Facial Expression Recognition via ResNet-18

Bin Li[1], Runda Li[2], and Dimas Lima[3](✉)

[1] School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, Henan, People's Republic of China
libin@home.hpu.edu.cn
[2] Nanjing Foreign Language School, Nanjing, Jiangsu, China
[3] Department of Electrical Engineering, Federal University of Santa Catarina, Florianópolis, Brazil
dimaslima@ieee.org

Abstract. As an important part of human-computer interaction, facial expression recognition has become a hot research topic in the fields of computer vision, pattern recognition, artificial intelligence, etc., and plays an important role in our daily life. With the development of deep learning and convolutional neural network, the research of facial expression recognition has also made great progress. Moreover, in the current face emotion recognition research, there are problems such as poor generalization ability of network model. The extraction of traditional facial expression recognition features is complex and the effect is not ideal. In order to improve the effect of facial expression recognition, we propose a feature extraction method for deep residual network, and use deep residual network ResNet-18 to extract the features of the data set. Through the experimental simulation of the specified data set, it can be proved that this model is superior to state-of-the-art methods model.

Keywords: Deep residual network · Facial expression recognition · ResNet-18

1 Introduction

With the rapid development of computer technology and neural network technology, people have a higher and higher demand for intelligent automation. We hope that the computer can acquire the changes of facial expressions [1] just like the communication between people. Only in this way can we achieve real intelligence and better human-computer interaction. So, in order to interpret human emotions, computers are required to have accurate facial expression recognition. Facial expression recognition is to separate the specific facial state from the given static image or dynamic video sequence, so as to determine the psychological emotion of the object to be recognized. On the other hand, facial expressions are a form of non-verbal communication and are the main means by which people express information. By looking at the changes in facial expressions, we

B. Li and R. Li—Those two authors contributed equally to this paper, and should be regarded as co-first authors.

W. Fu et al. (Eds.): ICMTEL 2021, LNICST 388, pp. 290–303, 2021.
https://doi.org/10.1007/978-3-030-82565-2_24

can better identify the emotional changes in the other person. Face recognition is also an important research field in computer vision. Accurate facial expression recognition will be helpful to human-computer interaction, security monitoring, auxiliary medical treatment, auxiliary driving and other work smoothly.

The purpose of facial expression recognition is to analyze a given facial expression and then classify the corresponding emotions. According to the basic facial emotions defined by American psychologists Ekman and Friesen in 1971, we can divide facial emotions into seven kinds: happy, sad, fearful, angry, surprised, disgusted and neutral.

Facial expression recognition can be divided into four stages: image acquisition, image preprocessing, feature extraction and classification recognition. The core of traditional facial expression recognition algorithms is feature extraction and classifier design. Artificial features include local binary mode (LBP), gradient histogram, etc., and classifier design includes neural network, support vector machine (SVM), K-nearest neighbor algorithm, etc. The problem is the facial expression of this kind of method of preprocessing. Feature extraction and classifier's generalization ability will influence the final classification results. Photo angle, light and shade, color of skin can cause different expression recognition difficulty. Thus, it needs for a specific case for image preprocessing and feature extraction and work more difficult. In recent years, convolutional neural network has developed rapidly, and its biggest advantage is that it can automatically learn the best features for specific tasks. Facial features such as local binary pattern (LBP), Active Shape Mode (ASM) and other features can be learned through deep learning network, and even higher-level abstract features can be learned automatically which is not available with traditional algorithms.

However, through reading recent literature, it can be found that most methods for extracting facial expression features are prone to lose the original emotional information. For example, Ali, et al. [2] proposed the use of support vector machine (SVM) method. Evans [3] presented to use Haar wavelet transform (HWT) method. In addition, the generalization and robustness of the network model are poor. When the operating environment on which the network model depends slightly changes, the network model is likely to get stuck in the recognition process, which directly affects the speed and accuracy of facial emotion recognition. Lu [4] used biorthogonal wavelet entropy as feature extractor, and employed fuzzy SVM as classifier. Phillips [5] employed Jaya algorithm to classify facial emotion recognition. Yang [6] utilized cat swarm optimization to recognize facial emotions. Li [7] chose to use biogeography-based optimization (BBO) for the same task.

The main content of this paper is to study the recognition of facial expressions, and propose an image facial expression recognition algorithm based on deep learning, and realize the recognition of image sequence facial expressions on the basis of static expression recognition. The main contributions of this paper are as follows: Study the input image pretreatment operation to improve the recognition rate of facial expressions; ResNet-18 is used as the backbone network for static facial expression recognition algorithm. Compared with other algorithms, the experimental results show that the recognition rate of this algorithm is improved to some extent.

2 Dataset

In order to make the experimental process easier to implement and the experimental results more comparable, the data set adopted in this paper is a new data set composed of a face model. The reference data set captures the facial expressions of objects of different ages, occupations, and races, including images of seven facial emotions: happy, sad, fearful, angry, surprised, disgusted, and neutral. Figure 1 displays samples of our dataset.

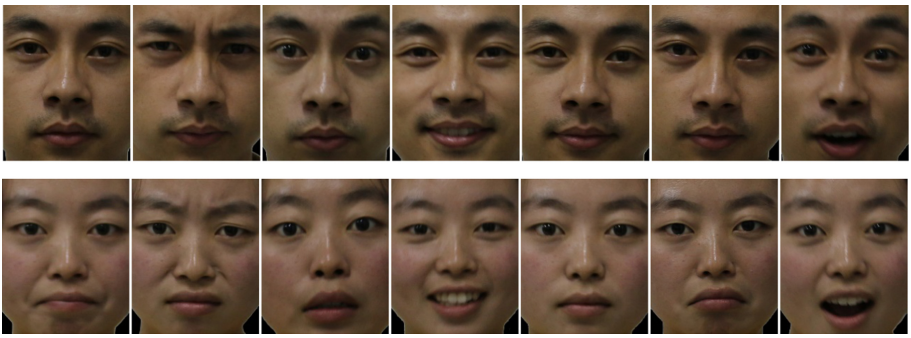

Fig. 1. Samples of our dataset [4]

3 Methodology

3.1 Convolution

The main function of convolutional layer is to extract key features from the input image and compress the image into a form that is easier to process, so as to obtain good prediction. In functional analysis, convolution is a definition of a function [8–10]. It is a mathematical operator that generates a third function by two functions f and g, representing the area of the overlap between the functions f and g after flipping and shifting. The calculation formula of convolution is as follows:

$$h(x) = f(x) * g(x) = \int_{-\infty}^{+\infty} f(t)g(x-t)dt \tag{1}$$

As is shown in (1), the output of the system at a certain time is the result of the superposition of multiple inputs. $f(x)$ and $g(x)$ are two integrable functions. And $h(x)$ is the result of the convolution operation [11]. In the image analysis, $f(x)$ can be understood as the original pixel points, all the original pixel points added up to be the original graph. $g(x)$ can be called the action point, and all the action points are collectively called the convolution kernel. After all the action points on the convolution kernel act on the original pixels in turn, the output result of linear superposition is the output of the final convolution, which is called destination pixel [12].

Convolutional neural network is a process of extracting features, selecting features and then classifying them. Some specified features of the original image can be extracted by mathematical operation with the convolution kernel [13]. The extracted features are different with different convolution kernels. The extracted features are the same, and different convolution kernels have different effects.

3.1.1 Standard Convolution

Each image can be regarded as a matrix composed of pixel values, and features can be extracted from the image through convolution. In the convolutional layer, the unit used for the convolution operation is called the convolution kernel [14–16]. Its parameters are what we want to learn, and the size should be smaller than the input image. In the process of convolution, each kernel carries out convolution computation with the input image [17], slides the convolution kernel on the image, multiplies the pixel value on the image with the corresponding value on the convolution kernel, and adds up all the multiplied values and finally slide each part of the image [18].

The Fig. 2 shows a two-dimensional convolution process with a 3 × 3 kernel and a stride of 1.When the convolution kernel is scanned on the input image, the value of the corresponding position in the input image is multiplied by the convolution kernel one by one, and the final sum is summarized to obtain the convolution result of the position [19]. By constantly moving the convolution kernel, the convolution results at each position can be calculated. For each point, we can take this point and convolve it with the 3 by 3 points around it.

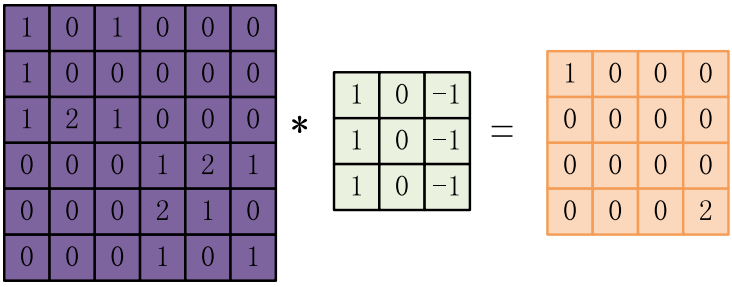

Fig. 2. Standard convolutional process

3.2 Pooling

Pooling is used to reduce the dimension of the data. In CNN, after convolution, the feature dimension of the output of the convolutional layer is usually reduced by Pooling, the size of the matrix generated by the convolutional layer is reduced, and the over-fitting phenomenon can be prevented while the network parameters are reduced [20–22]. In short, pooling is to remove the redundant information, retain the key information, reduce

the impact of noise, and make each feature more robust. The common pooling operations are Max pooling and Average pooling.

3.2.1 Average Pooling

Averaging pooling means averaging the characteristic points in the neighborhood. There are two errors in feature extraction of pooling method: first, the limitation of neighborhood size leads to the increase of estimated variance; Second, parameter error of convolutional layer causes deviation of the estimated mean value. The average pooling of image can reduce the first error and retain more background information of the image.

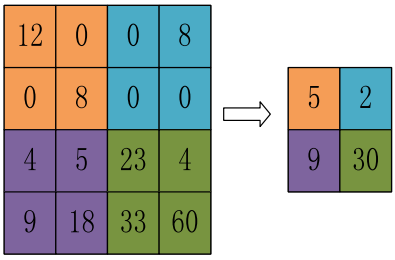

Fig. 3. Average pooling process

In Fig. 3, the input is a 4 × 4 matrix and the stride is 2. The kernel that performs the average pooling is a 2 × 2 matrix. We break up the 4 × 4 input into different areas and color these areas. For a 2 × 2 output, each element of the output is the average element value in its corresponding color region. The Average pooling result is obtained from Average (12, 0, 0, 8) = 5, Average (0, 0, 8, 0) = 2, Average (4, 5, 9,18) = 9, and Average (23, 4, 33, 60) = 30.

3.2.2 Max Pooling

The max pooling is to take the maximum of the characteristic points in the neighborhood. Features of Max pooling selection have better recognition and provide nonlinear features. In addition, maximum pooling can reduce the second error and retain more texture information [23–25]. The advantage of Max pooling is that only the maximum value (features) in the area is retained and other values are ignored to reduce the impact of noise and improve the robustness of the model. In addition, the hyper-parameters required for Max pooling are only filter size F and filter stepping length S, and no other parameters need to be obtained by model training, so the calculation amount is very small. If there are multiple channels, the Max Pooling operation is performed separately for each channel.

In Fig. 4, the input is a 4 × 4 matrix and the stride is 2. The kernel that performs the max pooling is a 2 × 2 matrix. We break up the 4 × 4 input into different areas and color these areas. For a 2 × 2 output, each element of the output is the maximum element value in its corresponding color region. The max pooling result is obtained from Max(12, 0, 0, 8) = 12, Max(0, 8, 0, 0) = 8, Max(4, 5, 9, 18) = 18, and Max(23, 4, 33, 60) = 60.

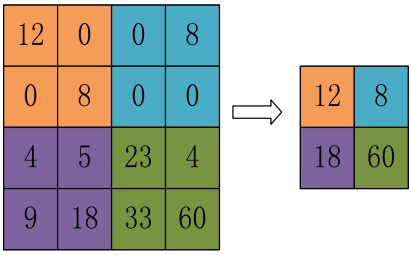

Fig. 4. Max Pooling Process

3.3 Batch Normalization

Due to the activation of the input value in deep neural network before nonlinear transform as the network depth deepening or in the process of training, its distribution deviates or changes gradually, the training convergence is slow and gradually approaches the upper and lower limits of the value interval of the nonlinear function, such as the sigmoid function, the activation input value Wx+ B is a large negative or positive value, so it leads to lower when back propagation neural network gradient disappear, this is a fundamental reason for the slow convergence is more and more deep neural network training, BN through certain means of standardization, this input neurons in each layer of the neural network value forced back to the distribution of the mean to 0 variance 1 standard normal distribution, distribution of more and more partial forced back to the standard distribution, in order to activate the input value falls in the area of nonlinear function is more sensitive to input, this will lead to small changes in input change a loss function, mean let gradient get bigger, so avoid gradient disappeared, and gradient bigger means learning convergence speed is fast. The large step towards the optimal value of loss function can accelerate the training speed greatly [25–27].

BN can prevent gradient explosion or dispersion, improve the robustness of the model to different hyperparameters (learning rate, initialization) during training, and keep most activation functions away from its saturated region. All of these properties of BN can help us to have a fast and robust training network. The real reason why BN can work is that BN changes the optimization problem again, making the optimization space very smooth.

In the process of image preprocessing, we usually standardize the image, which can accelerate the convergence of the network. As shown in the Fig. 5, regarding the Conv1, the distribution of the input is to satisfy a certain characteristic matrix, but, for Conv2, the feature map input may not satisfy a certain distribution, here to meet a certain distribution does not mean a certain feature map data to satisfy the distribution, the data of the feature map corresponding to the whole training sample set should meet the distribution. The purpose of Batch Normalization is to make feature map meet the distribution law of mean value 0 and variance 1.

As shown in the following formula,one layer has n dimensional input: $a = (a_1 \cdots a_n)$. μ_A is the average of the values of a. ${\sigma_A}^2$ is the variance of a. And ε is a small constant that prevents the denominator from being zero. O_i is the result by Batch

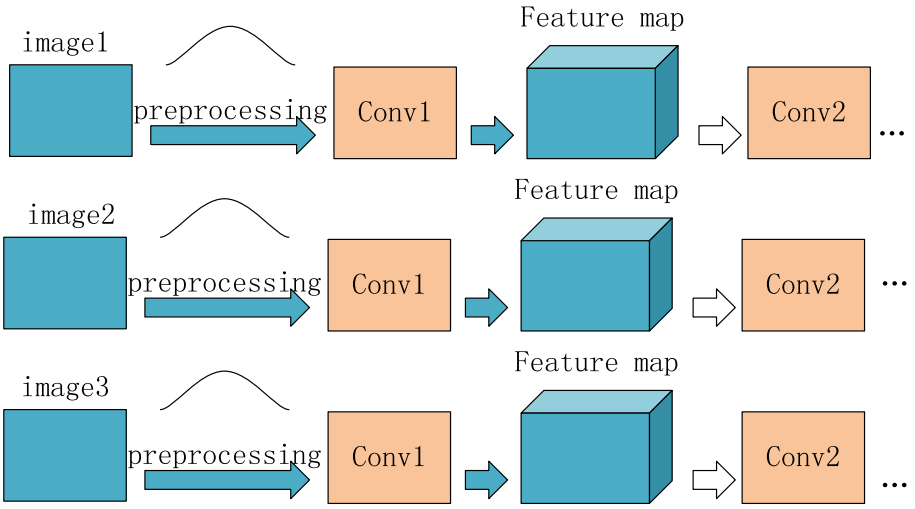

Fig. 5. Convolution example

Normalization process. C in (5) is used to adjust the variance size of the numerical distribution, and D is to adjust the location of the numerical mean. These two parameters are learned in the back propagation process, and the default value of C and D is 1 and 0.

$$\mu_A \leftarrow \frac{1}{n}\sum_{i=1}^{n} a_i \tag{2}$$

$$\sigma_A{}^2 \leftarrow \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu_A)^2 \tag{3}$$

$$\widehat{a_i} \leftarrow \frac{a_i - \mu_A}{\sqrt{\sigma_A{}^2 + \varepsilon}} \tag{4}$$

$$O_i \leftarrow C\widehat{a_i} + D = BN_{C,D}(a_i) \tag{5}$$

The training sample set corresponding feature map data to satisfy the distribution, to calculate the feature map of the entire training set and then standardizing, for a large data set is obviously not possible, so we calculate a Batch data feature map and then standardize, The larger the batch is, the closer it is to the distribution of the entire data set, the better the effect is. As is shown in (2), μ_A represents the mean value of the feature map, and each element of μ_A vector represents the mean value of a dimension (channel). $\sigma_A{}^2$ represents the variance of the feature map, and each element of $\sigma_A{}^2$ vector represents the variance value of a dimension (channel). Then according to μ_A and $\sigma_A{}^2$ and through Eq. (4), do the standardized calculation to get the final value.

The Fig. 6 shows the calculation process of the Batch Normalization with a batch size of 2, the feature1 and feature2 are feature matrix obtained by a series of convolution and pooling of image1 and image2 respectively. The channel of feature is 2, then $a^{(1)}$ represents the data of Channel 1 of all features of the batch ($a^{(1)} = \{1, 1, 1, 2, 0, -1, 2, 2\}$), Similarly, $a^{(2)}$ represents the data of Channel2 of all features

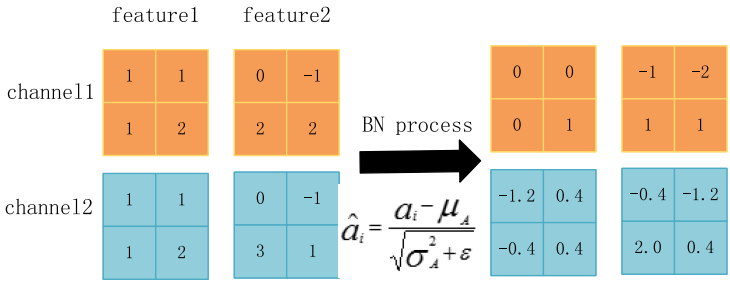

Fig. 6. Batch normalization process

of the batch ($a^{(2)} = \{-1, 1, 0, 1, 0, -1, 3, 1\}$). Then calculate the mean and variance of $a^{(1)}$ and $a^{(2)}$ respectively to get μ_A and ${\sigma_A}^2(\mu_A = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix}, {\sigma_A}^2 = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix})$.

The value of each channel is then calculated according to Eq. (4). In the process of training the network, the data is trained from one batch to one batch. In the course of the training we're going to keep counting the mean and the variance of each batch, and using the method of moving average to record the mean and the variance of the statistics, and after we've trained, we can approximate the mean and the variance of the statistics that we're going to be equal to the mean and the variance of our entire training set. Then, in our validation and prediction, we use the mean and variance of the mean and variance that we count to standardize the process.

3.4 Rectified Linear Unit

$$\sigma_{\mathrm{ReLU}}(z) = \max(0, z) \tag{6}$$

ReLU is an activation function. Why the activation function was introduced? The output of each layer is the linear function of the upper input, no matter how many layers of the neural network, the output is the linear combination of the input, and no hidden layer effect. Therefore, the introduction of the nonlinear function as the activation function [28–30], so that the deep neural network is intentional, no longer is a linear combination of input, which can approach any function. The original idea is the Sigmoid function or the Tanh function, which has a bounded output and can easily be entered as the next layer. In the ReLU [31], less than 0 is directly placed in 0, which is greater than 0 as input. This is the realization of the nonlinear transformation. The partial gradient of greater than 0 is 1. So the gradient of the activation function is always 1 for the information that needs to be passed from the input [32]. Even if continuous multiplication is not smaller, ReLU solves the problem of gradient disappearing [33], which is a powerful way to help train rapid convergence. According to the definition of ReLU, information can only be propagated in the region of ReLU greater than 0 (forward and backward), which brings another advantage is the sparseness [34], reducing the interdependence of the parameters and reducing the occurrence of the overfitting problem. The sparse ability not only helps the performance of the network, but also from the perspective of the neuroscience, the activation rate of the neurons is low [35], which is also a simulated simulation. In addition, the activation function such as Sigmoid has exponential operation in Forward

propagation and when calculating the error gradient in Back propagation, the derivative involves the division [36, 37]. While ReLU activation function saves a lot of calculation in the whole process [38].

As is shown in Fig. 7, when the input is less than 0, the output is 0. When the input is greater than 0, the output value is the value of the input.

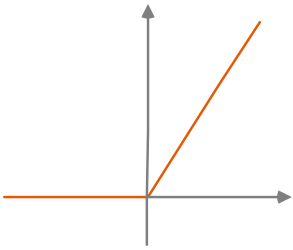

Fig. 7. Rectified linear unit

3.5 ResNet-18

As the number of layers in the network deepens, gradient disappearance or gradient explosion will occur. In other words, as the number of layers increases, the gradient of back propagation in the network will become unstable with continuous multiplication and become particularly large or small. It can be solved by data standardization, weight initialized, and Batch Normalization. However, with the deepening of layers, there is also a degradation problem, that is, deeper networks have higher training set errors, which can be solved by the residual structure in ResNet-18 [39, 40].

When we simply stack the network directly to a long length, the features inside the network have reached their best at a certain layer, and the remaining layers should not make any changes to the features and automatically learn the form of identity mapping. Compared with the shallow network, deeper networks should not have worse effects, but this is not the case for the degradation of network. What we need to do is to make the deep network achieve at least the performance of shallow networks under the network degradation, and make the layer behind the deep network achieve the identity mapping. Therefore, the residual structure is proposed to help the identity mapping of networks.

Due to the existence of many residual modules, the connections of some neural layers are weakened and reduced, and the linear transmission of the interlayer is realized instead of blindly pursuing nonlinear relations. The model itself can "tolerate" deeper neural networks. In terms of performance, additional residual modules will not degrade the performance of the Big NN.

As is shown in Fig. 8, the straight line forward is the residual mapping, and the curved line is the identity mapping, so the final output is $y = F(x) + x$. $F(x)$ is obtained from the input after a series of convolution, BN operations, and activation of the function. And x is the value of the input. The downsample uses the 1 * 1 convolution kernel, which makes the latitudes equal.

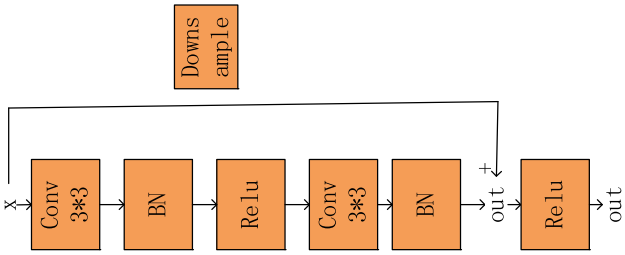

Fig. 8. Basic block

As is shown in Fig. 9, the ResNet-18 first goes through a 7 * 7 convolutional layer, then through a 3 * 3 maximum pooled subsampling, then through a series of residual structures, and finally through average pooled subsampling and full connection layer [41].

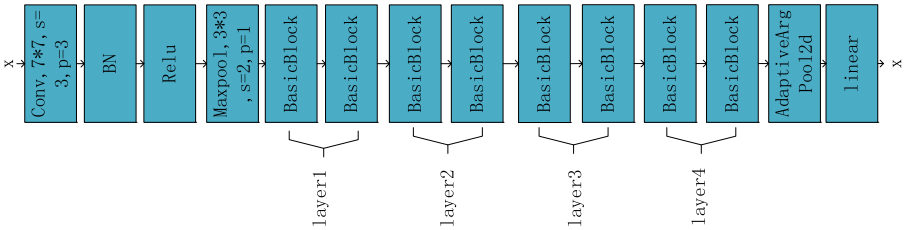

Fig. 9. ResNet-18

4 Experiment Result and Discussions

4.1 Statistical Analysis

The sensitivity analysis is shown in Table 1. The data is closely related to the facial muscles that correspond to expressions, with lip funneler and nose wrinkles making their early facial expressions similar, and expressions such as jaw drop and the upper lid raiser also make expressions based on the same muscle movement characteristics look similar.

Table 1 shows the sensitivity analysis of the seven emotion classes running for 10 times. According to the data from Table 1, the sensitivity of each expression is as follows: $95.20 \pm 2.35\%$, $95.70 \pm 2.16\%$, $94.90 \pm 1.66\%$, $93.60 \pm 2.88\%$, $95.30 \pm 2.67\%$, $93.20 \pm 3.58\%$, $95.70 \pm 1.34\%$. From this we can get: the expression of Disgust is the most sensitive and easy to recognize, followed by the expression of Surprise, and the third is the expression of Neutral. According to Table 2, the overall average accuracy of the system after 10 runs is $94.80 \pm 1.43\%$.

Table 1. Statistical analysis on the sensitivities of each class

	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
Run 1	97.00	92.00	94.00	89.00	97.00	95.00	95.00
Run 2	97.00	97.00	97.00	97.00	97.00	97.00	98.00
Run 3	92.00	92.00	96.00	93.00	98.00	87.00	97.00
Run 4	97.00	96.00	97.00	97.00	97.00	96.00	96.00
Run 5	97.00	96.00	95.00	96.00	93.00	96.00	94.00
Run 6	96.00	98.00	94.00	92.00	90.00	96.00	95.00
Run 7	95.00	95.00	96.00	94.00	96.00	95.00	97.00
Run 8	93.00	96.00	95.00	92.00	92.00	90.00	95.00
Run 9	91.00	97.00	92.00	90.00	96.00	89.00	94.00
Run 10	97.00	98.00	93.00	96.00	97.00	91.00	96.00
Average	95.20 ± 2.35	95.70 ± 2.16	94.90 ± 1.66	93.60 ± 2.88	95.30 ± 2.67	93.20 ± 3.58	95.70 ± 1.34

Table 2. Statistical analysis on the overall accuracies

Run	OA
1	94.14
2	97.14
3	93.57
4	96.57
5	95.29
6	94.43
7	95.43
8	93.29
9	92.71
10	95.43
Average	94.80 ± 1.43

4.2 Comparison with State-of-the-Art Approaches

The OA of the "ResNet-18" method used in this experiment was compared with that of the other three methods, which were HWT [3], CSO [6] and BBO [7]. The results are shown in Table 3: OA of HWT [3] is 78.37 ± 1.50%; OA of CSO [6] is 89.49 ± 0.76%; OA of BBO [7] is 93.79 ± 1.24%. We can clearly see that the method of "ResNet-18" has the highest accuracy (94.80 ± 1.43%), followed by BBO [7], and the third highest accuracy is CSO [6], while the lowest accuracy is HWT [3].

It can be seen from Table 1 that the highest OA obtained by "ResNet-18" method mainly depends on: (i) the ability of CNN to extract image features; (ii) the excellent training ability of ResNet-18. And the next best method is BBO [7], which comes from the theory of biogeography and is a swarm intelligence optimization algorithm based on the

general rules of migration and variation of different populations of different organisms in different habitats. The third best method is CSO [6], which mainly combines the seeking mode and the tracing mode in the algorithm through mixture ratio to achieve global optimization.

We should note that there are currently several variants of ResNet, such as Wide Residual Network (WRN), ResNeXt and MobileNet. In the future research, we will test their performances.

Table 3. Comparison with State-of-the-art methods

Method	OA
HWT [3]	78.37 ± 1.50
CSO [6]	89.49 ± 0.76
BBO [7]	93.79 ± 1.24
ResNet-18 (Ours)	94.80 ± 1.43

5 Conclusion

In this content, we propose an improved facial emotion recognition system. We use ResNet-18 for feature extraction. The facial emotion recognition system has achieved good recognition effect. In the future research, we will continue to focus on the research of facial emotion recognition and try to collect more emotional images than in this experiment, so as to optimize and propose a better algorithm to train the hyperparameter of the neural network, such as the weights and biases. And we will also try such optimization algorithms based on ResNet-18 to improve the performance of neural network.

References

1. Oji-Mmuo, C.N., Speer, R.R., Gardner, F.C., Marvin, M.M., Hozella, A.C., Doheny, K.K.: Prenatal opioid exposure heightens sympathetic arousal and facial expressions of pain/distress in term neonates at 24–48 hours post birth. J. Maternal-Fetal Neonatal Med. **33**, 3879–3886 (2020)
2. Ali, H., Hariharan, M., Yaacob, S., Adom, A.H.: Facial Emotion recognition based on higher-order spectra using support vector machines. J. Med. Imaging Health Inf. **5**, 1272–1277 (2015)
3. Evans, F.: Haar wavelet transform based facial emotion recognition. Adv. Comput. Sci. Res. **61**, 342–346 (2017)
4. Lu, H.M.: Facial emotion recognition based on biorthogonal wavelet entropy, fuzzy support vector machine, and stratified cross validation. IEEE Access **4**, 8375–8385 (2016)
5. Phillips, P.: Intelligent facial emotion recognition based on stationary wavelet entropy and Jaya algorithm. Neurocomputing **272**, 668–676 (2018)

6. Wang, S.-H., Yang, W., Dong, Z., Phillips, P., Zhang, Y.-D.: Facial emotion recognition via discrete wavelet transform, principal component analysis, and cat swarm optimization. In: Sun, Yi., Lu, H., Zhang, L., Yang, J., Huang, H. (eds.) IScIDE 2017. LNCS, vol. 10559, pp. 203–214. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67777-4_18
7. Li, X.: Facial emotion recognition via stationary wavelet entropy and biogeography-based optimization. EAI Endorsed Trans. e-Learn. **6**, Article ID: e4 (2020)
8. Lv, Y.-D.: Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling. J. Med. Syst. **42**, Article ID: 2 (2018)
9. Tang, C.: Twelve-layer deep convolutional neural network with stochastic pooling for tea category classification on GPU platform. Multimed. Tools Appl. **77**, 22821–22839 (2018)
10. Pan, C.: Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling. J. Comput. Sci. **27**, 57–68 (2018)
11. Hasebe, T., Ueda, Y.: Unimodality for free multiplicative convolution with free normal distributions on the unit circle. J. Oper. Theory **85**, 21–43 (2021)
12. Belinschi, S.T., Bercovici, H., Liu, W.H.: The atoms of operator-valued free convolutions. J. Oper. Theory **85**, 303–320 (2021)
13. Kumar, S., Mahadevappa, M., Dutta, P.K.: Lensless in-line holographic microscopy with light source of low spatio-temporal coherence. IEEE J. Sel. Top. Quantum Electron. **27**, 8, Article ID: 6800608 (2021)
14. Fujioka, T., Yashima, Y., Oyama, J., Mori, M., Kubota, K., Katsuta, L., et al.: Deep-learning approach with convolutional neural network for classification of maximum intensity projections of dynamic contrast-enhanced breast magnetic resonance imaging. Magn. Reson. Imaging **75**, 1–8 (2021)
15. Hou, X.-X.: Seven-layer deep neural network based on sparse autoencoder for voxelwise detection of cerebral microbleed. Multimed. Tools Appl. **77**, 10521–10538 (2018)
16. Pan, C.: Multiple sclerosis identification by convolutional neural network with dropout and parametric ReLU. J. Comput. Sci. **28**, 1–10 (2018)
17. Bercovici, H., Dykema, K., Nica, A.: Dan-virgil voiculescu at seventy. J. Oper. Theory **85**, 5–20 (2021)
18. Egger, H., Schmidt, K., Shashkov, V.: Multistep and Runge-Kutta convolution quadrature methods for coupled dynamical systems. J. Comput. Appl. Math. **387**, 14, Article ID: 112618 (2021)
19. Erbay, H.A., Erbay, S., Erkip, A.: A semi-discrete numerical method for convolution-type unidirectional wave equations. J. Comput. Appl. Math. **387**, 13, Article ID: 112496 (2021)
20. Katsagounos, I., Thomakos, D.D., Litsiou, K., Nikolopoulos, K.: Superforecasting reality check: evidence from a small pool of experts and expedited identification. Eur. J. Oper. Res. **289**, 107–117 (2021)
21. Huang, C.: Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling. Front. Neurosci. **12**, Article ID: 818 (2018)
22. Zhao, G.: Polarimetric synthetic aperture radar image segmentation by convolutional neural network using graphical processing units. J. Real-Time Image Proc. **15**, 631–642 (2018)
23. Muhammad, K.: Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimed. Tools Appl. **78**, 3613–3632 (2019)
24. Wang, S.-H., Sun, J.: Cerebral micro-bleeding identification based on a nine-layer convolutional neural network with stochastic pooling. Concurr. Comput. Pract. Exp. **32**, e5130 (2020)
25. Sangaiah, A.K.: Alcoholism identification via convolutional neural network based on parametric ReLU, dropout, and batch normalization. Neural Comput. Appl. **32**, 665–680 (2020)

26. Choi, S.H., Jung, S.H.: Stable acquisition of fine-grained segments using batch normalization and focal loss with L1 regularization in U-Net structure. Int. J. Fuzzy Logic Intell. Syst. **20**, 59–68 (2020)
27. Wang, S.-H.: DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. ACM Trans. Multimed. Comput. Commun. Appl. **16**, Article no. 60 (2020)
28. Olimov, B., Karshiev, S., Jang, E., Din, S., Paul, A., Kim, J.: Weight initialization based-rectified linear unit activation function to improve the performance of a convolutional neural network model. Concurr. Comput. Pract. Exp. **11** (2021). (Article; Early Access). https://doi.org/10.1002/cpe.6143
29. Zhang, Y.-D.: Advances in multimodal data fusion in neuroimaging: overview, challenges, and novel orientation. Inf. Fusion **64**, 149–187 (2020)
30. Wang, S.-H.: Covid-19 classification by FGCNet with deep feature fusion from graph convolutional network and convolutional neural network. Inf. Fusion **67**, 208–229 (2021)
31. Yaliniz, G., Ikizler-Cinbis, N.: Using independently recurrent networks for reinforcement learning based unsupervised video summarization. Multimed. Tools Appl. **80** (2021). (Article; Early Access). https://doi.org/10.1007/s11042-020-10293-x
32. Kawahara, D., Tang, X.Y., Lee, C.K., Nagata, Y., Watanabe, Y.: Predicting the local response of metastatic brain tumor to gamma knife radiosurgery by radiomics with a machine learning method. Front. Oncol. **10**, 8, Article ID: 569461 (2021)
33. Dubey, S.R., Chakraborty, S.: Average biased ReLU based CNN descriptor for improved face retrieval. Multimed. Tools Appl., 26 (2021)
34. Yamaguchi, M., Iwamoto, G., Nishimura, Y., Tamukoh, H., Morie, T.: An energy-efficient time-domain analog CMOS BinaryConnect neural network processor based on a pulse-width modulation approach. IEEE Access **9**, 2644–2654 (2021)
35. Farrell, M.H., Liang, T.Y., Misra, S.: Deep neural networks for estimation and inference. Econometrica **89**, 181–213 (2021)
36. Tripathi, D., Edla, D.R., Kuppili, V., Bablani, A.: Evolutionary extreme learning machine with novel activation function for credit scoring. Eng. Appl. Artif. Intell. **96**, 10, Article ID: 103980 (2020)
37. Satapathy, S.C.: A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. Mach. Vis. Appl. **32**, Article ID: 14 (2021)
38. Moon, S.: ReLU network with bounded width is a universal approximator in view of an approximate identity. Appl. Sci. **11**, 11, Article ID: 427 (2021)
39. Bernardo, P.P., Gerum, C., Frischknecht, A., Lubeck, K., Bringmann, O.: UltraTrail: a configurable ultralow-power TC-ResNet AI accelerator for efficient keyword spotting. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **39**, 4240–4251 (2020)
40. Alotaibi, B., Alotaibi, M.: A hybrid deep ResNet and inception model for hyperspectral image classification. PFG J. Photogramm. Remote Sens. Geoinf. Sci. **88**, 463–476 (2020)
41. Hammad, M., Plawiak, P., Wang, K.Q., Acharya, U.R.: ResNet-attention model for human authentication using ECG signals. Expert Syst., 17, Article ID: e12547 (2020)