Lightweight CNN-based Expression Recognition on Humanoid Robot

Guangzhe Zhao[1], Hanting Yang[1], Yong Tao[2*], Lei Zhang[1] and Chunxiao Zhao[1]
[1] Beijing University of Civil Engineering and Architecture
Beijing 100044, China
[e-mail:zhaoguangzhe@bucea.edu.cn]
[2] Beihang University
Beijing 100191, China
[e-mail :taoy@buaa.edu.cn]
*Corresponding author: Yong Tao

Received September 27, 2019; revised October 27, 2019; accepted November 11, 2019; published March 31, 2020

Abstract

The human expression contains a lot of information that can be used to detect complex conditions such as pain and fatigue. After deep learning became the mainstream method, the traditional feature extraction method no longer has advantages. However, in order to achieve higher accuracy, researchers continue to stack the number of layers of the neural network, which makes the real-time performance of the model weak. Therefore, this paper proposed an expression recognition framework based on densely concatenated convolutional neural networks to balance accuracy and latency and apply it to humanoid robots. The techniques of feature reuse and parameter compression in the framework improved the learning ability of the model and greatly reduced the parameters. Experiments showed that the proposed model can reduce tens of times the parameters at the expense of little accuracy.

Keywords: Humanoid Robot, Human-machine interaction, CNN, Emotion Recognition

This work was supported by the National Natural Science Foundation of China under Grant 61871021, supported by the fundamental Reaserch Funds for Beijing University of Civil Engineering and Architecture under grant X18002.

http://doi.org/10.3837/tiis.2020.03.015 **ISSN : 1976-7277**

1. Introduction

The expression implies the inner psychological activities of people when they experience events, and also it is a crucial factor leading communication in the social environment. Facial expression is caused by many reasons, including mood, personality, motivation, etc. Scientists believed that facial expression is an important signal transmission system for humans. Therefore, constructing an automatic recognition system can better understand how expressions are formed and analyze the potential meaning of expressions. It can contribute to the practical problems, such as, product recommendation, pain detection, fatigue detection and so on.

The earliest research on facial expression recognition was published in 1978, which is based on facial key point tracking algorithm [1]. Similar methods had been used in the same period, but they were limited by poor facial detection algorithms and facial registration algorithms, which had led to slow progress in this field. At the beginning of the 20th century, the leap in computer arithmetic and the publication of the first benchmark facial expression database Cohn-Kanade [2] encouraged many researchers to re-engage in research.

The mainstream methods are divided into two types. One assumes that people have mutually exclusive basic expressions, which are distributed in respective spaces; the other method considers that human expressions are composed of facial units, which are different facial muscle groups. Facial units contain a combination of single or multiple facial muscle groups, and there are more difficulties in calibrating the data. In contrast, basic expressions are widely accepted for their universality and comprehensibility.

Basic expression recognition generally uses two-dimensional image data, because the color information has little contribution to the extraction of facial features, so grayscale images are the most commonly used. There are generally two ways of facial feature extraction, one is manual design features, such as appearance features and geometric features, and the other is learning features, which are obtained by machine learning. It is worth mentioning that because the two-dimensional image does not contain depth information, so some researchers published a three-dimensional database like BU-3DFE [3]. However, 3D images need more computation, so we do not consider using them in this paper.

The method of extracting manual design features is gradually replaced by data-driven method on the premise of having a lot of data and efficient computing chip. Convolutional neural network is a typical data-driven method, which effectively extracts image features by weight sharing and downsampling [4]. As a result, well-funded laboratories build large models through the number of layers of the stacked network and train on several high-performance graphics cards. This makes it difficult for ordinary researchers to reproduce and it is almost impossible for this model to be deployed in real situations. To solve the above problems, we proposed a lightweight convolution neural network model, which included a densely connected convolution layer and a parameter compression layer to achieve better real-time performance at the expense of little accuracy.

Besides, the humanoid robot is the most ideal robot that can interact with people. Its shape and behavior are close to people, and it is an effective practice platform for human-computer interaction. The development of humanoid robots originated from Professor Kato Ichiro's research on robotic bipedal walking in 1973, and Honda developed the first humanoid robot P2. The popular humanoid robots currently on the market include ASIMO of Honda Corporation of Japan, Atlas of Boston Power, and NAO of Aldebaran Robotics of France. The

NAO robot has an easy-to-use appearance, simple operation, and a programmable interface. If the real-time expression recognition system is integrated, the emotional state of the head of the household can be understood, which will be of great significance to the design and development of the home robot. In particular, the accompanying robot, a robot for the elderly and children, can analyze the psychological state of the head of the household and make corresponding actions to achieve the effect of psychotherapy. In addition to innovation in the original basic expression and motion unit detection algorithms, it is also a research hotspot to identify more complex expression information. For example, self-expression recognition is used to detect whether the expression is deliberate or spontaneous; fatigue state detection is of great significance for assisting driving driving system [5]; depression, pain and depression detection can help doctors better analyze the patient's condition [6].There are also complex mental state analysis of virtual facial expressions, which are current and future research trends.

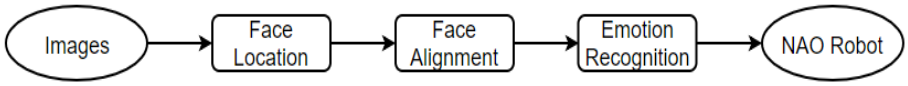


Fig. 1. Proposed Emotion Recognition Framework.

2. Related Work

According to this review [4], facial expression recognition mainly includes four processes, namely face detection, face registration, feature extraction and expression recognition.Feature extraction and expression recognition can be performed simultaneously in different frameworks.

Face detection locates the position of the face in the image through key points such as the nose and mouth of the eye. The classical algorithm is the Adaboost cascade classifier based on haar-like feature extraction proposed by Viola and Jones [7]. Combinations of support vector machines and gradient histograms are also common [8]. In order to solve the problem of attitude transformation, researchers have used a large amount of data to train a classifier based on convolutional neural networks, but the picture processing time is longer[9].

Face registration uses matrix transformation to stretch the original face to the position corresponding to the key point. Many studies have proved that face registration after face detection can effectively improve the accuracy of expression recognition [10, 11].

Feature extraction laying the foundation for subsequent expression recognition and the core of algorithm efficiency. It can be divided into two ways: pre-design features and learning features. The pre-design feature is the earliest method used. It takes the prior knowledge to manually design the operator to extract relevant information. Appearance features and geometric features are the subclassification.

The geometric feature measures the distance, curvature, and deformation based on the facial reference points found in the image, and then analyzes the emotional information of the face. M. Pantic and I. Patras [12] 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 of the distance. Although the representation ability of geometric features is sufficient to describe the overall face information, it is difficult to capture small changes such as wrinkles and skin texture. The researchers suggest to use appearance features based on image grayscale information, which is generally extracted on the entire face. This method can capture the facial deformation when the micro-expression is triggered. Abhinav et al., “[13] 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 which statistically analyzes the edge image direction gradient histogram at different levels leading to strong anti-noise performance and certain anti-rotation ability, but is subject to layering rules and lacks scale adaptability. In the work of [14], the improved LBPs feature Boosted-LBP is used to describe the local texture information, which effectively improves the discriminating ability for low-resolution images, but it is difficult to generalize to other datasets. G. Littlewor et al., "[15] used Gabor filters to extract image features which takes advantage of Gabor wavelets characteristics in processing texture and discrimination features and illumination invariance and posture invariance, but the disadvantage is that the calculation is complex and needs to go through Gaussian kernel function modulation and other steps.

Pre-designed features involve a large amount of prior knowledge and is difficult to modify, so researchers turn their attention to end-to-end learning methods which use a large amount of labelling data for supervised learning, mainly based on convolutional neural networks that are good at processing image data [16, 17], which utilizes the characteristics of local receptive fields and is similar to the way human eyes observe things. Recursive neural networks take additional timing information into account whose variant version can retain important information and abandon unwanted information [18-20]. Because of the advancement of big data, above mentioned method with strong data dependence has occupied most of the visual field problems, and the researchers continue to expand the depth or width of the network architecture in order to obtain more state-of-art results. However, there are two obvious shortcomings. One is that the characteristic expression of learning is the weight of the network layer neurons. There is no convincing statement to clarify its significance. Secondly, large-scale networks need thousands of trainable parameters which means that it is not feasible to apply it to practical application.

The purpose of expression classification is to analyze and make a pre-judgment based on the features extracted by the feature extraction link. It actually includes both classification and regression, depending on the expression of the target. The Bayesian network classifier is used in the work of [21]. Its working principle is to build a model using prior knowledge and then predict it according to Bayes' law. The Bayesian method can model on any descriptive problem, while the Bayesian learning process only includes two general steps of prior and integration, but it is difficult to a priori in practical problems. It is specified, and the calculation process is very complicated and non-automatic.

In addition, the more common classifiers are neural networks [22], support vector machines [23], and random forests [24], all of which belong to machine learning methods. Among them, the performance of neural networks continues to improve with the increase of data volume, while the growth of support vector machines is not obvious. Although random forests can handle large-scale data, they are better at handling simple numerical data. Therefore, researchers have used neural networks, especially convolutional neural networks, as expression classifiers. In order to obtain state-of-art results, the complexity and depth of the model continue to increase. However, deep models are able to achieve excellent results in competitions but are difficult to deploy in real-world applications. It led to the study of model compression techniques, such as convolutional layer optimization [25, 26], model parameter pruning [27, 28]. These methods can effectively reduce model parameters without compromising the model's learning ability.

In summary, the contribution of this paper was to use the dense network [29] to train the expression recognition classifier to balance the recognition accuracy and real-time of the algorithm. A dense network is essentially a convolutional neural network, but its densely

connected nature greatly reduces the trainable parameters of the network. Under the similar precision conditions of large-scale convolutional networks, it achieved the high real-time requirements in real-world applications. In addition, to test the feasibility of the algorithm, this work used the humanoid robot NAO as the application interaction platform for verification. Experiments showed that while maintaining acceptable accuracy, NAO can quickly respond to human-computer interactions for different expression classifications.

3. Emotion Recognition

Deep convolutional networks can handle the posture changes, illumination changes, and occlusion challenges in the natural environment if widely distributed datasets are provided. However, the continuous expansion of the network architecture for the pursuit of accuracy is meaningful for exploring the potential of the network architecture and does not help to build practical applications as the expression recognition in real scenes requires prediction in one second or less. Therefore, it is very important to use a network architecture with fewer parameters to ensure acceptable accuracy and high real-time performance. We proposed a method to meet the above requirements. Details of each part was followed.

3.1 Preprocess of Face Data

Face detection and face registration can be viewed as a data pre-processing process that processes the image into a tile input convolutional network model of the same size and correction. In the face detection section, we used the HOG feature and the SVM approach which balanced accuracy and speed better than other methods and was more suitable for online identification applications. In the face alignment section, we used an ensemble method of regression trees to detect 68 landmark points, **Fig. 2** and **Fig. 3** explained the processes.

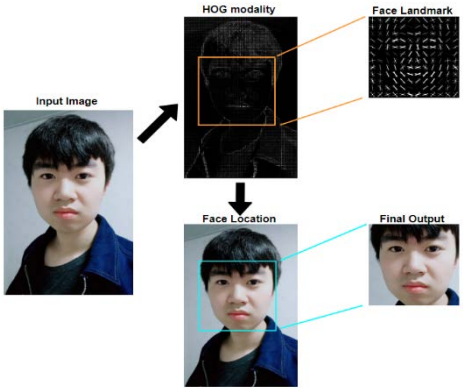


Fig. 2. Match the face patch by HOG features of standard face in the HOG modality image and get the bounding box.

Fig. 3. Match the face profile with 68 landmarks and then correct to the center.

3.2 Dense Emotion Neural Network

This article described the model in the following four aspects: architecture, convolutional layer, transition layer, and training strategy.

a. Architecture

The dense convolutional neural network used herein contains a total of 37 convolutional layers, 3 pooling layers, and a softmax layer. The input was a 48×48×1 gray image, then through a 3×3 convolution layer, followed by 3 dense blocks each containing 12 convolution layers. Connected at the end of each dense block, a transition layer consisted of an Average Pooling, a Bottleneck Layer, and a Compression Layer. Finally, according to the different target categories, the two final output layers of the 7-category Softmax layer and the 10-category Softmax layer were connected. The purpose of the Softmax layer was to map the output of multiple neurons into the interval of (0,1), which was calculated as:

$$p\left(y^{(i)} = j \middle| x^{(i)}; \theta\right) = \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}} \quad (1)$$

where $y^{(i)}$ represented the label of a certain type of expression, $x^{(i)}$ represented the input feature, θ was the total weight of the network. Above function's output was the confidence of a certain type of expression. It can also be considered as the predicted probability. The way the Softmax layer introduced the probability model forced the sum of the output vectors to be 1, which was more suitable for discretely defined expression classification problems because each category was mutually exclusive. In addition, the Softmax function maximized the gradient that the loss function obtains during training and made the network more discriminating. The total number of parameters of the network under the settings of the two Softmax layers were 95263 and 95302, respectively.

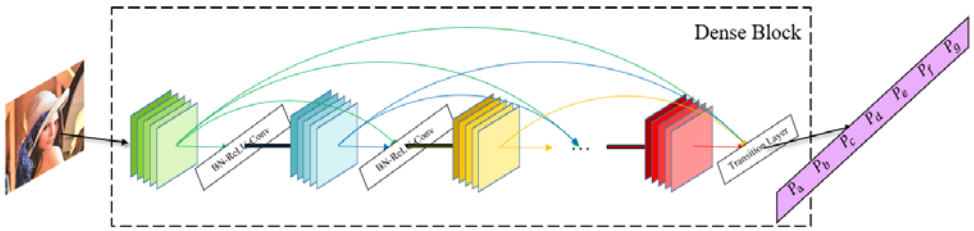


Fig. 4. Proposed Dense Emotion Neural Network.

b. Convolution Layer

The convolutional layer of a densenet was divided into two types, one was a first layer of feature extraction layers other than the dense block, and the other was a convolutional layer contained inside the dense block. The purpose of the former was to change the image from a tensor with large-sized and narrow channel to a tensor with a small-size and wide channel, which facilitated the computational of dense connections. The latter was the most important part of the architecture. Each layer of the convolutional layer in the dense block was connected to all subsequent convolutional layers as supplementary inputs, and in the case of the L-layer convolutional layer, a total of $L(L+1)/2$ connections were generated. The feature map of the product calculation used all the time, and the final convolutional layer accepted the features extracted from all previous convolutional layers. The growth rate k is a hyperparameter which specifies the growth rate of the number of feature maps for each layer of convolutional networks. When the number of channels of input data is assumed to be m, the layer 1 convolutional network have $m + k(l-1)$ feature maps.

Convolutional neural networks have two main extensions, deep expansion and convolution kernel expansion [30, 31]. Deep extension directly passes the input features to the next layer so that the network does not lose important information during the learning process, thus ensuring that the learning ability of the network is not affected by the disappearance of the

gradient. Convolution kernel expansion allows the network to learn multi-scale features by setting multi-size convolution kernels in the convolutional layer. The dense network referred to the method of depth expansion, and the features of each layer were input into each subsequent layer, so that the features were highly multiplexed. In the optimization aspect, since each layer of the feature map was directly connected to the last layer, the weight of each layer was optimized from the position where the gradient was generated. Therefore, the information flow of the network didn't terminate due to the large architecture which means vanishing gradient problem can be solved.

In fact, ReLU and Batch Normalization [32] were also included in the convolutional layer. 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 the unilateral suppression mapping to be closer to the biological signal transmission process, and has a wider excitation boundary, which also has a significant effect in overcoming the gradient disappeared problem. In addition, ReLU deliberately shields a large number of input signals, which is reflected in the negative half-axis part of the X-axis. This sparse activation is more suitable for extracting the sparse image features existing in the manifold space so that improving the precision and efficiency of learning. The purpose of batch normalization was to ensure that the input of each layer has zero mean and unit variance, which was originally derived from the initialization of the input layer and belongs to the network training skills. It speeded up the training of the network and added a certain regularization function. The specific details were described in the training strategy. The generalized calculation in the convolutional layer was shown in Eqs. (2).

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

c. Transition Layer

Transition layer existed in the middle of two dense blocks and had two purposes: reducing network parameters and facilitating the calculation of the next dense block. This layer mainly included an average pooling layer, a bottleneck layer, and a compression layer. The pooling layer is a nonlinear down sampling process, which divides the feature map outputted by the convolutional layer into a set of non-overlapping rectangles. It also performs a certain rule calculation for each sub-area to output a single value, which reduces the network parameters and the calculation amount. The average pooling layer is a kind of pooling, which calculates the average value in the sub-area and inputs the next layer. The size of the sub-area in this paper was set to 2*2. The essence of the bottleneck layer was a 1*1 convolutional layer, its main purpose is not to extract features, but to use the super parametric filter number d to controllable dimensionality reduction of the accumulated feature map. The purpose of the compression layer was to further reduce the parameters of the network and increase the compactness of network. The layer was connected behind the bottleneck layer and proportionally reduces the number of feature maps by setting a hyperparameter θ between 0 and 1. Whether the bottleneck layer and the compression layer need to be added depends on the complexity of the network and the amount of training data. They reduced the network parameters and force the network to extract more accurate features.

d. Training Strategy

The training strategy of this paper mainly focused on two aspects. One was whether the network can converge to an acceptable accuracy rate in the verification set, and the other was

to avoid over-fitting problems. The former was mainly reflected in the choice of optimization algorithm and network architecture. This paper inspired by the architecture of dense network to achieve top-level accuracy on CIFAR-10 dataset, and builds a network with a total length of 40, growth rate of 12 and three dense blocks. For the optimization algorithm, this paper used the nesterov momentum optimization method [33], which is 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.(3).

$$\begin{cases} v_t = \beta v_{t-1} + \alpha \nabla_\theta L(\theta) \\ \theta = \theta - v_t \end{cases} \tag{3}$$

This algorithm increases the momentum in the same direction as the gradient update, while reduces the vibration in the direction of the gradient change, thus can achieve 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, speed up the update rate, if the gradient has a decreasing trend, slow down the update speed rate, as shown in formula (4). 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 a faster and more stable convergence.

$$\begin{cases} v_t = \beta v_{t-1} + \alpha \nabla_\theta L(\theta - \beta v_{t-1}) \\ \theta = \theta - v_t \end{cases} \tag{4}$$

For overfitting problems, this paper used batch normalization and dropout. Over-fitting problems mean that the model over-analyzes a particular data set, so failure to fit other data or predict unknown data results in poor generalization. The batch normalization solves the internal covariate shift problem which causes by the neural network's layered parameterized manner. This manner results in the forward-propagating information flow mutating as the number of layers deepens, while the gradient is corrected through the backpropagation which lead to more information distortions. Therefore, batch normalization allows the output of each layer of the network layer to correct the distribution difference. The specific method is zero-meanization and unit variance, which makes the information flow of the network more accurate, and the back-propagation can be more accurately corrected. Dropout is similar to the way the compression layer reduces parameters. The difference is that random inactivation only randomly eliminates the output of neurons at a certain rate during training, and then reactivates the neurons when calculating the accuracy of the verification set. Under this kind of operation, the network at each iteration is actually different, and each neuron cannot rely on other neurons, avoiding the over-analysis or over-fitting caused by all neurons co-analyzing the data.

4. Experiment Result

This section presented our experimental environment and experimental results. The content of experimental environment part included hardware devices and training datasets. The content of experimental results part were the training results of different hyperparametric dense convolutional neural networks on three datasets generated by homologous samples.

4.1 Experimental Environment

a. Hardware Devices

All the model training in this paper was on the GTX1060 graphics card. It has 1280 CUDA units, 6GB GDDR5 memory, core frequency 1506MHz, and single-precision floating-point operation is 4.4TFlops. The test device used a screen-integrated 2-megapixel camera that is sufficient for facial expression recognition in images. In addition, the NAO robot was used to test the practicality of the expression recognition system. It has an upright height of 120cm, a head with two CMOS cameras and four loudspeakers, and supports a variety of programming languages.

b. Dataset

FER2013 used was originally derived from a facial expression image [34] taken from the video collected by the Kaggle team from the internet in 2013, which contains 35,887 gray images of 48 x 48 pixels. At the first publication time, the dataset labels were divided into 7 categories, including 4953 cases of "anger", 547 cases of "disgust", 5121 cases of "fear", 8989 cases of "happy", 6077 cases of "sadness", 4002 cases of "sadness" and "4002 cases of surprise" and " Neutral" 6198 cases. This labelling was later verified to be inaccurate, and we trained in the dataset, as well as the improved FER PLUS dataset [35] and FERFIN modified from FER PLUS.

4.2 Network Training Results

a. Training on the FER2013 database

For this dataset, the hyperparameters' setting as follows: added L2 regularization after the first convolutional layer, the penalty coefficient γ is set to 0.0001; added the Compression layer in the network transition layer. The parameter θ compressed the number of network feature maps to reduce the network parameters. In practice, θ was set to 0.5; the network optimization method was nesterov momentum, the learning rate ε was set to 0.1, the momentum parameter α was 0.1, and the attenuation step was 32000.

In terms of data augmentation, we used the standard 10-crop, which is to add 4 rows or columns of zero values around each image, and then intercept the top left, top right, bottom left, bottom right and middle five tiles. Then, flipping left and right to make this value doubled to 10. Compared with the large-scale networks used in the competition at the time, such as VGG16 and AlexNet, which contains parameters of several hundred thousand or even one million, the parameters of our network are only 92,407. After 18,000 steps, the accuracy of the network in the verification set was reached 67.01%, as shown in Figure 5. Final accuracy ranking of the competition as follows: first 71.16%, second 69.26%, third 68.82%. We believe that a dense network can achieve the top five results without using any aggregation method and a small number of parameters for two reasons: first, the way of feature reuse increased the size of the input of the subsequent convolution layer, which made the subsequent layers variable while accepting the previous knowledge of the network; second, dense connection and the setting of the bottleneck layer greatly reduced the parameters of the network, forcing the network to extract more compact and discriminating features.

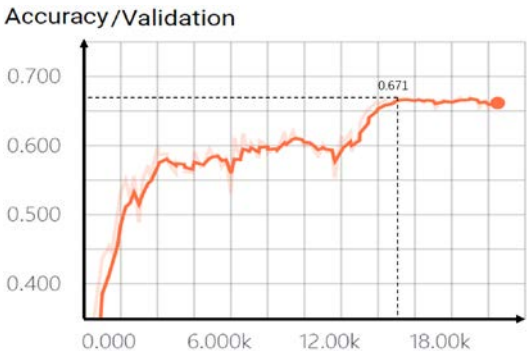


Fig. 5. Learning Curve for Model Training on FER2013.

b. Training on the FER PLUS database

Among the challenges suggested by Goodfellow et al., "[36] regarding neural network resolution classification problems, the performance degradation caused by the low accuracy of human labeler is included. The survey showed that the accuracy of the label is only about 65% due to the small size of the sample and its grayness in the FER2013 dataset, which means that even if the deep convolutional network has enough representation ability, it cannot achieve high training accuracy on the dataset. Therefore, we trained the second model on FER PLUS dataset. FER PLUS used the crowdsourcing method to improve the accuracy of the label, and added three categories of contempt, unknow and not a face.

In the original paper, four pre-designed way to handle the objective function setting problem. We only used majority vote for preprocessing as the main focus was on the update of framework. Due to the increased categories of the classification, the model become more complicated. Compared with the last training, we adjusted the penalty coefficient γ of L2 regularization to 0.001, and added the weight decay technique of the dropout layer at the end of each dense block to prevent over-fitting, the keep probability was set to 50%. The number of decay steps of nesterov momentum was adjusted to 10,000 steps with an attenuation rate of 0.1. The remaining parameters remain unchanged. The accuracy of the network in the verification set reaches 81.78%, as shown in Figure 6. The work of [35] used VGG13 to achieve an average accuracy of 83.97% under the use of the majority vote loss function strategy. Their network parameters were 8.7 million, about 97 times of our network.

Large-scale convolutional neural networks do achieve top results with a large amount of usable data, but in fact, each additional small order of accuracy after reaching a certain value requires a significant increase in network parameters.

Although the results of [35] cannot be exceeded, the dense convolutional neural network architecture minimized the amount of parameters that achieve a certain accuracy through the compression model architecture.

c. Training on the FERFIN dataset

After careful observation, we found that there were only 177 cases and 212 cases of not a face and unknown in the FER PLUS dataset. So, in order to remove the noise in the training set, we modified it to remove the not a face and unknown classes in the dataset. In addition, the contempt class and the disgust class had only 248 cases and 216 cases. In fact, the sample space similarity between the two types was very high, and if they were divided into two types, they are easily interfered by other types of tags. Therefore, the second modification we made was to combine the disgust class with the contempt class.

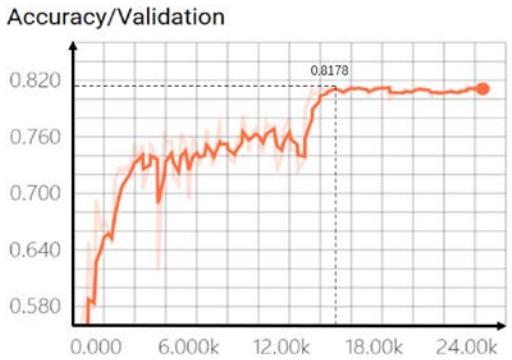


Fig. 6. Learning Curve for Model Training on FER PLUS.

The final data set is called FERFIN. In the final data set after the majority vote, there are "neutral" 12,858 cases, "happy"9354 cases, "surprised" 4462 cases, "sad" 4351 cases, "angry" 3082 cases, “disgust” 575 cases and "fear" 816 cases. A total of 35,498 cases reduced 390 noise compared to the original FER2013 dataset. According to the new sample classification, the number of attenuation steps optimized for nestrov momentum was reduced to 20,000 and the learning rate was reduced by 0.01. After training, the accuracy of the verification set reached 83.66%, which was 1.88% more than the training result on FERPLUS, as shown in Figure 7.

It is proved that the noise in the data set has a certain interference effect on the learning of the network, especially the "not a face" and "unknown" classes making the feature map less discriminating.

d.Expansion

In recent years, lightweight models have become the mainstream research direction for reducing model parameters, and many excellent CNN architectures have been proposed. To fully verify the availability of the proposed method, we compared it with the Moblienet families [25], Shufflenet [26] families and SqueezeNet. Experiments show that the accuracy performance of Densenet is the best at fer2013. The probable reason is that most of the lightweight models aim for large-scale datasets, such as ImageNet, and on small-scale datasets, the learning ability is not enough due to too few parameters. **Fig. 9** shows the comparison results.

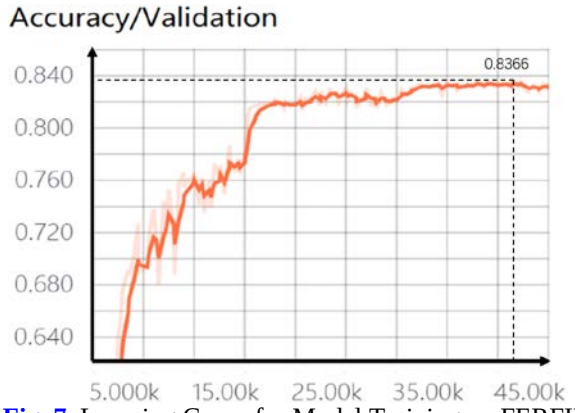


Fig. 7. Learning Curve for Model Training on FERFIN.

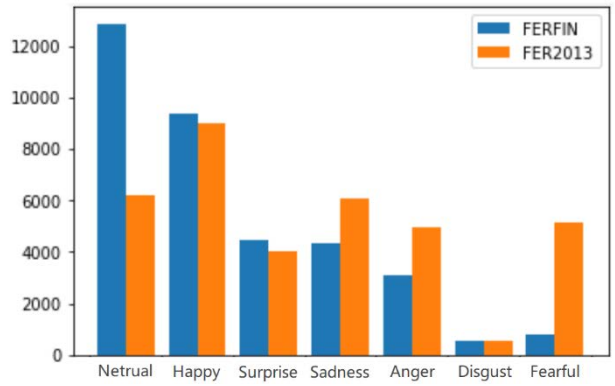


Fig. 8. Dataset Distribution between FERFIN and FER2013.

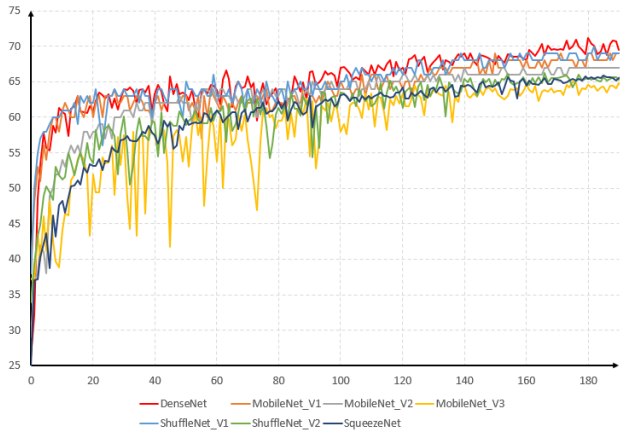


Fig. 9. Comparison results on learning curve

4.3 Expression Recognition Application Based on NAO Robot

To explore the application of expression recognition in actual scenes, we conducted experiments on NAO robots. First, the image frame read by the camera was identified to generate expression classification information, and then the information was transmitted to the NAO robot. Finally, after a certain amount of reasoning, NAO responded accordingly. **Fig. 10** showed the sequence of actions that several NAO robots make under the corresponding expressions, including sadness, happiness, disgust, and surprise. These actions are encapsulated in the intrinsic API of the NAO robot, and we selected the actions that are closer to the corresponding emoticons for testing. In the test, the speed of the expression recognition reached our expectations, and it was able to detect facial expressions in real time with acceptable accuracy.

For fear and disgust expressions, due to the lack of training samples, it was difficult for the model to detect subtle information, which was prone to missed detection and false detection. For the robot side, we used the network communication method to transmit the classification result as a single value or a vector, and then the robot executes the instruction according to the corresponding result. The NAO performed in the experiment is more natural and non-aggressive, and can meet the requirements of home companion robots.

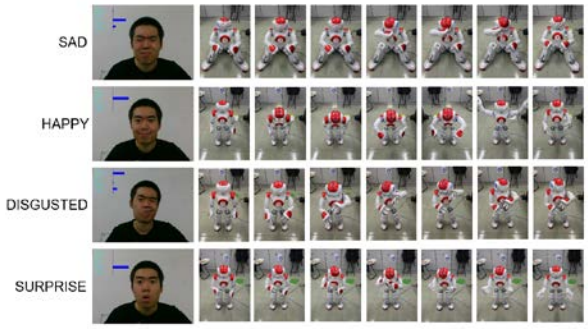


Fig. 10. Emotion recognition on NAO robot show in action series.

5. Conclusions and future work

There are classic pre-design feature's methods and emerging deep learning methods in the field of expression recognition. The former involves more prior knowledge and have less generalization ability. Early deep learning methods can achieve top-level accuracy but require millions of parameters. In order to train the expression recognition network parameters with the deep convolution model, this paper proposed to use the dense convolutional network as the new training network. Its multi-level connection and feature reuse feature reduces network parameters while enhancing network representation capability. It can reduce the need of the amount of trainable parameters as much as possible to achieve the expected accuracy.

In order to meet the requirements of real-time and accuracy in online expression recognition, we trained on the three data sets FER2013, FER PLUS and FERFIN. FER2013 is the most original competition dataset, and its 48*48 grayscale image format is very suitable for training convolutional networks. However, due to the error of manual labeling, the accuracy of the data label itself is only about 65%, so the dense network trained on the dataset had only 67% validation accuracy. FER PLUS was generated by using the crowdsourcing method to recalibrate on the FER2013 dataset. After adjusting the network hyperparameters, the dense network achieved an accuracy of 81.78%, although the accuracy decreased by 1% compared to the original text, but we reduced 147-fold parameters which meet our expected real-time requirements.

In addition, after carefully observing the data label of FERPLUS, we made a certain modification according to the noise existing in it and obtained FERFIN, which achieved an accuracy of 83.66\% on the validation set, which proved the influence of noise on network learning. In addition, the FERFIN data set has a highly unbalanced nature. Its maximum class has 10042 cases more than the minimum class, as shown in Figure 8, which means that it is difficult for the network to learn the exact characteristics of the small class. Therefore, we tried to introduce a weighted cross entropy loss function, which makes it slower to learn for a bigger class, and faster for a smaller class. Unfortunately, the accuracy of the validation set after the proposed network training was only 75% as the weighted loss function increases the training difficulty of the network, which means a deeper network architecture is required. If so, the parameters of the network model will rise dramatically, violating the real-time requirements of the proposed framework. All the results were shown in the **Table 1**.

Table 1. Accuracy on FER2013, FERPLUS and FERFIN

Model	FER2013	FERPLUS	FERFIN
Densenet	67.01%	81.78%	83.66%

The final expression recognition framework is embedded in the NAO robot and interactive experiments are performed. Experiments showed that the real-time performance had been improved. In future work, we will consider ways to further improve accuracy without sacrificing near real-time. The current feasible thinking as follow: first thinking is to supplement the small class of the dataset to balance the distribution of the data set; another thinking is to strengthen the training of specialized categories, such as the aversion and fear of difficult to detect, and then migrate to learn other databases; also if there are fast and accurate manual design methods exist, we will consider joint to the pre-training part of the network.

References

[1] M. Suwa, N. Sugie, and K. Fujimora, "A preliminary note on pattern recognition of human emotional expression," in *Proc. of Internationa Joint Conference on Pattern Recognition*, pp. 408-410, 1978. Article (CrossRef Link)

[2] Tian Y I, Kanade T, Cohn J F, "Recognizing action units for facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97-115, 2002. Article (CrossRef Link)

[3] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A high-resolution 3D dynamic facial expression database," in *Proc. of IEEE International Conference & Workshops on Automatic Face & Gesture Recognition*, pp. 1-6, 2008. Article (CrossRef Link)

[4] Corneanu C A, Simon M O, Cohn J F, et al., "Survey on RGB, 3D, Thermal, and Multimodal Approaches for Facial Expression Recognition: History, Trends, and Affect-Related Applications," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 38, no. 8, pp. 1548-1568, 2016. Article (CrossRef Link)

[5] Ji Q, Looney, "A probabilistic framework for modeling and real-time monitoring human fatigue," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 36, no. 5, pp. 862-875, 2006. Article (CrossRef Link)

[6] Ashraf A B, Prkachin K, Chen T, et al., "The painful face - pain expression recognition using active appearance models," in *Proc. of International Conference on Multimodal Interfaces. ACM*, pp.9-14, 2007. Article (CrossRef Link)

[7] Viola P, Jones M., "Rapid Object Detection using a Boosted Cascade of Simple Features," *IEEE Computer Society*, pp. I-511, 2001. Article (CrossRef Link)

[8] Dalal N, Triggs B, "Histograms of oriented gradients for human detection," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, USA*, pp. 886-893, 2005. Article (CrossRef Link)

[9] Osadchy M, Miller M, Lecun Y, "Synergistic face detection and pose estimation," *Journal of Machine Learning Research*, vol. 8, no. 1, pp. 1197-1215, 2006.

[10] Le V, Brandt J, Lin Z, et al., "Interactive Facial Feature Localization," in *Proc. of European Conference on Computer Vision 2012. Springer-Verlag*, pp. 679-692, 2012. Article (CrossRef Link)

[11] Sariyanidi E, Gunes H, Cavallaro A, "Automatic Analysis of Facial Affect: A Survey of Registration, Representation, and Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1113-1133, 2015. Article (CrossRef Link)

[12] Pantic M, Patras I, "Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences," *IEEE Transactions on Systems Man & Cybernetics Part B*, vol. 36, no. 2, pp. 433-449, 2006. Article (CrossRef Link)

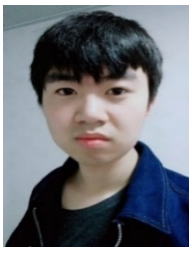
[13] Dhall, AbhinavAsthana, AkshayGoecke, et al, "Emotion recognition using PHOG and LPQ features," in *Proc. of IEEE International Conference on Automatic Face & Gesture Recognition & Workshops*, pp. 878-883, 2011. Article (CrossRef Link)

[14] Shan C, Gong S, Mcowan P W, "Facial expression recognition based on Local Binary Patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803-816, 2009. Article (CrossRef Link)

[15] Littlewort, Whitehill, Wu T, et al., "The computer expression recognition toolbox," in *Proc. of IEEE International Conference on Automatic Face \& Gesture Recognition & Workshops. IEEE*, pp. 298-305, 2011. Article (CrossRef Link)
[16] Ranzato M, Susskind J, Mnih V, et al., "On deep generative models with applications to recognition," *Computer Vision & Pattern Recognition. IEEE*, pp. 2857-2864, 2011. Article (CrossRef Link)
[17] Rifai S, Bengio Y, Courville A, et al., "Disentangling factors of variation for facial expression recognition," in *Proc. of European Conference on Computer Vision. Springer-Verlag*, pp. 808-822, 2012. Article (CrossRef Link)
[18] Caridakis G, Malatesta L, Kessous L, et al., "Modeling naturalistic affective states via facial and vocal expressions recognition," in *Proc. of International Conference on Multimodal Interfaces. ACM*, pp.146-154, 2006. Article (CrossRef Link)
[19] Wöllmer, Martin, Kaiser M, Eyben F, et al., "LSTM-modeling of continuous emotions in an audiovisual affect recognition framework," *Image & Vision Computing*, vol. 31, no. 2, pp. 153-163, 2013. Article (CrossRef Link)
[20] Gao, Lianli, et al., "Hierarchical LSTMs with adaptive attention for visual captioning," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1-1, 2019. Article (CrossRef Link)
[21] Sebe N, Lew M S, Sun Y, et al., "Authentic facial expression analysis," *Image & Vision Computing*, vol. 25, no. 12, pp. 1856-1863, 2007. Article (CrossRef Link)
[22] Li X, Song D, Zhang P, et al., "Emotion recognition from multi-channel EEG data through Convolutional Recurrent Neural Network," in *Proc. of 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE*, 2016. Article (CrossRef Link)
[23] Lemaire P, Ardabilian M, Chen L, et al, "Fully automatic 3D facial expression recognition using differential mean curvature maps and histograms of oriented gradients," in *Proc. of IEEE International Conference & Workshops on Automatic Face & Gesture Recognition. IEEE*, pp. 1-7, 2013. Article (CrossRef Link)
[24] Dapogny A, Bailly K, Dubuisson S., "Dynamic facial expression recognition by joint static and multi-time gap transition classification," in *Proc. of IEEE International Conference & Workshops on Automatic Face & Gesture Recognition*, pp. 1-6, 2015. Article (CrossRef Link)
[25] Howard, Andrew, et al., "Searching for MobileNetV3," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2019. Article (CrossRef Link)
[26] Ma, Ningning, et al., "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," in *Proc. of European Conference on Computer Vision*, pp. 122-138, 2018. Article (CrossRef Link)
[27] Liu Z, Sun M, Zhou T, et al., "Rethinking the value of network pruning," in *Proc. of International Conference on Learning Representations*, 2019. Article (CrossRef Link)
[28] Zhuang Liu, Jianguo Li, et al., "Learning Efficient Convolutional Networks through Network Slimming," in *Proc. of IEEE International Conference on Computer Vision*, 2017. Article (CrossRef Link)
[29] Huang, Gao, et al., "Densely Connected Convolutional Networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition IEEE Computer Society*, pp. 2261-2269, 2017. Article (CrossRef Link)
[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Identity Mappings in Deep Residual Networks," in *Proc. of The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016. Article (CrossRef Link)
[31] Szegedy C, Liu W, Jia Y, et al., "Going Deeper with Convolutions," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015. Article (CrossRef Link)
[32] Ioffe S, Szegedy C, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proc. of International Conference on International Conference on Machine Learning (JMLR)*, pp. 448-456, 2015. Article (CrossRef Link)
[33] Su W, Boyd S, Candes E J., "A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights," *Advances in Neural Information Processing Systems*, vol. 3, no. 1, pp. 2510-2518, 2014. Article (CrossRef Link)

[34] Kaggle team, FER2013 Dataset, https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge, 2013. Article (CrossRef Link)
[35] Barsoum, Emad, et al., "Training deep networks for facial expression recognition with crowd-sourced label distribution," in *Proc. of ACM International Conference on Multimodal Interaction ACM*, pp. 279-283, 2016. Article (CrossRef Link)
[36] Goodfellow I J, Erhan D, Carrier P L, et al., "Challenges in Representation Learning: A Report on Three Machine Learning Contests," in *Proc. of International Conference on Neural Information Processing. Springer, Berlin, Heidelberg*, pp. 117-124, 2013. Article (CrossRef Link)

GUANGZHE ZHAO received the PhD degree in computer science from Nagoya University, Japan in 2012. He is currently an associate professor with Beijing University of Civil Engineering and Architecture. His research interests include Image Processing and Pattern Recognition. E-mail: zhaoguangzhe@bucea.edu.cn



HANTING YANG received his B.S. degree in building electricity and intelligence from Beijing University of Civil Engineering and Architecture, China in 2013. His current research interests include Emotion Recognition, Fatigue Detection and Deep Learning. E-mail: 18810903925@163.com

YONG TAO received the Ph. D. degree in School of Mechanical Engineering and Automation, Beihang University, China in 2009. Currently, he is an associate professor at Beihang University, China. His research interests include intelligent robot advanced control technology and integrated applications, control of embedded mechanical and electrical integration. E-mail: taoy@buaa.edu.cn

Lei Zhang received his PhD degree in Beijing Institute of Technology, Beijing, China in 2007. He is currently professor of School of Electrical and Information Engineering, BUCEA. And he is currently deputy director of Beijing Key Laboratory of Robot Bionics and Function Research. His main research interests are humanoid robot, human-machine interaction and machine vision. E-mail: leizhang@bucea.edu.cn.

CHUNXIAO ZHAO received the PhD degree in computer science from Dongbei University, China in 2005. He is currently an professor with Beijing University of Civil Engineering and Architecture. His research interests include Machine Learning and Multi-Agent System. E-mail: chunxiao@bucea.edu.cn