IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 18, 2017, accepted February 2, 2018, date of publication February 14, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2805861

Adaptive Feature Mapping for Customizing Deep Learning Based Facial Expression Recognition Model

BING-FEI WU, (Fellow, IEEE), AND CHUN-HSIEN LIN, (Student Member, IEEE)
Electrical and Control Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan

Corresponding author: Chun-Hsien Lin (clifflin@cssp.cn.nctu.edu.tw)

This work was supported by the Ministry of Science Technology under Grant MOST 106-2622-E-009-009-CC2.

ABSTRACT Automated facial expression recognition can greatly improve the human–machine interface. The machine can provide better and more personalized services when it knows the human's emotion. This kind of improvement is an important progress in this artificial intelligence era. Many deep learning approaches have been applied in recent years due to their outstanding recognition accuracy after training with large amounts of data. The performance is limited, however, by the specific environmental conditions and variations in different persons involved. Hence, this paper addresses the issue of how to customize the generic model without label information from the testing samples. Weighted Center Regression Adaptive Feature Mapping (W-CR-AFM) is mainly proposed to transform the feature distribution of testing samples into that of trained samples. By means of minimizing the error between each feature of testing sample and the center of the most relevant category, W-CR-AFM can bring the features of testing samples around the decision boundary to the centers of expression categories; therefore, their predicted labels can be corrected. When the model which is tuned by W-CR-AFM is tested on extended Cohn-Kanade (CK+), Radboud Faces database, and Amsterdam dynamic facial expression set, our approach can improve the recognition accuracy by about 3.01%, 0.49%, and 5.33%, respectively. Compared to the competing deep learning architectures with the same training data, our approach shows the better performance.

INDEX TERMS Cross domain adaption, facial expression recognition, computer vision, pattern recognition, image processing.

I. INTRODUCTION

Facial expression recognition plays a vital role in the artificial intelligence era. According to the human's emotion information, machines can provide personalized services. Many applications, such as virtual reality, personalized recommendations, customer satisfaction, and so on, depend on an efficient and reliable way to recognize the facial expressions. This topic has attracted many researchers for years, but it is still a challenging topic since expression features vary greatly with the head poses, environments, and variations in the different persons involved.

To mitigate these variations, some approaches modified the handcrafted features to gain the better performance, like [32] and [33]. Mao *et al.* [34] make a Bayesian model by means of multiple head poses to conquer the feature variation caused by head poses. However, the handcrafted features have shown their limitations in practical applications, so deep learning methods are utilized to make the models learn to extract the complicated features from large amounts of facial expression data [5]–[9]. Most of the standard database for facial expression recognition are not candid since they are built under the controlled environment with coached expressions. Therefore, Li *et al.* [5], Mollahosseini *et al.* [8], and Peng *et al.* [9] apply data mining technique to search for the facial images on the internet to make the model more realistic. For deep learning neural networks, there is no clear rule to determine the architecture and learning parameters, so image pre-processing is often adopted to improve the neural network's performance. Lopes *et al.* [10] apply the spatial normalization, local intensity normalization, and facial image cropping to the Convolutional Neural Network (CNN). Mapped binary pattern method is utilized in [11]. They all have the better result after applying the pre-processing. In addition, some other approaches combine common machine learning models to gain the robustness and higher performance. Vo and Le [12] take the second-to-last output layer as the encoded features, and utilize Support Vector Machine (SVM) to be the label predictor. Hamester *et al.* [13] propose a 2-channel CNN, and the first convolutional layer in one of the channels is trained by Convolutional Auto-Encoder (CAE) to learn the better capability in order to extract better features. In order to mitigate the effect of head pose, a CNN learns the pose-robust features by regressing the features extracted from the Principal Component Analysis Network (PCANet) which has been trained by the frontal facial images with various expressions [14]. Different from traditional learning algorithms in CNN, the model in [15] learns the correlations among the training data. To mitigate the person-specific differences, Meng *et al.* [35] and Zhang *et al.* [36] propose a way to train an identity-aware structure to extract the person-specific features for recognizing the facial expressions. Rather than trying to recognize a single image, Zhang *et al.* [16], Jung *et al.* [17], and Byeon and Kwak [18] predict the expressions by passing a video, which seems more reasonable in practice; nevertheless, labeling video data is more labor intensive.

The approaches mentioned are all static after the learning procedure. If applying enough data for training, they can do well in general cases while the performance would be relatively low in the specific testing, like the experiment results in [9]. Moreover, CNNs are also weak with cross-domain data [6]. Consequently, adaptive learning may be a possible solution to tailor the generic model in specific cases. Feature adaption proposed in [19] tries to find a mapping in the universal Reproducing Kernel Hilbert Space (RKHS). With the mapping, the features of samples can be transferred into a new space where the feature distributions of testing and training samples are similar. Pan *et al.* [20] realize the domain adaption by transfer component analysis while Hsu *et al.* [21] aim to use closest common space learning for associating cross-domain data. Besides, Chu *et al.* [22] train the generic SVM by re-weighting the trained samples which are most relevant to the testing data. These methods have to calculate the relations between both training and testing data sets, so the computational complexity is so high that it is difficult to apply them to deep learning models.

In this research, three types of Adaptive Feature Mapping (AFM) are proposed to transfer the feature space of testing samples to that of training samples as closely as possible. Since AFM learns the data sequentially, it can be deployed to deep learning models easily, and does improve the performance.

There are two main contributions in this paper. First, a novel pre-processing method is proposed for the general facial image processing, and it does improve the performance. Second, the domain adaption methods, AFMs, for deep learning models with large number of training data is proposed, which can fine-tune the parameters efficiently and gain better recognition accuracy in the specific applications.

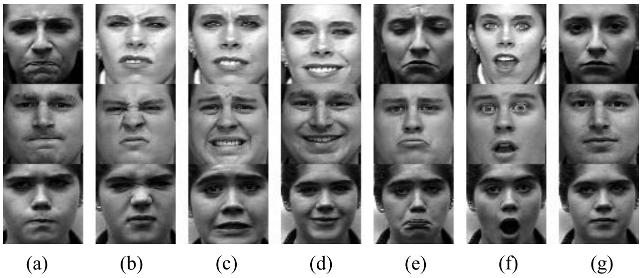

FIGURE 1. Samples in CK+. (a) Anger. (b) Disgust. (c) Fear. (d) Happiness. (e) Sadness. (f) Surprise. (g) Neutral.

This paper is organized as follows. Section II introduces the preparation of testing and training data. Section III and IV address the method of image pre-processing and the CNN architecture. Section V explains how AFMs work and their design principles. Section VI shows the experiments and the discussions. Section VII is the conclusion.

II. FACIAL EXPRESSION DATABASE

This section addresses the usage of the facial expression database and the process of preparing the training and testing data. Only seven common facial expressions, anger, disgust, fear, happiness, sadness, surprise, and neutral, are considered in this paper. Other expressions are ignored even if they are collected in the public domain database.

A. EXTENDED COHN-KANADE

Extended Cohn-Kanade (CK+) [2] has been widely utilized to research facial expression recognition for years. In each expression category for a person, there are about 15 images in a sequence, and the expression intensity changes from low to high. The first one or two images are regarded as the neutral expression while the last one or two images are selected as the expressions in full effect. Consequently, there are 630 images from CK+. The samples are shown in Fig. 1.

B. RADBOUD FACES DATABASE

The Radboud Faces Database (RaFD) [3] is a high quality database of faces, which contains pictures of 8 emotional expressions, including Caucasian males and females, Caucasian children, both boys and girls, and Moroccan Dutch males. Head poses vary from left side to right, and each pose is shot with three eye gazing directions. Compared to CK+, RaFD is more challenging to the recognition model. The samples are shown in Fig. 2.

C. AMSTERDAM DYNAMIC FACIAL EXPRESSION SET

Around 10 emotional expressions are collected in the Amsterdam Dynamic Facial Expression Set (ADFES) [4]. Most of them are videos with head pose variations, and the expression intensity also changes from low to high, like in CK+. The facial images are captured with fixed time steps when the expressions start to become obvious. The samples are shown in Fig. 3.

TABLE 1. The configuration of the testing and training data.

Type	Database	Anger	Disgust	Fear	Happiness	Sadness	Surprise	Neutral	Total
Testing	CK+	63	91	40	130	42	150	114	630
	RaFD	88	90	87	90	89	83	89	616
	ADFES	80	80	80	80	82	80	80	562
Training	RaFD	494	494	466	493	485	452	493	3,377
	ADFES	367	373	367	367	380	343	362	2,559
	Proprietary	407	244	411	6,545	2,105	941	7,002	17,655

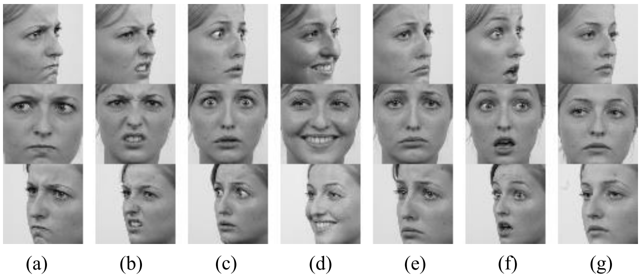

FIGURE 2. Samples in RaFD. (a) Anger. (b) Disgust. (c) Fear. (d) Happiness. (e) Sadness. (f) Surprise. (g) Neutral.

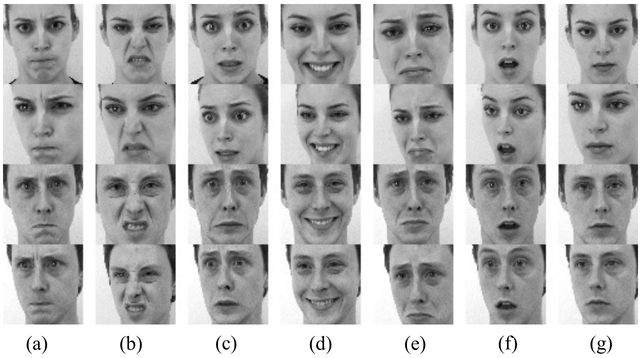

FIGURE 3. Samples in ADFES. (a) Anger. (b) Disgust. (c) Fear. (d) Happiness. (e) Sadness. (f) Surprise. (g) Neutral.

D. PROPRIETARY DATABASE

To make the deep model more robust and general, a home-grown/proprietary database is built to train the model. 372 videos are downloaded from YouTube, including movies, film reviews, variety shows, and some short videos. After that, a face detection method proposed by King [25] is employed to capture the face images with the time intervals set to 1, 2, or 3 second(s) to avoid repeating the images with similar expressions for one person. Then, 100,000 facial images are produced. Only the images that represent their corresponding categories are manually picked to be the training and testing samples. This database ended up with 17,655 images. Some samples are shown in Fig. 4.

E. TRAINING AND TESTING DATA REARRANGEMENT

Since CK+ is a well-known benchmark in facial expression recognition and the number of images is small, it will not be placed in the training set but instead in the testing set only in order to objectively show the performance of the proposed approach. Out of RaFD and ADFES, 10 and 4 persons' images are chosen to be the testing data respectively; therefore, people in the training and the testing sets are definitely different. Altogether, the number of testing images is 630 from CK+, 616 from RaFD, and 562 from ADFES while the number of training data is 23,591 including 17,655 from the proprietary database, 3,377 from RaFD and 2,559 from ADFES. The configuration of the testing and training data is listed in TABLE 1.

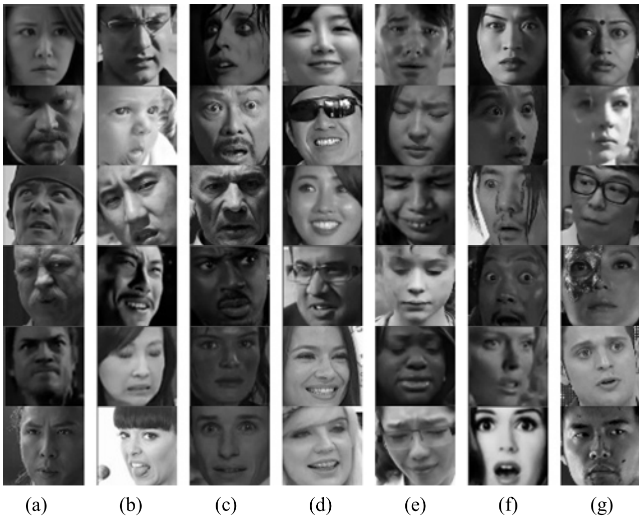

FIGURE 4. Samples in proprietary database (a) Anger. (b) Disgust. (c) Fear. (d) Happiness. (e) Sadness. (f) Surprise. (g) Neutral.

To balance the image count throughout all categories, the category with less images is supplemented by copying the randomly selected images before combining the training data. In this way, the total number of images in every category will be the same.

Researches show that if the data is augmented in a reasonable way, the model can perform much better [30]. Thus, the training set is mirrored and also augmented by two Gamma transformation, three Gaussian blur, and three sharpening filter, so one image is extended to 42 images. As a result, the total number of training data is increased to 2,315,544, and the resolution is set to 64 × 64 pixels in grayscales.

III. IMAGE PRE-PROCESSING

Previous researches [10], [11] have shown that if the image is pre-processed appropriately, the recognition performance

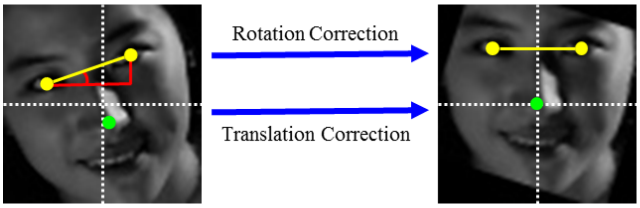

FIGURE 5. Spatial normalization in image pre-processing.

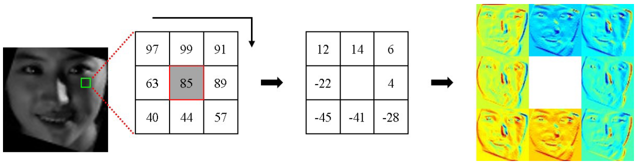

FIGURE 6. Neighbor-center difference images (NCDIs).

can be improved. In this chapter, a proposed pre-processing method which contains spatial normalization and feature enhancement is introduced.

A. SPATIAL NORMALIZATION

The purpose of spatial normalization is to adjust the alignment of the position and rotation angle of the detected facial images. An example is shown in Fig. 5.

A face alignment algorithm [23] is utilized to detect some landmarks on the face. The tip of the nose will be shifted to the center of the image so that the placement offset can be mitigated.

B. FEATURE ENHANCEMENT

Local Binary Pattern (LBP) [30] may be an efficient way to extract the features from images. Nonetheless, it may lose a lot of intrinsic information. Lu *et al.* [24] try to fix this problem by finding a mapping from the Neighbor-Center Difference Vector (NCDV) into the binary space so that the patterns can better represent the images in the original database, but it needs more computing effort.

Neighbor-Center Difference Image (NCDI) is presented to enhance the edges efficiently and retain the original information. The concept is the same as NCDV. NCDIs are extracted by subtracting the center pixel from the neighboring pixels, so the pixel values fall in the range from -255 to 255. An NCDI collects the subtraction results of the selected channel from all patches to reconstruct the image. Thus, eight images which have been sharpened in eight different directions are produced if the 8-channel NCDI is applied, Fig. 6.

After enhancing the edges, the facial contour and background become sharper, but they have nothing to do with facial expressions. Hence, facial image cropping should be applied. Since the facial contour is often confused with the background, the detected landmarks may drift between the facial contour and background. It is not recommended to crop the facial image by connecting the landmarks, which is considered as polygon cropping. Except for the contour, other landmarks are more stable. An elliptical region which regresses the suitable landmarks, as shown in Fig. 7, is the better way to crop the facial image effectively. The ellipse function is

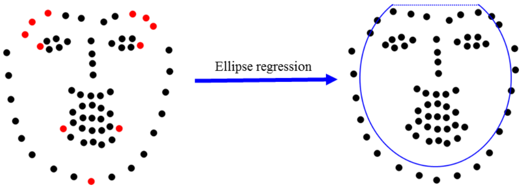

FIGURE 7. Ellipse regression. The red landmarks are the suitable points used to find the cropping boundary, the blue curve, by ellipse regression.

$$f\left(x, y \,|\mathbf{a}\right) = a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6, \quad (1)$$

where $\mathbf{a} = \left[a_1 \; a_2 \; \cdots \; a_6\right]^T$ and $4a_1a_3 - a_2^2 > 0$. To regress these landmarks, the cost function is defined as

$$\begin{aligned}&\Omega\left(\mathbf{a}\,|\mathbf{x}, \mathbf{y}\right)\\ &= \frac{1}{2N}\sum_{n=1}^{N} f^2\left(x_n, y_n \,|\mathbf{a}\right) - \frac{\delta}{2}\left(4a_1a_3 - a_2^2 + a_1 + a_3\right), \quad (2)\end{aligned}$$

where $\mathbf{x} = \left[x_1 \; x_2 \; \cdots \; x_N\right]^T$ and $\mathbf{y} = \left[y_1 \; y_2 \; \cdots \; y_N\right]^T$ are the selected positions of the landmarks, N is the sample number, and δ is the hyper parameter used to regularize the optimization. By setting the gradient of the cost function to zero, $\nabla\Omega\left(\mathbf{a}\,|\mathbf{x}, \mathbf{y}\right) = 0$, the equation becomes

$$\left(\mathbf{D} - \mathbf{\Psi}\right)\mathbf{a} = \mathbf{\Lambda}, \quad (3)$$

$$\mathbf{D} = \sum_{i=1}^{N}\begin{bmatrix} x_i^4 & x_i^3y_i & x_i^2y_i^2 & x_i^3 & x_i^2y_i & x_i^2 \\ x_i^3y_i & x_i^2y_i^2 & x_iy_i^3 & x_i^2y_i & x_iy_i^2 & x_iy_i \\ x_i^2y_i^2 & x_iy_i^3 & y_i^4 & x_iy_i^2 & y_i^3 & y_i^2 \\ x_i^3 & x_i^2y_i & x_iy_i^2 & x_i^2 & x_iy_i & x_i \\ x_i^2y_i & x_iy_i^2 & y_i^3 & x_iy_i & y_i^2 & y_i \\ x_i^2 & x_iy_i & y_i^2 & x_i & y_i & 1 \end{bmatrix}, \quad (4)$$

$$\mathbf{\Psi} = \begin{bmatrix} 0 & 0 & \frac{2\delta}{N} & 0 & 0 & 0 \\ 0 & -\frac{\delta}{N} & 0 & 0 & 0 & 0 \\ \frac{2\delta}{N} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5)$$

$$\mathbf{\Lambda} = \begin{bmatrix} \frac{N\delta}{2} & 0 & \frac{N\delta}{2} & 0 & 0 & 0 \end{bmatrix}^T, \quad (6)$$

where $\mathbf{D} - \mathbf{\Psi}$ is a symmetric matrix, so $\left(\mathbf{D} - \mathbf{\Psi}\right)^{-1}$ exists if it is of full rank. Through $\mathbf{a} = \left(\mathbf{D} - \mathbf{\Psi}\right)^{-1}\mathbf{\Lambda}$, there is an analytical solution to find the ellipse.

The schematic diagram is shown in Fig. 7. Only the pixels in the ellipse and those lower than the highest landmarks of the eyebrows are kept, as Fig. 8 shows. The differences

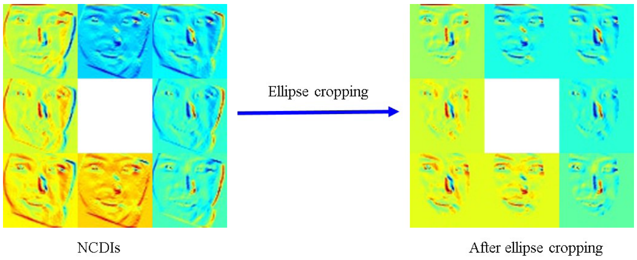

FIGURE 8. Ellipse cropping of NCDIs.

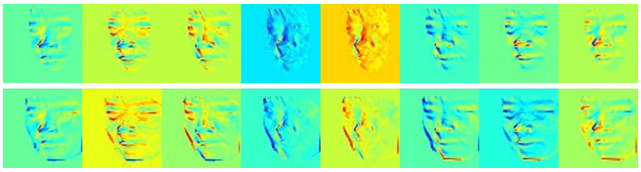

FIGURE 9. Comparison between ellipse cropping and polygon cropping. The images in the top row are the results of ellipse cropping NCDIs, and the images in the bottom row are the results of polygon cropping NCDIs.

between ellipse cropping and polygon cropping are shown in Fig. 9.

The pre-processing procedure follows the steps below. Detect the face and find the bounding box [25]. Resize the facial image into 64 × 64 pixels. Then, extract the landmarks on the face, and perform the spatial normalization and feature enhancement last.

IV. DEEP CONVOLUTIONAL NEURAL NETWORK

Based on Caffe framework [26], a CNN model is designed from the concept of [6] and [27]. There are parallel structures in the network to extract the features using different sizes of windows. The model consists of nine convolutional layers, two max pooling layers, one mean pooling layer, three fully connected layers, and a Local Response Normalization (LRN) layer. The activation functions are all set to rectified linear functions. Other details and the configuration of the model are shown in Fig. 10.

Like [12], the output of Full connection layer (L12) is regarded as the encoded features. The combination of Full connection layer (L13) and Softmax output layer (L14) is regarded as a classifier. Except for the classifier, the whole structure is a feature extractor for the input image. The Convolutional Feature Extractor (CFE) is defined as from Convolution layer (L1) to Mean pooling layer (L10) while the Fully Connected Feature Extractor (FCFE) is defined as from Full connection layer (L11) to Full connection layer (L12).

V. ADAPTIVE FEATURE MAPPING

This section addresses the design principle and the mechanism of AFM. In the following description, the trained and testing data sets are denoted as $\mathbf{X}^s = \left[\mathbf{x}_1^s\ \mathbf{x}_2^s \cdots \mathbf{x}_{N_s}^s\right]$ and $\mathbf{X}^t = \left[\mathbf{x}_1^t\ \mathbf{x}_2^t \cdots \mathbf{x}_{N_t}^t\right]$ respectively, and N_s is the number of trained samples while N_t is the batch size of the testing samples. The feature extractor consists of CFE and FCFE, and it is denoted as $\mathbf{h}(\mathbf{x}\,|\mathbf{W})$. $\mathbf{W}$ is the parameter set of the whole feature extractor while $\mathbf{x}$ is the input sample. In this research, $\mathbf{x}$ is the 8-channel NCDIs.

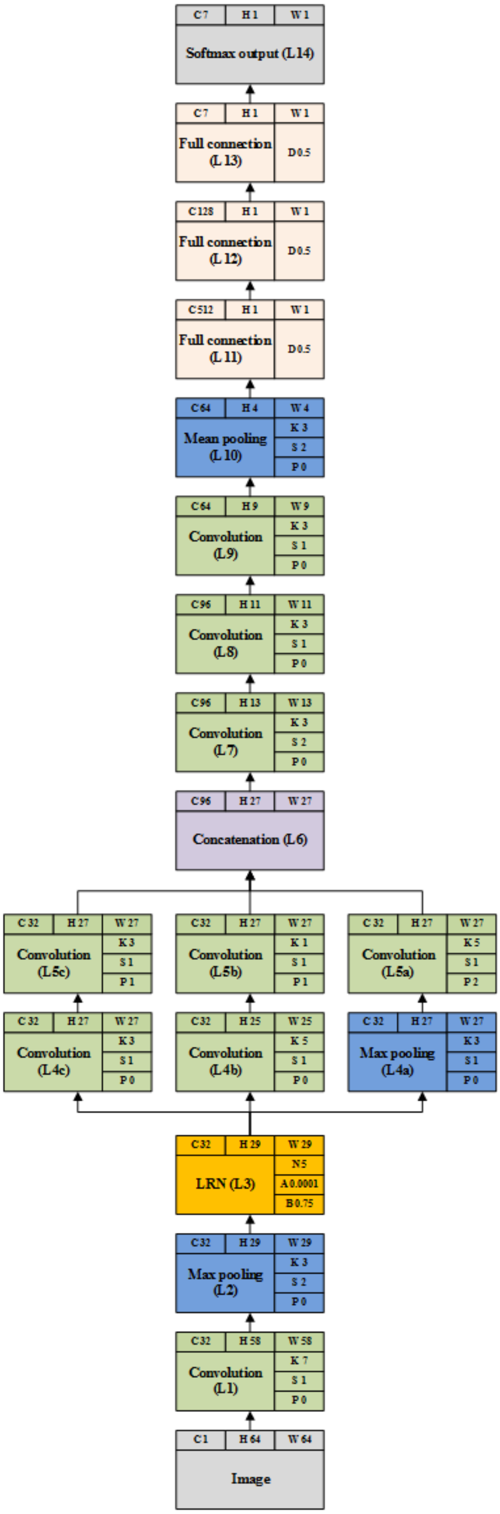

FIGURE 10. Convolutional neural network architecture. C, H, W are the output channel, height, and weight respectively. K, S, P stand for the kernel size, stride, and patch of convolution or pooling. N is the local size while A and B are the scale and the exponent parameters of LRN. D is the dropout ratio.

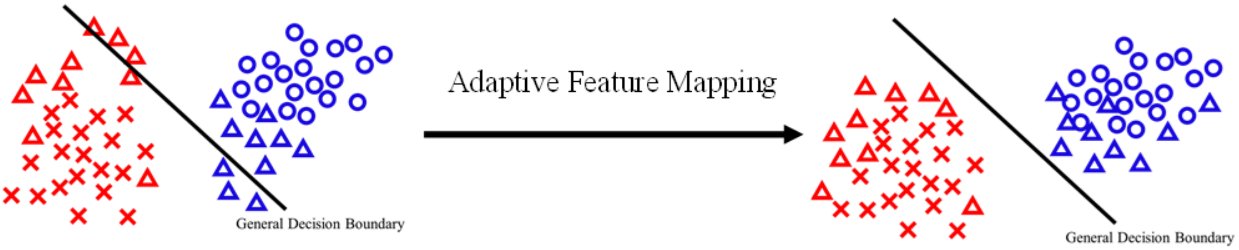

FIGURE 11. The purpose of AFM. There are red and blue categories in the example. The crosses and circles are the trained data. The triangles are the new testing subjects, and the color on each subject represents the ground truth.

A. COST FUNCTION

The main purpose of AFM is to tune the parameters of the feature extractor for the testing samples so that the tuned feature extractor can make the feature distribution of the testing samples similar to that of the trained samples. See Fig. 11. That is, $p\left(\mathbf{h}\left(\mathbf{x}^t\left|\tilde{\mathbf{W}}\right.\right)\right) \approx p\left(\mathbf{h}\left(\mathbf{x}^s\left|\mathbf{W}\right.\right)\right)$, where $\mathbf{W}$ is the generic parameter set, and $\tilde{\mathbf{W}}$ is the new parameter set for the testing samples. To accomplish this, the discrepancy between the means of the trained and testing samples must be minimized. According to [19], [20], and [22], the cost function can be written as

$$\begin{aligned}E&\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t,\mathbf{X}^s\right.\right)\\&=\left\|\frac{1}{N_t}\sum_{i=1}^{N_t}\varphi\left(\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)\right)-\frac{1}{N_s}\sum_{j=1}^{N_s}\varphi\left(\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)\right)\right\|_{\mathrm{H}}^2,\end{aligned}\tag{7}$$

where H stands for RKHS which can be defined by a kernel, $k\left(\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right),\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)\right)=\varphi\left(\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)\right)^T\varphi\left(\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)\right)$, to describe the relation between $\mathbf{h}\left(\mathbf{x}_i^t\left|\mathbf{W}\right.\right)$ and $\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)$. The linear kernel is chosen for simplicity, so the kernel is $k\left(\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right),\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)\right)=\left\|\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)-\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)\right\|^2$. Based on [20], the cost function shall be

$$E\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t,\mathbf{X}^s\right.\right)=Tr\left(\mathbf{KL}\right),\tag{8}$$

$$\mathbf{K}=\begin{bmatrix}\mathbf{K}_{s,s} & \mathbf{K}_{s,t}\\ \mathbf{K}_{t,s} & \mathbf{K}_{t,t}\end{bmatrix},\tag{9}$$

$$\mathbf{K}_{s,s}=\begin{bmatrix}k\left(\mathbf{h}_1^s,\mathbf{h}_1^s\right) & k\left(\mathbf{h}_1^s,\mathbf{h}_2^s\right) & \cdots & k\left(\mathbf{h}_1^s,\mathbf{h}_{N_s}^s\right)\\ k\left(\mathbf{h}_2^s,\mathbf{h}_1^s\right) & k\left(\mathbf{h}_2^s,\mathbf{h}_2^s\right) & \cdots & k\left(\mathbf{h}_2^s,\mathbf{h}_{N_s}^s\right)\\ \vdots & \vdots & \ddots & \vdots\\ k\left(\mathbf{h}_{N_s}^s,\mathbf{h}_1^s\right) & k\left(\mathbf{h}_{N_s}^s,\mathbf{h}_2^s\right) & \cdots & k\left(\mathbf{h}_{N_s}^s,\mathbf{h}_{N_s}^s\right)\end{bmatrix},\tag{10}$$

$$\mathbf{K}_{t,t}=\begin{bmatrix}k\left(\mathbf{h}_1^t,\mathbf{h}_1^t\right) & k\left(\mathbf{h}_1^t,\mathbf{h}_2^t\right) & \cdots & k\left(\mathbf{h}_1^t,\mathbf{h}_{N_t}^t\right)\\ k\left(\mathbf{h}_2^t,\mathbf{h}_1^t\right) & k\left(\mathbf{h}_2^t,\mathbf{h}_2^t\right) & \cdots & k\left(\mathbf{h}_2^t,\mathbf{h}_{N_t}^t\right)\\ \vdots & \vdots & \ddots & \vdots\\ k\left(\mathbf{h}_{N_t}^t,\mathbf{h}_1^t\right) & k\left(\mathbf{h}_{N_t}^t,\mathbf{h}_2^t\right) & \cdots & k\left(\mathbf{h}_{N_t}^t,\mathbf{h}_{N_t}^t\right)\end{bmatrix},\tag{11}$$

$$\mathbf{K}_{s,t}=\begin{bmatrix}k\left(\mathbf{h}_1^s,\mathbf{h}_1^t\right) & k\left(\mathbf{h}_1^s,\mathbf{h}_2^t\right) & \cdots & k\left(\mathbf{h}_1^s,\mathbf{h}_{N_t}^t\right)\\ k\left(\mathbf{h}_2^s,\mathbf{h}_1^t\right) & k\left(\mathbf{h}_2^s,\mathbf{h}_2^t\right) & \cdots & k\left(\mathbf{h}_2^s,\mathbf{h}_{N_t}^t\right)\\ \vdots & \vdots & \ddots & \vdots\\ k\left(\mathbf{h}_{N_s}^s,\mathbf{h}_1^t\right) & k\left(\mathbf{h}_{N_s}^s,\mathbf{h}_2^t\right) & \cdots & k\left(\mathbf{h}_{N_s}^s,\mathbf{h}_{N_t}^t\right)\end{bmatrix},\tag{12}$$

$$\mathbf{K}_{t,s}=\mathbf{K}_{s,t}^T,\tag{13}$$

where $\mathbf{h}_i^t$ and $\mathbf{h}_j^s$ are the abbreviations of $\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)$ and $\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)$ respectively. The elements of the matrix $\mathbf{L}$ is $\frac{-1}{N_s^2}$ if $\mathbf{x}_i,\mathbf{x}_j\in\mathbf{X}^s$, else $\frac{-1}{N_t^2}$ if $\mathbf{x}_i,\mathbf{x}_j\in\mathbf{X}^t$, otherwise, $\frac{1}{N_sN_t}$. Since we are only concerned with the cross relations between the features of trained and testing samples, other terms can be eliminated. Therefore, the cost function can be modified as

$$E\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t,\mathbf{X}^s\right.\right)=\sum_{i=1}^{N_t}\sum_{j=1}^{N_s}\frac{\alpha_{i,j}}{N_sN_t}\left\|\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)-\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)\right\|^2,\tag{14}$$

where $\alpha_{i,j}$ are the weights to represent the relevance between $\mathbf{h}\left(\mathbf{x}_i^t\left|\mathbf{W}\right.\right)$ and $\mathbf{h}\left(\mathbf{x}_j^s\left|\mathbf{W}\right.\right)$. A large $\alpha_{i,j}$ value is required when the features of trained and testing samples are relevant, i.e. error becomes small. On the contrary, a smaller $\alpha_{i,j}$ value is required when the features of trained and testing samples are less relevant, i.e. error becomes larger. If $\alpha_{i,j}$ is not appropriate, the cost function will oscillate drastically and may not converge during the training process. Moreover, the computational complexity will increase as the number of trained samples increases. The winner-take-all strategy, therefore, is considered, so the cost function can be re-written as

$$\begin{aligned}E&\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t,\mathbf{X}^s\right.\right)\\&=\frac{1}{2N_t}\sum_{i=1}^{N_t}\left\|\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)-\mathbf{h}\left(\mathbf{x}_r^s\left|\mathbf{W}\right.\right)\right\|^2,\end{aligned}\tag{15}$$

$$r=\arg\min\left(\left\|\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)-\mathbf{h}\left(\mathbf{x}_r^s\left|\mathbf{W}\right.\right)\right\|^2\right),\tag{16}$$

where r is the index of the trained sample which is most relevant to the i^{th} new sample in distance measure, and $r\in[1,N_s]$. By minimizing the nearest trained sample, the oscillation is damped, and the computational complexity is reduced.

Since some bad samples may limit the performance of AFM, the probability distribution on the output of the classifier can be taken into consideration to regularize the cost function. By simply multiplying the prediction confidence, the cost function can be written as

$$E\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t, \mathbf{X}^s, \mathbf{y}^s\right.\right) = \frac{1}{2N_t}\sum_{i=1}^{N_t} f_{y_r^s}\left(\mathbf{h}\left(\mathbf{x}_r^s\left|\mathbf{W}\right.\right)\right)\cdot\left\|\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)-\mathbf{h}\left(\mathbf{x}_r^s\left|\mathbf{W}\right.\right)\right\|^2, \tag{17}$$

where the entries of $\mathbf{y}^s = \left[y_1^s\ y_2^s \cdots y_{N_s}^s\right]^T$ are the labels of the trained samples. $\mathbf{f}(\cdot) = \left[f_1\ f_2 \cdots f_{N_k}\right]^T$ is the classifier, and N_k is the number of categories. This form of AFM is defined as Weighted Adaptive Feature Mapping (W-AFM).

The classifier in CNN is a linear transformation. After a CNN model is well trained, the features extracted by the model must be linearly separable. Therefore, there must be a unique center in each category. If the centers of the categories are considered, the person-specific bias can be mitigated further. The cost function, then, can be modified as

$$E\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t, \mathbf{X}^s, \mathbf{y}^s\right.\right) = \frac{1}{2N_t}\sum_{i=1}^{N_t} f_{y_r^s}\left(\mathbf{h}\left(\mathbf{x}_r^s\left|\mathbf{W}\right.\right)\right)\cdot\left\|\mathbf{h}\left(\mathbf{x}_i^t\left|\tilde{\mathbf{W}}\right.\right)-\mathbf{h}_{y_r^s}^c\right\|^2, \tag{18}$$

where $\mathbf{h}_{y_r^s}^c$ stands for the feature center of category y_r^s, and such form of AFM is regarded as Weighted Center Regression Adaptive Feature Mapping (W-CR-AFM).

These cost functions can be easily solved by the stochastic gradient descent which is written as follow:

$$\tilde{\mathbf{W}}^{k+1} = \tilde{\mathbf{W}}^k - \eta\cdot\left[\nabla E\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t, \mathbf{X}^s, \mathbf{y}^s\right.\right) + \lambda\tilde{\mathbf{W}}^k\right], \tag{19}$$

where η is the learning rate, and λ is the regularizing factor to prevent the parameters from going out of bound. $\nabla E\left(\tilde{\mathbf{W}}\left|\mathbf{X}^t, \mathbf{X}^s, \mathbf{y}^s\right.\right)$ is the gradient of the cost function.

B. SYSTEM OPERATION

After training the CNN model, the extracted features of training samples shall be stored as the feature database. In the testing phase, AFM can tune the weights based on the relationship between features of testing samples and the feature database in order to transform the features of testing samples into a new space so that its distribution can be similar to that of the feature database. Most of the parameters are distributed in the fully connected layers, so AFM is only applied to tune FCFE for higher efficiency. See Fig. 12. The premise of AFM is that the feature distribution of the testing samples is assumed to be similar to that of the training samples. The features around the decision boundary, therefore, shall be moved to the centers of categories. This way, the misclassified labels can be corrected. In addition, misclassified trained samples must be removed in advance so that the newly mapped features can be better. To make it more reliable, the testing samples with lower confidence of prediction can be ignored.

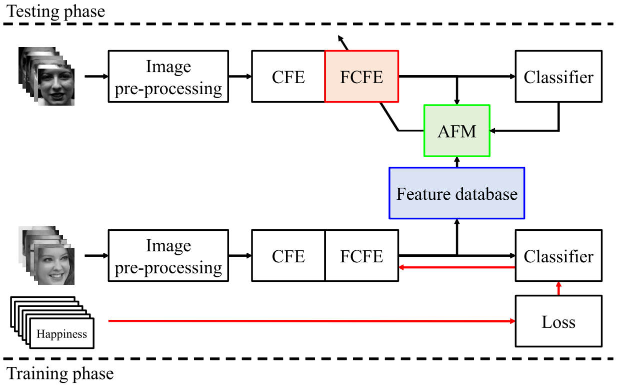

FIGURE 12. System architecture.

TABLE 2. Image pre-processing comparison.

Pre-processing	Accuracy (%) CK+	RaFD	ADFES
None	82.22	90.26	85.59
SN	81.75	94.16	83.10
SN + FE	**86.83**	**95.78**	**87.37**

SN stands for spatial normalization while FE stands for feature enhancement.

VI. EXPERIMENTS

A. IMAGE PRE-PROCESSING COMPARISON

TABLE 2 shows the results of the proposed model with different pre-processing methods. As can be seen in TABLE 2, the spatial normalization does not always seem to help the recognition accuracy since the edges of bounding box may appear and become the main feature of the image after spatial normalization is applied, which impairs the recognition function, causing the accuracy to be lower. Also, the model has been trained with many candid images from YouTube, so it can extract some features that are not affected by rotation. Thus, the recognition accuracy can be higher than when spatial normalization is applied. These may be the reasons why the spatial normalization appears ineffective in Table 2. The feature enhancement operation not only makes the facial edges more distinct but also removes the areas that are irrelevant to facial expressions, so the accuracy can be increased by about 4.61% in CK+, 5.52% in RaFD, and 1.78% in ADFES. These results demonstrate that the proposed pre-processing method is really effective.

B. THE EFFECT OF ADAPTIVE FEATURE MAPPING

The proposed CNN model with spatial normalization and feature enhancement is regarded as our Generic Model (GM). As for AFM, the learning rate η is set to 0.001 while the regularizing factor λ is set to 0.0005. The training iteration

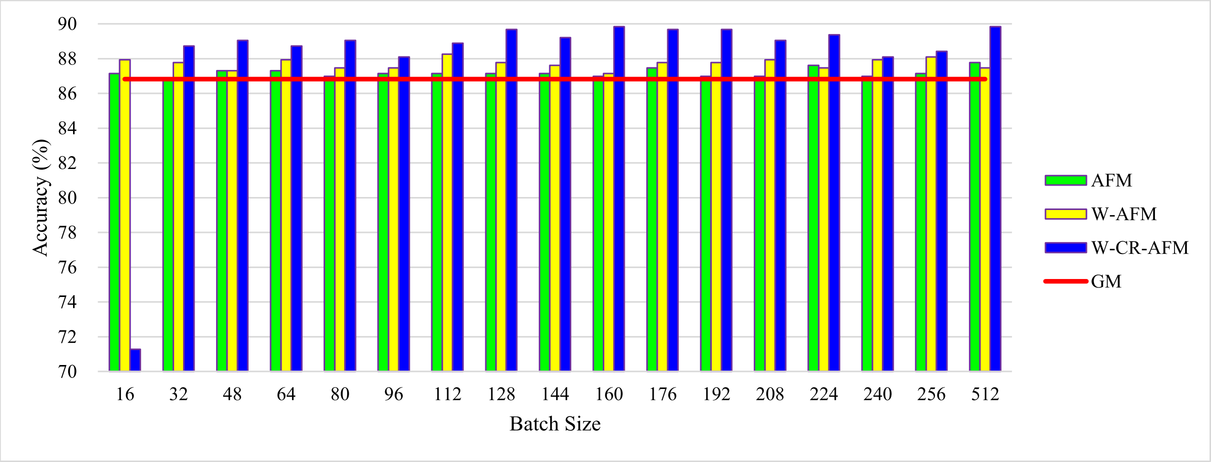

FIGURE 13. The improved accuracy of CK+ in different batch size.

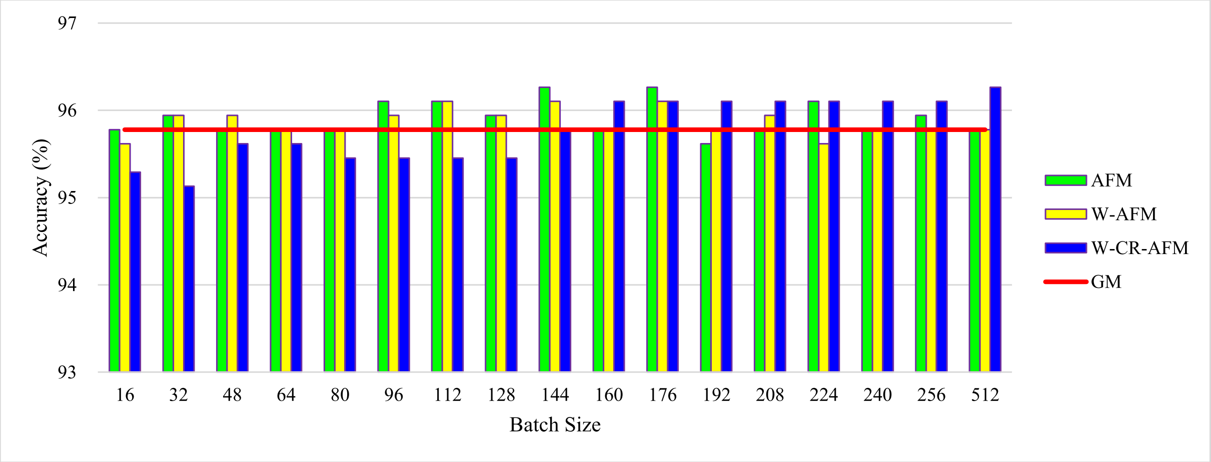

FIGURE 14. The improved accuracy of RaFD in different batch size.

is set to 1000. The batch size ranges from 16 to 512. The trained samples and testing samples are mirrored, and the trained samples which are misclassified should be removed in advance while the testing samples whose prediction confidence is lower than 90% are not taken into consideration when tuning the model with AFM. The results are shown in Fig. 13, Fig. 14, and Fig. 15. In most cases, W-AFM performs better than AFM, and W-CR-AFM is the best of all. The performance will be more stable when the batch size is large enough, otherwise the effect may be limited.

According to the experiments shown in Fig. 13, Fig. 14, and Fig. 15, the best result of each AFM is listed in TABLE 3. Based on the result of GM, the improved recognition accuracy is in TABLE 4. The category of happiness can always be predicted correctly in these three databases because its feature is obvious and its training data is ample. After applying AFM, most predicted labels are corrected. Compared to other categories, the number of images in anger, disgust and fear is less, so the ability of extracting features for these expressions is poor; hence, most features of testing samples do not fall into the center of the category, and will be drawn to other categories. Besides, the expression of anger is usually not explicit, so it is sometimes confused with neutral expression even if AFM or W-AFM is utilized. The main feature of surprise is the exaggerated mouth while the minor feature is the eyes, but the feature of eyes is difficult to extract properly because it varies greatly with the person. Sadeghi *et al.* [29] have proven that the mouth is the primary feature for facial expressions. However, some surprised faces do not clearly express the feature on the mouth in ADFES, so they are misclassified into neutral expression if AFM or W-AFM is applied.

For W-CR-AFM, since it minimizes the distance between the feature of the testing sample and the center of the most relevant category rather than the most relevant feature of the trained sample, the person-specific bias can be mitigated much more. Moreover, the feature distribution of neutral expression contains the largest area in the feature space,

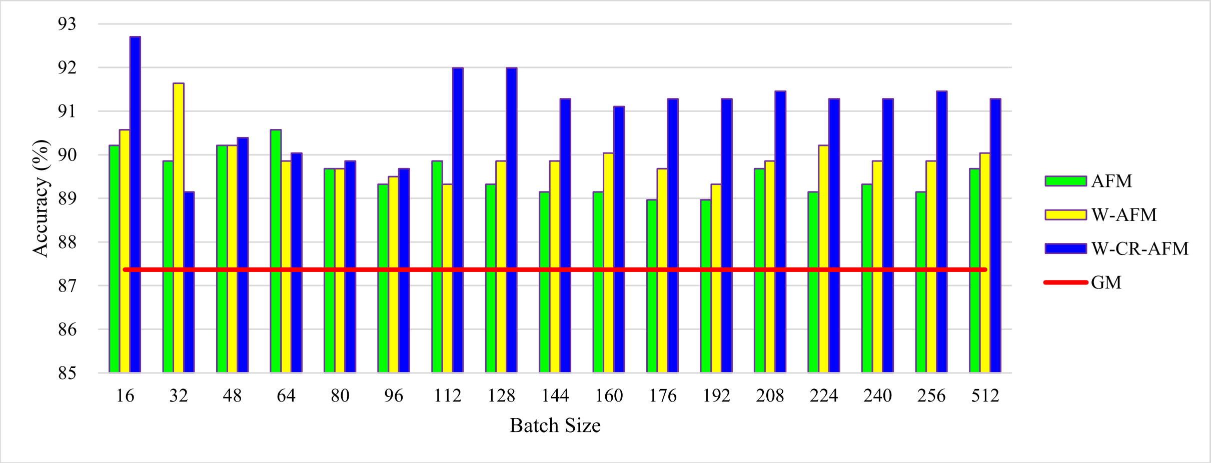

FIGURE 15. The improved accuracy of ADFES in different batch size.

TABLE 3. Accuracy in each database.

Database / Category	CK+ (%)				RaFD (%)				ADFES (%)			
	GM	**AFM**	**W-AFM**	**W-CR-AFM**	**GM**	**AFM**	**W-AFM**	**W-CR-AFM**	**GM**	**AFM**	**W-AFM**	**W-CR-AFM**
Anger	77.78	69.84	73.02	80.95	98.86	100.00	100.00	100.00	88.75	96.25	97.50	98.75
Disgust	69.23	70.33	73.63	91.21	97.78	98.89	98.89	100.00	88.75	88.75	92.50	100.00
Fear	42.50	52.50	52.50	77.50	87.36	85.06	85.06	88.51	71.25	68.75	68.75	72.50
Happiness	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Sadness	83.33	85.71	85.71	83.33	88.76	92.13	92.13	95.51	93.90	93.90	93.90	97.56
Surprise	94.67	96.67	96.67	100.00	98.80	98.80	98.80	100.00	97.50	90.00	92.50	100.00
Neutral	97.37	99.12	97.37	75.44	98.88	98.88	97.75	89.89	71.25	96.25	96.25	80.00
Total	**86.83**	**87.78**	**88.25**	**89.84**	**95.78**	**96.27**	**96.10**	**96.27**	**87.37**	**90.57**	**91.64**	**92.70**

TABLE 4. Improvement in each database.

Database / Category	CK+ (%)			RaFD (%)			ADFES (%)		
	AFM	**W-AFM**	**W-CR-AFM**	**AFM**	**W-AFM**	**W-CR-AFM**	**AFM**	**W-AFM**	**W-CR-AFM**
Anger	-7.94	-4.76	+3.17	+1.14	+1.14	+1.14	+7.50	+8.75	+10.00
Disgust	+1.10	+4.40	+21.98	+1.11	+1.11	+2.22	0.00	+3.75	+11.25
Fear	+10.00	+10.00	+35.00	-2.30	-2.30	+1.15	-2.50	-2.50	+1.25
Happiness	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Sadness	+2.38	+2.38	0.00	+3.37	+3.37	+6.75	0.00	0.00	+3.66
Surprise	+2.00	+2.00	+5.33	0.00	0.00	+1.20	-7.50	-5.00	+2.50
Neutral	+1.75	0.00	-21.93	0.00	-1.13	-8.99	+25.00	+25.00	+8.75
Total	**+0.95**	**+1.42**	**+3.01**	**+0.49**	**+0.32**	**+0.49**	**+3.20**	**+4.27**	**+5.33**

so the features of neutral expression that are far away from the center of neutral expression category will be brought to other categories. That is why the recognition accuracy of neutral expression is reduced after applying W-CR-AFM while accuracy of other expressions are raised.

According to the experiments, all three types of AFM can assist in improving the performance of a model in specific cases. For the overall recognition accuracy, W-CR-AFM works the best.

C. BENCHMARK COMPARISON

Some other deep learning approaches in facial expression recognition are introduced and compared with ours. They are trained with our training data for fair comparison.

TABLE 5. Benchmark comparison.

Approach	Accuracy (%) CK+	 RaFD	 ADFES
GoogLeNet [27]	85.71	95.45	86.48
AlexNet [28]	85.87	95.29	84.01
AlexNet + SVM [12]	86.83	95.13	88.43
CNN [7]	80.16	94.16	87.01
Our GM	86.83	95.78	87.37
Our GM + AFM	**87.78**	**96.27**	**90.57**
Our GM + W-AFM	**88.25**	**96.10**	**91.64**
Our GM + W-CR-AFM	**89.84**	**96.27**	**92.70**

To make GoogLeNet [27] and AlexNet [28] perform better, they have been trained with ImageNet previously. The second-to-last layer of the trained AlexNet is utilized to train a SVM [12]. To present the original performance of the competing models, the architectures and training parameters are set based on the original works. Since CK+ [2] is not included in the training data, it is reasonable that the recognition accuracy in TABLE 5 is lower than the results of state-of-the-arts. If the models are trained by CK+ [2], the recognition accuracy is expected to be much higher.

The results in TABLE 5 show that our approach performs better than others. The parameter quantities of GoogLeNet, AlexNet and the CNN designed by Li *et al.* [5] are around 40MB, 222MB, and 5MB respectively. Although our parameter quantity, around 3.5MB, which is much lower than others, the performance can be comparable to these state-of-the-art architectures through the use of proposed pre-processing method. Besides, AFMs can adapt the testing samples so that the model can perform better than other approaches.

VII. CONCLUSION

Two main contributions are presented in this paper. One contribution is that the proposed pre-processing method can assist the CNN model to gain the higher accuracy rate in the applications of facial image processing. The other contribution is that three types of AFMs can reformulate the features of new samples which do not have label information so that some misclassified samples can be corrected, which means it can tune a generic model to adapt to a specific condition. Moreover, AFMs can be deployed to real-time systems since it learns batch by batch rather than calculating all of the training and testing data in one batch. The concept drift problem is restrained because AFMs map the features of the testing samples to a static feature distribution. With the pre-processing and AFMs, a light CNN can outperform the state-of-the-art architectures.

REFERENCES

[1] R. E. Jack, O. G. B. Garrod, H. Yu, R. Caldara, and P. G. Schyns, "Facial expressions of emotion are not culturally universal," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 19, pp. 7241–7244, 2012.

[2] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn–Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, 2010, pp. 94–101.

[3] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg, "Presentation and validation of the radboud faces database," *Cognit. Emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[4] J. van der Schalk, S. T. Hawk, A. H. Fischer, and B. J. Doosje, "Moving faces, looking places: Validation of the Amsterdam dynamic facial expression set (ADFES)," *Emotion*, vol. 11, no. 4, pp. 907–920, 2011.

[5] W. Li, M. Li, Z. Su, and Z. Zhu, "A deep-learning approach to facial expression recognition with candid images," in *Proc. IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, Tokyo, Japan, May 2015, pp. 279–282.

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

[7] H. Jung *et al.*, "Development of deep learning-based facial expression recognition system," in *Proc. 21st Korea-Jpn. Joint Workshop Frontiers Comput. Vis. (FCV)*, 2015, pp. 1–4.

[8] A. Mollahosseini, B. Hassani, M. J. Salvador, H. Abdollahi, D. Chan, and M. H. Mahoor, "Facial expression recognition from World Wide Web," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2016, pp. 59–65.

[9] X. Peng, Z. Xia, L. Li, and X. Feng, "Towards facial expression recognition in the wild: A new database and deep recognition system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun./Jul. 2016, pp. 93–99.

[10] A. T. Lopes, E. de Aguiar, and T. Oliveira-Santos, "A facial expression recognition system using convolutional networks," in *Proc. 28th SIBGRAPI Conf. Graph., Patterns Images*, 2015, pp. 273–280.

[11] G. Levi and T. Hassner, "Emotion recognition in the wild via convolutional neural networks and mapped binary patterns," in *Proc. ACM Int. Conf. Multimodal Interaction*, 2015, pp. 503–510.

[12] D. M. Vo and T. H. Le, "Deep generic features and SVM for facial expression recognition," in *Proc. 3rd Nat. Found. Sci. Technol. Develop. Conf. Inf. Comput. Sci. (NICS)*, 2016, pp. 80–84.

[13] D. Hamester, P. Barros, and S. Wermter, "Face expression recognition with a 2-channel convolutional neural network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2015, pp. 1–8.

[14] F. Zhang, Y. Yu, Q. Mao, J. Gou, and Y. Zhan, "Pose-robust feature learning for facial expression recognition," *J. Frontiers Comput. Sci.*, vol. 10, no. 5, pp. 832–844, 2016.

[15] Y. Guo, D. Tao, J. Yu, H. Xiong, Y. Li, and D. Tao, "Deep neural networks with relativity learning for facial expression recognition," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–6.

[16] T. Zhang, W. Zheng, Z. Cui, Y. Zong, J. Yan, and K. Yan, "A deep neural network-driven feature learning method for multi-view facial expression recognition," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2528–2536, Dec. 2016.

[17] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2983–2991.

[18] Y.-H. Byeon and K.-C. Kwak, "Facial expression recognition using 3D convolutional neural network," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 12, pp. 107–112, 2014.

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

[20] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[21] T. M. H. Hsu, W. Y. Chen, C.-A. Hou, Y.-H. H. Tsai, Y.-R. Yeh, and Y.-C. F. Wang, "Unsupervised domain adaptation with imbalanced cross-domain data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4121–4129.

[22] W.-S. Chu, F. De la Torre, and J. F. Cohn, "Selective transfer machine for personalized facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 529–545, Mar. 2017.

[23] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1867–1874.

[24] J. Lu, V. E. Liong, and J. Zhou, "Cost-sensitive local binary feature learning for facial age estimation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5356–5368, Dec. 2015.

[25] D. E. King. (2015). "Max-margin object detection." [Online]. Available: https://arxiv.org/abs/1502.00046

[26] Y. Q. Jia *et al.* (2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: https://arxiv.org/abs/1408.5093

[27] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[29] H. Sadeghi, A.-A. Raie, and M.-R. Mohammadi, "Facial expression recognition using geometric normalization and appearance representation," in *Proc. 8th Iranian Conf. Mach. Vis. Image Process. (MVIP)*, 2013, pp. 159–163.

[30] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 958–963.

[31] D.-C. He and L. Wang, "Texture unit, texture spectrum, and texture analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 28, no. 4, pp. 509–512, Jul. 1990.

[32] Y. Ding, Q. Zhao, B. Li, and X. Yuan, "Facial expression recognition from image sequence based on LBP and Taylor expansion," *IEEE Access*, vol. 5, pp. 19409–19419, 2017.

[33] B. Ryu, A. R. Rivera, J. Kim, and O. Chae, "Local directional ternary pattern for facial expression recognition," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 6006–6018, Dec. 2017.

[34] Q. Mao, Q. Rao, Y. Yu, and M. Dong, "Hierarchical Bayesian theme models for multipose facial expression recognition," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 861–873, Apr. 2017.

[35] Z. Meng, P. Liu, J. Cai, S. Han, and Y. Tong, "Identity-aware convolutional neural network for facial expression recognition," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit.*, May/Jun. 2017, pp. 558–565.

[36] C. Zhang, P. Wang, K. Chen, and J.-K. Kämäräinen, "Identity-aware convolutional neural networks for facial expression recognition," *J. Syst. Eng. Electron.*, vol. 28, no. 4, pp. 784–792, Aug. 2017.

BING-FEI WU (M'92–SM'02–F'12) received the B.S. and M.S. degrees in control engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1992. He joined the Institute of Electrical and Control Engineering, NCTU, as the Director, in 2011. Since 1992, he has been with the Department of Electrical and Computer Engineering, where he was promoted to a Professor in 1998 and a Distinguished Professor in 2010. His research interests include image recognition, vehicle driving safety and control, intelligent robotic systems, intelligent transportation systems, and multimedia signal analysis. He is a Fellow of the IET and CACS. He founded and served as the Chair for the Taipei Chapter of the IEEE Systems, Man, and Cybernetics Society (SMCS) in 2003. He has been serving as the Chair for the Technical Committee on Intelligent Transportation Systems of the IEEE SMCS since 2011. He is currently an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics: Systems and the Editor-in-Chief of the *International Journal of Computer Science and Artificial Intelligence*.

Dr. Wu received many research honors, including the Outstanding Research Award of the Ministry of Science and Technology, Taiwan, in 2015; the Technology Invention Award of the Y. Z. Hsu Scientific Award from the Y. Z. Hsu Foundation in 2014; the National Invention and Creation Award of the Ministry of Economic Affairs, Taiwan, in 2012 and 2013, respectively; the Outstanding Research Award of the Pan Wen Yuan Foundation in 2012; the Best Paper Award from the 12th International Conference on ITS Telecommunications in 2012; the Best Technology Transfer Contribution Award from the National Science Council, Taiwan, in 2012; and the Outstanding Automatic Control Engineering Award from the Chinese Automatic Control Society in 2007.

CHUN-HSIEN LIN (S'15) was born in Taipei, Taiwan. He received the B.S. degree in electrical and computer engineering and the M.S. degree in electrical and control engineering from National Chiao Tung University, Taiwan, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree. His research interests include computer vision, machine learning, neural networks, and control systems.