IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 31, 2019, accepted January 4, 2020, date of publication January 8, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964752

Expression Recognition Method Based on a Lightweight Convolutional Neural Network

GUANGZHE ZHAO[1], HANTING YANG[1], AND MIN YU[2]
[1]College of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
[2]Department of General Surgery, Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, Guangzhou 510080, China

Corresponding author: Min Yu (yumin@gdph.org.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61871021 and Grant 61531006.

ABSTRACT Effective emotion recognition algorithms can help machines better understand people and promote the development of human-computer interaction applications. In recent years, many research efforts have used benchmark expression data to train deep neural network models to achieve state-of-art results. These high-accuracy models usually contain hundreds of layers, so they require complex calculations and may not be suitable for real-world scenarios. This paper proposes a lightweight emotion recognition (LER) model to handle the latency problem under natural conditions. The three main contributions of this paper are as follows. 1) The LER model incorporates a densely connected convolution layer and model compression techniques into a framework that eliminates redundancy parameters. 2) Multichannel input is introduced in our work to preprocess the image data, which improves the learning ability of the model. 3) Experiments show that the proposed LER model has better performance on the FER2013 and FERPLUS datasets compared with other lightweight models. Compared with the VGG13 used in previous work, the LER model achieves higher accuracy and reduces the number of parameters by 97 times. Finally, the FERFIN dataset is created, which had fewer noise data and more accurate labels than the FERPLUS dataset.

INDEX TERMS Emotion recognition, convolutional neural network, lightweight.

I. INTRODUCTION

Emotion is the cognitive experience that human beings produce under intense psychological activities. It provides cues of potential affection by observing facial expressions when people communicate with each other [1]. Building a system that can automatically recognize emotion has tremendous significance and can be applied to many scenarios such as pain detection, patient care, driver alert systems and detection of false statements [2].

The fact that facial expression can be produced by electrical stimulation suggests that characterizing features in the face is the most effective way to analyze emotions [3]. The first research on emotion recognition was published in 1978 and tracked the position of the key points in a continuous set of face images [4]. Additionally, the facial motion coding system (FACS) which measured human facial movements by defining facial action units (AUs) was published [5]. The FACS the initial method that attempted to describe all states of the face. However, unadvanced preprocessing algorithms and low computational power limited its development. In 2000, Kanade and Cohn published a CK dataset, which contained hundreds of face sequences with variable postures and overcame this situation [6].

Early research focused on extracting handcrafted features, which involved prior knowledge from 2D images and can roughly be divided into geometry and appearance [7]. Geometric features are good at characterizing primary expressions by studying the correlation between the coordinates of facial landmarks. For example, Pantic *et al.* [8] detected limited facial landmarks with particle filtering and calculated the distance between them to measure the AUs. Comparatively, appearance features are good at finding subtle color and texture changes in the face by computing the mathematics of the intensity value of pixels [9]. A typical example is the Gabor filter which is a linear filter used for texture analysis. Bartlett *et al.* [10] convolved the input data with Gabor filters to obtain frequency and orientation representations for basic emotion recognition.

2D images are computationally convenient for extracting features, while 3D images contain more intrinsic information [11]. To supplement the depth information on the gray

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqing Zhang.

or RGB data, researchers focused on new modalities such as 3D data and heat figures. Yin *et al.* published a BU-3DFE database that contains 2,500 3D facial expressions of 100 subjects [12]. They claim that 3D models can handle large head rotations, subtle skin movements, and light changes more stably than 2D models. However, modeling of stereoscopic head portraits requires reliable software to achieve certain photorealistic effects including intensive numeric calculation [2]. Regarding the heat figure, the difficulty of interpretation makes it hard to extract facial information and often has to combine with the RGB dataset [11].

The primary challenge in methods utilizing hand-craft features is performance decline in naturalistic environments [13], which is caused by head pose variations, illumination, and occlusions. In contrast, data-driven methods have benefited from the development of discrete graphics and big data technologies. Recently, many excellent works have employed a machine learning (ML) algorithm to fulfill end-to-end tasks [14], [15]. Typical examples are SVMs [16], [17] and random forests [18], which are supervised algorithms and k-nearest neighbors [19], which reduce the dimensions of the input and is an unsupervised algorithm. Furthermore, as a branch of ML, deep learning, especially convolutional neural networks (CNN), led the majority task, such as classification and segmentation in the computer vision community. CNN is inspired by biological processes and uses relatively little preprocessing compared to other image classification algorithms. In addition, the shared weight architecture and translation invariance characteristics make it specialized in image recognition applications [20].

Although many emotion recognition methods have incorporated CNNs into their framework, the lack of labeled data and computational inefficiency are the two main problems to be considered [2]. In applications such as fatigue detection in driver assistant systems, real-time data processing is an inevitable requirement. Therefore, large-scale CNN could be useless in such a scenario. To take advantage of the high generalization the performance of the CNN and to apply a well-established model in practice, we demonstrate how this tradeoff can be realized by presenting a strategy based on a dense convolutional neural network that not only eliminates millions of parameters, but also achieves accuracy comparable to a large-scale CNN. DenseNet was designed by Huang *et al.* [21] and achieved state-of-art results in many benchmark image classification datasets. Of note, DenseNet can considerably decreased trainable parameters by feature reuse and a compression feature map produced by convolutional layers [22].

Inspired by their work, we propose a lightweight emotion recognition (LER) model that incorporats a densely connected convolution layer and model compression techniques into a framework that eliminates redundancy parameters. After preprocessing the image data with the multichannel input method, the LER model can achieve higher accuracy compared to previous work and reduce the parameters by 97 times. Finally, according to the distribution of the FER2013 and FERPLUS datasets, we created the FERFIN dataset by removing noise data and combining two similar categories. Details are illustrated in section 3.

II. RELATED WORK

A complete emotion recognition system must have three steps: face detection, face alignment and emotion recognition. The strategy used in each step is different, depending on the modality of the data. There are 2D, 3D and thermal data formations employed in current research community. The main focus of the present study is 2D images, since 3D models require complex computations and thermal images have many limitations, such as a lack of geometric information.

A. FACE DETECTION

The purpose of face detection is to identify faces in images and mark them for subsequent procedures. The marking method can be divided into two categories: a detection method that aims for the bounding box of the face and a segmentation method that specifies the outline with a binary label of the pixel.

Viola and Jones proposed that a cascade classifier applies over Haar-like features selected by AdaBoost and is still one of the most prevalent methods [23]. A Haar-like feature considers adjacent rectangular regions at a specific location and can be calculated in constant time for any size image. Although it has high efficiency, it cannot address occlusion and large posture variations. The linear support vector machine (SVM) to detect humans with histograms of gradients (HOG) is also a typical method [24]. The primary step divides an image into small connected regions and then obtain the distribution of intensity gradients or edge directions. Because the HOG descriptor operates on local cells, its invariance to geometry makes it suitable for human detection. Osadchy built a convolutional neural network model to map input images to points on the manifold to integrate face detection and pose estimation together. If sufficient data are available, it achieves remarkable accuracy on a variety of pose images [25].

B. FACE ALIGNMENT

The purpose of face alignment is to solve cases when a face is not frontal which can lead to inaccurate recognition results. The mainstream approach finds the facial landmarks based on the located face and then carries out rotation or deformation. The landmark numbers vary depending on how many sample points, which are used around eyes, nose, mouth and face contour.

Active appearance models (AAM) are the extension of active shape models (ASM) and attempt to construct a statistical model by learning the correlation between estimation of appearance and the target image [26]. The matching process is optimized by the least squares algorithm which is a standard regression analysis method. In addition, AAM takes advantage of extra texture information other than shape features. Matthias and Juergen proposed a real-time method

using conditional regression forests to learn intensity features from facial image patches [27]. A conditional model is a machine learning framework that augments learning with declarative constraints that incorporate prior expressive knowledge. Together with random forests which can alleviate the overfitting problem, they can effectively improve accuracy. A recently suggested method utilizing the ensemble of regression trees optimized by gradient boosting to locate the facial landmarks processes pictures at the millisecond-level [28]. In essence, the regression tree is a decision tree with continuous target values that can achieve better predictive performance by ensemble learning techniques.

C. EMOTION RECOGNITION

Emotion recognition strategies depend on two factors: the definition of facial expressions and extracting features that can be divided into handcrafted and learned.

1) DEFINITION OF FACIAL EXPRESSIONS

There are two methods for defining facial expressions, which are continuous and discrete. The continuous definition relies on FACS, where AU represents the contraction of one or more muscles in the face [5]. In this setting, researchers have attempted to detect the four phases of AU, which are neutral, onset, apex and offset. These four phases have time coherence and can represent the marking process from start to finish of AUs.

The discrete definition divides the facial expression space and generates the most basic expression. Early methods only identified six basic expressions: disgust, fear, happiness, surprise, sadness, and anger. Later, to find a more refined classification, researchers added more basic expressions. Discrete definitions are widely used in expression recognition research due to their universality and comprehensibility.

2) HAND-CRAFTED FEATURES

Geometry and appearance are the two main classes under the categories. Geometric features measure the distance, curvature, and deformation based on the facial reference points found in the image. Appearance features capture specific information by analyzing the relations of pixels.

Optical flow is the pattern of apparent motion and focuses on the distribution of velocities of movement of brightness patterns in an image. Some works used optical flow to detect AUs and recognized corresponding primary emotions. While AUs are robust to background changes, they are susceptible to intense light. Pantic and Patras [8] proposed a particle filter to track the position of 15 feature points of the face, and automatically recognize the action units (AUs) in the face contour according to the change in distance. Sandbach and Zafeiriou [29] proposed a local normal binary mode to recognize expressions by calculating the 2D-characterized local binary pattern (LBP) features extracted from the 3D image information.

Dhall *et al.* [30] used PHOG (pyramid of histogram of gradient) features and LPQ (local phase quantization) features to describe facial appearance and shape. The PHOG feature is an improvement of the HOG feature. It statistically analyzes the edge image direction gradient histogram at different levels leading to strong antinoise performance and certain anti-rotation ability, but is subject to layering rules and lacks scale adaptability. Littlewor [10] used Gabor filters to extract image features that take advantage of Gabor wavelet characteristics in processing texture and discrimination features and illumination invariance and posture invariance, but the disadvantage is that the calculations are complex and require time to go through Gaussian kernel function modulation and other steps.

3) LEARNED FEATURES

Handcrafted features involve a large amount of prior knowledge and are difficult to modify, so researchers have turned their attention to end-to-end learning methods. These methods use a large amount of labeling data for supervised learning, mainly based on convolutional neural networks, which are good at processing image data [31], [32] and utilizing the characteristics of local receptive fields and are similar to the way that human eyes observe things. Recursive neural networks consider additional timing information whose variant version can retain important information and abandon unwanted information [33], [34].

Because of the advancement of big data, the aforementioned method with strong data dependence has occupied most of the visual field problems, and researchers have continued to expand the depth and width of the network architecture to obtain better results. However, there are two obvious shortcomings. One is that the upper limit of neural network performance depends on the reliability of the labeler. If the label is wrong, then the model that learns from it cannot achieve high accuracy. Second, large-scale networks need thousands of trainable parameters which means that it is not feasible to apply it to practical applications.

In summary, the handcrafted features have the advantage of low-complexity and fast calculation speed, but they require prior knowledge and have poor generalization ability. Learned features referring to the deep-learning method can handle large-scale variance, but as the model architecture increases, the flops increase exponentially. We tried to seek an approach that can address head and background diversity means while having computational efficiency. Therefore, we adopted the dense convolutional neural networks that employ many parameter compression layers to reduce the model complexity and still in a data-driven way. Details are illustrated in the next section.

III. PROPOSED EMOTION RECOGNITION FRAMEWORK

In this section, we briefly introduce the face detection and face alignment pipeline and focus on illustrating our DenseNet model with different hyperparameter deployment.

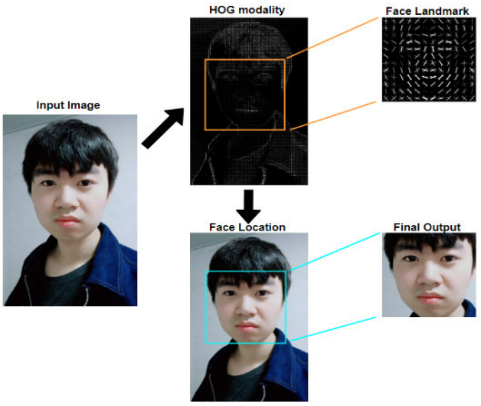


FIGURE 1. Face detection based on HOG features.

FIGURE 2. Face alignment based on landmarks.

A. FACE DETECTION AND ALIGNMENT

In the face detection part, we used the SVM method applied over HOG features [24] which constructed feature vectors by calculating the histograms of gradient of the local regions of the image and then put them into the classifier. If the result was positive, it returned the position of the detection area which is the coordinates of the upper left corner of the bounding box (X_l, Y_l) and the coordinates of the lower right corner (X_r, Y_r). This method can achieve better balances in terms of accuracy and speed compared with other methods and is more suitable for online identification applications. Details of computing the gradients of pixels in the image are shown in Eq.1, where m and θ are the magnitude and direction, respectively.

$$\begin{aligned} f_x(x,y) &= f(x+1,y) - f(x-1,y) \\ f_y(x,y) &= f(x,y+1) - f(x,y-1) \\ m(x,y) &= \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \\ \theta(x,y) &= arctan(f_x(x,y)/f_y(x,y)) \end{aligned} \tag{1}$$

In the face aliment part, we used the millisecond ensemble method proposed in [28] to train several regression trees using gradient boosting, and then regressed the 68 landmark points include eyes contour, bridge of the nose and mouth contour, by the ensemble of decision trees. Figure.1 and Figure.2 illustrate the face detection and alignment process in the proposed system.

B. LIGHTWEIGHT EMOTION RECOGNITION MODEL

DenseNet is a unique convolutional neural network(CNN) architecture that maximally reduces trainable parameters through an intensive connected pattern and many parameter reduction layers.

Unlike the depth expansion CNN architecture ResNet [35], which employs the identity function to extend the effective optimized distance and the width expansion CNN architecture Inception [36], which uses different sizes of convolution filters to perform feature extraction on different scales, DenseNet employs heavy feature reuse to allow any former layers' feature maps directly link to subsequent layers as shown in Figure.3.

In essence, DenseNet has two key hyperparameters: the growth rate k and dense block number n. The growth rate specifies the accumulated speed of the feature maps product by convolutional layers. For example, if the input data with m channels go through l convolutional layers, then the lth layer has $m + k(l-1)$ input feature maps. To conveniently understand various DenseNet architectures and adjust the hyperparameter flexibly, DenseNet sets another hyperparameter dense block.

In addition, the described convolutional layer includes not only the convolution calculation of the filtering window but also the activation function ReLU and batch normalization [37]. ReLU is a typical nonlinear activation function that maps the input signal into the feature space with the formula f(x) = max(0,x). Compared with the traditional sigmoid activation function, ReLU uses unilateral suppression mapping, which is more similar to the biological signal transmission process and has a broader excitation boundary and also has a significant effect in overcoming the disappearing gradient problem.

Additionally, ReLU deliberately shields a large number of input signals, which are reflected in the negative half-axis of the X-axis. This sparse activation is more suitable for extracting the sparse image features existing in the manifold so that it improves the precision and efficiency of learning. The purpose of batch normalization is to ensure that the input of each layer has zero mean and unit variance, which is originally derived from the initialization of the input layer

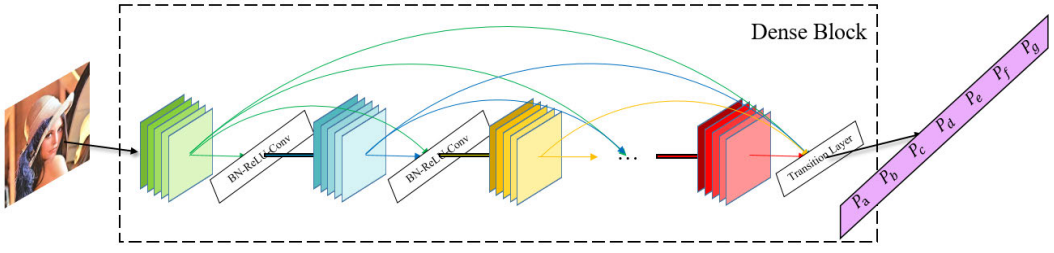


FIGURE 3. Architecture of DenseNet employed in emotion recognition.

and belongs to the network training skills, which speeds up the training of the network and adds a certain degree of regularization. The generalized calculation in the convolutional layer is shown in Eq.2.

$$\begin{aligned} f_1(x_i) &= max(0, x_i), \\ f_2(x_i) &= conv_{3*3}(f_i(x_i)), \\ f_3(x_i) &= \frac{f_2(x_i) - E[f_2(x_i)]}{\sqrt{Var[f_2(x_i)]}}, \\ F_{output} &= f_3([x_1, x_2, x_3, \ldots, x_{l-1}]) \end{aligned} \tag{2}$$

Rather than the middle of the dense block, the pooling layer sits between them. Along with the bottleneck and compression layer, they are called the transition layer. All the convolutional layer pad zero pixels around the input tensor before the convolution function so that the feature map sizes are consistent, as shown in Eq.3, where w and h are the width and height of the feature maps, F indicates the filter's size, s indicates the filter moving stride and p represents the zero padding pixels.

$$Size_{(w,h)} = \frac{w(h) - F + 2p}{s} + 1 \tag{3}$$

In a typical CNN architecture, the pooling layer is followed by every convolution layer to gradually subsample the tensor of weights. However, in the DenseNet architecture, the pooling layer sits between two dense blocks that fully utilize the feature extraction function of the convolution layer.

Specifically, we took the 2×2 average pooling rather than 2×2 max pooling as it is more suitable for the convolution structure by enforcing correspondences between feature maps and categories. The max pooling discards three-quarters of the information while average pooling considers all information. In addition, average pooling sums the spatial information; thus, it is more robust to spatial translations of the input. The mean normalization is actually a generalization function that can prevent the dense connection from falling into the overfitting problem. After average pooling, the feature map size is shown in Eq.4.

$$Size_{(w,h)} = \frac{w(h) - F}{s} + 1 \tag{4}$$

The idea of the bottleneck layer was first suggested in Lin's work [38], and they proposed a micro neural network with 1×1 convolution to enhance model discriminability for local patches. Furthermore, 1×1 convolution can compress the trainable parameters by setting fewer convolutional filters to reduce the model size.

The compression layer further improves the model compactness. As the final layer in the transition layer, hyperparameter θ decreases the feature maps generated by the dense block, where $0 \leq \theta \leq 1$ referrs to the compression factor.

C. THREE MINI-SIZE DENSENETS

As mentioned above, the DenseNet has some practical means for reducing the parameters. The growth rate and the number of convolutional layers in the dense block are the key elements that affect the size of the network model. Therefore, we designed three mini-sizes DenseNets to train a real-time emotion classifier with acceptable accuracy. The architecture details are shown in Table.1.

DenseNet-1 has three dense blocks with the a growth rate set to 12, and each block has 12 convolutional layers. DenseNet-2 has four dense blocks with the growth rate set to 16, and each block has 12 convolutional layers. DenseNet-3 has four dense blocks with the growth rate set to 12, and each block has 6, 12, 24, 16 convolutional layers separately inspired by the original work [21].

For the optimization algorithm, we used the Nesterov momentum optimization method [39], which based on the improvement of momentum. The momentum method is an improvement for the local minimum point oscillation problem in the optimization space for stochastic gradient descent.

It adds the weighted update vector generated by the previous iteration to the current update vector, as shown in Eq.5.

$$\begin{aligned} v_t &= \beta v_{t-1} + \alpha \nabla_\theta L(\theta) \\ \theta &= \theta - v_t \end{aligned} \tag{5}$$

This algorithm increases the momentum in the same direction as the gradient update while reducing the vibration in the direction of the gradient change, thus achieving a faster convergence rate. However, blindly following the gradient acceleration update also brings instability. The Nesterov momentum gives the approximate gradient trend information after the optimization function by calculating $\theta - \beta v_(t-1)$. If the gradient has an increasing trend, the update rate is speeded up. If the gradient has a decreasing trend, the update speed rate is slowed down, as shown in Eq.6. In essence, the second-order information of the loss function is introduced so that the optimization function has a predictive function in the optimization space and faster and more stable convergence.

$$\begin{aligned} v_t &= \beta v_{t-1} + \alpha \nabla_\theta L(\theta - \beta v_{t-1}) \\ \theta &= \theta - v_t \end{aligned} \tag{6}$$

IV. EXPRIMENT

In this section, we briefly introduce our experiment platform and specify our training dataset and results.

A. EXPERIMENT PLATFORM

Our model training processing was performed on a NVIDIA Titan X graphics card with 3,584 CUDA units, 12GB of GDDR5X memory, a core frequency of 1,531MHz, and the single-precision floating-point operation is 7.0 TFlops. We designed our algorithm based on Python3.6 and the TFlearn deep learning toolkit.

B. DATASET

The FER2013 dataset initially intercepted facial expression images from videos collected by the Kaggle team from the

TABLE 1. DenseNet architectures for FER2013, FERPLUS and FERFIN. The growth rate for DenseNet-1 and DenseNet-3 is k = 12, for DenseNet-2 is k = 16.

Layers	Output Size	DenseNet-1	DenseNet-2	DenseNet-3
Convolution	48 × 48	3 × 3 conv		
Dense Block	48 × 48	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 6$
Transition Layer	48 × 48	1 × 1 conv		
	24 × 24	2 × 2 average pool		
Dense Block	24 × 24	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$
Transition Layer	24 × 24	1 × 1 conv		
	12 × 12	2 × 2 average pool		
Dense Block	12 × 12	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 24$
Transition Layer	12 × 12	1 ∗ 1 conv		
	6 × 6	2 × 2 average pool		
Dense Block	6 × 6	/	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times 1 conv \\ 3\times 3 conv \end{bmatrix} \times 16$
Transition Layer	6 × 6	/	1 × 1 conv	
	3 × 3	/	2 × 2 average pool	
Classification Layer	1 × 1	6 × 6 global pool	3 × 3 global pool	3 × 3 global pool
		7D-Softmax	10D-Softmax	7D-Softmax

internet in 2013, which contains 35,887 gray images of 48 × 48 pixels, and used it as a challenge [40]. At the first publication, the dataset labels were divided into 7 categories, including 4,953 cases of ''anger'', 547 cases of ''disgust'', 5,121 cases of ''fear'', 8,989 cases of ''happy'', 6,077 cases of ''sadness'', 4,002 cases of surprise'' and ''neutral'' 6,198 cases.

However, the FER2013 labeling was later proven to be inaccurate due to the low performance of human labelers [41]. In this case, Barsoum *et al.* [42] used the crowdsourcing method to improve the accuracy of the labeler and added three categories of contempt, unknown and not a face. The improved dataset has 12,906 cases of ''neutral'', 9,355 cases of ''happy'', 4,462 cases of ''surprise'', 4,371 cases of ''sadness'', 3,111 cases of ''anger'', 248 cases of ''disgust'', 819 cases of ''fear'' 216 cases of ''contempt'', 222 cases of ''unknown'', and 177 cases of ''not a face''.

After careful observation, we found that ''not a face'' and ''unknown'' was quite rare compared to other classes and they may be noise in training the neural network. Therefore, we modified the dataset to remove these two classes. In addition, ''contempt'' and ''disgust'' classes only had 248 cases and 216 cases respectively. In fact, the sample space similarity between the two types was very high, and they easily interfered with each other. Therefore, the second modification we made was to combine these two classes. The final dataset was called FERFIN which, after the majority vote, contained 12,858 cases of ''neutral'', 9,354 cases of ''happy'', 4,462 cases of ''surprised'', 4,351 cases of ''sad'', 3,082 cases of ''angry'', 575 cases of ''disgust'' and 816 cases of ''fear''. A total of 35,498 cases eliminated 390 noise cases compared to the original FER2013 dataset.

C. RESULTS

As the statement in section 3, we employed the three DenseNet architecture training on the FER2013, FERPLUS and FERFIN datasets. The learning curves of the proposed three models on three datasets are shown in Figure.4. For discrete situations, the hyperparameter setting is slightly different. First, a 7-softmax layer or a 10-softmax layer followed after the final fully-connected layer depends on the class number of the dataset.

For other hyperparameters, the Nesterov momentum learning rate ε was set to 0.1, the momentum parameter α was 0.1, and the attenuation step was 15,000. The compression factor and the bottleneck reduction rate were set to 0.5. In terms of training strategy, we used the standard 10-crop data augmentation, which added four rows or columns of zero values around each image, and then intercepted the top left, top right, bottom left, bottom right and middle five tiles. Then, they were flipped left and right to double this value to ten. Batch normalization as an accepted method, was also used in the training process to ensure that the input of each layer had zero mean and unit variance.

On the FER2013 dataset, DenseNet-3 achieved validation accuracy of 71.73%, which surpassed the first team result in the challenge of 71.16% [41]. We believed that DenseNet-3 could achieve this result without using any ensemble method and a small number of parameters for two reasons. First, the feature reuse method increases the input size of subsequent convolutional layers, and makes the subsequent layers learnable while accepting the previous knowledge of the network. Second, dense connections and the setting of the bottleneck layer significantly reduces the parameters of the network, forcing it to extract more compact and discriminating features.

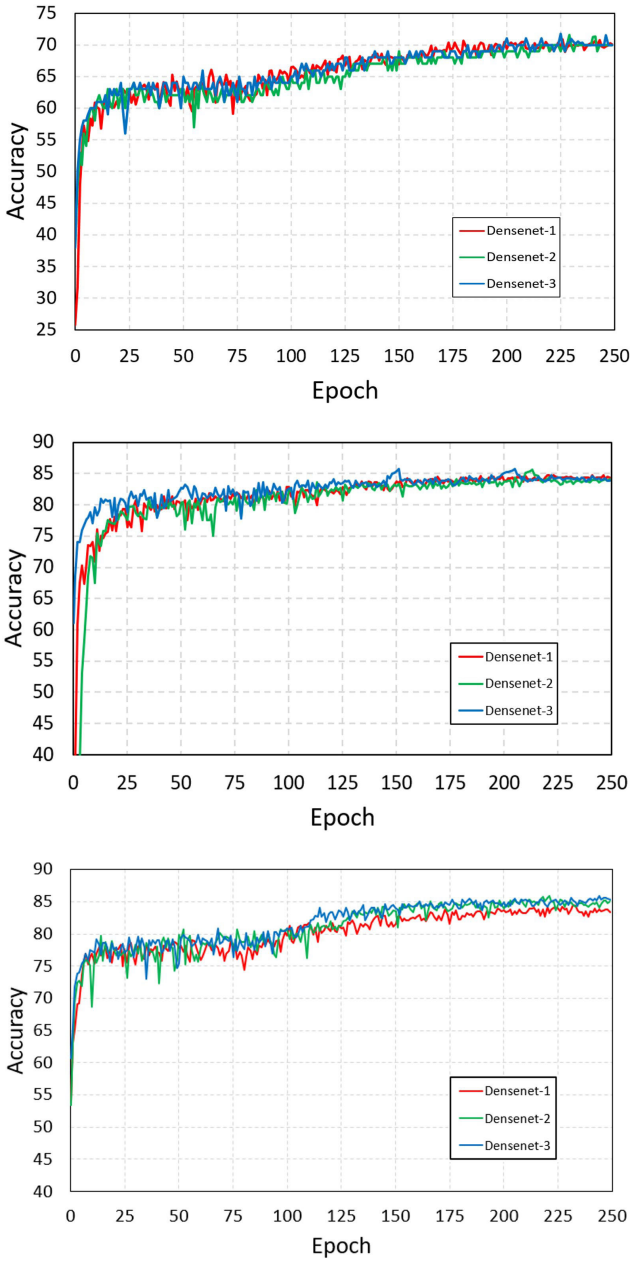


FIGURE 4. Learning curve on the FER2013, FERPLUS, and FERFIN dataset.

On the FERPLUS dataset, DenseNet-2 achieved a validation accuracy of 85.58%. It was 0.69% beyond the result of Barsoum's work [42], and DenseNet-2 only had 41 times fewer parameters than VGG13 employed. The number increased to 92 times when the DenseNet-1 architecture was employed losing 0.52% accuracy. Large-scale convolutional neural networks can achieve state-of-art results with a large amount of labeled data, but in fact, every additional small order of accuracy after reaching a particular value requires more network parameters. DenseNet minimizes the redundant parameters in the convolutional network and maximizes the representation ability of retained parameters. Therefore, our model can achieve a better balance in terms of accuracy and algorithm complexity. The feature map variation is illustrated in Figure.5.

TABLE 2. Experiment results.

Model	FER2013	FERPLUS	FERFIN
DenseNet-1	70.911%	84.06%	84.25%
DenseNet-2	71.55%	85.58%	**85.89%**
DenseNet-3	71.73%	85.67%	85.90%

On the FERFIN dataset, the same DenseNet-2 achieves a validation accuracy of 85.89% which supports our assumption of noise classes. Because the categories within the database are more distinct, DenseNet learned more robust representation features. The best results on each datasets of the model are presented in Table 2. The results of five trials on the FERFIN dataset for the three models are listed in Table 6.

D. COMPARISON WITH STATE-OF-THE-ART

We compared our proposed DenseNet model with other state-of-the-art methods with the original work which published the FERPLUS dataset in the following aspects: training method, model, validation accuracy and optimization function.

The original work adopted four different schemes to process the data with ten labels that determine the learning efficiency of the neural network from data. Since the aim was a tradeoff between accuracy and real-time capability, this paper only used the majority vote, which took the most frequent tag as the final label, to define the loss function. Comparisons with excellent methods were conducted to assess the advantages and disadvantages of our methods.

Specifically, the original work employed VGG13 with ten convolution layers and the dropout technique as the training model. According to the description, the VGG13 network has a total of 8.7 million trainable parameters and achieves an

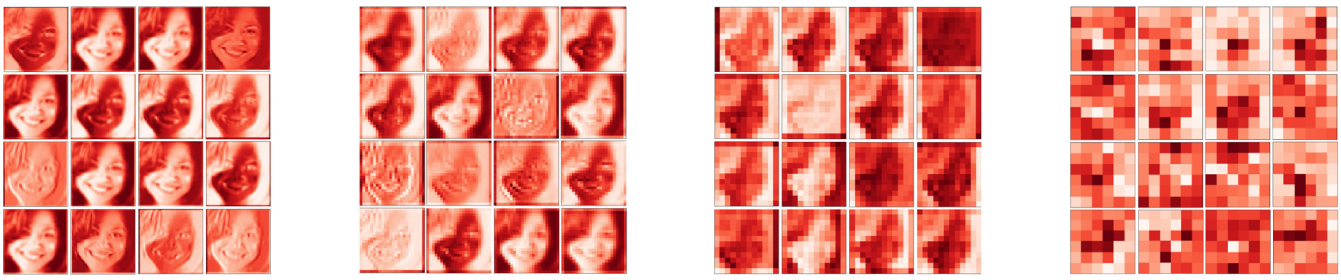


FIGURE 5. The level of features abstraction from low to high. From left to right represent the feature maps of (a) the first convolution layer, (b) DenseBlock1, (c) DenseBlock2, (d) DenseBlock3. All feature maps have 16 channels.

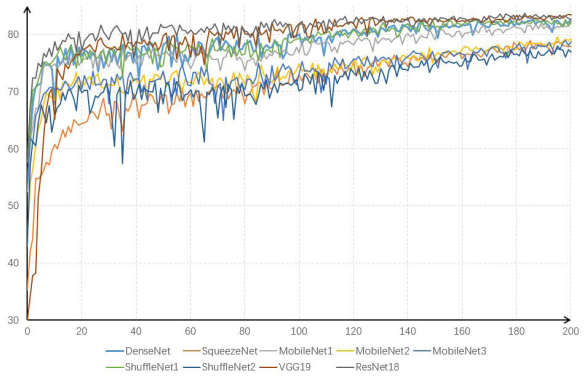


FIGURE 6. Learning curve comparison diagram, the best results are VGG19 and ResNet18 which are large-scale networks. The third blue curve is the proposed DenseNet, whose parameter is only 1/113 of VGG19.

average validation accuracy of 83.85%. The hyperparameter settings are shown in Table.3.

Comparatively, the three proposed DenseNet models require 0.09 million, 0.21 million and 0.17 million trainable parameters respectively. DenseNet-2 achieved the best average validation accuracy of 85.58%, and DenseNet-1 contains only 0.09 million parameters with a decrease in 0.52% accuracy.The learning curves of all models in the validation set are shown in Figure 4, and Figure 8 shows the degree of overfitting on the FERFIN dataset.

Finally, the original work utilized the standard gradient descent algorithm to optimize the neural network while the present paper used the Nesterov momentum. The Nesterov outperformed gradient descent, but was more suitable for DenseNet architecture with dense connections.

In addition to the VGG13 networks used by Barsoum *et al.*, we also tested the performance of other lightweight networks on FERPLUS datasets, such as SqueezeNet [43], MobileNet [44] and ShuffleNet [45]. Additionally, we utilized large-scale CNNs, such as ResNet and VGG19. The results suggested that the dense emotion

TABLE 3. Hyperparameter comparison between VGG13 of Barsoum's work [42] and proposed DenseNets.

	Conv	Pooling	FC	Parameters
VGG13	10	4	3	87566680
DenseNet-1	36	4	1	95263
DenseNet-2	48	5	1	218839
DenseNet-3	58	5	1	178939

TABLE 4. FLOPS represents the number of addition and multiplication operations required by the model to perform an input and output. CPU time represents the CPU run time required for a single input and output.

Model	Parameters	FLOPS	Accuracy	CPU time
DenseNet	0.17M	0.17B	84.285%	199.363ms
VGG13	9.41M	0.52B	84.369%	266.755ms
VGG19[46]	20.04M	0.90B	84.369%	287.343ms
ResNet18[47]	11.17M	1.26B	84.787%	292.355ms
SqueeezeNet[43]	0.74M	0.02B	80.134%	85.081ms
Mobilenet1	1.09M	0.03B	83.505%	240.771ms
Mobilenet2	2.23M	0.02B	81.443%	596.515ms
Mobilenet3[44]	3.88M	0.02B	81.666%	483.830ms
ShuffleNet1	0.86M	0.95B	84.062%	226.128ms
ShuffleNet2[45]	1.26M	0.01B	80.440%	213.504ms

neural network proposed in this paper achieved the best trade-off between accuracy and latency. Figure.6 and Table.4 show the learning curves and related parameters of each model.

E. DISCUSSION

As shown in Table.2 and Figure.7, the accuracy of our models is comparatively satisfactory in the corresponding datasets. We assumed that the success of the current models may be attributed to the following factors. First, the convolutional neural network is good at analyzing image data because the local receptive fields share the knowledge. Second, the unique architecture of the DenseNet and compressed layers minimize the parameters in the model. Third, correct and pure labeling data make the convergence process easily performed. Still, our method has shortcomings. From Table 5, it can be found that the model has a poor recognition accuracy for the

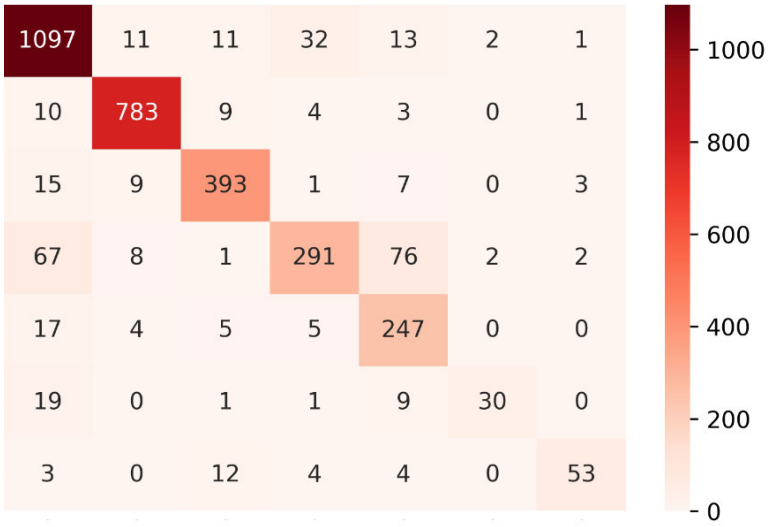


FIGURE 7. Confusion matrix of DenseNet-2 on the FERFIN dataset, the labels are in order of neutral, happiness, surprise, sadness, anger, disgust, fear.

TABLE 5. Precision, Recall, and F1 score of DenseNet-2 on FERFIN dataset.

Emotion	Neutral	Happy	Surprise	Sad	Angry	Disgust	Fear
Precision	0.8933	0.9607	0.9097	0.8609	0.6880	0.8824	0.8833
Recall	0.9400	0.9667	0.9182	0.6510	0.8885	0.5000	0.6974
F1 Score	0.9161	0.9636	0.9139	0.7418	0.7755	0.6383	0.7838

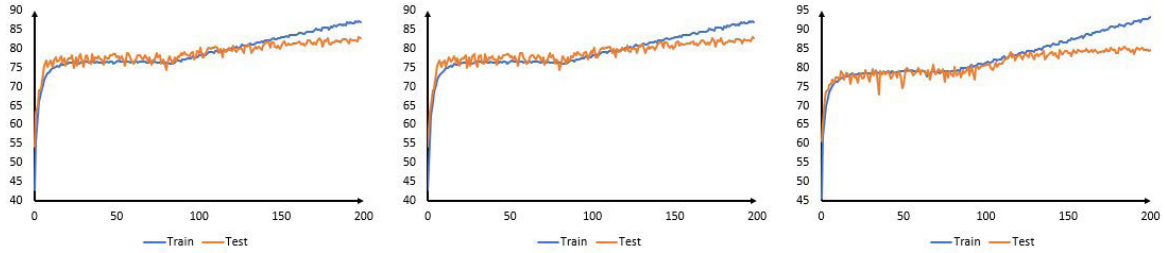


FIGURE 8. Learning curve of three DenseNet models on both FERFIN training set and testing set.

TABLE 6. Testing accuracy from training three DenseNet models on FERFIN.

Model	Trials 1	2	3	4	5	Accuracy
DenseNet-1	83.51%	**84.25%**	83.17%	83.84%	83.75%	83.71% $\pm$ 0.54%
DenseNet-2	84.36%	84.42%	**85.89%**	84.73%	85.76%	85.125% $\pm$ 0.765%
DenseNet-3	**85.90%**	84.59%	84.78%	84.28%	85.37%	85.09% $\pm$ 0.81%

"Sad", "Angry", "Disgust" and "Fear" classes. This may be due to the small number of samples and the large intra-class variation. We will try to solve this problem in the future.

Deep learning methods have disadvantages that force researchers to pursue large labeling datasets and build many very large models so that they can achieve state-of-art results in competitions. Due to the weak learning ability of a shallow convolutional network, small-sized and efficient architectures such as DenseNet deserve more attention and investigation. Therefore, the advantage of automatic and high-accuracy deep learning methods can apply to real-time applications.

V. CONCLUSION AND FEATURE WORK

To directly recognize emotion from the input image in real-time with high accuracy, this work proposed a lightweight emotion recognition (LER) model that utilizes a densely connected convolution layer and model compression techniques. In addition, to improve accuracy, this work utilized a multichannel input method to preprocess the gray images and created a more concise dataset FERFIN that was adjusted from the FERPLUS dataset.

In the original FER2013 dataset, our DenseNet-3 achieved 71.73% accuracy in the validation set which is 0.57% beyond the first team's result. In the crowd-sourced labeled dataset FERPLUS, our DenseNet-2 achieved 85.58% accuracy in the validation set. Under the same loss function setting our model was 0.69% improved compared with the result in Barsoum's work [42], and there were 41 time fewer parameters in DenseNet-3 than VGG13. After removing the noise date and combining similar classes, we created the FERFIN dataset. In this dataset, our DenseNet-2 model with 0.21 million parameters archived 85.89% accuracy in the validation set.

Many researchers believe that dynamic modality can extract more useful features to recognize spontaneous facial expression which is the next inevitable topic in emotion recognition. In the future, we plan to detect spontaneous emotion by considering temporal information while still utilizing the lightweight algorithm for real-time application.

REFERENCES

[1] S. Li and W. Deng, "Deep facial expression recognition: A survey," 2018, *arXiv:1804.08348*. [Online]. Available: http://arxiv.org/abs/1804.08348

[2] P. V. Rouast, M. Adam, and R. Chiong, "Deep learning for human affect recognition: Insights and new developments," *IEEE Trans. Affective Comput.*, to be published.

[3] G.-B. D. de Boulogne and R. A. Cuthbertson, *The Mechanism of Human Facial Expression*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[4] M. Suwa, N. Sugie, and K. Fujimora, "A preliminary note on pattern recognition of human emotional expression," in *Proc. Int. Joint Conf. Pattern Recognit.*, 1978, pp. 408–410.

[5] P. Ekman and W. V. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Palo Alto, CA, USA: Consulting Psychologists Press, 1978.

[6] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 97–115, Feb. 2001.

[7] B. Martinez, M. F. Valstar, B. Jiang, and M. Pantic, "Automatic analysis of facial actions: A survey," *IEEE Trans. Affective Comput.*, vol. 10, no. 3, pp. 325–347, Jul. 2019.

[8] M. Pantic and I. Patras, "Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 2, pp. 433–449, Apr. 2006.

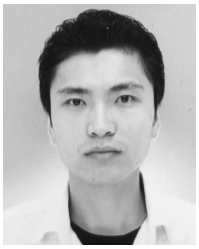
[9] F. Noroozi, D. Kaminska, C. Corneanu, T. Sapinski, S. Escalera, and G. Anbarjafari, "Survey on emotional body gesture recognition," *IEEE Trans. Affective Comput.*, to be published.

[10] M. Bartlett, G. Littlewort, T. Wu, and J. Movellan, "Computer expression recognition toolbox," in *Proc. 8th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Sep. 2008, pp. 298–305.

[11] C. A. Corneanu, M. O. Simon, J. F. Cohn, and S. E. Guerrero, "Survey on RGB, 3D, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1548–1568, Aug. 2016.

[12] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A high-resolution 3D dynamic facial expression database," in *Proc. 8th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Sep. 2008, pp. 1–6.

[13] L. Zhang, B. Verma, and D. Tjondronegoro, "Facial expression analysis under partial occlusion: A survey," *Facial Expression Anal. Under Partial Occlusion, A Survey*, vol. 52, no. 2, p. 25, 2018.
[14] B. Tu, X. Zhang, X. Kang, J. Wang, and J. A. Benediktsson, "Spatial density peak clustering for hyperspectral image classification with noisy labels," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5085–5097, Jul. 2019.
[15] B. Tu, X. Yang, N. Li, C. Zhou, and D. He, "Hyperspectral anomaly detection via density peak clustering," *Pattern Recognit. Lett.*, vol. 129, pp. 144–149, Jan. 2020.
[16] P. Lemaire, M. Ardabilian, L. Chen, and M. Daoudi, "Fully automatic 3D facial expression recognition using differential mean curvature maps and histograms of oriented gradients," in *Proc. 10th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit. (FG)*, Apr. 2013, pp. 1–7.
[17] S. A. M. Al-Sumaidaee, M. A. M. Abdullah, R. R. O. Al-Nima, S. S. Dlay, and J. A. Chambers, "Multi-gradient features and elongated quinary pattern encoding for image-based facial expression recognition," *Pattern Recognit.*, vol. 71, pp. 249–263, Nov. 2017.
[18] A. Dapogny, K. Bailly, and S. Dubuisson, "Dynamic facial expression recognition by joint static and multi-time gap transition classification," in *Proc. 11th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit. (FG)*, May 2015, pp. 1–6.
[19] 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, Jan. 2012.
[20] K. Suzuki, "Overview of deep learning in medical imaging," *Radiol. Phys. Technol.*, vol. 10, no. 3, pp. 257–273, Sep. 2017.
[21] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
[22] H. Ma and T. Celik, "FER-Net: Facial expression recognition using densely connected convolutional network," *Electron. Lett.*, vol. 55, no. 4, pp. 184–186, Feb. 2019.
[23] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Aug. 2005, p. I-511.
[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jul. 2005, pp. 886–893.
[25] M. Osadchy, M. Miller, and Y. Lecun, "Synergistic face detection and pose estimation," *J. Mach. Learn. Res.*, vol. 8, no. 1, pp. 1197–1215, 2006.
[26] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.
[27] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2578–2585.
[28] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1867–1874.
[29] G. Sandbach, S. Zafeiriou, and M. Pantic, "Local normal binary patterns for 3D facial action unit detection," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 1813–1816.
[30] A. Dhall, A. Asthana, R. Goecke, and T. Gedeon, "Emotion recognition using PHOG and LPQ features," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Workshops*, Mar. 2011, pp. 878–883.
[31] M. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On deep generative models with applications to recognition," in *Proc. CVPR*, Jun. 2011, pp. 2857–2864.
[32] S. Rifai *et al.*, "Disentangling factors of variation for facial expression recognition," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 808–822.
[33] G. Caridakis, L. Malatesta, L. Kessous, N. Amir, A. Raouzaiou, and K. Karpouzis, "Modeling naturalistic affective states via facial and vocal expressions recognition," in *Proc. Int. Conf. Multimodal Interfaces ACM*, 2006, 146–154.
[34] M. Wöllmer, M. Kaiser, F. Eyben, B. Schuller, and G. Rigoll, "LSTM-modeling of continuous emotions in an audiovisual affect recognition framework," *Image Vis. Comput.*, vol. 31, no. 2, pp. 153–163, Feb. 2013.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (JMLR)*, 2015, pp. 448–456.
[38] M. Lin, Q. Chen, and S. Yan, "Network in network," *Comput. Sci.*, to be published.
[39] W. Su, S. Boyd, and E. J. Candes, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, vol. 3, no. 1, pp. 2510–2518.
[40] Kaggle Team. (2013). *FER2013 Dataset*. [Online]. Available: https://www.kaggle.com
[41] I. J. Goodfellow, D. Erhan, and P. L. Carrier, "Challenges in representation learning: A report on three machine learning contests," in *Proc. Int. Conf. Neural Inf. Process.* Berlin, Germany: Springer, 2013, pp. 117–124.
[42] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang, "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *Proc. 18th ACM Int. Conf. Multimodal Interact. (ICMI)*, 2016, pp. 279–283.
[43] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360
[44] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for MobileNetV3," 2019, *arXiv:1905.02244*. [Online]. Available: http://arxiv.org/abs/1905.02244
[45] N. Ma *et al.*, "Shufflenet v2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Comput. Sci.*, to be published.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.



GUANGZHE ZHAO received the Ph.D. degree in computer science from Nagoya University, Japan, in 2012. He is currently an Associate Professor with the Beijing University of Civil Engineering and Architecture. His research interests include image processing and pattern recognition.

HANTING YANG received the B.S. degree in building electricity and intelligence from the Beijing University of Civil Engineering and Architecture, China, in 2013. His current research interests include emotion recognition, fatigue detection, and deep learning.

MIN YU graduated from Sun Yat-Sen University with an eight-year program of clinical medicine. He is currently an MD of the Pancreatic Tumor Center of Guangdong Provincial People's Hospital. He is not only a Secretary of the Group of Pancreas Surgery, Guangdong Provincial Physician Association, but also on the Youth Committee of the Anticancer Association, Guangdong Provincial Medical Association. So far, he has published over 45 SCI articles about the treatment and prevention of pancreatic cancer and liver cancer, deep learning, and bioinformatics. He is proficient in basic scientific research, such as treatment and prevention of pancreatic cancer and liver cancer, deep learning, and bioinformatics. Moreover, he acts as a Peer Reviewer of many famous medical journals, such as *BMC Genetic and Frontier oncology*.