Sparse deep feature learning for facial expression recognition

Weicheng Xie[a,b], Xi Jia[a], Linlin Shen[a,b,*], Meng Yang[c]

[a] *School of Computer Science & Software Engineering, Shenzhen University, China*
[b] *Guangdong Key Laboratory of Intelligent Information Processing*
[c] *School of Data and Computer Science, Sun Yat-sen University, China*

Abstract

While weight sparseness-based regularization has been used to learn better deep features for image recognition problems, it introduced a large number of variables for optimization and can easily converge to a local optimum. The L2-norm regularization proposed for face recognition reduces the impact of the noisy information, while expression information is also suppressed during the regularization. A feature sparseness-based regularization that learns deep features with better generalization capability is proposed in this paper. The regularization is integrated into the loss function and optimized with a deep metric learning framework. Through a toy example, it is showed that a simple network with the proposed sparseness outperforms the one with the L2-norm regularization. Furthermore, the proposed approach achieved competitive performances on four publicly available datasets, i.e., FER2013, CK+, Oulu-CASIA and MMI. The state-of-the-art cross-database performances also justify the generalization capability of the proposed approach.

Keywords: Expression recognition; Feature sparseness; Deep metric learning; Fine tuning; Generalization capability.

[*] Corresponding author. Tel: 86-0755-86935089. Fax: 86-0755-26534078.
Email address: `llshen@szu.edu.cn` (Linlin Shen)

1. Introduction

With the development of human-machine interaction, FER (Facial Expression Recognition) has been popular topic in recent decades. As reviewed by Zeng et al. [1], various algorithms have been proposed to directly model the recognition of expression images. A sparse linear model with Gabor phase shifts computed from facial videos is proposed to learn bases of activity for FER [2]. Common features and expression-specific features were disentangled with feature selection in [3] and de-expression residue learning [4]. For 2D FER, deep learning-based algorithms have surpassed the traditional methods that employ hand-crafted features [5]. However, the generalization ability of the recognition network is still limited. The limited discriminative ability of the traditional softmax loss function could be one of the reasons. Overfitting due to the large number of parameters is another problem. The generalization performance can be low when the variations in the geometry and texture among different persons and databases are large.

To improve the discrimination ability of the softmax loss function, a number of so-called deep metric learning approaches have been proposed to decrease the intra-class variance and increase the inter-class difference. Liu et al. [6] proposed the large-margin softmax loss to improve the intersection angle θ in the representation. More examples are the center loss [7], SphereFace [8], lifted structured loss [9] and tuplet loss with hard negative mining [10]. While these strategies encourage accurate identification, feature (FC input) norm regularization is another way to improve the discrimination ability. The study [11] has argued that enlarged $||x||_2$ for correctly classified samples is beneficial to push the feature vector away from the origin to encourage inter-class separability. L2-norm feature normalization [12, 13] was employed to decrease the influence of noise, the image resolution variation and the expression. However, the improved deep losses can increase the influence of noisy samples and the possibility of overfitting. Feature normalization useful for face verification also eliminates the information about the facial expressions.

When the diversity of the expression database used for training is not large enough, deep learning is likely to result in overfitting, and low generalization performance will occur on other databases. Currently, different algorithms, such as data augmentation [14], dropout [15], DropConnect [16], deep salient feature extraction [17], and sparse regularization [18, 19], have been proposed to address the overfitting to improve the network's generalization ability.

Deep networks with sparseness constraints can be used to decrease the possibility of overfitting [19]. Sparse learning, namely, compressed sensing, was proposed to reconstruct the original signal with sparse sampling [20]. Sparse representations are robust to noisy samples, can decrease the redundancy and extract common features among different databases, and can help to decrease the computational complexity of the training, which has been studied for face recognition [21], [22]. Algorithms such as discriminant non-negative matrix factorization [23] and atlas sparse features [24] have been applied for FER. At the same time, sparse features are beneficial to the generalization performance [25]. Two forms of sparsity, i.e., network weight sparsity and feature sparsity, are often employed for generalization ability improvement.

Some studies in the literature achieved weight sparseness by imposing the sparseness constraint on the weight matrix W of a deep network [25, 26, 27]. Ranzato et al. [28] imposed feature sparseness constraints on both the code vector and the weight coefficient for deep belief networks. Yan et al. [29] proposed sparse kernel reduced-rank regression to weight the contributions of the training data samples. Yu et al. [26] added L_1 sparseness to the weight matrix to reduce the number of nonzero weight coefficients. Liu et al. [30] introduced sparse decomposition to largely decrease the number of network parameters without a significant decrease in the performance. However, a network with weight s-parseness can be considered to be an additional optimization problem, whose optimization variables W can have high dimension. An optimization problem with such a large number of variables is likely to result in the gradient vanishing problem [31] for a network with very deep layers, and it can be trapped in a local optimum.

The sparseness of deep features is a promising direction to decrease the possibility of overfitting and to boost the network generalization ability, at the cost of optimization with a relatively smaller number of variables [32]. Ji et al. [33] studied the sparseness of the hidden response in deep belief network to achieve more powerful discriminative ability. Zou et al. [34] and Zeng et al. [35] imposed a sparseness constraint on the hidden unit output for autoencoder. Alam et al. [36] used dropout learning on all the hidden units of deep simultaneous recurrent networks for FER performance improvement and model size reduction. Li et al. [37] imposed mixed l_1/l_2 sparseness on the activation output of each hidden layer in deep stacking network for image classification.

However, the hidden unit sparseness imposes a constraint on each hidden layer [33, 34, 35, 37]. As a result, a large number of hidden units sparseness need to be optimized in the loss function, which may largely increase the computational burden and the possibility of stagnating to a local optimum. However, the network output layers contain moderate number of variables, whose dimensions are also independent of the network depth. Thus, sparseness on the network output layers can be easily integrated into deep convolutional neural network (CNN), without significant increase of model complexity. Meanwhile, compared with network hidden layers, the output layers determine the performance of the discrimination or classification models [38]. The inputs or outputs of the last two fully connected layers are often directly used as feature representations or are mathematically transformed for discrimination ability improvement. Jung et al. [39] introduced a fine-tuning network for FER by integrating the geometry features into a texture feature network and fine-tuned the last two layers of the network. Liu et al. [40] used the last three FC layers to construct the (N+M)-tuple clusters loss for FER. The studies [5, 41] employed the regression loss on the FC layers to use the information of face recognition network or hand-crafted features. Ioffe and Szegedy [42] conducted batch normalization on the FC layers to accelerate the network training and improve the feature discrimination ability by reducing the internal covariance shift. The L_2 norm of the FC vectors was fixed in [12, 13] to improve the network discrimination ability. Szegedy et al. [43] suggested that the sparseness of the FC layers can decrease the risk of overfitting and save on computational resources.

Furthermore, considering that the general feature sparseness or active patch selection of common features is beneficial to improve the FER and boost the generalization ability [44], the feature sparseness is adopted in this work by directly appending the sparseness of the FC input into the network loss function. Unlike dropout [15] and similar algorithms, the feature sparseness is optimized by minimizing the proposed loss function.

1.1. Motivation

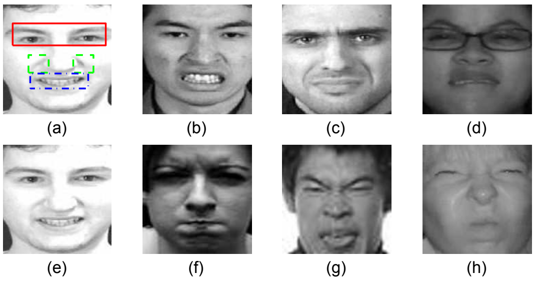


Figure 1: The 'disgust' expression of different persons in different databases. (a)-(d) the 'disgust' expression in MMI database [45], (e)-(h) the same expression in MMI [45], CK+ [46], FER2013 [47], Oulu-CASIA [48] databases, respectively. Red solid, green dashed, blue dash-dot rectangles, i.e. nose root, nasolabial and mouth denote the expression sensitive regions.

As shown in Figure 1, the expression features among different persons differ a large amount in the same database. The difference is even more significant for different databases, as shown in the 2nd row. The expression sensitive regions concentrate on the eyes, the nasolabial area and the mouth. An algorithm with good generalization ability should not only identify the expression non-sensitive regions but also emphasize a few discriminative regions for each expression. For example, it is more reasonable to extract the sparse feature on the nose root and the nasolabial region for 'disgust' recognition.

The network weight sparseness is formulated as follows

$$\begin{cases} \min \mathcal{L}_S + \lambda \sum_j ||W_j||_2^2, \\ \mathcal{L}_S = -\frac{1}{N} \sum_{i=1}^{N} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_j e^{W_j^T x_i + b_j}}, \end{cases} \tag{1}$$

where N is the number of samples; W, b are the weight matrix and the bias terms between the FC layers; x_i is the FC input of the i-th sample, and y_i is its class label; λ is the regularization parameter; $\mathcal{L}_S$ is Softmax loss function. For the weight sparseness in equation (1), the parameter space has a very high dimension. For example, when the numbers of the neurons in the last two layers are 4096 and 4096, respectively, the number of variables for the optimization of the weight sparseness term in equation (1) will be larger than 1.6×10^7. However, the optimization for adjusting a large number of parameters suffers from instability [32], and there is a large possibility of converging to a local optimum. At the same time, for a network with very deep layers, the gradients with respect to the network weights for back-propagation are small [31] in the latter evolution when the bipolarizing probabilities are input for a softmax loss.

Thus, a model that imposes feature sparseness directly on the FC layer input is proposed in this work to select active and common components among different expression databases, which is presented as follows

$$\min \mathcal{L}_S + \lambda \frac{1}{N} \sum_{i=1}^{N} ||x_i||_2^2, \tag{2}$$

where x_i and N are the same as that in equation (1). With this feature sparseness, the proposed algorithm can extract the common expression features, decrease the parameter space for optimization, and boost the generalization ability of the trained deep features.

As a closely related approach to the feature sparseness, the study in [12, 13] revealed that the L2 norm of deep features, i.e., the value of the variable α in equation (3), should be a large value for performance improvement for face recognition tasks.

$$\min \mathcal{L}_S, s.t.||x_i||_2 = \alpha, \text{for } i = 1, \cdots, N, \tag{3}$$

where α is a hyper-parameter that determines the radius of the parameter perturbation region, i.e. the L_2-norm of the feature vector x_i. Equation (2) is similar to the Lagrange representation of the optimization (3), i.e. $\mathcal{L}_S + \frac{1}{N}\sum_{i=1}^{N}\lambda_i(||x_i||_2^2 - \alpha^2)$, while the constraints on the regularization coefficients are different. The regularization coefficient λ in equation (2) should be positive, while the introduced regularization coefficients $\{\lambda_i\}$ of the Lagrange transformation of equation (3) should satisfy the KKT condition [49], whose signs are not constrained. The L2-norm regularization can decrease the influence of noise and the image resolution variation, which is beneficial for face recognition tasks. However, justified by the following toy example and experiments, useful expression features could be suppressed during the normalization since they can also change the feature norm, which is not desirable for the FER problem.

1.2. Contributions

In this work, a new framework for deep feature regularization based on FC input sparseness is proposed, in which the input vectors of the last two FC layers are made sparse to generalize the discrimination ability of Wx. The feature sparseness is then solved by deep metric learning.

The main contribution of this work is summarized as follows:

- A new framework of facial expression recognition (FER) is proposed, where different feature sparseness strategies are embedded and investigated;
- Feature sparseness of the FC input is embedded into a deep network to boost the feature generalization ability;
- The deep metric learning achieved competitive recognition rates on four benchmark expression databases.

This paper is structured into the following sections. The proposed approach is demonstrated in Section 2, where the feature sparseness is introduced in Section 2.2. The experimental results and the corresponding illustrations are demonstrated in Section 3. Finally, the conclusions and a discussion are presented in Section 4.

2. The proposed algorithm

In the following section, the proposed algorithm is introduced. First, the employed preprocessing for face alignment and augmentation is introduced. Then, the proposed feature sparseness and the corresponding optimization algorithm are demonstrated. Finally, the employed network configurations and the fine-tuning strategy between different databases are introduced.

2.1. Preprocessing

For the face alignment, the five key points are first located on the eyes, nose and mouth tips. Then, the faces are aligned, cropped and scaled with the three key points presented in Figure 2 (b). The database is augmented by cropping different regions, as shown in Figure 2(c).

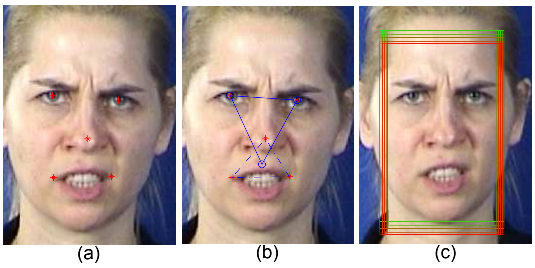


Figure 2: The data preprocessing. (a) the five landmark points. (b) the transformed key points for image alignment, cropping and scaling. (c) the image regions with different colors are cropped.

To guarantee the similar alignment among different databases, the face size of the CK+ database is used to scale the faces in the other databases. In other words, the average lengths of the lines in the blue solid triangle and the distances from the four boundaries to the solid triangle are first recorded for the CK+ database, and they are used to scale the corresponding lengths of the other databases. Then, the same margin beyond the solid triangle region in Figure 2(b) is extracted as a reference to generate multiple regions (Figure 2(c)). Finally, each expression image I is normalized in its gray level and is mirrored and scaled to the size 227 × 227 for the training.

2.2. The feature sparseness

To improve the generalization ability of the learned features, the sparseness of the FC input is integrated into the framework of deep metric learning for network training. The structure of the proposed feature sparseness is presented in Figure 3 together with the hidden unit sparseness [35] and the weight sparseness structures. The key symbols employed in the proposed algorithm are illustrated in Table 1.

The proposed sparseness losses of the FC features are formulated as follows:

$$\begin{cases} \mathcal{L}_{FC1} = \frac{1}{p}\sum_i \mathcal{N}_p^p(z_i), \\ \mathcal{L}_{FC2} = \frac{1}{p}\sum_i \mathcal{N}_p^p(x_i), \end{cases} \tag{4}$$

where $\mathcal{N}_p^p(z_i)$ denotes the p-th power of the L_p-norm $\mathcal{N}_p(z_i)$ of z_i, which is $\sqrt{\sum_j z_{i,j}^2}$ when $p = 2$ and $\sum_j |z_{i,j}|$ when $p = 1$.

Figure 3 shows that the proposed feature sparseness is largely different with the weight sparseness, which imposes sparseness on the weight matrix between layers, and it is additionally different from the hidden unit sparseness [35, 36, 37], which imposes sparseness on the hidden unit outputs of multiple layers.

Compared with the regularization term of the weight sparseness, the parameter dimension of the sparseness term in the proposed algorithm is not larger than the number of neurons of FC1 (4096) and FC2 (4096), i.e., $4096 + 4096 \approx 8.2 \times 10^3$ for the VGG model. For the network learned using the weight V sparseness, the parameter dimension of the sparseness term is not less than the product of the number of neurons, i.e., $4096 \times 4096 \approx 1.6 \times 10^7$, which is significantly larger than that of the proposed approach and the number of training

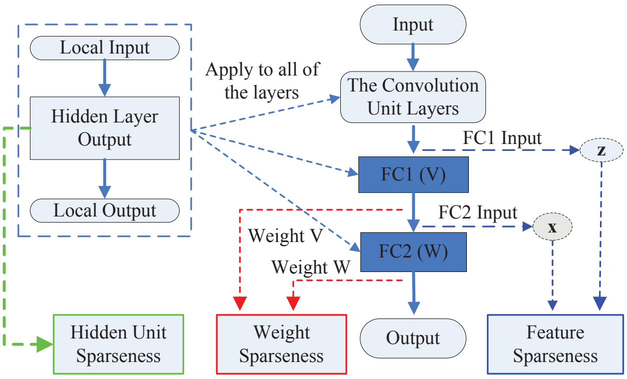


Figure 3: The hidden unit, weight and the proposed feature sparseness. V and W are the weight matrices linking the FC layer input with the corresponding output. z and x are the inputs of FC1 and FC2, respectively.

samples ($\leq 10^5$).

Compared with hidden unit sparseness, the proposed loss imposes sparseness on the inputs of the last two FC layers, while hidden unit sparseness imposes constraints on the activation output of each hidden layer [37] or the hidden layer of each sub-network [35, 36]. Thus, the proposed sparseness largely simplifies the model complexity for sparseness optimization.

While the feature sparseness is proposed to constrain the features, the softmax loss is then employed to optimize the recognition accuracy. Thus, the loss function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_S + \lambda_{FC1}\mathcal{L}_{FC1} + \lambda_{FC2}\mathcal{L}_{FC2}, \tag{5}$$

where λ_{FC1}, λ_{FC2} are the regularization parameters, and the softmax loss $\mathcal{L}_S$ is presented as follows:

$$\mathcal{L}_S = -\sum_i log\frac{e^{W_{y_i}^T x_i}}{\sum_j e^{W_j^T x_i}} = -\sum_i log\frac{e^{W_{y_i}^T V^T z_i}}{\sum_j e^{W_j^T V^T z_i}}, \tag{6}$$

where the network bias parameters $\{b_j\}$ are set to 0 in the formula for simplification. The fusion of center loss [7] for performance improvement will be introduced in Section 2.3.

For better presentation of the algorithms implementation, the entire network is divided into two sub-networks, i.e., the sub-network before the last but one FC layer, **MI**, and the sub-network with the last two FC layers, **MII**. Their loss criteria are denoted as **CI** and **CII**, respectively.

Table 1: Illustration of the employed symbols.

Name	Remark	Name	Remark
$FC1$	The last but one FC layer	$FC2$	The last FC layer
MI	The layers before $FC1$	**CI**	Use fused loss in equation (9)
MII	Use **MI** output as input	**CII**	Use SoftMax loss
$\mathcal{L}_{FC1}$	The feature loss of $FC1$	$\mathcal{L}_{FC2}$	The feature loss of $FC2$
W	The weights linking FC2 and the network output (**MII** weight)	V	The weights linking FC1 and FC2 ($n_{FC1} \times n_{FC2}$-dim)
z_i	The input of the i-th sample of FC1 (**MI** output)	x_i	The input of the i-th sample of FC2 ($x_i = V^T z_i$)
ϖ	The weight of **MI**	y_i	The expression label of z_i

For the network optimization by back propagation, the gradients of the loss in equation (5) w.r.t. the features are presented as follows:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial z_i} = \frac{\partial \mathcal{L}_S}{\partial z_i} + \lambda_{FC1} \frac{\partial \mathcal{L}_{FC1}}{\partial z_i} + \lambda_{FC2} V \frac{\partial \mathcal{L}_{FC2}}{\partial x_i}, \\ \frac{\partial \mathcal{L}}{\partial x_i} = \frac{\partial \mathcal{L}_S}{\partial x_i} + \lambda_{FC2} \frac{\partial \mathcal{L}_{FC2}}{\partial x_i}, \\ \frac{\partial \mathcal{L}_{FC1}}{\partial z_i} = \begin{cases} z_i & p = 2 \\ (SIGN(z_{i,1}), \cdots, SIGN(z_{i,n_{FC1}})) & p = 1 \end{cases} \end{cases} \tag{7}$$

where $SIGN(\cdot)$ is the sign function. The derivative $\frac{\partial \mathcal{L}_{FC2}}{\partial x_i}$ can be similarly induced. Since the Cross-Entropy function is employed for the softmax loss $\mathcal{L}_S$, the partial derivatives $\frac{\partial \mathcal{L}_S}{\partial x_i}$, $\frac{\partial \mathcal{L}_S}{\partial z_i}$, $\frac{\partial \mathcal{L}_S}{\partial W}$, $\frac{\partial \mathcal{L}}{\partial W}$ and $\frac{\partial \mathcal{L}}{\partial \varpi}$ are automatically obtained by the network backward of **MII** or **MI**. The backward computation is often embedded in the deep learning framework, such as Caffe, which can be directly used for the network optimization.

With the obtained gradients in equation (7), the network parameters are iteratively updated with the optimization according to Stochastic Gradient Descent (SGD), as follows:

$$\begin{cases} z^{t+1} = z^t - \mu_2^t \frac{\partial \mathcal{L}}{\partial z^t}, x^{t+1} = x^t - \mu_2^t \frac{\partial \mathcal{L}}{\partial x^t}, \\ V^{t+1} = V^t - \mu_2^t \frac{\partial \mathcal{L}}{\partial V^t}, W^{t+1} = W^t - \mu_2^t \frac{\partial \mathcal{L}}{\partial W^t}, \\ \varpi^{t+1} = \varpi^t - \mu_1^t \frac{\partial \mathcal{L}}{\partial \varpi^t}, \end{cases} \tag{8}$$

where μ_1^t, μ_2^t are the learning rates w.r.t. **MI**, **MII**, respectively.

The optimization framework of the proposed feature sparseness is illustrated in Algorithm 1, where the implementation of the network training is demonstrated. For the network optimization in Algorithm 1, the forward and backward operations of each model (**MI** or **MII**) and criterion (**CI** or **CII**) are used. The model forward operation gives the output for each layer; then, the criterion for the forward operation computes the final loss function based on the network output; the criterion for the backward operation obtains the derivatives of the loss function w.r.t. the network output; finally, the model backward operation computes the derivatives of the loss function w.r.t. the network input and the weight parameters.

Algorithm 1 Feature sparseness-based network training.

```
Set the parameters $\lambda_{FC1}$, $\lambda_{FC2}$, $MaxIter = 3e3$.
Initialize the network parameters $\varpi$ of MI, the weight vector $V$ and $W$ of MII.
for $s = 0, \cdots, MaxIter$ do
  Perform MI forward to obtain the output $z_i$.
    Perform MII forward to obtain $x_i$ and $W_j^T x_i$.
    Perform CII forward to obtain the loss value $\mathcal{L}_S$ in equation (6).
    Perform CII backward to obtain the gradient of the output, i.e. $\frac{\partial \mathcal{L}_S}{\partial (W_j^T x_i)}$.
    Perform MII backward to compute the gradients, i.e., $\frac{\partial \mathcal{L}_S}{\partial x_i}$, $\frac{\partial \mathcal{L}_S}{\partial z_i}$, $\frac{\partial \mathcal{L}_S}{\partial W}$ and $\frac{\partial \mathcal{L}_S}{\partial V}$.
    Perform SGD in equation (8) to update $V, W$.
  Perform CI forward to obtain the entire loss function $\mathcal{L}$ in equation (5).
  Perform CI backward to obtain $\frac{\partial \mathcal{L}}{\partial z_i}$ in equation (7).
  Perform MI backward to compute the gradients $\frac{\partial \mathcal{L}}{\partial \varpi}$ of model MI.
  Perform SGD in equation (8) to update $\varpi$.
end for
Output the trained network for testing.
```

In the following sections, the loss with the L_2-norm ($p = 2$) or L_1-norm ($p = 1$) sparsity on the input of the last FC layer is denoted as L2 or L1. When imposing the constraint on only the input of the FC2 layer, the regularization coefficient λ_{FC1} is set to 0. When L1 or L2 sparsity is applied to both the layers, i.e., $\lambda_{FC1} \neq 0$, $\lambda_{FC2} \neq 0$, the loss function is denoted as L1L1 and L2L2, respectively.

2.3. Fusion with center loss

With the proposed feature sparseness, the center loss [7] can be fused into the proposed loss in equation (5) to further improve the discriminant ability of the feature x_i. The minimization optimization of the fused loss is formulated as follows:

$$\mathcal{L}_{fuse} = \mathcal{L} + \lambda_C \mathcal{L}_C, \tag{9}$$

where λ_C is the regularization parameter of the center loss, and the loss $\mathcal{L}_C$ is presented as follows:

$$\mathcal{L}_C = \tfrac{1}{2} \textstyle\sum_i ||x_i - c_{y_i}||_2^2. \tag{10}$$

where c_{y_i} denotes the y_i-th center vector.

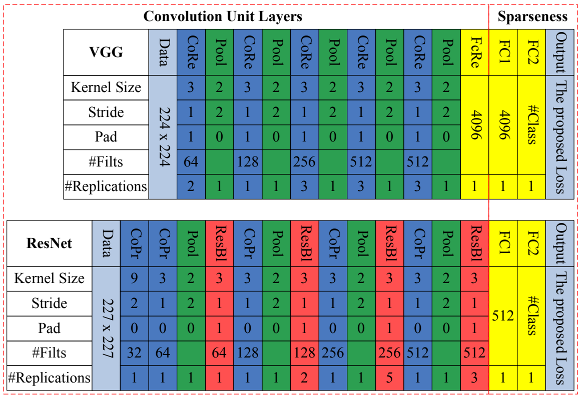
For the network optimization in equation (9), the gradients of the center loss w.r.t. x_i and z_i are added into the corresponding formulas in equation (7), which are presented as follows:

$$\begin{cases} \frac{\partial \mathcal{L}_C}{\partial x_i} = x_i - c_{y_i}, \\ \frac{\partial \mathcal{L}_C}{\partial z_i} = V \frac{\partial \mathcal{L}_C}{\partial x_i}. \end{cases} \tag{11}$$

To implement Algorithm 1 for the optimization in equation (9), the parameter λ_C and weight vector c_j should be initialized first. At the end of each iteration, the centers c_j should be updated using the gradient $\frac{\partial \mathcal{L}_{fuse}}{\partial c_j}$ [7].

2.4. The networks and fine tuning

Two networks, i.e., the VGG [50] and ResNet [31] with slight modification, are used for the training and evaluation in this work. For VGG, an additional FC layer is appended to the original network. The new VGG network employed a small convolution kernel size and more network layers to increase its non-linear capability. The dropout strategy in the FC layers is removed. The modified VGG configuration is presented in Figure 4.



	Convolution Unit Layers												Sparseness		
VGG	Data	CoRe	Pool	CoRe	Pool	CoRe	Pool	CoRe	Pool	CoRe	Pool	FcRe	FC1	FC2	Output
Kernel Size	224 x 224	3	2	3	2	3	2	3	2	3	2	4096	4096	#Class	The proposed Loss
Stride		1	2	1	2	1	2	1	2	1	2				
Pad		1	0	1	0	1	0	1	0	1	0				
#Filts		64		128		256		512		512					
#Replications		2	1	1	1	3	1	3	1	3	1	1	1	1	

ResNet	Data	CoPr	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	CoPr	Pool	ResBl	FC1	FC2	Output
Kernel Size	227 x 227	9	3	2	3	3	2	3	3	2	3	3	2	3	512	#Class	The proposed Loss
Stride		2	1	2	1	1	2	1	1	2	1	1	2	1			
Pad		0	0	0	1	0	0	1	0	0	1	0	0	1			
#Filts		32	64		64	128		128	256		256	512		512			
#Replications		1	1	1	1	1	1	2	1	1	5	1	1	3	1	1	

Figure 4: The structures of two networks. $CoRe, CoPr$ denote the convolutions followed by the ReLU, PReLU activation functions, respectively. $Pool$ is the MaxPooling function. $ResBl$ is a residual block with output $ResOutput = PoolOutput + CoPr(CoPr(PoolOutput))$. $FcRe$ denotes the module of FC layer, followd by ReLU layer. $\#Replications$ denotes the number of times the same block is replicated. $\#Filts$ denotes the number of feature maps. The 'Convolution Unit Layers' are used in Figure 3.

The residual network (ResNet) appends the residual mapping $\mathcal{F}$ to the identity mapping x to estimate the output $\mathcal{H} = \mathcal{F} + x$, rather than fitting the output $\mathcal{H}$ directly. ResNet was reported to be able to decrease the possibility of weight gradient vanishing when the network is very deep [51]. The kernel size of the 1st convolution is modified to 9, and the number of neurons for the last but one FC layer is modified for FER. The configuration of the modified ResNet network is presented in Figure 4.

To fully make use of the already trained models, a fine tuning strategy between the databases is presented in Figure 5. Although the network trained for face recognition is different from that for FER, the learned network parameters can be used as the initialization for transfer learning. The two models are first fine-tuned using FER2013, and then, they are further tuned for other databases.

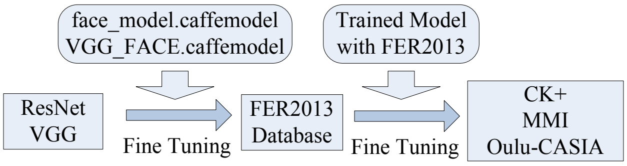


Figure 5: The fine tuning of models for different databases, i.e. FER2013, CK+, Oulu-CASIA and MMI. face_model.caffemodel and VGG_FACE.caffemodel are two CAFFE model trained for face recognition corresponding to networks ResNet and VGG, respectively.

When applying the model trained for seven-class expressions to a six-class problem, the weight matrix that links the last two FC layers was initialized and learned from scratch since the numbers of the FC neurons are different.

For the recognition of each testing sample, majority voting of the probability of augmented face regions in Figure 2(c) is employed, which is presented as follows:

$$Label_i = arg \max_{1 \leq k \leq \#class} \sum_{j=1}^{n_i} v_{i_j,k}, \tag{12}$$

where n_i is the number of augmented faces of the i-th sample with Figure 2 (c), and $v_{i_j,k}$ is the k-th output probability of the j-th augmented face. $Label_i$ is the finally assigned label of the i-th testing sample.

3. Experimental results

A four-kernel Nvidia TITAN GPU Card and Caffe platform are used for the algorithm testing. For the SGD optimization of the proposed algorithm, the batch size is set to 64; the momentum is 0.9; the coefficient $\lambda_{FC1} = \lambda_{FC2}$ is set to 1e-5 for the two-layer sparseness, and 0 for one-layer sparseness; the coefficients λ_{FC2} of L2 and λ_{FC1} of L1 sparseness terms are 1e-5; λ_C is fixed to 8e-3; the base learning rate is 1e-2; the weight decay factor of the learning rate is 5e-4; and the step size is 4000.

3.1. Databases

The expression databases of FER2013 [47], CK+ [46], Oulu-CASIA [48] and MMI [45] are used for the performance evaluation.

The FER2013 database [47] consists of 35887 grayscale face images with size 48x48, which are collected from the internet and used for a challenge. The faces were labeled with one of seven categories, i.e., angry (An), disgust (Di), fear (Fe), happy (Ha), sad (Sa), surprise (Su) and neutral (Ne). The training set consists of 28,709 examples. The public test (validation) set used for the algorithm development consists of 3,589 examples. The final test (testing) set, which was used to determine the winner of the competition, consists of another 3,589 examples.

The CK+ database consists of 593 expression sequences from 123 subjects, where 327 sequences are labeled with one of seven expressions, i.e., angry, disgust, fear, happy, sad, surprise and contempt, and 'contempt' is not considered. Each sequence was captured when the subject changed his expression, with 1033 expression images, i.e., the neutral and three peak frames sampled from each expression sequence were used for testing. For the testing, 415 expression sequences from 160 person identities were selected, which was further augmented as in Figure 2 to include 22410 images.

The Oulu-CASIA NIR&VIS expression database [48] contains videos of 80 subjects, and each acts with the six typical expressions to generate the corresponding expression sequence. The images are captured with two imaging systems, NIR (Near Infrared) and VIS (Visible light), under three different illumination conditions, i.e., normal (strong) indoor illumination, weak illumination (only the computer display is on) and dark illumination (all lights are off). The three peak expressions in each sequence of the database of NIR and *Strong* are used, followed by a simple data augmentation with 16 different crops for each image, to include 23040 images.

The MMI database [45] includes more than 20 subjects of both genders (44% female), which range in age from 19 to 62, with either a European, Asian, or South American ethnicity. Each expression sequence consists of a neutral expression at the beginning and end, while the between images present one of the typical expressions with different deformation intensities. Three peak frames in each of 205 expression sequences are employed for testing, and the augmented database includes 15375 images.

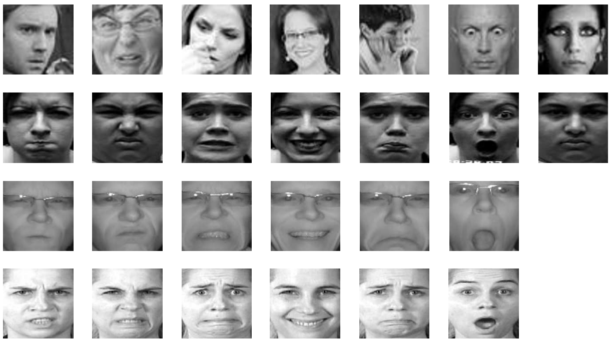


Figure 6: Example images of FER2013, CK+, Oulu-CASIA and MMI. The columns represent expressions of An, Di, Fe, Ha, Sa, Su and Ne, respectively.

Example expressions of each database are presented in Figure 6. The databases CK+, Oulu-CASIA and MMI are aligned using the process shown in Figures 2(b), 2(c). The popular employed data partition strategy, i.e., ten-fold person-independent cross-validation is conducted for the experiments. When one-fold samples are used for the testing, the samples of the other nine folds are used for the training, and the average of the ten-fold recognition rates is used as the final performance.

3.2. Performance evaluation with a toy model

In this section, a simple network with a convolution layer and two FC layers is designed to show the advantages of the proposed L2 feature sparseness approach described in equation (2) against the L2-norm regularization in equation (3).

For the simple network, the number of features, kernel size and stride are 1, 9 and 2, respectively. To train the network for the six-expression classification problem, the numbers of neurons for the two FCs are set as 32 and 6. Figure 7(a) shows the 1st FC layer and the convolution layer connected to it. After the network was trained using 13500 images, a number of 150 face images with a fear expression was used to elaborate the importance of the learned features extracted at different facial parts for this particular expression. For the m-th facial part ($PART_m$), such as the nose root, eyes and brows, cheek and mouth, a quantitative metric is designed to measure their importance for identifying the fear expression:

$$RespondRatio_{e,m} = \frac{\sum_{i\in SET_e}\sum_{j\in I^e_{50\%}} e^{\sum_{k\in PART_m} V_{j,k} z_{i,k}+b_j}}{\sum_{i\in SET_e}\sum_{j\in I^e_{50\%}} e^{\frac{\#PART_m}{\sum_m \#PART_m} V_j^T z_i+b_j}} \qquad (13)$$

where $\#PART_m$ denotes the number of pixels of region $PART_m$, V is the weight matrix that connects the convolution layer and the 1st FC layer, b is the bias, z_i is the i-th feature map, x_i is the FC2 input of the i-th sample in the set of face images with the e-th feature expression SET_e and $I^e_{50\%}$ records the top 50% components of x_i, in terms of the response value.

For a larger quantitative value $RespondRatio_{e,m}$, the ratio between the sum of the key response values and the sum of all of the response values is larger. The greater the contribution of the considered part to the feature representation is, the more important the part is to the final recognition.

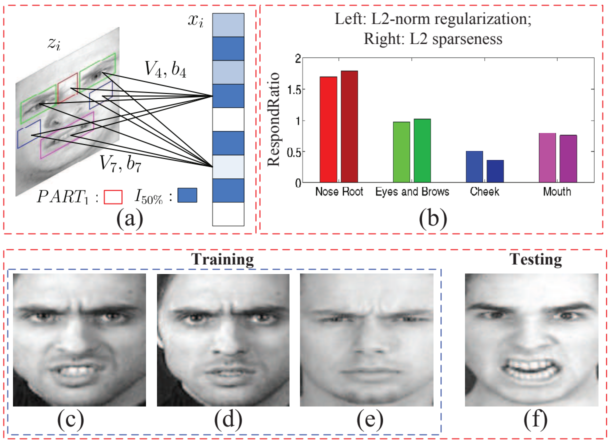


Figure 7: The comparison of the L2-norm regularization and the L2 sparseness. (a) the toy network. (b) the importance of different regions. (c)-(f) example training and testing 'fear' expressions.

Table 2: The response ratios of different expressions w.r.t. different facial parts in the MMI database. NM and SP denote the L2-norm and L2-sparseness, respectively.

Exp.	Nose root		Eyes and brows		Cheek		Mouth	
	NM	SP	NM	SP	NM	SP	NM	SP
An.	1.03	**1.08**	1.11	**1.14**	**0.54**	0.49	**0.85**	0.84
Di.	1.22	**1.31**	**1.19**	1.17	**0.77**	0.73	0.92	**0.97**
Fe.	1.69	**1.78**	0.97	**1.01**	**0.50**	0.36	**0.79**	0.76
Ha.	**0.91**	0.89	1.04	**1.07**	**0.83**	0.80	1.11	**1.16**
Sa.	1.45	**1.51**	1.03	**1.06**	**0.71**	0.64	1.05	**1.09**
Su.	**0.96**	0.91	1.10	**1.18**	**0.79**	0.75	1.07	**1.1**

Figure 7(b) shows the calculated metrics for the nose root, eyes and brows, cheek and mouth of the 'fear' expression, when the network was trained using the L2-norm regularization and L2 sparseness. One can observe from the histogram that the network trained using L2 sparseness puts more emphasis on the nose root, eyes and brows for 'fear' recognition. Because the feature trained using L2-norm regularization puts more emphasis on the cheek and mouth with training data (e.g., Figures 7(c)-7(e)), the feature can get easily confused when the faces with the fear expression present large variations in the mouth region (e.g., Figure 7(f)) and the faces with different expressions present small differences in the cheek region. The quantitative values of all of the six expressions are presented in Table 2, where similar findings are observed on the Cheek, Eye and Brow regions.

3.3. Feature sparseness analysis

In this section, different sparse models are tested using the FER2013 database, and then, the best model is selected for the following evaluation. Afterward, the proposed L2 sparseness is compared to the hidden unit sparseness, the network weight sparseness and L2-norm regularization. Finally, the proposed feature sparseness is compared to the center loss and dropout [15].

To see the performance of different sparse strategies in equation (5) on the ResNet and VGG models, Table 3 demonstrates the recognition rates of four different sparse models for the FER2013 database. Center loss is not considered in this stage, i.e., λ_C is zero, and the softmax loss is employed. One can observe from Table 3 that the proposed feature sparsity improves the performance, i.e., the L2 sparseness imposed on the VGG and ResNet models achieves improvements of 3.2% and 2.2% over SoftMax, respectively. At the same time, the best recognition rate, i.e., 71.91%, is achieved by ResNet with the L2 sparsity. Though L1L1 achieved similar performance, it requires much more parameters and computational complexity for gradient calculation.

Table 3: The recognition rates (%) of different sparsity models for the FER2013 database. L2 (or L2L2) denotes the loss with the L_2-norm regularization on the FC2 input (or both FC1 and FC2 inputs). $L2^{H,S}$ and $L2^{H,D}$ denote the $L2$ sparseness imposed on the activation output of all the hidden layers, where $L2^{H,S}$ employs the 'same' regularization coefficient as the default setting; $L2^{H,D}$ employs the 'divided' coefficient, i.e. $Default/\#layer$. $L1^{H,S}$ and $L1^{H,D}$ are similarly defined. Center loss is not used.

Model	SoftMax	Feature Sparseness				Hidden Unit Sparseness			
		L2	L1	L2L2	L1L1	$L2^{H,S}$	$L1^{H,S}$	$L2^{H,D}$	$L1^{H,D}$
VGG	66.84	**70.08**	68.18	68.04	**70.08**	68.49	69.38	69.16	68.57
ResNet	69.71	**71.91**	71.32	71.38	71.88	71.27	71.52	71.69	71.36

While sections 1 and 2.2 have shown the differences between hidden unit sparseness and the proposed feature sparseness, the performance of the hidden unit sparseness is further compared to that of the proposed feature sparseness on the FER2013 database in Table 3. Although hidden unit sparseness largely reduces the number of nonzero numbers with L_1-norm or the number of large elements with L_2-norm, it introduces large number of variables in the sparseness term. For example, the number of hidden units of 12 hidden layers in ResNet, i.e. 582,912, is significantly larger than that of the proposed feature sparseness, i.e. 512. Larger number of variables in the hidden unit sparseness yields higher computational complexity and larger possibility of trapping in a local optimum. Although significantly smaller number of variables are used in the proposed sparseness, the feature sparseness on the FC input still achieves better performance than the hidden unit sparseness in Table 3, where improvements of 0.7% and 0.22% are observed for VGG and ResNet, respectively.

To statistically evaluate the performance of the proposed feature sparseness, two statistical measures, i.e. sensitivity (i.e. the true positive rate) and specificity (i.e. the true negative rate = 1 - the false positive rate), proposed for two-class classification problems [52], are employed for the comparison between the Softmax loss and the L2-sparseness loss on the FER2013 database. To apply the two measures for seven-class problem, an independent computation of the two measures are conducted for each expression class, while the activation value at the dimension of ground-truth label after Softmax function is used as the prediction probability of each sample. The average values [52] are used to evaluate and compare the performance, as presented in Table 4. Meanwhile, the average Receiver Operating Characteristic (ROC) curve for the seven expressions and the corresponding Area under Curve (AUC) are demonstrated in Figure 8.

Table 4: The statistical measures (%) of the Softmax and L2-sparseness losses for the FER2013 database.

Loss	Softmax loss		L2-sparseness and Softmax losses	
Statistical measure	Sensitivity	Specificity	Sensitivity	Specificity
Value	83.71	81.88	**84.77**	**85.72**

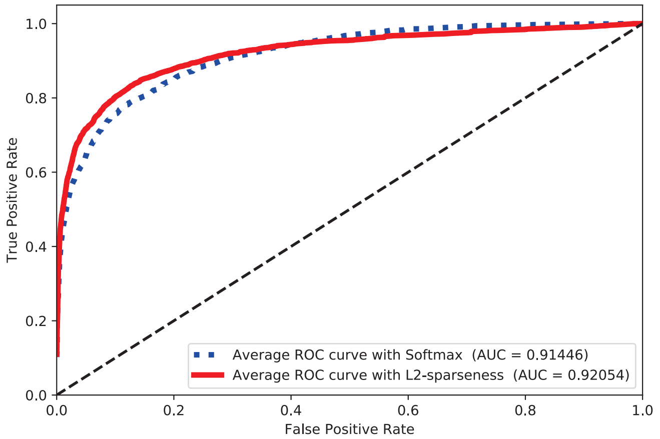


Figure 8: The average ROC curves and the corresponding values of AUC.

Compared with Softmax loss, the larger AUC obtained by the proposed L2-sparseness further justifies its better overall performance over the Softmax loss, while the advantage of the proposed L2-sparseness is more obvious in the case of small false positive rate. Since sensitivity and specificity measures are the balance points of the missed- and false-recognition rates on the ROC curve, the better sensitivity and specificity achieved by the proposed L2-sparseness in Table 4 shows that the proposed sparseness can better balance the missed- and false-recognition rates.

To compare the performance of the proposed L2 feature sparseness with the weight sparseness, the sparseness of the weight matrixes V, W of the last three FC layers is considered. The regularization coefficient of the weight sparseness is 1e-5, and the other parameters are fixed as the same as the proposed L2 feature sparseness. The recognition rates of the two sparse models on FER2013 are presented in Table 5.

Table 5: The performance (%) of the L2 feature, weight and connection pruning (c.p.) [26] based sparseness on the FER2013 database. W-L2-c.p. denotes the connection reduction ratio for the weight matrix W with L2 sparseness. Center loss is not used.

Models	W-L2	V-L2	V-W-L2	W-L1-c.p.	W-L2-c.p.	V-L1-c.p.	V-L2-c.p.	L1	L2
VGG	66.87	67.07	67.07	66.95	67.29	67.79	68.01	68.18	**70.08**
ResNet	68.24	69.41	68.51	69.52	69.77	70.30	70.33	71.32	**71.91**

The weight matrixes V, W in the weight sparseness model contain $512 \times 6 \times 6 \times 512$, $\#classes \times 512$ elements, respectively, which are significantly larger than those of the proposed feature sparseness (512). The larger number of optimization parameters in the V weight sparseness implies a larger possibility of convergence to a local optimum. While the output probability vector used for SoftMax is made sparse, it has little effect on the features. In contrast, the proposed L2 sparseness imposes the feature sparsity directly on the FC neurons with a moderate number of variables. Table 5 reveals that the proposed L2 sparseness significantly outperforms the sparseness strategies of the weights W, V and both W and V. Improvements of 3.67%, 2.5%, 3.4% with ResNet (3.21%, 3.01%, 3.01% with VGG) were observed on the FER2013 database.

The sparseness model [26] improved the weight sparseness by mandatory connection pruning (c.p.). For this comparison, the weight sparseness with a 30% reduction suggested in [26] is employed for the evaluation, where a binary mask matrix is used to set the pruned weights to zero. Although better performance than the V-sparseness was achieved, the c.p. sparseness [26] was mainly proposed to reduce the model size. Table 5 shows that the feature sparseness achieved better performance than the c.p. L2 V-sparseness, and improvements of 1.58% with ResNet and 2.07% with VGG were observed on the FER2013 database.

As a closely related approach, the L2-norm regularization was proposed in [12, 13] to regularize the learned deep feature for face recognition. However, as illustrated in section 1.1, the FER problem is different, and the strategy to keep the norm of the learned feature constant might not be working in this case. In this testing, the L2-norm regularization and L2-sparseness approaches are compared using ResNet and FER2013. The results on the validation and testing data are shown in Figure 9, where one can observe that the L2 sparseness significantly outperforms the L2-norm regularization with α being fixed to 2 or 0.5. While the self adaptive algorithm [1] of the feature normalization [12, 13] uses adaptive α to improve the network performance, the proposed sparse deep feature still achieved better performance on the testing dataset.

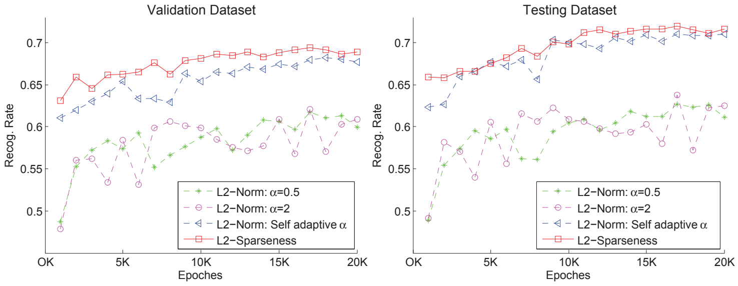


Figure 9: The validation and testing accuracies of L2-norm regularization and L2-sparseness for the FER2013 database under different parameter settings, where 1K=1000.

Considering the difference of the L2-norm regularization and L2 sparseness for face recognition and FER, the expression feature is the perturbed information of face recognition. The studies in [12, 13] stated that feature normalization

[1] $https://github.com/happynear/caffe-windows/blob/$

can decrease the influence of the factors that change the feature norm, such as the image resolution, for face recognition. Since the expression features can change the feature norm, they are also suppressed during the feature normalization. While L2 feature normalization can suppress the expression variation and benefit face recognition, it is not desirable for the FER problem. However, the proposed L2 sparseness can extract the common features among different personal identities and databases, which is beneficial to FER.

While L2 sparseness has been shown to achieve better performance than weight sparseness and L2-norm regularization, the proposed L2 sparse loss with the dropout strategy [15] and center loss [7] are now further combined for performance improvement. Table 6 lists the accuracy of the ResNet and VGG networks using different combinations. Dropout with probability 0.5 was applied to the input of the last two FC layers. The comparison between the 4th and 6th rows (or the 5th and 8th rows, or the 1st and 2nd rows) of Table 6 shows that the dropout strategy is not beneficial to the performance of both networks. The results in the 4th and 8th rows (or the 2nd and 7th rows) of Table 6 show that the L2 sparseness outperforms the dropout strategy on FER2013. One reason is that the proposed approach made the FC input sparse by L2 sparsity constraint, while the dropout strategy makes the FC input sparse randomly. Another reason is that the proposed L2 sparseness actively selects the common features among the different databases. While the L2 sparse loss (71.91%) achieved better performance than center loss (71.58%) for ResNet (6th and 7th rows), its accuracy (70.08%) is slightly lower than that of center loss (70.33%) for VGG. The combination of L2 sparseness and center loss achieved the best performance for both networks.

3.4. Results on four databases

While the proposed L2 feature sparsity has been shown to give better results than other regularizations such as L2-norm regularization, ResNet performs better than VGG on the FER2013 database; the performance is tested with ResNet trained using the combination of center loss and L2 sparseness loss (Eq.

Table 6: The accuracy (%) for different combinations of the softmax loss (SoftMax), dropout, center loss (center) and feature sparseness (L2) on the FER2013 database.

Id	SoftMax	Dropout	Center	L2	ResNet	VGG
1	Yes	No	No	No	69.71	66.84
2	Yes	Yes	No	No	69.57	66.56
3	Yes	Yes	No	Yes	71.72	69.85
4	Yes	Yes	Yes	No	71.24	70.15
5	Yes	Yes	Yes	Yes	71.83	70.1
6	Yes	No	Yes	No	71.58	70.33
7	Yes	No	No	Yes	71.91	70.08
8	Yes	No	Yes	Yes	**72.14**	**70.43**

(9)).

To evaluate the performance of the proposed L2 sparse loss against the weight sparseness, dropout and center loss methods, the performances of these methods on the four databases are further compared in Table 7. Table 7 shows that ResNet with the proposed L2 feature sparse loss achieved much higher accuracy than the approach with the center loss (without using the proposed loss), i.e., improvements of 3.96% and 4.4% are achieved for the databases of Oulu-CASIA and MMI, respectively. At the same time, the proposed L2 feature sparse loss also outperforms the other methods, i.e., large improvements are achieved on FER2013 and MMI compared with W-L2, V-L2 weight sparseness, and on CK+ and Oulu-CASIA compared with the dropout method.

Table 7: The overall accuracy of ResNet with different settings on the four databases, where the softmax loss is used. Center and L2 denote center loss and the proposed L2 sparseness.

Method	FER2013	CK+	Oulu-CASIA	MMI
SoftMax only	69.71	94.7	77.29	72.69
W-L2 and Center	68.9	97.11	81.04	74.63
V-L2 and Center	69.55	96.63	81.46	75.61
Dropout and Center	71.24	95.9	80.83	77.07
Center	71.58	96.14	78.75	74.15
L2	71.91	97.35	**82.92**	77.56
L2 and Center	**72.14**	**97.59**	82.71	**78.54**

Tables 8, 9, 10, and 11 compare the performance and the other testing protocols of the proposed approach with the state-of-the-art approaches in the literature, for all of the four databases.

For the database FER2013, the work [53] achieved the highest recognition rate of 75.1%. However, it requires external data sources, i.e. social relation dataset for the bridging layer construction in the network modeling. As shown in Table 8, the proposed algorithm employs only one existing network for the fine-tuning, can easily be transferred to other databases. Though better accuracy was reported, the work [14] requires a face frontalization module and multiple deep networks were fused.

However, the proposed algorithm outperforms the work [14] when the same experimental setting, i.e., a single deep network, is employed.

For the CK+ database, the proposed algorithm ranked 2nd among the algorithms for seven-class FER. While the work [36] using randomly approximated high-dimension metric functions achieved a higher recognition rate, the proposed algorithm achieved a competitive performance of 97.59%, which is slightly lower than the best performer [36], i.e. 97.68%, which adopted recurrent hidden output sparseness and deep metric learning is not employed. Meanwhile, as shown in Table 9, the proposed algorithm employs only one network, which has smaller model complexity than the work [36] that employed 11 recurrent layers for performance improvement. The fine tuning network [39] integrated the expression images and their landmarks for joint learning, which achieved the competitive performance of 97.25%. However, landmark points might not be visible for faces present in the wild, due to pose and occlusion.

For the Oulu-CASIA database, compared with the study [5] that achieved the highest recognition rate, the proposed algorithm employed a slightly different dataset, i.e., $Strong - NIR$ for the training. However, the work [5] needs to train an additional network for pre-fine tuning based on a regression loss, which uses the convolution layer feature of face recognition network to fine-tune the feature generation of FER. The proposed algorithm achieves an improvement of 5.42% over the baseline in Table 7, which is comparable to the improvement, i.e. 4.45% achieved in the work [5]. The proposed algorithm achieved an accuracy of 82.71%, which was approximately 5% higher than that of the algorithm [54] where the same network, i.e. the spatial network without fusing the temporal expression images, was employed.

For the MMI database, the proposed algorithm achieved the best performance among the algorithms without temporal data for training. The proposed algorithm achieved better performance than the algorithm in [54] when only the spatial network with static images is employed. Compared with the study [40] that employed adaptive deep metric for identity-aware hard negative mining, the proposed algorithm does not use the adaptive strategies or regularization parameters on the loss function, and still achieves competitive performance.

Compared with the algorithms that achieved the best performances on the four databases in Tables 8, 9, 10, and 11, the proposed algorithm balances the performances on the four databases, i.e., it almost ranked in the 2nd position on all of the databases. At the same time, the proposed algorithm is tested on four popular databases, which is different from the best algorithms tested on one of the databases [36, 53] or two of the databases [5]. Moreover, compared with the algorithm [14] fusing aligned and non-aligned face information, the algorithm in [40] used adaptive deep metrics on the CK+ database, the algorithm in [39] fine-tuned the network with geometry and texture information on the Oulu-CASIA database, and the algorithm [54] used a multi-signal convolutional neural network on the MMI database. The proposed algorithm is easy to implement, and additional face alignments, temporal expression sequences or multiple network configurations are not needed.

Although the preceding comparison is not strictly fair since different fine-tuning strategies and network models are employed, these results still reveal the competitiveness of the proposed algorithm, compared with the state-of-the-art algorithms under the same experimental settings.

Table 8: Performance of different algorithms on the FER2013 databases. Symbol '#Net' denotes the number of networks.

Methods	Fine Tuning	#Net	Recog. rate (%)
Deeper DNN 2016 [55]	No fine tuning	1	66.4
DNN with SVM 2013 [56]	Fine tuning with SVM's objective	1	71.2
Fusing aligned faces 2016 [14]	Fusing multiple networks based on aligned and frontalized databases	4	73.73 (71.86 by single DCN)
Fusing multiple data sources 2015 [53]	Using external database	2	**75.1**
Ours	Public face recognition models	1	72.14

Table 9: Performance of different algorithms on the CK+ databases. Symbol '-' denotes Not Reported. Symbol '*' denotes that neural is replaced with contempt expression; '#C' denotes the number of expression classes; '10F' denotes 10-fold setting.

Methods	Data	Fine Tuning	#C	Sub.	Proto.	#Net	Recog. rate (%)
Adaptive deep metric 2017 [40]	Three peak	CMU Multi-pie	7^*	118	10F	1	97.1
Fine tuning 2015 [39]	Temporal	Geometry and texture losses	7^*	106	10F	2	97.25
Sparse autoencoders 2018 [35]	Four peak	-	8	123	10F	1	95.79
Margin projection 2013 [57]	The peak	-	7	100	5F	Non	89.2
Radial feature 2012 [58]	Five images	-	7	94	10F	Non	91.51
Dropout and randomized DMLs 2018 [36]	Five peak	-	7	118	10F	11 recurrent layers	**99.11** (97.68)
Ours	Three peak	FER2013 model	7	106	10F	1	97.59

Table 10: Performance of different algorithms on the Oulu-CASIA databases.

Methods	Data	Fine Tuning	#C	Sub.	Proto.	#Net	Recog. rate (%)
AdaLBP 2011 [48]	Temporal ($Strong - VIS$)	-	6	480	10F	Non	73.54
Fine tuning 2015 [39]	Temporal ($Strong$)	Geometry and texture losses	6	480	10F	2	81.46
Spatial and temporal networks 2017 [54]	Temporal ($Strong$)	-	6	480	10F	1	86.25 (77.67)
Face net regularization 2016 [5]	Three peak ($Strong - VIS$)	Face recognition net	6	480	10F	1	**87.71**
Ours	Three peak ($Strong - NIR$)	FER2013 model	6	480	10F	1	82.71

Table 11: Performance of different algorithms on the MMI databases.

Methods	Data	Fine Tuning	#C	Sequence	Proto.	#Net	Recog. rate (%)
Fine tuning 2015 [39]	Temporal	Geometry and texture losses	6	205	10F	2	70.24
Spatial and temporal networks 2017 [54]	Temporal	-	6	205	10F	1	**81.18** (77.05)
AU network 2013 [59]	Three peak	Logistic regression	7	205	10F	1	74.76
Deeper DNN 2016 [55]	-	No fine tuning	6	79	5F	1	77.6
Multiscale active learning 2015 [44]	Three peak	-	6	-	10F	Non	77.39
Adaptive deep metric 2017 [40]	Three peak	CMU Multi-pie	6	205	10F	1	**78.53**
Ours	Three peak	FER2013 model	6	205	10F	1	**78.53**

3.5. Generalization performance

The generalization performance of the proposed approach is now tested by cross-database training and testing. Table 12 shows the accuracy when one of the four databases is used for training and the remaining ones are used for testing. Because the images in database FER2013 are all collected in the wild, the accuracies of the approach trained using other databases are relatively low. The model trained using CK+ achieved only a 39.72% accuracy on FER2013. The highest accuracy (84.47%) is achieved for the CK+ dataset when the model was trained using the Oulu-CASIA dataset.

Tables 13 and 14 show the confusion matrixes of the approach for the CK+

Table 12: The generalization performance (%) of the proposed approach.

Training	Testing			
	FER2013	CK+	Oulu-CASIA	MMI
FER2013	-	63.86	39.79	57.07
CK+	39.72	-	42.08	60.48
Oulu-CASIA	54.19	**84.47**	-	**61.46**
MMI	**60.19**	77.02	**50.83**	-

Table 13: Confusion matrix (%) of the CK+ database for the proposed approach trained on the FER2013 database.

Exp.	An	Di	Fe	Ha	Sa	Su	Ne
An	**44.44**	0	20	0	13.33	0	22.22
Di	81.36	**11.86**	1.69	3.39	0	0	1.69
Fe	0	0	**24**	44	4	16	12
Ha	0	0	0	**100**	0	0	0
Sa	0	0	17.86	0	**60.71**	0	21.43
Su	6.02	0	7.23	21.69	3.61	**53.01**	8.43
Ne	0	0	0.94	2.83	0	0	**96.23**

Table 14: Confusion matrix (%) of the CK+ database for the proposed approach trained on the Oulu-CASIA database.

Exp.	An	Di	Fe	Ha	Sa	Su
An	**84.44**	4.44	0	0	8.89	2.22
Di	18.64	**79.66**	0	1.69	0	0
Fe	4	0	**56**	20	20	0
Ha	0	0	0	**100**	0	0
Sa	42.86	3.57	0	0	**50**	3.57
Su	0	2.41	1.20	1.20	0	**95.18**

database when FER2013 and Oulu-CASIA are used for training. Table 13 shows that the 'Di' expression is wrongly recognized as the 'An' expression with a probability of 81.36%, which suggests that the 'Di' expression between the database FER2013 collected in the wild and the database CK+ captured under controlled conditions differ substantially, as demonstrated in Figure 1. At the same time, for the 'Fe' expression, the 'mouths' in the FER2013 database are 'open', which are similar to the 'Ha' and 'Su' expressions. However, the proposed algorithm achieves reasonable recognition rates (average 77%) for the 'Ha', 'Sa', 'Su' and 'Ne' expressions. In contrast to FER2013, Table 14 shows that the features trained with the Oulu-CASIA database present more powerful generalization ability on the CK+ database, and the recognition rates of the 'Di' and 'Fe' expressions are largely improved. These promising accuracies for all of the six expressions show the good generalization performance of the learned features.

To evaluate the generalization performance against other sparseness methods, the performance of W-L2 sparseness, V-L2 sparseness and dropout based on center loss for each database is compared to that of the proposed algorithm in Table 15.

Table 15: The best generalization performance (%) of different sparseness methods.

Testing	Training	W-L2	V-L2	Dropout	Ours
FER2013	MMI	54.14	54.83	55.07	**60.19**
CK+	Oulu-CASIA	83.81	82.17	82.52	**84.47**
Oulu-CASIA	MMI	**51.04**	49.79	48.96	50.83
MMI	Oulu-CASIA	58.53	57.56	**61.46**	**61.46**

Table 15 shows that the proposed feature sparseness achieved the best performance on three of the four databases and comparable performance on the Oulu-CASIA database, which illustrates the competitive generalization ability of the proposed feature sparseness.

Table 16: Comparison of the cross-database performance (%). The database in the brackets denotes the training data.

Methods	FER2013	CK+	Oulu-CASIA	MMI
LBP 2009 [60]	-	-	-	51.1 (CK+)
Radial feature 2012 [58]	-	54.05 (JAFFE)	-	-
Deeper DNN 2016 [55]	34.0 (CK+ & MMI)	64.2 (MMI)	-	55.6 (CK+)
Ours	**39.72** (CK+) **60.19** (MMI)	**77.02** (MMI) **84.47** (Oulu-CASIA)	**50.83** (MMI)	**60.0** (CK+) **61.46** (Oulu-CASIA)

Table 16 compares the generalization performance of the proposed approach with three state-of-the-art methods in the literature. While no cross-database results were reported in the literature for Oulu-CASIA, the proposed approach achieved significantly better results on the FER2013 and CK+ datasets when MMI was used for training. The accuracies of the proposed approach are approximately 26% and 13% higher than those of the DNN proposed in [55] on FER2013 and CK+, respectively. The accuracy of hand-crafted LBP features is approximately 9% lower than the proposed approach on the MMI dataset when CK+ is used for training.

4. Discussion and conclusions

In this work, feature sparseness is embedded into deep feature learning to boost the generalization ability of the convolutional neural network for FER. Different from the weight sparseness and hidden unit sparseness, which introduce a large number of parameters in an additional optimization, the proposed algorithm imposes the sparseness directly on the FC layers to select common features among the different persons and expression databases. Compared with the-state-of-the-art algorithms on the public expression databases, i.e., FER2013, CK+, Oulu-CASIA and MMI, the proposed algorithm not only improved the person-independent recognition performance for the same database, but also boosted the generalization ability for cross-database recognition.

Although competitive results are achieved by the proposed algorithm, there remains room for further improvement. First, because the deformation intensities of 'An', 'Di', 'Sa' are significantly smaller than those of the 'Ha' and 'Su' expressions, the generalization performance between databases could be further improved by considering multiscale features and multiple network fusion. Second, two-layer sparseness can introduce three regularization parameters for the three loss terms, and a self-adaptive model should be devised to set the best parameter values. Third, the database was collected under controlled conditions, i.e., MMI can be augmented with expression synthesis and used for the fine tuning of the FER2013 database. Fourth, since different sparseness strategies and regularization coefficients might be appropriate for different databases, multiple sparseness losses can be dynamically adjusted to customize the sparseness strategy and regularization parameter setting for a specific database. Fifth, more insight should be placed on the theoretical analysis of the weight sparseness and the feature sparseness for FER. Finally, the feature sparseness will be expanded in other applications such as handwritten character recognition.

Acknowledgments

The authors thank the anonymous reviewers for their helpful comments and constructive suggestions, Dr. Feng Wang in University of Electronic Science and Technology for the public codes of deep feature regularization, and Prof. Wenming Zheng in Southeast University and Prof. Michel Valstar in University of Nottingham for their helpful comments and suggestions. The work was supported by Natural Science Foundation of China under grands no. 61602315 and 61672357, the Science and Technology Innovation Commission of Shenzhen under grant no. JCYJ20170302153827712, the Tencent "Rhinoceros Birds"-Scientific Research Foundation for Young Teachers of Shenzhen University, the School Startup Fund of Shenzhen University under grands no. 2018063.

References

[1] Z. Zeng, M. Pantic, G. I. Roisman, T. S. Huang, A Survey of affect recognition methods: Audio, visual, and spontaneous expressions, IEEE Trans. Pattern Anal. Mach. Intell. 31 (1) (2009) 39-58.

[2] E. Sariyanidi, H. Gunes, A. Cavallaro, Learning bases of activity for facial expression recognition, IEEE Trans. Image Process. 26 (4) (2017) 1965-1978.

[3] P. Liu, J. T. Zhou, W. H. Tsang, Z. Meng, S. Han, Y. Tong, Feature disentangling machine-a novel approach of feature selection and disentangling in facial expression analysis, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 84-90.

[4] H. Yang, U. Ciftci, L. Yin, Facial expression recognition by de-expression residue learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2168-2177.

[5] H. Ding, S. K. Zhou, R. Chellappa, FaceNet2ExpNet: Regularizing a deep face recognition net for expression recognition, in: Proceedings of the

IEEE International Conference on Automatic Face Gesture Recognition, 2016, pp. 118-126.

[6] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 507-516.

[7] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 499-515.

[8] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, SphereFace: deep hypersphere embedding for face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 212-220.

[9] H. O. Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4004-4012.

[10] X. Liu, B. V. K. V. Kumar, P. Jia, J. You, Hard negative generation for identity-disentangled facial expression recognition, Pattern Recog. 88 (2019) 1-12.

[11] M. Tonomura, K. Nakayama, A hybrid learning algorithm for multilayer perceptrons to improve generalization under sparse training data conditions, in: Proceedings of the International Joint Conference on Neural Networks, 2001, pp. 967-972.

[12] F. Wang, X. Xiang, J. Cheng, A. L. Yuille, NormFace: L2 hypersphere embedding for face verification, in: Proceedings of the ACM International Conference on Multimedia, 2017, pp. 1041-1049.

[13] R. Ranjan, C. D Castillo, R. Chellappa, L2-constrained softmax loss for discriminative face verification, arXiv:1703.09507 (2017).

[14] B. K. Kim, S. Y. Dong, J. Roh, G. Kim, S. Y. Lee, Fusing aligned and non-aligned face information for automatic affect recognition in the wild: A deep learning approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2016, pp. 1499-1508.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929-1958.

[16] L. Wan, M. D. Zeiler, S. Zhang, Y. Lecun, R. Fergus, Regularization of neural networks using DropConnect, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 1058-1066.

[17] Y. Liu, X. Yuan, X. Gong, Z. Xie, F. Fang, Z. Luo, Conditional convolution neural network enhanced random forest for facial expression recognition, Pattern Recog. 84 (2018) 251-261.

[18] S. Scardapane, D. Comminiello, A. Hussain, A. Uncini, Group sparse regularization for deep neural networks, Neurocomputing 241 (2017) 81-89.

[19] P. Murugan, S. Durairaj, Regularization and optimization strategies in deep convolutional neural network, arXiv:1712.04711v1 (2017).

[20] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798-1828.

[21] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210-227.

[22] L. Zhang, M. Yang, Sparse representation or collaborative representation: Which helps face recognition?, in: Proceedings of the International Conference on Computer Vision, 2011, pp. 471-478.

[23] I. Buciu, I. Pitas, A new sparse image representation algorithm applied to facial expression recognition, in: Proceedings of the Machine Learning for Signal Processing Workshop, 2004, pp. 539-548.

[24] Y. Guo, G. Zhao, M. Pietikainen, Dynamic facial expression recognition with atlas construction and sparse representation, IEEE Trans. Image Process. 25 (5) (2016) 1977-1992.

[25] J. Mutch, D. G. Lowe, Multiclass object recognition with sparse, localized features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 11-18.

[26] D. Yu, F. Seide, G. Li, L. Deng, Exploiting sparseness in deep neural networks for large vocabulary speech recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2012, pp. 4409-4412.

[27] S. Zhang, J. Wang, X. Tao, Y. Gong, N. Zheng, Constructing deep sparse coding network for image classification, Pattern Recog. 64 (C) (2017) 130-140.

[28] M. A. Ranzato, Y. L. Boureau, Y. Lecun, Sparse feature learning for deep belief networks, in: Proceedings of the International Conference on Neural Information Processing Systems, 2007, pp. 1185-1192.

[29] J. Yan, W. Zheng, Q. Xu, G. Lu, H. Li, B. Wang, Sparse kernel reduced-rank regression for bimodal emotion recognition from facial expression and speech, IEEE Trans. Multimedia 18 (7) (2016) 1319-1329.

[30] B. Liu, M. Wang, H. Foroosh, M. Tappen, M. Penksy, Sparse convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 806-814.

[31] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[32] Z. Zheng, Y. Fan, J. Lv, High dimensional thresholded regression and shrinkage effect, J. R. Stat. Soc. Series B Stat. Methodol. 76 (3) (2014) 627-649.

[33] N. N. Ji, J. S. Zhang, C. X. Zhang, A sparse-response deep belief network based on rate distortion theory, Pattern Recog. 47 (9) (2014) 3179-3191.

[34] W. Y. Zou, A. Y. Ng, K. Yu, Unsupervised learning of visual invariance with temporal coherence, in: Proceedings of the International Conference on Neural Information Processing Systems Workshop, 2011.

[35] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, A. M. Dobaie, Facial expression recognition via learning deep sparse autoencoders, Neurocomputing 273 (C) (2018) 643-649.

[36] M. Alam, L. S. Vidyaratne, K. M. Iftekharuddin, Sparse simultaneous recurrent deep learning for robust facial expression recognition, IEEE Trans. Neural. Netw. Learn. Syst. 29 (10) (2018) 4905-4916.

[37] J. Li, H. Chang, J. Yang, Sparse deep stacking network for image classification, in: Proceedings of the Association for the Advancement of Artificial Intelligence, 2015, pp. 3804-3810.

[38] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, R. Chellappa, An all-in-one convolutional neural network for face analysis, in: Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition, 2017, pp. 17-24.

[39] H. Jung, S. Lee, J. Yim, S. Park, J. Kim, Joint fine-tuning in deep neural networks for facial expression recognition, in: Proceedings of the International Conference on Computer Vision, 2015, pp. 2983-2991.

[40] X. Liu, B. V. K. V. Kumar, J. You, P. Jia, Adaptive deep metric learning for identity-aware facial expression recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2017, pp. 522-531.

[41] G. Zeng, J. Zhou, X. Jia, W. Xie, L. Shen, Hand-crafted feature guided deep learning for facial expression recognition, in: Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition, 2018, pp. 423-430.

[42] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448-456.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1-9.

[44] L. Zhong, Q. Liu, P. Yang, J. Huang, D. N. Metaxas, Learning multiscale active facial patches for expression analysis, IEEE Trans. Cybern. 45 (8) (2015) 1499-1510.

[45] M. Pantic, M. Valstar, R. Rademaker, L. Maat, Web-based database for facial expression analysis, in: Proceedings of the IEEE International Conference on Multimedia & Expo (ICME), 2005, pp. 317-321.

[46] T. Kanade, J. F. Cohn, Y. Tian, Comprehensive database for facial expression analysis, in: Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition, 2002, pp. 46-53.

[47] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D. H. Lee, Challenges in representation learning: A report on three machine learning contests, in: Proceedings of the International Conference on Neural Information Processing, 2013, pp. 117-124.

[48] G. Zhao, X. Huang, M. Taini, S. Z. Li, Facial expression recognition from near-infrared videos, Image Vis. Comput. 29 (9) (2011) 607-619.

[49] S. Boyd, Convex optimization, IEEE Trans. Automat. Contr. 51 (11) (2006) 1859.

[50] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations, 2015.

[51] L. Shen, X. Jia, Y. Li, Deep cross residual network for HEp-2 cell staining pattern classification, Pattern Recog. 82 (2018) 68-78.

[52] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2005) 861-874.

[53] Z. Zhang, P. Luo, C. C. Loy, X. Tang, Learning social relation traits from face images, in: Proceedings of the International Conference on Computer Vision, 2015, pp. 3631-3639.

[54] K. Zhang, Y. Huang, Y. Du, L. Wang, Facial expression recognition based on deep evolutional spatial-temporal networks, IEEE Trans. Image Process. 26 (9) (2017) 4193-4203.

[55] A. Mollahosseini, D. Chan, M. H. Mahoor, Going deeper in facial expression recognition using deep neural networks, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2016, pp. 1-10.

[56] Y. Tang, Deep learning using linear support vector machines, in: Proceedings of the International Conference on Machine Learning Workshop, 2013.

[57] S. Nikitidis, A. Tefas, I. Pitas, Maximum margin projection subspace learning for visual data analysis, IEEE Trans. Image Process. 23 (10) (2014) 4413-4425.

[58] W. Gu, C. Xiang, Y. V. Venkatesh, D. Huang, H. Lin, Facial expression recognition using radial encoding of local Gabor features and classifier synthesis, Pattern Recog. 45 (1) (2012) 80-91.

[59] M. Liu, S. Li, S. Shan, X. Chen, AU-aware deep networks for facial expression recognition, in: Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition, 2013, pp. 1-6.

[60] C. Shan, S. Gong, P. W. Mcowan, Facial expression recognition based on local binary patterns: A comprehensive study, Image Vis. Comput. 27 (6) (2009) 803-816.