Adaptive Weighting of Handcrafted Feature Losses for Facial Expression Recognition

Weicheng Xie, Linlin Shen, and Jinming Duan

***Abstract*—Due to the importance of facial expressions in human–machine interaction, a number of handcrafted features and deep neural networks have been developed for facial expression recognition. While a few studies have shown the similarity between the handcrafted features and the features learned by deep network, a new feature loss is proposed to use feature bias constraint of handcrafted and deep features to guide the deep feature learning during the early training of network. The feature maps learned with and without the proposed feature loss for a toy network suggest that our approach can fully explore the complementarity between handcrafted features and deep features. Based on the feature loss, a general framework for embedding the traditional feature information into deep network training was developed and tested using the FER2013, CK+, Oulu-CASIA, and MMI datasets. Moreover, adaptive loss weighting strategies are proposed to balance the influence of different losses for different expression databases. The experimental results show that the proposed feature loss with adaptive weighting achieves much better accuracy than the original handcrafted feature and the network trained without using our feature loss. Meanwhile, the feature loss with adaptive weighting can provide complementary information to compensate for the deficiency of a single feature.**

***Index Terms*—Deep feature loss, expression recognition, handcrafted feature, loss adaptive weighting.**

Manuscript received March 17, 2019; revised May 18, 2019 and June 15, 2019; accepted June 18, 2019. Date of publication August 2, 2019; date of current version April 15, 2021. This work was supported in part by the Natural Science Foundation of China under Grant 61602315, Grant 61672357, and Grant U1713214, in part by the Science and Technology Project of Guangdong Province under Grant 2018A050501014, in part by the Science and Technology Innovation Commission of Shenzhen under Grant JCYJ20170302153827712, in part by the China Post-Doctoral Science Foundation under Grant 2019T120751, in part by the Tencent "Rhinoceros Birds"-Scientific Research Foundation for Young Teachers of Shenzhen University, and in part by the School Startup Fund of Shenzhen University under Grant 2018063. This paper was recommended by Associate Editor J. Su. *(Corresponding author: Linlin Shen.)*

W. Xie is with the Computer Vision Institute, School of Computer Science and Software Engineering, Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, China (e-mail: wcxie@szu.edu.cn).

L. Shen is with the Computer Vision Institute, School of Computer Science and Software Engineering, Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518060, China (e-mail: llshen@szu.edu.cn).

J. Duan is with the School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: j.duan@bham.ac.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2019.2925095.

Digital Object Identifier 10.1109/TCYB.2019.2925095

I. Introduction

DIFFERENT from the nondeep network that employs handcrafted features like Gabor, local binary pattern (LBP), and scale-invariant feature transform (SIFT) [1], [2] to extract features for recognition, deep neural network (DNN) uses end-to-end models to automatically learn feature representation with multiple combinations of nonlinear filters. As the optimization space of DNN can be very large due to the large number of network parameters, local searching or warm start strategies have been proposed to speed up the searching efficiency.

Zhong *et al.* [3] proposed to learn active expression sensitive patches with multitask learning for facial expression recognition (FER). Ionescu *et al.* [1] proposed a local feature learning approach to use neighbors around each testing sample for FER. Jung *et al.* [4] integrated the geometry features into texture feature network and fine-tuned the last two layers of a network with a fine-tuning network for FER. Sikka *et al.* [2] proposed to extract multiscale dense SIFT features with locality-constrained linear coding and max-pooling strategies for FER. Zadeh *et al.* [5] incorporated a set of appearance prototypes of different poses and expressions, based on a local model of a convolutional expert constraint. Pan and Jiang [6] added an orthogonal projection constraint on the loss function to reduce the model size and redundancy of convoluted features, by replacing the pooling layer with an orthogonal projection layer. Wang *et al.* [7] introduced the similarity between the deep features of the visible images and the augmented thermal images for deep network training. Although a fine-tuning strategy or a constraint was applied to the deep network, these algorithms rarely use the merits of handcrafted features.

Handcrafted features have been fused into deep features to improve the performance of DNN. Liu *et al.* [8] embedded the handcrafted HOG and dense SIFT features into deep CNN for FER. Paul *et al.* [9] fused handcrafted quantitative features into the deep features of the top-five layers for survival prediction. Suggu *et al.* [10] concatenated several handcrafted features with the output of the last convolution layer, while Majtner *et al.* [11] fused the discrimination probabilities of handcrafted and deep features before SVM.

In these algorithms, the handcrafted features are fused directly with deep features to explore the advantages of both features, while their entangled information is not employed in deep network [12]. Actually, Khorrami *et al.* [13] showed that the features of a deep CNN are analogous to the appearance of basic facial expression action units, and Zeiler and

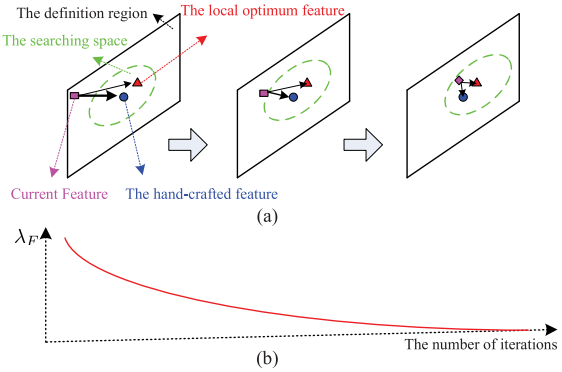

Fig. 1. Evolution of the searching space as the algorithm proceeds. (a) Line thickness denotes the intensity of the influence of the corresponding guidance loss. (b) Parameter λ_F denotes the weight coefficient of the proposed feature loss.

Fergus [14] revealed that the feature maps in the shallow layers of CNN are similar to the handcrafted feature, that is, Gabor feature [15]. Juefei-Xu *et al.* [16] explored the similarity between the convolution layer and LBP-like operator to reduce the model complexity without largely decreasing the performance. Chen *et al.* [17] used the feature generation of the joint Bayesian model to guide and fine-tune the deep network training. These studies implied that the handcrafted and deep features were similar and complementary. To embed constraint information into deep network to improve the performance, a few loss functions, such as the Center loss [18], SphereFace loss [19], and adaptive deep metric learning (DML) [20], have been proposed based on DML. Thus, we propose a DML-based algorithm to perform local searching around the handcrafted feature to improve the discrimination ability of deep network.

In recent FER algorithms, both handcrafted features, such as maximum margin projection [21] and radial feature [22], and deep features like AU deep network [23], short-term memory networks [24], DNN [25], and spatial–temporal recurrent DNN [26], have been proposed. However, works that fused handcrafted feature information into deep network for FER were rarely found.

A. Motivation

The motivation of the feature loss embedding the handcrafted feature information is presented in Fig. 1.

Considering that handcrafted features can provide complementary information for deep features, a feature loss is proposed to use the complementary information and decrease the optimization space in the early network training, that is, the information of handcrafted features is embedded for local searching, as shown in Fig. 1(a). Thus, the feature loss encourages a deep network to find the local optimum around the handcrafted feature. While a handcrafted feature can guide the network training during the early training, it often performs worse than the deep feature for the final recognition. Thus, a weight decaying strategy is employed to gradually decrease the influence of the feature loss as the algorithm proceeds, as shown in Fig. 1(b).

Zeng *et al.* [27] has preliminarily studied the feature loss for performance improvement, while it is restricted to one handcrafted feature on a single fully connected (FC) layer. Meanwhile, the performance for FER is strictly limited since each feature loss maybe suitable to only some specific databases. Self-adaptive loss weighting or selection of the handcrafted features can be devised to balance the influence of different losses on the considered database.

For multiple-loss weighting with regularization, Xu *et al.* [28] proposed to pretrain the model with the single-core-loss function and then warm start the whole multiloss DNN with the convolutional parameters transferred from the pretrained model. Liu *et al.* [20] proposed the adaptive DML among multiple losses for FER. Chen *et al.* [29] proposed a self-adaptive algorithm for loss weight regularization in multitask learning by adjusting the gradient norms during training.

In this paper, a self-adaptive weighting based on the newly proposed alternative loss is proposed to select the suitable guidance information of a feature loss for a specific database. Meanwhile, different loss weighting strategies are evaluated and compared on four public databases.

B. Contribution

In this paper, a new deep network based on handcrafted feature guidance is proposed for FER, that is, the handcrafted feature information is embedded into the ResNet network by imposing a new loss metric into the final loss function. The handcrafted feature embedded in the proposed loss is used to guide the network learning, reduce the searching space during the early training, and add complementary prior knowledge into the learning of DNN features. Moreover, loss weighting strategies based on branched and alternative feature losses are introduced to balance the influences of different feature losses for a specific database, which are significantly different from related state-of-the-art works that directly fuse multiple deep losses without dynamic weight adjustment.

This paper makes the following contributions.

1) A novel loss metric, namely, feature loss, is proposed to use the complementary information of handcrafted and deep features during early network training.
2) Different loss weighting strategies are introduced and compared on four public databases, where the alternative loss weighting that differs from the traditional loss fusion is newly proposed.
3) While a toy model was designed to elaborate and justify the motivation of the idea, the proposed algorithm has also been evaluated with the handcrafted approaches, the deep network without the proposed loss and state-of-the-art. The results on four public expression databases show that the performance of the proposed algorithm is very competitive.

This paper is structured as follows. Section II gives a description about the proposed feature loss and self-adaptive weighting of alternative loss step by step. The experimental results of the proposed algorithm on public databases

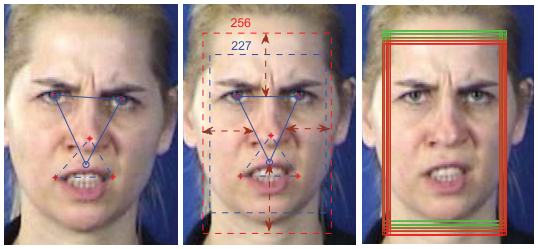

Fig. 2. Data preprocessing. (a) Five landmark points. (b) Transformed key points for image alignment, cropping, and scaling. Double arrows label the margins beyond the triangle. (c) Image regions with different colors are cropped.

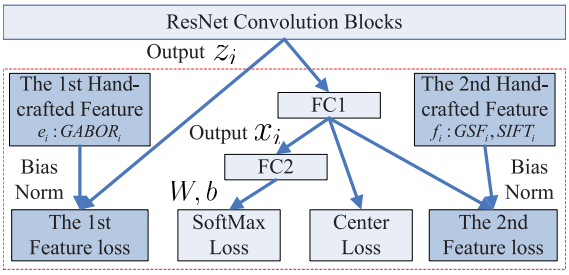

Fig. 3. Proposed network based on SoftMax, Center, and feature losses. FC1 denotes the first FC layer; and e, f, *GABOR*, *GSF*, and *SIFT* are the employed handcrafted features.

are presented in Section III. Finally, the discussions and conclusions are addressed in Section IV.

II. Proposed Algorithm

In this section, the proposed feature loss, the loss adaptive weighting strategies, and the used deep network structure and its training are introduced.

A. Preprocessing

For the face alignment, the five key points are first located on the eyes, nose, and mouth tips [30]. Then, the faces are aligned with the three circle points presented in Fig. 2(a), that is, the landmarks of two eyes and the barycenter of the lower three landmark points. The face region is then rotated and scaled according to the side lengths of the blue solid triangle of a standard template. Then, a region of 256×256 is cropped out based on the four distances (i.e., margins) with reference to the triangle shown in Fig. 2(b), where a central face region of size 227×227 is extracted. For data augmentation, four more regions with size 227×227 located in upper left, upper right, low left, and low right of the central face region, are cropped out in Fig. 2(c). The locations of upper left, upper right, low left, and low right bounding boxes are shifted about 15 pixels with reference to the location of center box. The cropped regions, together with the mirrored versions, become the augmented training data. Finally, each expression image I is normalized in gray level and mirrored for the training of DNN.

B. Feature Loss

In the proposed network, in addition to the SoftMax loss and the Center loss [18], a new metric, namely, feature loss, is proposed to measure the guidance force of handcrafted feature, which is presented in Fig. 3.

For comparison, the employed SoftMax, Center, and the proposed feature losses are formulated as follows:

$$\begin{cases} \mathcal{L}_S = -\sum_i \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_j e^{W_j^T x_i + b_j}} \\ \mathcal{L}_C = \frac{1}{2}\sum_i \|x_i - c_{y_i}\|_2^2 \\ \mathcal{L}_{F1}(z) = \frac{1}{p}\sum_i \mathcal{N}_p^p(z_i - \|z_i\|_2 \cdot h_i) \\ \mathcal{L}_{F2}(x) = \frac{1}{p}\sum_i \mathcal{N}_p^p(x_i - \|x_i\|_2 \cdot g_i) \end{cases} \tag{1}$$

where z_i and x_i denote the outputs of convolution blocks and FC1, as shown in Fig. 3; $\mathcal{N}_p(z_i - \|z_i\|_2 \cdot h_i)$ denotes the L_p-norm of $z_i - \|z_i\|_2 \cdot h_i$, while L_2-norm ($p = 2$) and L_1-norm ($p = 1$) are considered in this paper; $\mathcal{L}_S$, $\mathcal{L}_C$, $\mathcal{L}_{F1}(z)$, and $\mathcal{L}_{F2}(x)$ are the SoftMax, Center, and feature losses corresponding to the inputs of two FC layers, respectively; W and b are the weight matrix and the bias term of the layer FC2; y_i is the expression label of the ith sample; c_{y_i} is the center vector of FC2 inputs with respect to (w.r.t.) the y_ith class; h_i and g_i are the normalization of two handcrafted features of the ith sample, that is, e_i and f_i, which are formulated as follows:

$$\begin{cases} h_i = e_i / \|e_i\|_2 \\ g_i = f_i / \|f_i\|_2. \end{cases} \tag{2}$$

For the presentation convenience, the L_p-norm in (1) is simplified to L_2-norm in the following sections.

Since the handcrafted feature, that is, the Gabor feature, has been widely used in FER and achieved good performance, the Gabor surface feature (GSF) [31], [32] before and after projection is employed in this paper. More precisely, the feature of the ith expression sample is first represented as

$$fo_i = \left(pf_{p_1}, \ldots, pf_{p_n}\right) \tag{3}$$

where $n \times n$ is the number of face patches obtained by uniformly dividing each face into blocks with size 80×80, pf_{p_j} is the GSF representation of the jth patch. With the patch feature pf_{p_k}, handcrafted features e_i and f_i of the ith sample are formulated as follows:

$$\begin{cases} \text{GABOR}_i \leftarrow e_i = \text{Hist}(fo_i) \\ \text{GSF}_i \leftarrow f_i = P_{\text{LDA}} P_{\text{PCA}} e_i \end{cases} \tag{4}$$

where Hist denotes the histogram operator, P_{PCA} and P_{LDA} are the projection matrices of principal components analysis (PCA) and linear discriminant analysis (LDA), respectively. In order to match the dimension of handcrafted feature h_i with that of FC1 input z_i to construct $\mathcal{L}_{F1}(z)$, the histogram feature before projection, that is, e_i in (4) is specially designed. First, a number of 8×4 Gabor maps are generated for each patch [32]. Each Gabor map is then transformed to a 16-D feature with the histogram operator. Finally, a feature with the dimension of $8 \times 4 \times 16 \times n \times n$ (n is reset to 5) is extracted for the entire face to form e_i in (2). This feature is abbreviated as *GABOR* in the following presentation. To construct the handcrafted feature f_i for the loss $\mathcal{L}_{F2}(x)$, the histogram e_i is further reduced to $\#class - 1$ dimensions using PCA and LDA [33]. Thus, the

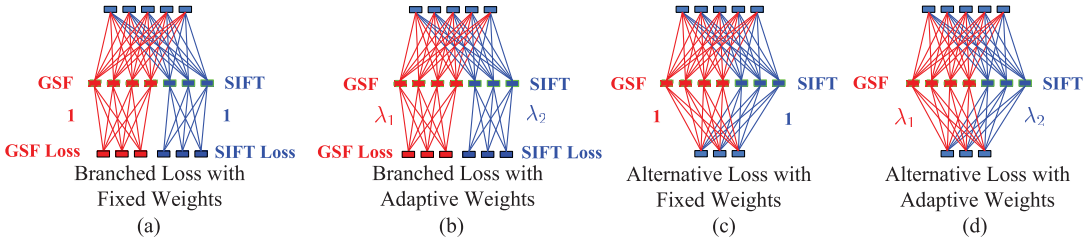

Fig. 4. Structures of [(a) and (b)] branched and [(c) and (d)] alternative losses with fixed and self-adaptive weights.

dimension of handcrafted feature g_i in (2) is $(\#class-1)\times n\times n$ (n is reset to 8), where $\#class$ is the number of expression categories.

Another handcrafted feature, that is, dense SIFT [1], [2] is also used as feature f_i in (2) for testing. Based on the divided $n \times n$ patches of a face (n is reset to 7), a feature with a dimension of $128 \times n \times n$ is first constructed [1], [2], and further projected to $(\#class - 1) \times n \times n$-dim with PCA and LDA presented in (4).

With the proposed feature loss, the final optimization of the proposed network is formulated as follows:

$$\min_{\mathcal{N}} \mathcal{L} = \mathcal{L}_S + \lambda_C \mathcal{L}_C + \lambda_{F1}\mathcal{L}_{F1}(z) + \lambda_{F2}\mathcal{L}_{F2}(x) \tag{5}$$

where $\mathcal{N}$ denotes the set of network parameters; λ_C, λ_{F1}, and λ_{F2} are the regularization parameters; and λ_{F1} and λ_{F2} are decreased with the same decaying factor ρ as follows:

$$\lambda_{F1}^{(t+1)} = \rho\lambda_{F1}^{(t)} \tag{6}$$

where t is number of iteration epochs.

The minimization of the loss function in (5) is formulated as a fitting form, while the gradients of $\mathcal{L}$ w.r.t. the variables z_i and x_i are calculated as follows:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial z_i} = \lambda_{F1}\frac{\partial \mathcal{L}_{F1}}{\partial z_i} + \left[\frac{\partial \mathcal{L}_S}{\partial z_i} + \lambda_C \frac{\partial \mathcal{L}_C}{\partial z_i} + \lambda_{F2}\frac{\partial \mathcal{L}_{F2}}{\partial z_i}\right] \\ \frac{\partial \mathcal{L}}{\partial x_i} = \lambda_C \frac{\partial \mathcal{L}_C}{\partial x_i} + \lambda_{F2}\frac{\partial \mathcal{L}_{F2}}{\partial x_i} + \left[\frac{\partial \mathcal{L}_S}{\partial x_i}\right] \\ \frac{\partial \mathcal{L}_C}{\partial x_i} = x_i - c_{y_i} \\ \frac{\partial \mathcal{L}_{F1}}{\partial z_i} = \begin{cases} z_i - \|z_i\|_2 \cdot h_i & p = 2 \\ \left(, \ldots, \text{SIGN}\left(z_{i,j} - \|z_i\|_2 \cdot h_{i,j}\right), \ldots,\right) & p = 1 \end{cases} \end{cases} \tag{7}$$

where SIGN$(\cdot)$ is the sign function. While cross-entropy function is employed for $\mathcal{L}_S$, the partial derivatives in $[\,\cdot\,]$, such as $(\partial\mathcal{L}_S/\partial x_i)$ in $(\partial\mathcal{L}/\partial x_i)$, are automatically obtained by the network backward propagation. The gradient $(\partial\mathcal{L}_{F2}/\partial x_i)$ is similarly induced as $(\partial\mathcal{L}_{F1}/\partial z_i)$.

With the obtained gradients in (7), the network parameters $\mathcal{N}$ are iteratively updated with optimization of stochastic gradient descent (SGD).

C. Adaptive Loss Weighting

As illustrated in Fig. 1, a handcrafted feature can guide the deep feature toward a local optimum in a local region around the handcrafted feature. However, a handcrafted feature suitable for a specific database may not work for other databases. Meanwhile, the loss fused with two handcrafted features can better guide the deep feature learning when the local optimum feature lies between them.

Different from the traditional loss fusion, four different settings with two loss structures and weighting strategies (three of them are newly proposed in this paper) are introduced in Fig. 4 and illustrated as follows:

1) branched loss structure [28] with fixed loss weights;
2) branched loss structure with self-adaptive loss weights;
3) alternative loss structure with fixed loss weights;
4) alternative loss structure with self-adaptive loss weights.

Different from the two-layer feature loss in (1), two feature constraints of GSF and SIFT are both imposed on the input of FC2, which are formulated as follows:

$$\begin{cases} \mathcal{L}_g = \frac{1}{2}\sum_i \|xs_i - \|xs_i\|_2 \cdot g_i\|_2 \\ \mathcal{L}_s = \frac{1}{2}\sum_i \|xg_i - \|xg_i\|_2 \cdot s_i\|_2 \end{cases} \tag{8}$$

where s_i and g_i denote the normalized handcrafted features of GSF and SIFT, and xs_i and xg_i are the corresponding components in x_i, that is, $x_i = (xs_i, xg_i)$.

To avoid multiple restart training from scratch to obtain the most suitable handcrafted feature loss, the online loss weighting is proposed by transforming the minimization (5) to an optimization problem as follows:

$$\begin{cases} \min_{\mathcal{N},\lambda_1\in[0,2],\lambda_2\in[0,2]} \mathcal{L} = \mathcal{L}_S + \lambda_1\lambda_b\mathcal{L}_s + \lambda_2\lambda_b\mathcal{L}_g \\ \text{s.t. } \lambda_1 + \lambda_2 = 2 \end{cases} \tag{9}$$

where λ_b is the base regularization weight, different settings of parameters λ_1 and λ_2 correspond to different strategies of feature losses.

To solve the optimization (9), a heuristic strategy is employed to modify the loss weights after each batch of epochs, for example, the loss weights are updated after every 1e3 epochs. More precisely, once a feature loss performs better than the other, the corresponding loss weights are amplified and normalized as follows:

$$\begin{cases} (\lambda_1, \lambda_2) \leftarrow \begin{cases} (\gamma\cdot\lambda_1, \lambda_2) & \text{if } \text{obj}_1 > \text{obj}_2 + \text{eps} \\ (\lambda_1, \gamma\cdot\lambda_2) & \text{if } \text{obj}_1 + \text{eps} < \text{obj}_2 \\ (\lambda_1, \lambda_2) & \text{otherwise.} \end{cases} \\ (\lambda_1, \lambda_2) \leftarrow \left(\frac{2\lambda_1}{\lambda_1+\lambda_2}, \frac{2\lambda_2}{\lambda_1+\lambda_2}\right) \end{cases} \tag{10}$$

where the amplification factor γ is 1.2 for FER2013 database and 1.05 for the other three databases; eps $= 1e-8$; and obj_1 and obj_2 denote the objective function values for loss weight update. Generalization ability is a main concern to transfer the trained model from the validation dataset to the testing dataset across different circumstances, which could be enhanced with algorithms of sparse representation [34], domain transfer [35], [36], and cross-database transfer [37]. In this paper, a relatively easy-to-implement sparse representation, that is,

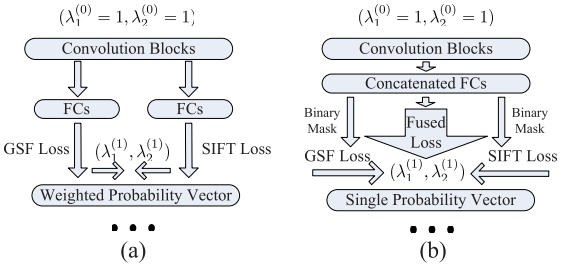

Fig. 5. Loss weighting processes for (a) branched loss and (b) alternative loss.

a L_1-sparseness term motivated from [34] is integrated with the accuracy of the validation dataset to produce the objective function as follows:

$$\begin{cases} \text{obj}_1 = \text{acc}_1 - \frac{\kappa}{\sqrt{\#xg}}\|xg\|_1 \\ \text{obj}_2 = \text{acc}_2 - \frac{\kappa}{\sqrt{\#xs}}\|xs\|_1 \end{cases} \tag{11}$$

where acc_1 is the accuracy of the validation dataset w.r.t. GSF feature loss, $\#xg$ denotes the dimension of the L_2-normalized feature xg defined in (8), and κ is the regularization coefficient.

Two loss weighting structures, that is, the branched loss and the alternative loss, as presented in Fig. 4, are employed for the evaluation. For the branched loss, each branch contains two FC layers, that is, FC1 and FC2 in Fig. 3, which corresponds to SoftMax, Center, and feature loss, and outputs an accuracy value, as illustrated in Fig. 4(a) and (b). For the alternative loss, multiple handcrafted feature losses are constructed in (8), as shown in Fig. 4(c) and (d). The entire structure contains only one branch with the extended FC2 layer, and outputs an accuracy corresponding to the fused feature loss.

Based on the network structures presented in Fig. 4, the returned two validation accuracies and feature sparseness are then used to update the loss weights with (10). The weighting processes of the branched and alternative losses are also presented in Fig. 5. As shown in Fig. 5, the returned accuracy of each branch for the validation dataset is directly used as the validation accuracy for the branched loss. While for the alternative loss, the nonrelated FC components and linking weights should be set to 0 with a binary mask [Fig. 4(c) and (d)], then the returned accuracy on the validation dataset is used as the validation accuracy of the considered feature.

For clarity, the proposed training algorithm with loss weighting is illustrated in Algorithm 1.

Algorithm 1 Proposed Network Training Algorithm With Loss Weighting

Function: Restart training
$\mathcal{M}^{(t+1)}$=*RestartTrain* $(\{\mathcal{M}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}\})$:
for $s = 0, \cdots, MaxIter$ **do**
  Perform network forward to obtain the entire loss function $\mathcal{L}$ in equation (9).
  Perform network backward to obtain $\frac{\partial \mathcal{L}}{\partial z_i}$ and $\frac{\partial \mathcal{L}}{\partial x_i}$ similar to equation (7).
  Perform SGD to update z_i, x_i and output $\mathcal{N}^{(t+1)}$.
end for

Obtain GSF and SIFT features $\{s_i, g_i\}$ of all the samples.
Set the parameters λ_C, λ_F and $MaxIter$.
Initialize the model $\mathcal{M}^{(0)}$;
for $t = 0, \cdots, T$ **do**
  if Weighting with Branched Loss: then
    Perform $\mathcal{M}^{(t+1)}$=*RestartTrain* $(\{\mathcal{M}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}\})$ and obtain $obj_1^{(t)}$, $obj_2^{(t)}$ for validation dataset;
    Update the loss weights $\{\lambda_1^{(t+1)}, \lambda_2^{(t+1)}\}$ with equation (10).
  else if Weighting with Alternative Loss: then
    Perform *RestartTrain* $(\{\mathcal{M}^{(t)}, 1, 0\})$ to obtain $obj_1^{(t)}$;
    Perform *RestartTrain* $(\{\mathcal{M}^{(t)}, 0, 1\})$ to obtain $obj_2^{(t)}$;
    Update the loss weights with equation (10).
    Perform
      $\mathcal{M}^{(t+1)}$=*RestartTrain* $(\{\mathcal{M}^{(t)}, \lambda_1^{(t+1)}, \lambda_2^{(t+1)}\})$;
  end if
end for
Output model and weights $\{\mathcal{M}^T, \mathcal{N}^T, \lambda_1^T, \lambda_2^T\}$.

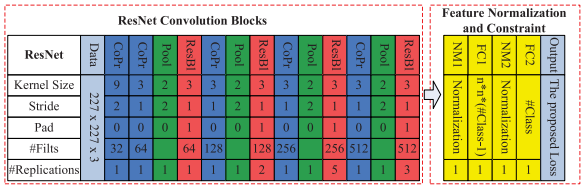

Fig. 6. Network structure of ResNet. *CoPr* denotes the convolutions followed by the PReLU activation function. *Pool* is the MaxPooling function. *ResBl* is a residual block with output $ResOutput = PoolOutput + CoPr(CoPr(PoolOutput))$. *NM*1 denotes the first normalization layer [40]. *#Replications* denotes the times the same block is replicated. *#Filts* denotes the number of feature maps. $n \times n$, *#class* denote the numbers of face patches and expression classes.

D. Network Structure and Training

The proposed feature losses are further used in the ResNet [38] for FER. While Yao *et al.* [39] also devised an efficient model (HoloNet) for FER, the model trained on the face database is unavailable. Thus, the ResNet [38] with slight modification, pretrained for face recognition, is employed for the fine tuning.

The residual network (ResNet) fits the residual mapping $\mathcal{F}$ and then appends it 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. The kernel size of the first convolution is modified to 9 and the neuron numbers of the last two FC layers are modified for FER. The configuration of the modified ResNet network is presented in Fig. 6.

To fully make use of the pretrained models, the fine-tuning strategy based on face recognition model [41] is employed, that is, "face_model.caffemodel" [38] is used for the fine tuning. Although the network trained for face recognition is different from that for FER, the learned network coding the key face parts can be set as the initialization for transfer learning.

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

$$\text{Label}_i = \arg \max_{1 \le k \le \#\text{class}} \sum_{j=1}^{n_i} v_{ij,k} \tag{12}$$

where n_i is the number of augmented faces of the ith sample with Fig. 2(c), $v_{ij,k}$ is the kth output probability of the jth augmented face with the dimension of $\#class$. Label_i is the finally recognized label of the ith testing sample. For the branched loss network, $v_{ij,k}$ in (12) is set as the kth average probability of multiple branched outputs.

Furthermore, weight balancing strategy is employed for the unbalanced databases, that is, the weight of each class is inversely proportional to the number of samples as follows:

$$w_i = \frac{1/\#S_i}{\sum_i 1/\#S_i} \tag{13}$$

where $\#S_i$ denotes the number of expression samples of the ith class. The SoftMax loss function in (1) is then formulated as follows:

$$\mathcal{L}_S = -\sum_i w_{y_i} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_j e^{W_j^T x_i + b_j}}. \tag{14}$$

Consequently, the recognition rates of the classes with significantly different numbers of samples are balanced.

E. Time Complexity Analysis

Since the numbers of scales and directions of the Gabor filters are constant, the time complexity for extracting the feature GABOR is $O(n_{\text{pixel}} \cdot \log(n_{\text{pixel}}))$, where n_{pixel} is the number of pixels of an image. The time complexity for extracting the SIFT feature is $O(n_{\text{pixel}} \cdot n_{\text{tile}})$, where n_{tile} is the number of pixels of each segmented tile. Since projection is included in the extraction of the GSF feature, the time complexities of PCA and LDA are both $O(n_{\text{var}}^2 n_{\text{samp}} + n_{\text{var}}^3)$ for n_{samp} samples with n_{var}-dim variables.

For deep network training, the time complexity of the convolution blocks is $T_{\text{conv}} \sim O(\sum_{l=1}^{d_{\text{conv}}} n_{\text{map},l}^2 \cdot n_{\text{ker},l}^2 \cdot n_{\text{cha},l-1} \cdot n_{\text{cha},l})$, where d_{conv} is the number of convolution layers, and $n_{\text{map},l}$, $n_{\text{ker},l}$, and $n_{\text{cha},l}$ are the feature map size, the kernel size, and the channel number in the lth layer. While the time complexities for the layers of pooling, ReLU, and batch normalization are negligible compared with that of the convolution layers. For the FC layers, the time complexity is $T_{fc} \sim O(\sum_{l=1}^{d_{fc}} n_{\text{neu},l-1} \cdot n_{\text{neu},l})$, where $n_{\text{neu},l}$ is the number of neurons of the lth FC layer, and d_{fc} is the number of FC layers. For the employed ResNet, the time complexity approximates to the number of $T_{\text{conv}} \approx 7.8 \times 10^8$ floating-point operations (FLOPs) for the convolution blocks, while $T_{fc} \approx 8.7 \times 10^6$ for the last two FC layers. Thus, the time complexities of handcrafted feature extraction and FC layers are negligible compared with that of the convolution blocks. When the numbers of epochs n_{epoch} and samples n_{samp} are considered, the time complexity of the network training turns out to be $T \sim O(n_{\text{epoch}} \cdot n_{\text{samp}} \cdot T_{\text{conv}})$.

TABLE I
PARAMETER SETTINGS OF THE PROPOSED ALGORITHM

Parameter Name	Value	Parameter Name	Value
λ_C in equation (5)	8e-3	$\lambda_{F1} = \lambda_{F2}$ in equation (5)	1e-6
Learning rate	5e-3	Batch size	64
Momentum	0.8	ρ in equation (6)	0.98
Image size	227x227	λ_b in equation (9)	1e-6
γ in equation (10)	1.2 or 1.05	κ in equation (11)	1e-5

Fig. 7. Example images of FER2013, CK+, Oulu-CASIA, and MMI. The columns represent expressions of An, Di, Fe, Ha, Sa, Su, and Ne, respectively.

Compared to the training of the original network, the time complexity remains T for the branched or alternative loss with fixed weights (ALFWs). For the branched loss with adaptive weights (BLAWs), the time complexity of the training network approximates to T. While for the alternative loss with adaptive weights (ALAWs), the time complexity of the training network approximates to $3T$, since three trials with different sets of loss weights are performed.

III. EXPERIMENTAL RESULTS

In this section, the experimental setting and employed databases are first presented; then, the proposed feature loss is studied based on a toy model. After that the proposed algorithm with different layer constraints and handcrafted features is tested on the FER2013 database. Furthermore, different loss weighting structures and strategies are evaluated and compared on the four databases. Finally, the performance of the proposed algorithm is compared to the state-of-the-art algorithms on the four public databases.

A. Experimental Setting and Databases

We perform the experiments using four-kernel Nvidia TITAN GPU Card, CAFFE package. The parameter settings of the proposed feature loss and loss weighting strategy are presented in Table I.

The proposed algorithm is tested on the FER2013 database [40], the Extended Cohn-Kanade dataset (CK+) [42], the Oulu-CASIA NIR&VIS database [43], and the MMI database [44], whose examples are presented in Fig. 7. The database is categorized to neutral and six basic expressions, that is, angry (An), disgust (Di), fear (Fe), happy (Ha), sad (Sa), and surprise (Su). The CK+ database consists of 123 subjects with 593 expression sequences, where 327 sequences are labeled with one of seven expressions (angry, disgust, fear, happy, sad, surprise, and contempt). Each sequence contains a set of captured frames when the

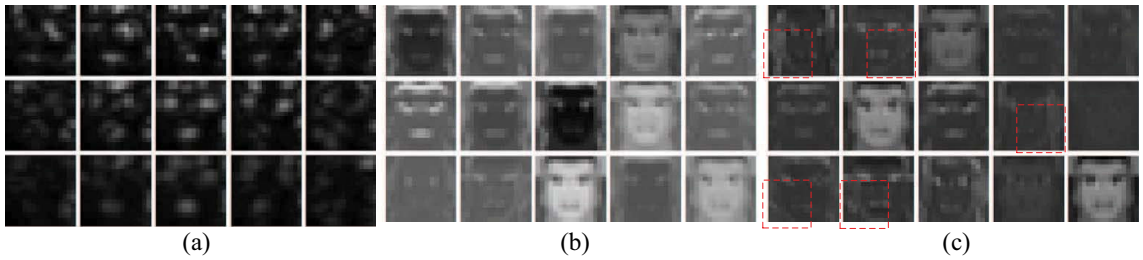

Fig. 8. Comparison among (a) GABOR filtered images, feature maps (FC1) learned with the (b) SoftMax loss and the (c) proposed feature loss. The dashed rectangles in (c) mark out the learned multidirection information that is different from (b).

subject changes her/his expression. One thousand and thirty three expression images, that is, the neutral and three non-neutral peak frames sampled from the labeled sequences are used for testing. The FER2013 database [40] is collected from the Internet and used for a challenge, which consists of 35 887 grayscale face images with the size of 48×48. While both of the validation and testing sets consist of 3589 samples, the remaining 28 709 examples are used as the training set. Each face was also labeled with one of seven categories.

The Oulu-CASIA NIR&VIS expression database [43] contains videos of 80 subjects, each acts with the six typical expressions. The images were captured with two imaging systems, namely, near infrared (NIR) and visible light (VIS), under three different illumination conditions, that is, normal indoor illumination, weak illumination (only computer display is on), and dark illumination (all lights are off). The databases of *NI* and *Strong* are employed for evaluation. When the entire expression sequence is used for the training, the frames with the preceding three largest deformation intensities are used for the testing.

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

For a fair comparison in the following experiments, the ten-fold cross-validation with the person-independent strategy is employed for testing. Thus, each considered database, except FER2013, is divided into ten groups with approximately equal number of person IDs. While nine of them were used for training, that is, eight and one groups are used as the training and validation datasets, the remaining group was used for testing. The validation accuracy is used for the weight tuning, while the testing accuracy is used for performance evaluation and comparison. The process was repeated for each of ten-fold and the overall accuracy, that is, *#SummedCorrectTesting*/*#AllTesting* is recorded as the final result. For the batch-based SGD optimization, the training samples are randomly perturbed. For PCA in the GSF and SIFT construction (4), the eigenvectors with the accumulative contribution rate of 0.99 are used.

GABOR Feature:	Fear	Fear	Happy	Angry	Fear
Softmax Loss:	Sad	Fear	Fear	Surprise	Neutral
Feature Loss:	Neutral	Sad	Surprise	Fear	Sad
Ground Truth:	Neutral	Sad	Surprise	Fear	Sad

Fig. 9. Example faces that are correctly recognized using the feature trained with the proposed feature loss, while wrongly recognized using the GABOR feature and feature trained with the SoftMax loss.

B. Feature Loss Analysis With Toy Model

Motivated from the studies [13], [23], [45] that provided semantic explanations of features, the effect of the proposed feature loss on the network learning is studied with a toy model. The toy model mainly consists of "CoPr," "Pool," and CoPr blocks with (Kernel Size, Stride, Pad, #Filts) of (3, 3, 0, 128), (2, 2, 0, 128), and (2, 2, 1, 512), respectively. The "Feature Normalization and Constraint" shown in Fig. 6 is applied. The toy model is then tested on the FER2013 database, where the input image size is set as 48×48. The handcrafted Gabor features are generated by convolving a face image with a set of Gabor filters.

Fig. 8 shows a number of 3×5 Gabor filtered images for an example face, together with the same number of feature maps learned by the toy model, using SoftMax loss (b) and the proposed feature loss (c). As shown in the figure, while some of the feature maps learned by both losses look similar with the Gabor filtered images, the feature maps learned by the proposed feature loss seems to integrate the information from both (a) and (b). Compared to the deep features learned by SoftMax, our approach integrates more discriminative information similar to the Gabor features. Compared with the Gabor feature that equally weights the features extracted at different scales and orientations, our feature imposes adaptive weights for them. The feature maps learned by the proposed feature loss are thus complementary to both Gabor features and deep features learned by SoftMax.

While Fig. 9 shows five example expressions correctly classified by the proposed feature loss, Fig. 10 shows the heat maps of the toy network after the training of 30 000 epochs for them. The heat maps show that more responses with large values are located on the expression sensitive regions, due to the guidance of the proposed feature loss. For example, the eyes of the first to fourth faces and the mouths of the 1st, 4th,

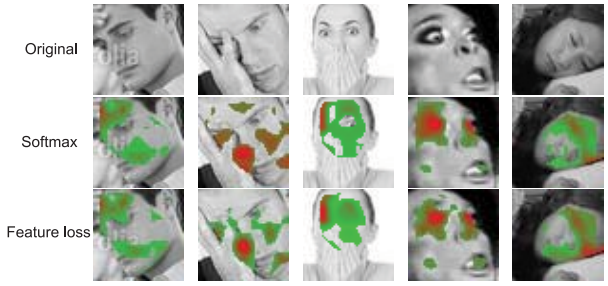

Fig. 10. Heat maps of the average feature maps of the deep networks trained with and without the proposed feature loss, where the top 20% largest response values are demonstrated in colors. Red color denotes the larger response values.

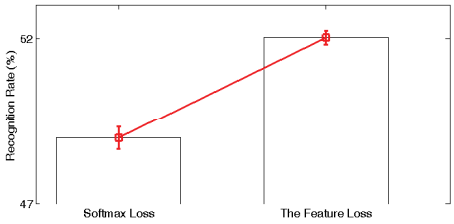

Fig. 11. Means and variances of the recognition rates of ten runs with the SoftMax and proposed feature losses.

and 5th faces are more emphasized by the proposed feature loss.

The performances of the features learned with and without the proposed feature loss are also evaluated using FER2013. The model was trained for ten times and the means and variances of the accuracies are shown in Fig. 11. As shown in the figure, the performance of the model trained using the proposed feature loss is significantly higher than that trained using the SoftMax. Due to the guidance of the Gabor feature, the performance of the model trained using the proposed feature loss is also much more stable, that is, the variance is much smaller than that trained using SoftMax.

C. Performance Without Adaptive Loss Weighting

In this section, the feature loss with different settings is evaluated on the FER2013 database. First, the performances of the handcrafted features of GSF, SIFT, and GABOR are compared to those of the corresponding feature loss. Second, different combinations of SoftMax, Center loss [18], feature regularization [40], and the proposed GSF feature loss are evaluated. Third, the proposed feature loss with different layer constraints and norms is studied on both validation and testing datasets. Lastly, the GSF and SIFT feature losses are evaluated on the four public databases.

To study the motivation of the proposed feature loss, Table II lists the performance of the employed handcrafted features with the SVM classifier [46], together with that of the deep network with the corresponding feature loss and without any feature loss on the FER2013 database. SVM classifier [46] is also used to test the performance of the learned deep features for comparison. Table II shows that the proposed feature losses largely improve the performances of the original handcrafted features and the network without any feature loss.

TABLE II
RECOGNITION RATES (%) OF THE HANDCRAFTED FEATURES AND THE PROPOSED DEEP LEARNING ON THE FER2013 DATABASE WITH (W.I.) AND WITHOUT (W.O.) THE PROPOSED FEATURE LOSS, WHERE GSF, GABOR, AND SIFT ARE EMBEDDED ON THE FC2, FC1, AND FC2 LAYERS, RESPECTIVELY

Recog. Rate (%)	W.O.	Features			-
		GSF	GABOR	SIFT	
Hand-crafted Feat. (SVM)	-	48.12	46.06	44.36	W.O.
Deep Feat. (SVM)	67.87	69.94	69.77	69.43	-
Deep Feat. (Softmax)	**69.71**	**72.22**	**72.08**	**71.64**	-

TABLE III
PERFORMANCES OF DIFFERENT COMBINATIONS OF LOSSES FOR THE FER2013 DATABASE

Id	SoftMax	Center	Normalization	GSF loss	Recog. Rate (%)
1	Yes	No	No	No	69.71
2	Yes	Yes	No	No	71.58
3	Yes	No	Yes	No	71.36
4	Yes	Yes	Yes	No	71.77
5	Yes	No	No	Yes	72.22
6	Yes	Yes	No	Yes	72.28
7	Yes	No	Yes	Yes	72.28
8	Yes	Yes	Yes	Yes	**72.36**

It seems that deep network achieves much better accuracy than the handcrafted features. For example, the deep feature achieved 67.87% accuracy, which is about 19.75% higher than that (48.12%) of the best handcrafted feature, GSF, when SVM is used for classification. When SoftMax increased the performance of deep feature from 67.87% to 69.71%, the proposed feature loss with GSF further improved the accuracy to 72.22%.

To test the performance of the proposed loss function, the network in Fig. 3 is trained with different loss functions, that is, the SoftMax, the Center loss, the proposed feature loss, and their different combinations. The recognition rates for different loss functions are presented in Table III, where the performance with only the SoftMax is listed as the baseline. Table III shows that both Center and feature losses are beneficial to the performance and the fused loss metric achieves the best performance, that is, the combination of three losses achieves an improvement of 2.65% on the FER2013 database, over the SoftMax loss. Meanwhile, the loss with GSF feature guidance performs better than the Center loss by comparing the accuracies in the second and fifth rows or the fourth and seventh rows, that is, improvements of 0.64% and 0.51% are achieved.

To test the performance of the employed feature normalization [40] (Fig. 6) of the proposed network, different configurations with and without the feature normalization are employed for the training on the FER2013 database and the performances are shown in Table III. The table shows that the proposed fused feature loss outperforms the feature normalization with an improvement of 0.86%, while the algorithm configuration with both the normalization and the GSF loss achieved an improvement of 2.57% over the network with only SoftMax loss.

To study the performance of the feature losses with different layers, one-layer [$\lambda_{F1} > 0$ or $\lambda_{F2} > 0$ in (5)] and two-layer ($\lambda_{F1} > 0$ and $\lambda_{F2} > 0$) feature losses are tested on

TABLE IV
RECOGNITION RATES (%) OF FEATURE LOSSES WITH DIFFERENT LAYERS FOR THE FER2013 DATABASE

FC1	FC2	λ_{F1}	λ_{F2}	Recog. Rate (%, validation)	Recog. Rate (%, testing)
-	-	-	-	68.68	69.71
GABOR	GSF	1e-7	1e-7	69.66	72.14
GABOR	GSF	1e-6	1e-6	69.63	72.0
GABOR	-	1e-7	-	70.08	72.14
GABOR	-	1e-6	-	**70.19**	72.25
-	GSF	-	1e-7	70.05	72.30
-	GSF	-	1e-6	70.13	**72.36**

TABLE V
RESULTS OF FEATURE LOSSES WITH DIFFERENT NORMS ON THE FC2 LAYER FOR THE FER2013 DATABASE

Feature loss	Norm	Recog. Rate (%, validation)	Recog. Rate (%, testing)
Non	-	68.68	69.71
SIFT	L_1	69.3	71.66
SIFT	L_2	69.41	71.64
GSF	L_1	69.57	71.85
GSF	L_2	**70.13**	**72.36**

the FER2013 database under different parameter settings. The recognition results are shown in Table IV.

Table IV shows that the performance of the feature loss with GABOR feature is comparable to that of the GSF feature on the validation dataset. However, the dimension of GABOR, that is, $(12\,800 = 512 \times 5 \times 5)$ is significantly larger than that (not larger than 384) of GSF, which needs significantly more computation resources and increases the model complexity. Thus, one layer feature loss with feature constraint imposed on the layer of FC2 is considered in the following evaluation and comparison. Meanwhile, Table IV shows that the recognition rates on the validation and testing datasets are positively correlated. Thus, the loss weights of λ_{F1} and λ_{F2} corresponding to the statistically best performance on the validation dataset are applied to the testing dataset. More precisely, the loss weight setting of $1e-6$ performs better than $1e-7$ on the validation dataset for three pairs of comparison, which becomes the default loss weight setting in Table I.

To study the performance of different norms of the feature loss (1), the L_1 and L_2 norms of GSF and SIFT on the FC2 layer are tested on the FER2013 database. The results are shown in Table V. Although SIFT feature loss with the L_1- and L_2-norms achieves similar performance on the FER2013 database, Table V shows that the L_2-norm of GSF constraint performs better than the corresponding L_1-norm feature loss, where the margins of 0.56% and 0.51% are achieved on the validation and testing datasets, respectively. Consequently, the L_2-norm feature loss is used for the following evaluation and comparison.

Based on the performance evaluation with different combinations of losses, norms, and feature layers on the FER2013 database, the feature loss with GSF/SIFT L_2 constraint on the FC2 layer is further evaluated on the four public databases, and the results, as well as the baselines with the SoftMax loss are presented in Table VI.

Table VI shows that GSF loss performs better than SIFT loss on three of the four databases, while the accuracy of GSF loss

TABLE VI
RECOGNITION RATES (%) OF DIFFERENT FEATURE LOSSES ON THE FOUR DATABASES

Loss	FER2013	CK+	Oulu-CASIA	MMI
Softmax (Baseline)	69.71	94.7	77.29	72.69
Feature loss with GSF	**72.36**	**97.35**	82.71	**78.53**
Feature loss with SIFT	71.66	96.63	**84.58**	76.59

TABLE VII
RECOGNITION RATES (%) OF DIFFERENT WEIGHTING STRATEGIES ON THE FOUR DATABASES

Database	BLFW	BLAW	ALFW	ALAW
FER2013	72.27	72.08	72.58	**72.67**
CK+	97.11	96.39	**97.35**	97.11
Oulu-CASIA	83.33	83.33	83.96	**84.17**
MMI	74.15	78.05	**78.53**	**78.53**

is lower than that of the SIFT loss with a margin of 1.87% for the Oulu-CASIA database. Meanwhile, the proposed feature loss with GSF achieves large improvements over the baseline on the four databases, that is, improvements of 5.42% and 5.84% are achieved on the Oulu-CASIA and MMI databases, respectively. Table VI also reveals that the handcrafted feature loss suitable for a specific database may not work for other databases, which motivated us to propose self-adaptive algorithm for loss weighting and selection.

D. Performance With Adaptive Loss Weighting

In this section, the fused feature losses with the newly proposed weighting strategies in Fig. 4, that is, the branched loss with fixed weights (BLFWs), the BLAWs, the ALFWs, and the ALAWs, are evaluated on the four public databases. The recognition results are presented in Table VII.

Table VII shows that the alternative loss performs better than the branched feature loss on the four public databases. Compared with the branched feature loss which encourages independent FC layers and SoftMax loss, the alternative loss can entangle the information from different feature losses to provide complementary information for network training. Compared with the strategy with fixed weights, the adaptive loss weighting can self-adaptively change the weights of different feature losses according to the performance on the validation dataset. Although the validation accuracy, together with feature sparseness, does not directly reflect the performance of the feature loss for the testing dataset, the ALAWs achieves better performance than the fixed weight strategy on the FER2013 and Oulu-CASIA databases.

Compared with the performance of a single feature loss in Table VI, the alternative loss with fixed or adaptive weights achieves competitive performance by using the complementary information of the GSF and SIFT losses, and a recognition rate of 84.17% is achieved on the Oulu-CASIA database, which is close to the best achieved performance of 84.58% by the single SIFT feature loss. Meanwhile, while the alternative loss obtains the best performances on the other two databases compared with the single GSF or SIFT feature loss, it achieves better performance on the FER2013 database than both single feature losses, that is, an improvement of 0.31% is obtained, compared with the best GSF feature loss (72.36%).

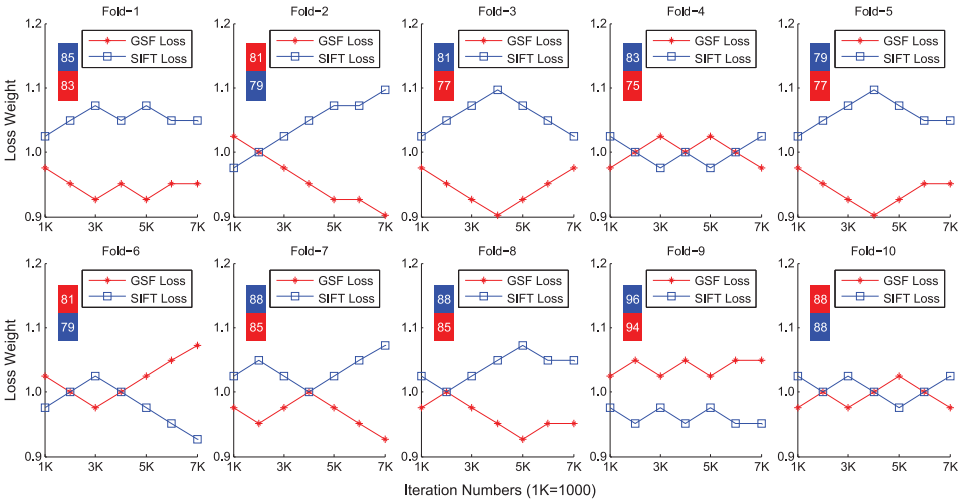

Fig. 12. Weight evolution of the alternative feature loss for ten-fold training on the Oulu-CASIA database. The colored rectangles record the recognition rate (%) of each feature loss for each testing dataset.

While Table VII shows that the alternative feature loss with adaptive weights achieves the most competitive performance on the Oulu-CASIA database, the evolutions of loss weight for the ten-fold experiments are presented in Fig. 12. Since the weights are directly related with the validation accuracy in (10) and (11), the testing accuracies are demonstrated. Fig. 12 shows that the two loss weights changed adaptively for different testing datasets. While Table VI has shown that the SIFT loss outperforms the GSF loss on the Oulu-CASIA database, Fig. 12 further reveals that the loss weights of the SIFT loss are set bigger for most folds of the database. As the weight is mainly associated with the performance of the network on validation set, Table VI and Fig. 12 show that the network performs almost consistently on the testing set. However, due to the big differences between validation and testing datasets, the weight evolutions of the second-fold and ninth-fold do not match with the testing accuracy, that is, the better testing accuracy of the loss, the smaller weight was set. However, the loss weights for the two feature losses are matched with the testing accuracy for the other folds, which illustrates the reasonability of the defined objective function in the loss weighting for the Oulu-CASIA database.

To evaluate the overall performance, the confusion matrices of the proposed algorithm with the alternative loss on the four databases are presented in Fig. 13.

One can observe from these figures that "fear" is the most difficult expression, which has the lowest recognition rates on the databases of FER2013, Oulu-CASIA and MMI. While "sad" is the most difficult expression for the CK+ database, it is largely recognized as the "neutral" expression for the FER2013 and CK+ databases, and the "angry" expression for the Oulu-CASIA and MMI databases. Meanwhile, Fig. 13 also reveals that the expressions angry, disgust, fear, and sad with smaller deformation intensities are relatively more difficult than the other three expressions.

Regarding to the parameter selection, the proposed algorithm introduces several hyperparameters in Table I, that is, λ_{F1}, λ_{F2}, γ, and κ, where the values of the parameters λ_{F1}, λ_{F2}, and γ are determined according to the performance

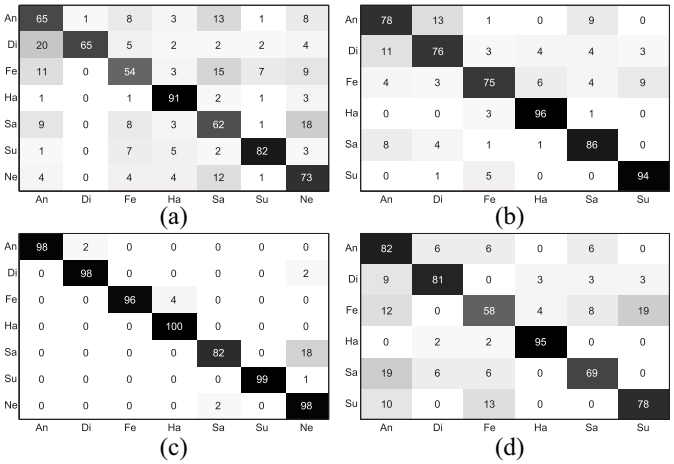

Fig. 13. Confusion matrix (%) of the proposed algorithm for the (a) FER2013, (b) Oulu-CASIA, (c) CK+, and (d) MMI databases.

TABLE VIII
RECOGNITION RATES (R.R. %) WITH DIFFERENT PARAMETER κ ON THE FER2013 DATABASE

κ	1e-4	5e-5	1e-5	5e-6	1e-6
R.R. (Validation)	70.3	70.27	**70.35**	70.24	70.21
R.R. (Testing)	72.58	**72.72**	72.67	72.53	72.56

on the validation dataset. While Table IV has shown that slight perturbation of the parameters λ_{F1} and λ_{F2} does not make significant change on the recognition performance, the performance sensitivity against the perturbation of the hyperparameter κ in (11) around the default setting is evaluated on the FER2013 database, and the results are shown in Table VIII. While further improvement is possible by fine tuning the hyperparameter κ, Table VIII shows that the performance is insensitive to the setting of κ.

E. Comparison With Other Algorithms

Tables IX–XII compare the performances and the testing protocols of our approach with the state-of-the-art approaches in the literature, for all of the four databases.

TABLE IX
PERFORMANCE OF DIFFERENT ALGORITHMS ON THE FER2013 DATABASE

Method	Network	External Data	Baseline (%)	Recog. rate (%)
Feature loss with HoloNet 2018 [27]	One network	Non	57.7	61.86
Deeper DNN 2016 [25]	One network	Non	-	66.4
DNN with SVM 2013 [54]	One network	Non	70.1	71.2
Fusing aligned faces 2016 [48]	Four networks	Aligned dataset	-	73.73 (71.86 by single DCN)
Fusing external database 2015 [47]	Two networks	Social relation dataset	-	**75.1**
Fusing deep and hand-crafted features 2019 [49]	Six networks	Face recognition model	71.89	74.92
Ours (ALFW)	One network	Face recognition model	69.71	72.58
Ours (ALAW)	One network	Face recognition model	69.71	72.67

TABLE X
PERFORMANCE OF DIFFERENT ALGORITHMS ON THE CK+ DATABASE. SYMBOL "-" DENOTES NOT REPORTED. SYMBOL "*" DENOTES THAT NEURAL IS REPLACED WITH CONTEMPT EXPRESSION; "10F" DENOTES "TENFOLD"; "SUB." AND "PROTO." ARE THE ABBREVIATIONS OF "SUBJECT" AND "PROTOCOL"

Method	Data	#class	Sub.	Proto.	Network	Baseline (%)	Recog. rate (%)
Deeper DNN 2016 [25]	The Peak	6	106	5F	One network	-	93.2
Multiscale active learning 2015 [3]	Three Peak	6	96	10F	Non	Non	91.53
Discriminant graph structures 2008 [55]	-	6	-	5F	Non	Non	97.1
Feature selection and disentangling 2014 [50]	Three Peak	6	106	8F	Non	Non	**97.7**
Adaptive deep metric 2017 [20]	Three Peak	7*	118	10F	One network	-	97.1
Fine tuning 2015 [4]	Temporal	7*	106	10F	Two networks	96.94	97.25
Spatial or temporal network 2017 [51]	Temporal	7*	118	10F	One network	-	96.36
Spatial-temporal recurrent network 2017 [26]	Temporal	7*	118	10F	Two networks	-	95.4
Sparse autoencoders 2018 [56]	Four Peak	8	123	10F	One network	-	95.79
Feature map selection 2017 [45]	Three Peak	8	123	10F	One network	-	**96.95**
Face net regularization 2016 [41]	Three Peak	8	123	10F	One network	89.9	96.8
Margin projection 2013 [21]	The Peak	7	100	5F	Non	Non	89.2
Radial feature 2012 [22]	Five images	7	94	10F	Non	Non	91.51
AU network 2013 [23]	Three Peak	7	118	10F	One network	-	92.05
Feature loss with HoloNet 2018 [27]	Three Peak	7	106	10F			97.35
Dropout and randomized DMLs 2018 [52]	Five Peak	7	118	10F	11 recurrent layers	-	**99.11** (97.68 for one recurrent layer)
De-expression residual learning 2018 [53]	Three Peak	7	118	10F	Two networks	89.5	97.3
Ours (ALFW)	Three Peak	7	106	10F	One network	94.7	97.35
Ours (ALAW)	Three Peak	7	106	10F	One network	94.7	97.11

For the FER2013 database, the study [47] achieved the best performance of 75.1%. However, it requires multiple external databases such as the social relation database for training. The work [48] achieved a high recognition rate of 73.73%, where images were preprocessed with face frontalization and multiple networks were employed. The proposed algorithm achieves a recognition rate of 72.67%, which is 0.81% higher than the work [48] when the same experimental setting, that is, a single network, is employed for the training. Though the work [49] achieved a higher recognition rate of 74.92%, it needs to fuse the bag of visual words into three different pretrained and fine-tuned networks for local SVM discrimination, which requires larger model complexity than the proposed algorithm using only single network. Compared with the algorithms that achieved better performance, the proposed algorithm only modifies the network loss function, without the requirement of external databases.

For the CK+ database, Table X shows that the proposed algorithm achieves a competitive recognition rate among 18 state-of-the-art algorithms. The work [50] achieved an accuracy of 97.7% by boosting generalization ability with feature selection and disentangling. However, it is tested for six-expression categories. Zhou and Shi [45] achieved the best performance of 96.95% for eight-expression recognition, while the additional information of action units is needed for the feature map selection. Compared with the study [51], the proposed algorithm obtains about 1% higher accuracy than the spatial and temporal models [51] trained with single network. Considering the algorithms for seven-class FER, the proposed algorithm achieved an accuracy of 97.35%, which ranked the second among seven different algorithms. The recurrent deep learning embedded with dropout and random DML [52] achieves the best performance of 99.11%. However, 11 recurrent layers are needed, which leads to large computational complexity. However, the recognition rate of 97.35% achieved by the proposed algorithm is rather competitive, compared with 97.68% of the study [52] when one recurrent layer is adopted.

For the Oulu-CASIA database, the proposed algorithm achieves a competitive performance under the lighting condition of *Strong-VIS* among the state-of-the-art algorithms presented in Table XI. The algorithm [41] achieved a better performance of 87.71% by proposing a regression loss to learn the deep feature of face recognition to guide the training of the expression feature generation, and an additional model for feature learning is pretrained. In addition, the proposed algorithm achieves an improvement of 6.88% over the baseline, which outperforms that of [41] (improvement of 4.45%). The study [53] achieved the best performance, that is, 88% by using the de-expression information with multiple loss functions generated from multiple hidden layers, which may benefit from better generalization ability of the trained

TABLE XI
PERFORMANCE OF DIFFERENT ALGORITHMS ON THE OULU-CASIA DATABASE

Method	Data	#class	Sub.	Proto.	Network	Baseline (%)	Recog. rate (%)
AdaLBP 2011 [43]	Temporal (*Strong-VIS*)	6	480	10F	Non	Non	73.54
Fine tuning 2015 [4]	Temporal (*Strong*)	6	480	10F	Two networks	75.63	81.46
Spatial or temporal network 2017 [51]	Temporal (*Strong*)	6	480	10F	One network	-	78.96
Face net regularization 2016 [41]	Three Peak (*Strong-VIS*)	6	480	10F	One network	83.26	87.71
De-expression residual learning 2018 [53]	Three Peak (*Strong-VIS*)	6	480	10F	Two networks	72.92	**88**
Ours (ALFW)	Three Peak (*Strong-NIR*)	6	480	10F	One network	77.29	83.96
Ours (ALAW)	Three Peak (*Strong-NIR*)	6	480	10F	One network	77.29	84.17
Ours (ALFW)	Three Peak (*Strong-VIS*)	6	480	10F	One network	77.29	85.42
Ours (ALAW)	Three Peak (*Strong-VIS*)	6	480	10F	One network	77.29	85.83

TABLE XII
PERFORMANCE OF DIFFERENT ALGORITHMS ON THE MMI DATABASE, "SEQ." DENOTES THE "SEQUENCE"

Method	Data	#class	Seq.	Proto.	Network	Baseline (%)	Recog. rate (%)
Fine tuning 2015 [4]	Temporal	6	205	10F	Two networks	67.8	70.24
Spatial or temporal network 2017 [51]	Temporal	6	205	10F	One network	-	77.05
AU network 2013 [23]	Three Peak	7	205	10F	One network	-	74.76
Deeper DNN 2016 [25]	-	6	79	5F	One network	-	77.6
Multiscale active learning 2015 [3]	Three Peak	6	-	10F	Non	Non	77.39
De-expression residual learning 2018 [53]	Three Peak	6	208	10F	Two networks	57	73.23
Adaptive deep metric 2017 [20]	Three Peak	6	205	10F	One network	-	**78.53**
Ours (ALFW)	Three Peak	6	205	10F	One network	72.69	**78.53**
Ours (ALAW)	Three Peak	6	205	10F	One network	72.69	**78.53**

features. However, the proposed algorithm achieved more balanced performance on the databases of CK+, Oulu-CASIA, and MMI, where an improvement of 5.3% is obtained by the proposed algorithm over the study [53] on the MMI database in Table XII. On *Strong-NIR*, the accuracy of the proposed algorithm is 2.71% higher than that of the work [4] using both the geometry and temporal losses, and 5.21% higher than [51] when single network is employed. Meanwhile, compared to the algorithms [41], [53], the proposed algorithm achieves better performance on the CK+ database in Table X.

For the MMI database, the proposed algorithm achieves the same recognition rate, that is, 78.53% with the approach of adaptive deep metric [20] that adopted identity-aware hard-negative mining, beating the other algorithms using static expression frames.

Compared with the algorithms that achieved the best performances on the four databases in Tables IX–XII, the proposed algorithm achieves balanced performances on the four databases, and comparable accuracies with the state-of-the-art on three of the four public databases. While different image preprocessing strategies and network models are employed, the competitive performance still reveals the effectiveness of the proposed algorithm.

IV. DISCUSSION AND CONCLUSION

This paper proposed a general framework for embedding a handcrafted feature constraint into a deep loss for feature learning. As different feature losses may be suitable to only some specific databases, four feature fusion methods with different structures and loss weighting strategies are introduced and evaluated. The experimental results on FER2013, CK+, Oulu-CASIA, and MMI databases show that the proposed algorithm achieved significantly better performances than the original handcrafted features and the features learned without using the proposed feature loss. Competitive performances compared with the state-of-the-art approaches on the four public databases have also been observed. The proposed algorithm provides a novel feature loss and a newly devised self-adaptive loss weighting to use multiple handcrafted features to improve the recognition performance.

However, there is a large room for further improvement. First, for the alternative loss with adaptive weighting, two additional training are needed to update the weights, which could be simplified by fixing the lower-layer parameters and fine tuning only the higher-layer weights to decrease the computational complexity. Second, handcrafted feature with large dimension can make the training model very large due to the number of neurons of the FC layers, which can be made sparse before embedded into the proposed feature loss for guidance. Third, considering that better validation accuracy and feature sparseness do not necessarily lead to better testing accuracy, more efficient objective functions for measuring the feature loss could be devised for self-adaptive loss weighing and generalization ability improvement. Fourth, several hyperparameters are introduced in the proposed algorithm, which could be made self-adaptive in the future work. Fifth, the proposed algorithm can be fused with image preprocessing strategy, for example, face frontalization [48], for overall performance improvement. Lastly, the proposed algorithm is general, more network structures and handcrafted features will be tested on other applications like face recognition.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments and constructive suggestions.

REFERENCES

[1] R. T. Ionescu, M. Popescu, and C. Grozea, "Local learning to improve bag of visual words model for facial expression recognition," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2013, pp. 1–6.

[2] K. Sikka, T. Wu, J. Susskind, and M. Bartlett, "Exploring bag of words architectures in the facial expression domain," in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2012, pp. 250–259.

[3] L. Zhong, Q. Liu, P. Yang, J. Huang, and D. N. Metaxas, "Learning multiscale active facial patches for expression analysis," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1499–1510, Aug. 2015.

[4] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *Proc. Int. Conf. Comput. Vis.*, 2016, pp. 2983–2991.

[5] A. Zadeh, T. Baltrusaitis, and L. P. Morency, "Convolutional experts constrained local model for facial landmark detection," in *Proc. Comput. Vis. Pattern Recognit. Workshop*, 2017, pp. 2051–2059.

[6] H. Pan and H. Jiang, "Learning convolutional neural networks using hybrid orthogonal projection and estimation," *CoRR*, vol. abs/1606.05929, 2016.

[7] S. Wang, B. Pan, H. Chen, and Q. Ji, "Thermal augmented expression recognition," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2203–2214, Jul. 2018.

[8] M. Liu, R. Wang, S. Li, S. Shan, Z. Huang, and X. Chen, "Combining multiple kernel methods on Riemannian manifold for emotion recognition in the wild," in *Proc. ACM Int. Conf. Multimodal Interact.*, 2014, pp. 494–501.

[9] R. Paul, S. H. Hawkins, L. O. Hall, D. B. Goldgof, and R. J. Gillies, "Combining deep neural network and traditional image features to improve survival prediction accuracy for lung cancer patients from diagnostic CT," in *Proc. Int. Conf. Syst. Man Cybern.*, Budapest, Hungary, 2016, pp. 2570–2575.

[10] S. P. Suggu, K. N. Goutham, M. K. Chinnakotla, and M. Shrivastava, "Hand in glove: Deep feature fusion network architectures for answer quality prediction in community question answering," in *Proc. Int. Conf. Comput. Linguist. (COLING)*, 2016, pp. 1429–1440.

[11] T. Majtner, S. Yildirim-Yayilgan, and J. Y. Hardeberg, "Combining deep learning and hand-crafted features for skin lesion classification," in *Proc. Int. Conf. Image Process. Theory Tools Appl.*, 2016, pp. 1–6.

[12] O. Araque, I. Corcuera-Platas, J. F. Snchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017.

[13] P. Khorrami, T. L. Paine, and T. S. Huang, "Do deep neural networks learn facial action units when doing expression recognition?" in *Proc. Int. Conf. Comput. Vis. Workshop*, 2015, pp. 19–27.

[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[15] L. L. Shen and L. Bai, "Gabor feature based face recognition using kernel methods," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, Seoul, South Korea, 2004, pp. 170–176.

[16] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, "Local binary convolutional neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 4284–4293.

[17] D. Chen, C. Xu, J. Yang, J. Qian, Y. Zheng, and L. Shen, "Joint Bayesian guided metric learning for end-to-end face verification," *Neurocomputing*, vol. 275, pp. 560–567, Jan. 2017.

[18] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.

[19] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 212–220.

[20] X. Liu, B. V. K. V. Kumar, J. You, and P. Jia, "Adaptive deep metric learning for identity-aware facial expression recognition," in *Proc. Comput. Vis. Pattern Recognit. Workshop*, Honolulu, HI, USA, 2017, pp. 522–531.

[21] S. Nikitidis, A. Tefas, and I. Pitas, "Maximum margin projection subspace learning for visual data analysis," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4413–4425, Oct. 2014.

[22] W. Gu, C. Xiang, Y. V. Venkatesh, D. Huang, and H. Lin, "Facial expression recognition using radial encoding of local Gabor features and classifier synthesis," *Pattern Recognit.*, vol. 45, no. 1, pp. 80–91, 2012.

[23] M. Liu, S. Li, S. Shan, and X. Chen, "AU-aware deep networks for facial expression recognition," in *Proc. Int. Conf. Autom. Face Gesture Recognit. Workshop*, Shanghai, China, 2013, pp. 1–6.

[24] P. Rodriguez *et al.*, "Deep pain: Exploiting long short-term memory networks for facial expression classification," *IEEE Trans. Cybern.*, to be published.

[25] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proc. IEEE Win. Conf. Appl. Comput. Vis.*, 2016, pp. 1–10.

[26] T. Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, "Spatial–temporal recurrent neural network for emotion recognition," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 839–847, Mar. 2019.

[27] G. Zeng, J. Zhou, X. Jia, W. Xie, and L. Shen, "Hand-crafted feature guided deep learning for facial expression recognition," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, Xi'an, China, 2018, pp. 423–430.

[28] C. Xu *et al.*, "Multi-loss regularized deep neural network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 12, pp. 2273–2283, Dec. 2016.

[29] Z. Chen, V. Badrinarayanan, C. Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 793–802.

[30] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[31] L. Shen and L. Bai, "A review on Gabor wavelets for face recognition," *Pattern Anal. Appl.*, vol. 9, nos. 2–3, pp. 273–292, 2006.

[32] K. Yan, Y. Chen, and D. Zhang, "Gabor surface feature for face recognition," in *Proc. Asian Conf. Pattern Recognit.*, Beijing, China, 2011, pp. 288–292.

[33] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 12, pp. 1357–1362, Dec. 1999.

[34] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, no. 1, pp. 1457–1469, 2004.

[35] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, "Domain generalization via conditional invariant representation," in *Proc. Conf. AAAI Artif. Intell.*, 2018, pp. 3579–3587.

[36] R. Zhu, G. Sang, and Q. Zhao, "Discriminative feature adaptation for cross-domain facial expression recognition," in *Proc. Int. Conf. Biometr.*, 2016, pp. 1–7.

[37] Y. Zong, W. Zheng, X. Huang, J. Shi, Z. Cui, and G. Zhao, "Domain regeneration for cross-database micro-expression recognition," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2484–2498, May 2018.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[39] A. Yao, D. Cai, P. Hu, S. Wang, L. Sha, and Y. Chen, "HoloNet: Towards robust emotion recognition in the wild," in *Proc. ACM Int. Conf. Multimodal Interact.*, 2016, pp. 472–478.

[40] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv Preprint, arXiv:1703.09507*, 2017.

[41] H. Ding, S. K. Zhou, and R. Chellappa, "FaceNet2ExpNet: Regularizing a deep face recognition net for expression recognition," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2016, pp. 118–126.

[42] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proc. Int. Conf. Autom. Face Gesture Recognit.*, 2000, p. 46.

[43] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietikainen, "Facial expression recognition from near-infrared videos," *Image Vis. Comput.*, vol. 29, no. 9, pp. 607–619, 2011.

[44] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," in *Proc. Int. Conf. Multimedia Expo*, 2005, p. 5.

[45] Y. Zhou and B. E. Shi, "Action unit selective feature maps in deep networks for facial expression recognition," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 2031–2038.

[46] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[47] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning social relation traits from face images," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 3631–3639.

[48] B.-K. Kim, S.-Y. Dong, J. Roh, G. Kim, and S.-Y. Lee, "Fusing aligned and non-aligned face information for automatic affect recognition in the wild: A deep learning approach," in *Proc. Comput. Vis. Pattern Recognit. Workshop*, Las Vegas, NV, USA, 2016, pp. 1499–1508.

[49] M. I. Georgescu, R. T. Ionescu, and M. Popescu, "Local learning with deep and hand crafted features for facial expression recognition," *IEEE Access*, vol. 7, pp. 64827–64836, 2019.

[50] P. Liu, J. T. Zhou, W.-H. Tsang, Z. Meng, S. Han, and Y. Tong, "Feature disentangling machine—A novel approach of feature selection and disentangling in facial expression analysis," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 84–90.

[51] K. Zhang, Y. Huang, Y. Du, and L. Wang, "Facial expression recognition based on deep evolutional spatial–temporal networks," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4193–4203, Sep. 2017.

[52] M. Alam, L. S. Vidyaratne, and K. M. Iftekharuddin, "Sparse simultaneous recurrent deep learning for robust facial expression recognition," *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4905–4916, Oct. 2018.
[53] H. Yang, U. Ciftci, and L. Yin, "Facial expression recognition by de-expression residue learning," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 2168–2177.
[54] Y. Tang, "Deep learning using linear support vector machines," in *Proc. Int. Conf. Mach. Learn.*, vol. 28, 2013.
[55] S. Zafeiriou and I. Pitas, "Discriminant graph structures for facial expression recognition," *IEEE Trans. Multimedia*, vol. 10, no. 8, pp. 1528–1540, Dec. 2008.
[56] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders," *Neurocomputing*, vol. 273, pp. 643–649, Jan. 2018.

Weicheng Xie received the B.S. degree in statistics from Central China Normal University, Wuhan, China, in 2008, and the M.S. degree in probability and mathematical statistics and the Ph.D. degree in computational mathematics from Wuhan University, Wuhan, China, in 2010 and 2013, respectively.

He is currently an Assistant Professor with the School of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He has been a Visiting Research Fellow with the School of Computer Science, University of Nottingham, Nottingham, U.K. His current research interests include image processing and facial expression synthesis and recognition.

Linlin Shen received the B.Sc. degree from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree from the University of Nottingham, Nottingham, U.K., in 2005.

He is currently a Professor with the School of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He was a Research Fellow with the Medical School, University of Nottingham, where he is researching on brain image processing of magnetic resonance imaging. His current research interests include pattern recognition, medical image processing, and deep learning.

Jinming Duan received the Ph.D. degree in computer science from the University of Nottingham, Nottingham, U.K.

From 2017 to 2019, he was an Research Associate with Imperial College London, London, U.K. He is currently a Lecturer with the University of Birmingham, Birmingham, U.K. His current research interests include deep neural networks, variational methods, partial/ordinary differential equations, numerical optimization, and finite difference/element methods, with applications to image processing, computer vision, and medical imaging analysis.