et al.: Suppressing uncertainties for large-scale facial expression recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6897–6906 (2020)
5. Gui, S., Zhu, Y., Qin, X., et al.: Learning multi-level domain invariant features for sketch re-identification. Neurocomputing **403**, 294–303 (2020)
6. Pampouchidou, A., Pediaditis, M., Kazantzaki, E., et al.: Automated facial video-based recognition of depression and anxiety symptom severity: cross-corpus validation. Mach. Vis. Appl. **31**(4), 1–19 (2020)
7. Bai, Z., Cui, Z., Rahim, J. A., et al.: Deep facial non-rigid multi-view stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5850–5860 (2020)
8. Lucey, P., Cohn, J. F., Kanade, T., et al.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE computer society conference on computer vision and pattern recognition-workshops, pp. 94–101 (2010)
9. Valstar, M., Pantic, M.: Induced disgust, happiness and surprise: an addition to the mmi facial expression database. In: Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect, pp. 65 (2010)
10. Zhao, G., Huang, X., Taini, M., et al.: Facial expression recognition from near-infrared videos. Image Vis. Comput. **29**(9), 607–619 (2011)
11. Chen, S., Wang, J., Chen, Y., et al.: Label distribution learning on auxiliary label space graphs for facial expression recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13984–13993 (2020)
12. Zeng, N., Zhang, H., Song, B., et al.: Facial expression recognition via learning deep sparse autoencoders. Neurocomputing **273**, 643–649 (2018)
13. Li, S., Deng, W., Du, J.: Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2852–2861 (2017)
14. Dhall, A., Goecke, R., Lucey, S., et al.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 2106–2112 (2011)
15. Mollahosseini, A., Hasani, B., Mahoor, M.H.: Affectnet: A database for facial expression, valence, and arousal computing in the wild. IEEE Trans. Affect. Comput. **10**(1), 18–31 (2017)
16. Li, Y., Zeng, J., Shan, S., et al.: Occlusion aware facial expression recognition using CNN with attention mechanism. IEEE Trans. Image Process. **28**(5), 2439–2450 (2018)
17. Ding, H., Zhou, P., Chellappa, R.: Occlusion-adaptive deep network for robust facial expression recognition. In: 2020 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–9 (2020)
18. Boucher, J.D., Ekman, P.: Facial areas and emotional information. J. Commun. **25**, 21–29 (1975)
19. Misra, D., Nalamada, T., Arasanipalai, A.U., et al.: Rotate to attend: Convolutional triplet attention module. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3139–3148 (2021)
20. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737. (2017)
21. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: a comprehensive study. Image Vision Comput. **27**(6), 803–816 (2009)
22. Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 915–928 (2007)
23. Zhi, R., Flierl, M., Ruan, Q., et al.: Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition. IEEE Trans. Syst. Man Cybern. Part B (Cybernetics) **41**(1), 38–52 (2010)
24. Zhong, L., Liu, Q., Yang, P., et al.: Learning active facial patches for expression analysis. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2562–2569 (2012)
25. Liu, P., Han, S., Meng, Z., et al.: Facial expression recognition via a boosted deep belief network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1805–1812 (2014)
26. Meng, Z., Liu, P., Cai, J., et al.: Identity-aware convolutional neural network for facial expression recognition. In: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), pp. 558–565 (2017)
27. Mollahosseini, A., Chan, D., Mahoor, M. H.: Going deeper in facial expression recognition using deep neural networks. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–10 (2016)
28. Zhang, F., Zhang, T., Mao, Q., et al.: Joint pose and expression modeling for facial expression recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3359–3368 (2018)
29. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
30. Yang, H., Ciftci, U., Yin, L.: Facial expression recognition by de-expression residue learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2168–2177 (2018)
31. Cai, J., Meng, Z., Khan, A. S., et al.: Identity-free facial expression recognition using conditional generative adversarial network. arXiv preprint arXiv:1903.08051 (2019)
32. Liu, P., Lin, Y., Meng, Z., et al.: Point adversarial self mining: A simple method for facial expression recognition in the wild. arXiv preprint arXiv:2008.11401. (2020)
33. Jiang, P., Wan, B., Wang, Q., et al.: Fast and efficient facial expression recognition using a Gabor convolutional network. IEEE Signal Process Lett. **27**, 1954–1958 (2020)
34. Luan, S., Chen, C., Zhang, B., et al.: Gabor convolutional networks. IEEE Trans. Image Process. **27**(9), 4357–4366 (2018)
35. Bruna, J., Zaremba, W., Szlam, A., et al.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
36. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. Adv. Neural Inf. Process. Syst. **29**, 3844–3852 (2016)
37. Kipf, T. N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

38. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI Conference on Artificial Intelligence (2018)
39. Zhao, L., Peng, X., Tian, Y., et al.: Semantic graph convolutional networks for 3d human pose regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3425–3435 (2019)
40. Wang, G. a., Yang, S., Liu, H., et al.: High-order information matters: Learning relation and topology for occluded person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6449–6458 (2020)
41. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
42. Dong, X., Yan, Y., Ouyang, W., et al.: Style aggregated network for facial landmark detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 379–388 (2018)
43. Yovel, G., Duchaine, B.: Specialized face perception mechanisms extract both part and spacing information: Evidence from developmental prosopagnosia. J. Cognit. Neurosci. **18**(4), 580–593 (2006)
44. Zanfir, A., Sminchisescu, C.: Deep learning of graph matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2684–2693 (2018)
45. Wang, R., Yan, J., Yang, X.: Learning combinatorial embedding networks for deep graph matching. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3056–3065 (2019)
46. Dhall, A., Goecke, R., Lucey, S., et al.: Collecting large, richly annotated facial-expression databases from movies. IEEE Multimedia **19**(03), 34–41 (2012)
47. Wang, K., Peng, X., Yang, J., et al.: Region attention networks for pose and occlusion robust facial expression recognition. IEEE Trans. Image Process. **29**, 4057–4069 (2020)
48. Paszke, A., Gross, S., Massa, F., et al.: Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. **32**, 8026–8037 (2019)
49. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
50. Ji, Y., Hu, Y., Yang, Y., et al.: Cross-domain facial expression recognition via an intra-category common feature and inter-category distinction feature fusion network. Neurocomputing **333**, 231–239 (2019)
51. Gogić, I., Manhart, M., Pandžić, I.S., et al.: Fast facial expression recognition using local binary features and shallow neural networks. Vis. Comput. **36**(1), 97–112 (2020)

Lei Liao received his B.S. degree from East China University of Science and Technology in 2018. He is currently a postgraduate at the school of information science and engineering, East China University of Science and Technology. His research interests include image and video classification, deep learning, and pattern recognition.

Yu Zhu received the Ph.D. degree from Nanjing University of Science and Technology, China, in 1999. She is currently a professor in the department of electronics and communication engineering of East China University of Science and Technology. Her research interests include image processing, computer vision, multimedia communication, and deep learning, especially, for the medical auxiliary diagnosis by artificial intelligence technology. She has published more than 90 papers in journals and conferences.

Bingbing Zheng obtained the B.S. degree in Information Science and Engineering from East China University of Science and Technology in 2015 and is pursuing the Ph.D. degree in East China University of Science and Technology. His main research interests include deep learning for medical image processing and computer vision. His experience includes the identification and detection of pulmonary nodules on CT images, the classification and segmentation of prostate on MRI, and the classification and segmentation of COVID-19. He has published in journals and conferences in the crossing field of medical and computer vision and has been involved in publicly and privately funded projects.

Xiaoben Jiang is pursuing the Ph.D. degree in East China University of Science and Technology. His current research interests include digital image processing and computer vision. His experience includes the denoising method on chest X-ray images and CT images and detection of COVID-19 cases from denoised CXR images. He has published in journals in the crossing field of medical science and computer vision and has been involved in publicly and privately funded projects.

Jiajun Lin obtained his Ph.D. degree from Tsinghua University, Beijing. He is a professor at School of Information Science and Engineering, East China University of Science and Technology. His research interests include intelligent information processing and security of industry control systems.