EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

A facial expression recognizer using modified ResNet-152

Wenle Xu[1,*], Rayan S Cloutier[2]

[1]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, P R China
[2]Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada

Abstract

In this age of artificial intelligence, facial expression recognition is an essential pool to describe emotion and psychology. In recent studies, many researchers have not achieved satisfactory results. This paper proposed an expression recognition system based on ResNet-152. Statistical analysis showed our method achieved 96.44% accuracy. Comparative experiments show that the model is better than mainstream models. In addition, we briefly described the application of facial expression recognition technology in the IoT (Internet of things).

Keywords: Facial expression recognition, ResNet-152, Recognition system.

Received on 28 February 2022, accepted on 08 April 2022, published on 27 April 2022

doi: 10.4108/eetiot.v7i28.685

1. Introduction

Facial expression is the result of position and movement of facial muscles. It conveys human emotional information through these movements. Therefore, facial expression is regarded as a form of non-verbal communication. In short, facial expression is a physical and psychological response of the body, which is usually used to convey emotion. In interpersonal communication, human can enhance the effect of communication by controlling their facial expressions. Facial expression recognition (FER) plays an important role in the field of human-computer interaction. In order to make the interaction more convenient, the computer must have FER ability. For the cutting-edge pages that need to communicate with human users, we need to utilize a FER system for situational understanding [1]. In this paper, we construct a facial expression recognizer based on ResNet-152.

Neural network has a strong nonlinear fitting ability for all kinds of data. But for row data, such as speech information, image information. It is difficult to get the ideal result by machine learning. The traditional method to extract features from images is to use the features created by hand. When we encounter the problem image processing, it is hard for us to extract features from the original pixels to describe the image. Until the emergence of Deep Neural Network (DNN), the problem of image processing has been well solved. However, one of the disadvantages of DNN is that it has a large number of parameters, so it is a great challenge to update parameter. Another disadvantage is that the number of parameters is easy to cause overfitting problems. Then comes the Convolution Neural Network (CNN). It has the characteristic of parameter sharing, which can greatly reduce parameters. As a result, CNN can not only extract features, but also control the amount of computation. This may also be the reason why CNN [2] has always been popular in the field of image processing [3, 4]. In general, CNN consists input layer, convolution layers, pooling layers, fully-connected layers. The convolution layer is mainly used to extract the features of image, which may be local features or combine local features. The pooling layer is used to reduce the data dimension and does not involve additional parameters. The fully-connected layer prepares for out. Simonyan [5] utilized an model with small convolution kernels to increase the depth of network. These works achieved very good results in the classification and localization tasks. He [6] proposed ResNet for classification tasks. This network model solves the problem of degradation caused by the deepening of the network.

FER aims to take an image as input, then analyze the image through a model, and finally divide the image to

[*]Corresponding author. Email: xwl@home.hpu.edu.cn

specific class. Classification of facial expression according to reference [7], expressions can be grouped into seven types: happy, sadness, fear, anger, surprise, disgust, and neutral. The whole process of recognition can be roughly divided into four parts: The first part is that we need to collect images for training. In the second part, these images are processed. The third part uses a model for feature extraction. The fourth part classifies the images. The third part and the fourth part are particularly important. Some researchers have done many experiments. For example, Ali [8] employed the radon transform (RT) and the traditional SVM method. Lu and Evans [9] proposed to use Haar wavelet transform (HWT) method. Yang [10] introduced cat swarm optimization (CSO) and achieved 89.49% accuracy. Li used ResNet-18 [11] and ResNet-50 [12] for feature extraction and classification. The methods mentioned above further improve the accuracy of model. Through the summary of the above literature, we find that that the model will lose some image information in the process of feature extraction. Therefore, assuming that we can extract more complete features from the image, the recognition accuracy can be further improved.

In this paper, we propose a novel FER algorithm based on deep learning. The main contributions of this paper are as follows:

(i) A Modified ResNet-152 is proposed.

(ii) Retraining weights were used for facial data.

(iii) The recognition system was better than the state-of-the-art methods.

The improvement of facial recognition technology in accuracy and stability shows great power in the field of IoT [13, 14]. For example, in health care, finance, transportation and so on. Thus, we must create some algorithms and master some core technologies. Only in this way can we promote the development of the IoT field.

2. Dataset

We adopted the data set [15]. The dataset includes seven facial emotion images: happy, sadness, fear, anger, surprise, disgust and neutral. There are 100 images for each type of expression. We have 700 images in total. These images are taken by professional photographer using Canon digital camera. Figure 1 displays seven emotion classes of a male and a female faces.

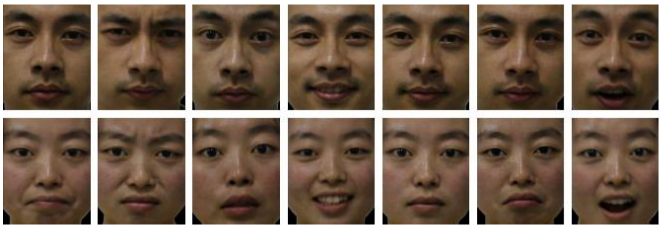


Figure 1. Samples of our dataset

3. Methodology

3.1. Input layers

When we recognize an image, we use the image as input, but we don’t input all the raw data of an image. Before inputting the data, we preprocess the data. Preprocessing operations usually include de-averaging, normalization, de-correlation and de-whitening.

3.1.1 De-averaging operation

There are two main ways for De-averaging. Suppose we have ten images with a size of $100\times100\times3$, then we sum the corresponding pixels of ten images and calculate the average. This is a way. We directly calculate the mean value of the three color channels of RGB. When there is a new image as input, subtract the corresponding average from the R channel. Other channels are also same operation. This is another way. If we don’t de-averaging, the model will be easily fitted. This operation result in all dimensions of the input data to be centralized to 0.

3.1.2 Normalization

If the range of each feature is different, it will have a bad impact on the optimization algorithm. For example, the data of one feature is between 1000 and 1500, and the data of another feature is between 1 and 10. This gap will have a negative impact on training, so we will scale the image. This operation includes two types, one is the normalization of the maximum value, the other is the normalization of the mean variance. The former is suitable for data distributed in a limited range. For example, the maximum is normalized to 1, and the minimum is normalized to 0. The whole process will follow the following formula:

$$x_{i,j}^{*}=\frac{x_{i,j}-x_{j}^{\min}}{x_{j}^{\max}-x_{j}^{\min}} \qquad (1)$$

x_j^{min} represents minimum of j th column in the pixel matrix. x_j^{max} represents maximum of j th column in the pixel matrix. $x_{i,j}$ represents pixel at the position of the pixel matrix (i,j) . $x_{i,j}^{*}$ represents the value at the normalized position of pixel matrix (i,j).

The latter is suitable for cases where the distribution has no obvious boundary. In most cases, the mean is normalized to 0, and the variance is normalized to 1. The whole process satisfies the following formula:

$$x_{new}=\frac{x_i-x_{mean}}{\sigma(x)} \qquad (2)$$

x_{new} represents the normalized value. σ(x) represents the standard deviation. x_{mean} represents the mean value. x_i represents the set of i th features.

The advantage of this operation is that each characteristic scale is controlled in the same range, so that the convergence time can be shortened and the optimal solution can be found easily. This operation is also the most common preprocessing operation.

3.1.3 De-correlation and De-whitening

Principal Component Analysis (PCA) is used to De-correlation. The core idea of PCA is to use a small number of representative and unrelated features to replace a large number of related features, so as to accelerate the training process. PCA can also achieve the purpose of reducing dimension.

In a static image, each pixel has a certain correlation. Therefore, it is redundant for a large of pixels. At this time, the whitening operation can be used to de-correlation. This operation has the following advantages. 1. Reduce the correlation between features. 2. Features have the same variance.

3.2. Convolution layers

When the variables of convolution are functions $f(x)$ and $g(x)$ the convolution formula is

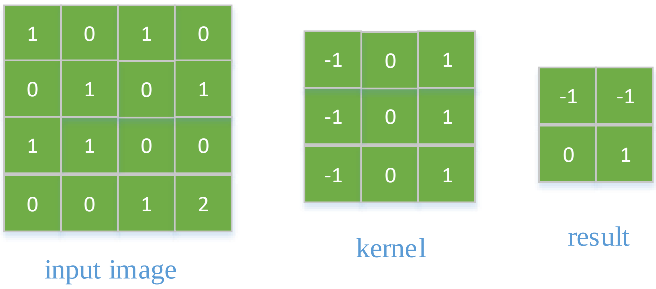
$$h(x) = f(x) * g(x) = \int_{-\infty}^{+\infty} f(\mu) g(x-\mu) \mathrm{d}\mu \qquad (3)$$

μ is the integral variable, x is the amount of displacement of function $g(-\mu)$, and the "*" denotes convolution. When the variables of convolution are sequence $l(y)$ and $k(y)$, the convolution formula is

$$s(y) = l(y) * k(y) = \sum_{\mu=-\infty}^{\infty} l(\mu) k(y-\mu) \quad (4)$$

If the variable y is zero, $k(-\mu)$ is the result of inverse sequence μ of $k(\mu)$, the "*" denotes convolution.

Convolution operation plays an essential role in neural networks. As is shown in (3), $f(x)$ and $g(x)$ denote two integrable functions [16]. In the application of image processing, $f(x)$ denotes the pixels from image, $g(x)$ denotes parameters from convolution kernels [17, 18]. In fact, the result of convolution processing is to take into account the surrounding pixels of each pixel, or even the entire image pixels, and carry out some kind of weighted processing on the current pixels to achieve a certain purpose. The explanations of the use of the convolution operation can be found in Refs [19-21]. The process is shown in Figure 2.



1	0	1	0
0	1	0	1
1	1	0	0
0	0	1	2

-1	0	1
-1	0	1
-1	0	1

-1	-1
0	1

Figure 2. Convolution process

According to Figure 2, the left represents the pixel matrix of the image, with a size of 4×4. The middle represents the convolution kernel, with a size 3×3 and a stride of 1. The right represents the output. For the specific calculation process, we take the first pixel "-1" in the upper left of the result matrix as an example.

$$\sum_{i=1}^{3} \sum_{j=1}^{3} x_{ij} w_{ij} = -1 \qquad (5)$$

x_{ij}represents the corresponding pixel at the position of input image (i, j), w_{ij} represents the corresponding weight at the position of kernel (i, j).

When we understand the convolution operation, the reason why the number of CNN parameters is much lower than that of DNN parameters is self-evident. The reason for too many DNN parameters is that when the neuron is connected with each node in the previous layer, there will be a different parameter, so the number of parameters is huge. But for CNN, the parameters inside the convolution kernel are fixed. We will convolution kernel to traverse every position of the image. The number of our parameters does not change during the traversal. This is the property that convolution neural network has shared parameters. These parameters need to be learned. Generally speaking, the convolution operation not only reduces the number of parameters, but also extracts features from the image.

3.3. Pooling layer

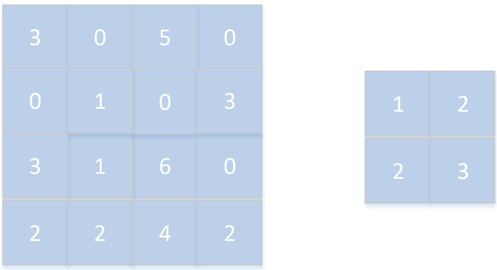
The convolution layer is followed by pooling layer. Why can't the results after convolution be directly used for classification? After convolution operation, we get some feature maps. The dimensions of these feature maps are high, and if they are directly connected to the fully-connected layer, it will lead a large amount of computation that can't be underestimated. Therefore, the pooling layer reduces the dimension of features and solves this problem very well [22, 23]. The other reason is adjacent pixels in the image tend to have similar values, so adjacent output pixels in the convolution layer usually have similar values. This means that most of the information contained in the output of the convolution layer is redundant. We also use pooling layer to solve this problem. What′s more, pooling can prevent overfitting to a certain extent and make optimization more convenient [24]. Last but not the least, pooling can achieve some invariance, such as rotation invariance, translation invariance and contraction invariance. Translation invariance [25-27] means that the vector translation of the output remains basically unchanged to the input. For example, the input is vector (1, 2, 5), the result of max-pooling is 5. If we shift the input one to the right to get (0, 1, 5), the result of the output is still 5. To put it simply, CNN can identify same results for the image and its translated version. Therefore, this property is beneficial to the classification task. In addition, the pooling layer can improve the ability of model feature extraction [28, 29]. We introduce two kinds of pooling below.

3.3.1 Average Pooling

The average pooling [30] operation is to take the average of each block, extract the information of all the features from feature maps. Then pass average value to the next layer. When the image contains a lot of useful information, we usually use average pooling. For example, average pooling is often used before the full connection layer. This is because the last layers contain a wealth of semantic information. If we use max pooling, we will lose a lot of important information.

According to Figure 3, the left represents the pixel matrix of the image, with a size of 4×4. The right represents the result after average pooling. From the result, we can see that the size of kernel is 2×2 and the stride is 2. In addition, the dimension of the feature map changes from 4 to 2. For the specific calculation process, we take the first pixel "1" in the upper left of the result matrix as an example.

$$(x_{11} + x_{12} + x_{21} + x_{22}) / 4 = (3+0+0+1) / 4 = 1 \quad (6)$$



3	0	5	0
0	1	0	3
3	1	6	0
2	2	4	2

1	2
2	3

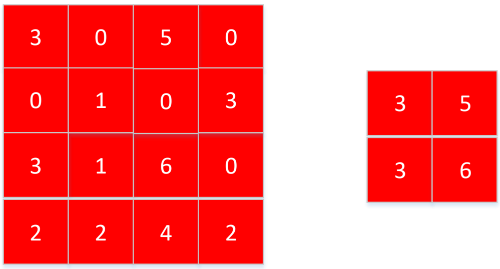
Figure 3. Average Pooling process

3.3.2 Max Pooling

The max pooling operation takes the maximum value in each block, while other pixels will not pass to the next layer. Academically, max pooling is to extract a feature in any quadrant [31-33], then its maximum value will be obtained. If this feature is not mentioned, it may not exist in this quadrant, then the maximum value is still very small [34-36]. When the image contains a small amount of useful information, we usually use max pooling. For example, in the first few layers of the network, images with noise, etc. The reason why people use max pooling is that this method works well in many experiments.

According to Figure 4Figure 4 , the left represents the pixel matrix of the image, with a size of 4×4. The right represents the result after max pooling. From the result, we can see that the size of kernel is 2×2 and the stride is 2. In addition, the dimension of the feature map changes from 4 to 2. For the specific calculation process, we take the first pixel "3" in the upper left of the result matrix as an example.

$$\max(x_{11}, x_{12}, x_{21}, x_{22}) = \max(3,0,0,1) = 3 \qquad (7)$$



3	0	5	0
0	1	0	3
3	1	6	0
2	2	4	2

3	5
3	6

Figure 4. Max Pooling process

In the process of forward propagation, the maximum pooling needs to record the location of the maximum value. Because the gradient only comes from this maximum value, the gradient update only updates this maximum value, and the gradient at other locations is 0. It is not necessary for average pooling. The loss comes from each element in the feature map above. So the gradient is divided by the size of the block.

3.4. Activation function

Activation functions are divided into two categories: one is saturated function, the other is non-saturated function. Saturated functions include Tanh, sigmoid and so on. Non-saturated functions include ReLU and its variants. Why the activation function was introduced? The reason for introducing activation function: If activation functions are not used, in this case, the output is a linear combination of inputs. The result is equivalent to the effect of no hidden layers [37]. This is the most primitive Perceptron situation [38]. Let's introduce the saturated activation function:

Let $h(x)$ be an activation function. When x approaches positive infinity and the derivative of the activation function approaches 0, we call it right saturation. The formula is as follows

$$\lim_{x\to+\infty} h'(x) = 0 \quad (8)$$

When x approaches negative infinity and the derivative of the activation function approaches 0, we call it left saturation. The formula is as follows

$$\lim_{x\to-\infty} h'(x) = 0 \quad (9)$$

When a function satisfies both left and right saturation, we call it saturation. A function that does not meet the above conditions is called a non-saturated activation function. The most commonly used is the ReLU function. It is shown in Figure 5 and (10).

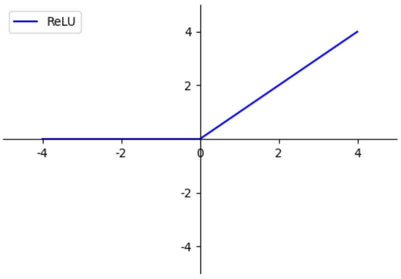


Figure 5 ReLU function

$$y = \max(0, x) \quad (10)$$

As is shown in (10), x denotes the output of the previous convolution layer, y denotes the output through the activation function. In Figure 5, the horizontal axis denotes x and the vertical axis denotes y. In addition, in the process of training weights, we need to utilize the derivative of activation function. The derivative of ReLU is as follows:

$$y' = \begin{cases} 0, & x < 0 \\ 1, & \text{otherwise} \end{cases} \qquad (11)$$

ReLU function has the following advantages: 1. The problem of gradient vanishing gradient in back propagation is solved. 2. The speed of promoting the convergence of neural network is much faster than sigmoid and Tanh. 3. The calculation speed is very fast. The user needs to check if the input is greater than 0.

3.5. Fully-connected layer

The fully-connected layer means that all neurons have weighted connections between the two layers. Because of the large amount of calculation, it is usually placed at the tail of the neural network. In practical application, the fully connected layer can be regarded as implemented by convolution operation. That is, 1×1 convolutions performed on the previous layer. In addition, it will act as a classifier. The structure is shown in Figure 6.

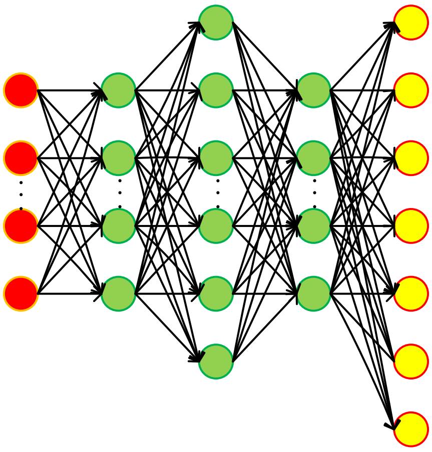


Figure 6. The structure of fully-connected layer

3.6. The method of training weights

We first define a loss function

$$L = \sum_{i=1}^{n} (y_i - f(x, w))^2 \quad (12)$$

where y_i denotes the true value, and f denotes the predicted value. In general, the smaller the loss function, the higher the model's accuracy. If we want to improve the accuracy of the learning model, we should reduce the value of the loss function as much as possible. We use gradient descent to optimize the value of the loss function. In the gradient descent process, we use the following two equations to iterate until the convergence of the loss function remains unchanged.

$$w_{i+1} = w_i - \alpha * \frac{\partial L}{\partial w_i} \quad (13)$$

There are many ways to train neural networks. For example, Stochastic Gradient Descent (SGD) algorithm, Back Propagation (BP) algorithm. SGD [39] is more effective than BP. If we use SGD for training, then be sure to use batch normalization. With the process of the training process, the distribution of input data in subsequent layers will change. We call this phenomenon as "Internal Covariate Shift" [40]. Batch normalization can alleviate this problem. What's more, it can also solve the problem, which needs to set some hyper-parameter artificially.

3.7. Modified ResNet-152

When the network is deepened, the representation ability of the model will become stronger. In other words, the model can extract more features from image. But network is hard to train. The first problem encountered during training is gradient vanishing/ exploding. This is because with the increase of the number of layers, the amount of calculation increases rapidly. The gradient becomes unstable in the back propagation process. For the problem of gradient vanishing, researchers proposed many solutions, such as batch normalization, initialization of MSRA+BN and so on. Another problem is the network degradation, that is, with the increase of depth of network, the performance of model becomes poor. Specially, the accuracy of the training set will be reduced. We can be sure that this is not caused by overfitting. Because in the case of overfitting, the accuracy of the training set should be very high. ResNet solve this obstacle through residual learning [6]. A building block is shown in Figure 7.

According to Figure 7, we can see that ResNet provides two choices, identity mapping, and residual mapping. We suppose the input of network has reached the optimal level. In other words, x is optimal. When we continue to deepen the network, the residual mapping will be set to 0. At this time, the network only has identity mapping. This is the reason why the network has always been in an optimal state [41]. The most important thing is that deepening of the network will not decrease the performance of the model.

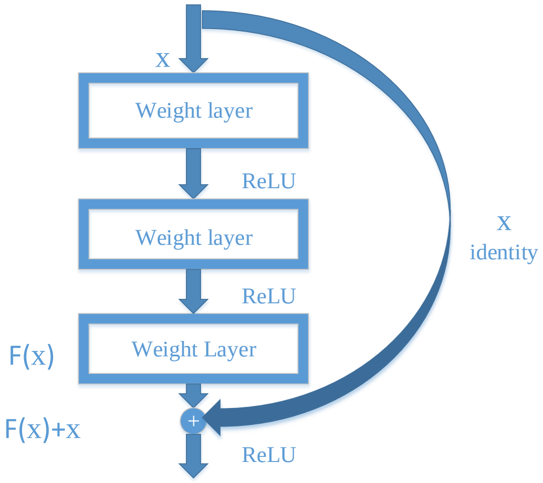


Figure 7. The building block

Formally, $H(x)$ denots the original mapping, we let the stacked nonlinear layers fit another mapping of $F(x) := H(x) - x$. Here $F(x)$ denotes the residual mapping. The original mapping transform into $F(x) + x$. To the extreme, if an identity mapping are optimal, at this time $H(x) = x$.

This paper uses Modified ResNet-152, a variant of ResNet-152, as a recognizer to classify facial expressions. In the Modified ResNet-152 architecture, the first layer is 7*7 convolutions. Then there are four building blocks with 9 layers, 24 layers, 108 layers and 9 layers respectively. Finally, perform a global average pooling, a 7-way fully connected layer and softmax. It is shown in Figure 8

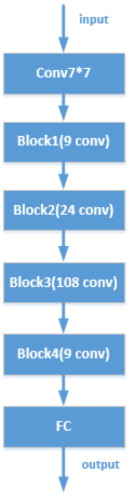


Figure 8. Modified ResNet-152

3.8. Measure

Cross-validation is also called Rotation estimation, which is a statistical method to cut data samples into smaller subsets. The core idea of cross-validation is to group the original dataset, one part as the training set and the other as the validation set or test set. The whole process is divided into two parts. In the first part, the classifier is trained with the training set. In the second part, the model is tested with the validation set. The purpose of using this technology is to obtain a reliable and stable model. In this way, the generalization ability of the model is greatly improved. In this paper, we used ten-fold cross validation.

The dataset has 700 images in total. Each fold includes 70 images. Each type of expression includes 10 images. We used 8 folds for training, 1 fold for validation, 1 fold for testing. In order to visualize the performance of the algorithm, we use the format of confusion matrix. We suppose

$$W = (w_{ij}), 1 \le i, j \le 7, i, j \in Z, w_{ij} \ge 0 \quad (14)$$

In the confusion matrix, r denotes the number of runs, d denotes the number of folds. When $r = 1, d = 10$, the ideal result is as follows:

$$W(r=1, d=10) = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (15)$$

w_{ii} denotes the correct number of expressions that are recognized for each type of expression. Therefore, in the ideal state, $w_{ii} = 100, i = 1, \ldots, 7$. In general, we run 10 times to avoid random and improve the generalization ability of the model. In other words, when $r = 10, d = 10$, we can get 10 confusion matrices. Ideally, we add up these 10 confusion matrices. The results are as follows:

$$W(r=10, d=10) = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \quad (16)$$

In practical problems, it's hard for us to make the model in an ideal state. We use the sensitivity and overall accuracy (OA) as the evaluation index of the model. We use the following two formulas to measure.

$$\mathrm{E}(t) = \frac{w_{tt}}{\sum_{n=1}^{7} w_{tn}} \quad (17)$$

$$\mathrm{OA} = \frac{\mathrm{tr}(W)}{\sum_{i=1}^{7} \sum_{j=1}^{7} w_{ij}} \quad (18)$$

$E(t)$ denotes the sensitivity of each class $t (1 \le t \le 7, t \in Z)$, which means the w_{tt} divided by the sum of elements in t th row of the confusion matrix. OA denotes the overall accuracy, which means the $\mathrm{tr}(W)$ divided by the sum of all elements of confusion matrix. $\mathrm{tr}(W)$ denotes the trace of the confusion matrix W. w_{ij} denotes the number of images belonging to class i and identified as class j.

4. Experiment Result and Discussions

4.1. Statistical Analysis

Table 1 shows the sensitivity analysis of each class. Figure 9 shows the trend of the sensitivity of class. From Table 1 and Figure 9, we can get the sensitivity of each expression is: 96.20+1.03%(anger), 96.40+1.58% (disgust), 96.80+1.03% (fear), 96.60+1.51% (happy), 96.20+1.48% (neutral), 96.50+1.35% (sadness), 96.40+1.96% (surprise). From this, we can get: that fear expression is most sensitive. In other words, the fear expression is easily recognized. The second most sensitive expression is happy. The third most sensitive expression is sadness. From Table 2, we can see the final overall accuracy is 96.44+0.56%.

Table 1. Statistical analysis on the sensitivity of each class

	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
Run1	97.00	97.00	97.00	97.00	96.00	97.00	96.00
Run2	96.00	97.00	96.00	95.00	95.00	98.00	97.00
Run3	94.00	96.00	96.00	94.00	97.00	94.00	98.00
Run4	96.00	95.00	97.00	96.00	97.00	97.00	98.00
Run5	97.00	98.00	97.00	98.00	97.00	97.00	98.00
Run6	96.00	94.00	98.00	98.00	97.00	97.00	98.00
Run7	97.00	98.00	98.00	95.00	95.00	98.00	97.00
Run8	95.00	94.00	95.00	98.00	98.00	95.00	95.00
Run9	97.00	98.00	96.00	97.00	97.00	97.00	92.00
Run10	97.00	97.00	98.00	98.00	93.00	95.00	95.00
Average	96.20+1.03	96.40+1.58	96.80+1.03	96.60+1.51	96.20+1.48	96.50+1.35	96.40+1.96

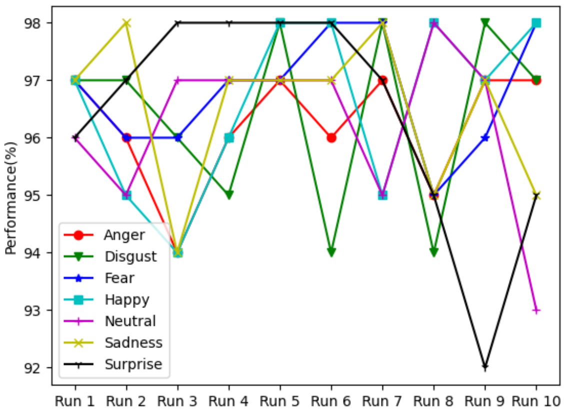


Figure 9. The trend of the sensitivities of class

Table 2. Statistical analysis on the overall accuracies

Run	OA
1	96.71
2	96.29
3	95.57
4	96.57
5	97.43
6	96.86
7	96.86
8	95.71
9	96.29
10	96.14
Average	96.44+0.56

4.2. Comparison with State-of-art Approaches

The state-of-art methods are HWT [9], CSO [10], and BBO [42], and the corresponding OA is 78.37+1.50%, 89.49+0.76%, and 93.79+1.24% respectively. The result of comparison is shown in Table 3. According to the data from Figure 10, we can obviously see that the method of “Modified ResNet-152” has 96.44+0.56% accuracy. The second-highest accuracy is BBO, which achieve 93.79 +1.24% accuracy. The third-highest accuracy is CSO, which achieve 89.49+0.76% accuracy. The lowest accuracy is HWT, which achieve 78.37+1.50% accuracy.

According to Table 1, “Modified ResNet-152” method get the highest OA mainly depends on (i) CNN can extract features from images at different scales; (ii) the problem of degradation can be well solved.

Furthermore, the method in second place is BBO, which solves the optimization problem. The core of BBO algorithm is migration and variation, which are important step to solve problem. In addition, the method in third place is CSO, which is a global optimization algorithm. The idea of CSO algorithm is to mimic the behaviors of cat.

We can use other classic network architectures, such as VGGNet and ResNet and their variants, and unexpected results may be.

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

Method	OA
HWT [9]	78.37+1.50
CSO [10]	89.49+0.76
BBO [42]	93.79+1.24
Modified ResNet-152	96.44+0.56

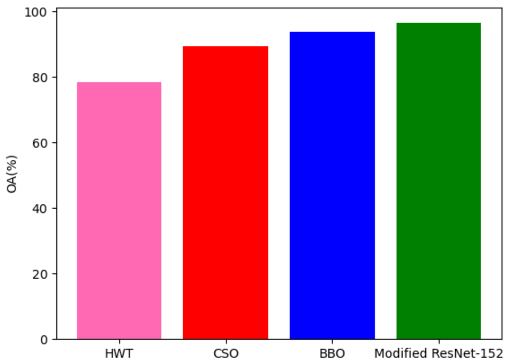


Figure 10. OA of State-of-the-art methods

4.3. Comparison with other ResNet variants

ResNet solves the problem of network degradation very well. Can we get a better model by deepening the ResNet network in the classification task? Li, et al. used ResNet-18 [11] and ResNet-50 [12] as facial expression classifiers to achieve the accuracy of 94.80+1.43%, 95.39+1.41%, respectively. Based on Table 4 and Figure 11, we can see that the “Modified ResNet-152” achieves higher accuracy.

Table 4. Comparison with other ResNet variants

Method	OA
ResNet-18 [11]	94.80+ 1.43
ResNet-50 [12]	95.39+ 1.41
Modified ResNet-152	96.44+0.56

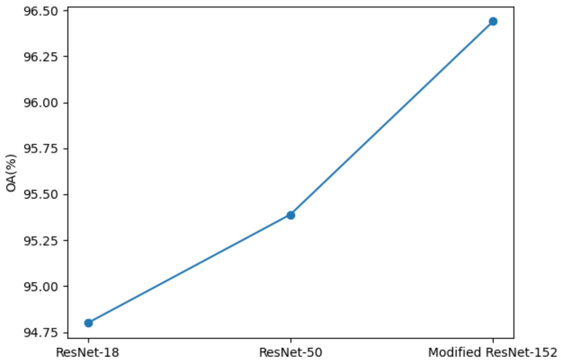


Figure 11. The trend of OA of ResNet variants

Swarm intelligence is a potential way to improve performance by optimizing hyper-parameters. We will test particle swarm optimization [43-45] and other global optimization methods [46, 47]. Software-defined networking

(SDN) [48] is an approach to networking that uses software-based controllers or application programming interfaces. In the future, we shall combine SDNs with this task.

5. Conclusion

This paper suggests a facial expression recognizer based on Modified ResNet-152. We show that our recognizer can classify human facial expression accurately. The low accuracy of some categories is considered that the number of images for the category is imbalanced.

Our further work starts from two aspects: on the one hand, we will continue to deepen the residual network's depth to improve the model's performance. On the other hand, human may have multiple expression types in one expression, we try to construct a system to recognize compound expressions. What' more, we try to optimize algorithms [49-51] to reduce the training time and improve the accuracy. Last but not least, if we can get a recognition model with high accuracy and good stability, we believe it will further promote the development of the IoT field.

References

[1] J. Jeon *et al.*, "A real-time facial expression recognizer using deep neural network," in *proceedings of the 10th international conference on ubiquitous information management and communication*, 2016, pp. 1-4.

[2] Y.-D. Zhang and Z.-C. Dong, "Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation," *Information Fusion,* vol. 64, pp. 149-187, 2020/12/01/ 2020.

[3] B. Peng, Y.-X. Liang, J. Yang, and K. So, "Image processing methods to elucidate spatial characteristics of retinal microglia after optic nerve transection," *Scientific Reports,* vol. 6, 2016, Art. no. 21816.

[4] J. Yang, "Pathological brain detection in MRI scanning via Hu moment invariants and machine learning," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 29, no. 2, pp. 299-312, 2017.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computer Science,* 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 770-778, 2016.

[7] P. Ekman and W. Friesen, "Facial Action Coding System (FACS): A technique for the measurement of facial action," *Rivista Di Psichiatria,* vol. 47, no. 2, pp. 126-38, 1978.

[8] H. Ali, M. Hariharan, S. Yaacob, A. H. J. J. o. M. I. Adom, and H. Informatics, "Facial Emotion Recognition Based on Higher-Order Spectra Using Support Vector Machines," 2015.

[9] S. Lu and F. Evans, "Haar Wavelet Transform Based Facial Emotion Recognition," in *2017 7th International Conference on Education, Management, Computer and Society (EMCS 2017)*, 2017.

[10] W. Yang, "Facial Emotion Recognition via Discrete Wavelet Transform , Principal Component Analysis, and Cat Swarm Optimization," *Lecture Notes in Computer Science,* vol. 10559, pp. 203-214, 2017.

[11] B. Li, R. Li, and D. Lima, *Facial Expression Recognition via ResNet-18*. Multimedia Technology and Enhanced Learning, Third EAI International Conference, ICMTEL 2021, Virtual Event, April 8–9, 2021, Proceedings, Part II, 2021.

[12] B. Li and D. Lima, "Facial expression recognition via ResNet-50," *International Journal of Cognitive Computing in Engineering*, pp. 57-64, 2021.

[13] S. Alnefaie, A. Cherif, and S. Alshehri, "A Distributed Fog-based Access Control Architecture for IoT," *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS,* vol. 15, no. 12, pp. 4545-4566, DEC 31 2021.

[14] A. A. Alaboudi, "Fifteen Deadly Cybersecurity Threats Aimed Covid-19," *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND NETWORK SECURITY,* vol. 21, no. 12, pp. 123-130, DEC 30 2021.

[15] H. M. Lu, "Facial Emotion Recognition Based on Biorthogonal Wavelet Entropy, Fuzzy Support Vector Machine, and Stratified Cross Validation," *IEEE Access,* vol. 4, pp. 8375-8385, 2016.

[16] H. Bazai, E. Kargar, and M. Mehrabi, "Using an encoder-decoder convolutional neural network to predict the solid holdup patterns in a pseudo-2d fluidized bed," *Chemical Engineering Science,* vol. 246:Article ID.116886, 2021.

[17] C.-F. Hsu, T.-W. Chien, and Y.-H. Yan, "An application for classifying perceptions on my health bank in Taiwan using convolutional neural networks and web-based computerized adaptive testing: A development and usability study," *Medicine,* vol. 100, no. 52:Article ID. e28457, 2021.

[18] N. H. T. Nguyen, S. Perry, D. Bone, H. T. Le, and T. T. Nguyen, "Two-stage convolutional neural network for road crack detection and segmentation," *Expert Systems with Applications,* vol. 186, 2021.

[19] X. Zhang, "DSSAE: Deep Stacked Sparse Autoencoder Analytical Model for COVID-19 Diagnosis by Fractional Fourier Entropy," *ACM Transactions on Management Information Systems,* vol. 13, no. 1, 2021, Art. no. 2.

[20] M. A. Khan, "Pseudo Zernike Moment and Deep Stacked Sparse Autoencoder for COVID-19 Diagnosis," *CMC-Computers, Materials & Continua,* vol. 69, no. 3, pp. 3145–3162, 2021.

[21] Z. Zhu, "PSCNN: PatchShuffle Convolutional Neural Network for COVID-19 Explainable Diagnosis," *Frontiers in Public Health,* vol. 9, 2021, Art. no. 768278.

[22] D. Kiruithiga, V. Manikandan, and Ieee, "Time Series Load Forecasting Using Pooling Based Multitasking Deep Neural Network," presented at the 2021 IEEE POWER & ENERGY SOCIETY INNOVATIVE SMART GRID TECHNOLOGIES CONFERENCE (ISGT), 2021.

[23] R. Ahmad, "Reviewing the relationship between machines and radiology: the application of artificial intelligence," *Acta Radiologica Open,* vol. 10, no. 2, Feb 2021, Art. no. 2058460121990296.

[24] J. Huang, L. Qu, R. Jia, and B. Zhao, "O2u-net: A simple noisy label detection approach for deep neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3326-3334.

[25] R. Blything, V. Biscione, Vankov, II, C. J. H. Ludwig, and J. S. Bowers, "The human visual system and CNNs can both support robust online translation tolerance following extreme displacements," *Journal of Vision,* vol. 21, no. 2, Feb 2021, Art. no. 9.

[26] C. Evans, C. Paz, and G. Mascialino, ""Infeliz " or "Triste ": A Paradigm for Mixed Methods Exploration of Outcome Measures Adaptation Across Language Variants," *Frontiers in Psychology,* vol. 12, Aug 2021, Art. no. 695893.

[27] M. Bader, L. J. Jobst, I. Zettler, B. E. Hilbig, and M. Moshagen, "Disentangling the Effects of Culture and Language on Measurement Noninvariance in Cross-Cultural Research: The Culture, Comprehension, and Translation Bias (CCT) Procedure," *Psychological Assessment,* vol. 33, no. 5, pp. 375-384, May 2021.

[28] K. Bhattacharjee, A. Tiwari, M. Pant, C. W. Ahn, and S. Oh, "Multiple Instance Learning with Differential Evolutionary Pooling," *Electronics,* vol. 10, no. 12, Jun 2021, Art. no.

1403.

[29] P. K. Nalajam and R. Varadarajan, "A Hybrid Deep Learning Model for Layer-Wise Melt Pool Temperature Forecasting in Wire-Arc Additive Manufacturing Process," *IEEE ACCESS,* vol. 9, pp. 100652-100664, 2021.

[30] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:*. 2013.

[31] X. Jiang, "Multiple Sclerosis Recognition by Biorthogonal Wavelet Features and Fitness-Scaled Adaptive Genetic Algorithm," (in English), *Frontiers in Neuroscience,* Original Research vol. 15, no. 1098, 2021-September-13 2021, Art. no. 737785.

[32] Z. Zhang and X. Zhang, "MIDCAN: A multiple input deep convolutional attention network for Covid-19 diagnosis based on chest CT and chest X-ray," *Pattern Recognition Letters,* vol. 150, pp. 8-16, 2021.

[33] K. Wu, "SOSPCNN: Structurally Optimized Stochastic Pooling Convolutional Neural Network for Tetralogy of Fallot Recognition," *Wireless Communications and Mobile Computing,* vol. 2021, p. 5792975, 2021/07/02 2021, Art. no. 5792975.

[34] E. Parcham, M. Ilbeygi, and M. Amini, "CBCapsNet: A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks," *Expert Systems with Applications,* vol. 185, p. 115649, 2021.

[35] C. Sitaula, T. B. Shahi, S. Aryal, and F. J. S. r. Marzbanrad, "Fusion of multi-scale bag of deep visual words features of chest X-ray images to detect COVID-19 infection," vol. 11, no. 1, pp. 1-12, 2021.

[36] L. Zwingmann, M. Zedler, S. Kurzner, P. Wahl, and J.-P. J. F. i. s. Goldmann, "How Fit Are Special Operations Police Officers? A Comparison With Elite Athletes From Olympic Disciplines," *Frontiers in sports active living,* vol. 3, 2021.

[37] Y. Fushimura *et al.*, "Orotic acid protects pancreatic 13 cell by p53 inactivation in diabetic mouse model," *Biochemical biophysical research communications,* vol. 585, pp. 191-195, 2021.

[38] E. KARADURMUŞ, G. Eda, N. TAŞKIN, and M. YÜCEER, "BROMATE REMOVAL PREDICTION IN DRINKING WATER BY USING THE LEAST SQUARES SUPPORT VECTOR MACHINE (LS-SVM)," *Sigma Journal of Engineering and Natural Sciences,* vol. 38, no. 4, pp. 2145-2153, 2020.

[39] V. Patel, "Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning," *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2620-2648, 2016.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448-456: PMLR.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*, 2016, pp. 630-645: Springer.

[42] X. Li, C. Tang, and J. Sun, "Facial emotion recognition via stationary wavelet entropy and Biogeography-based optimization," *ICST Transactions on education e-Learning,* vol. 6, no. 19, p. 165702, 2020.

[43] J. An, X. Li, Z. Zhang, W. Man, and G. Zhang, "Joint Trajectory Planning of Space Modular Reconfigurable Satellites Based on Kinematic Model," *International Journal of Aerospace Engineering,* vol. 2020, 2020.

[44] H. Delavari and S. Naderian, "Design and HIL implementation of a new robust fractional sliding mode control of microgrids," *IET Generation, Transmissionu&Distribution,* vol. 14, no. 26, pp. 6690-6702, 2021.

[45] S. Etedali, "Ranking of design scenarios of TMD for seismically excited structures using TOPSIS," *Frontiers of Structural and Civil Engineering,* vol. 14, no. 6, pp. 1372-1386, 2020.

[46] J. F. Yang and P. Sun, "Magnetic resonance brain classification by a novel binary particle swarm optimization with mutation and time-varying acceleration coefficients," (in English), *Biomedical Engineering-Biomedizinische Technik,* Article vol. 61, no. 4, pp. 431-441, Aug 2016.

[47] X.-X. Hou, "Alcoholism detection by medical robots based on Hu moment invariants and predator-prey adaptive-inertia chaotic particle swarm optimization," *Computers and Electrical Engineering,* vol. 63, pp. 126-138, 2017.

[48] U. Ghosh, P. Chatterjee, and S. Shetty, "Securing SDN-enabled smart power grids: SDN-enabled smart grid security," in *Research Anthology on Smart Grid and Microgrid Development*: IGI Global, 2022, pp. 1028-1046.

[49] A. S. Amiss *et al.*, "Modified horseshoe crab peptides target and kill bacteria inside host cells," *Cellular and Molecular Life Sciences,* vol. 79, no. 1, Jan 2022.

[50] S. Banerjee, M. A. Afzal, P. Chokshi, and A. S. Rathore, "Mechanistic modelling of Chinese hamster ovary cell clarification using acoustic wave separator," *CHEMICAL ENGINEERING SCIENCE,* vol. 246, DEC 31 2021.

[51] F. Bonchi, D. Garcia-Soriano, A. Miyauchi, and C. E. Tsourakakis, "Finding densest k-connected subgraphs," *DISCRETE APPLIED MATHEMATICS,* vol. 305, pp. 34-47, DEC 31 2021.