Facial Expression Recognition in Real Environment

Yiqun Zhang[1]

Research School of Computer Science,
ustralian National University
u7102332@anu.edu.au

Abstract. Facial expression recognition plays an important role in many fields, such as human-computer interaction, emotion monitoring, emotion detection and so on. But the position of human face in real environment is not fixed, so it is still difficult to recognize facial expression in real environment. To solve this problem, we propose a model, which can effectively solve the problem of face recognition in real environment. Our model uses the Viola-Jones[3] face detection method to locate the face position, and then uses ResNet18[1] to classify the facial expression. We evaluate our model in a real environment facial expression dataset *Static Facial Expressions in the Wild* (SFEW) [2], this data set contains 675 face images extracted from movies, with a total of 7 different expressions. Through experiments, the accuracy of our model in the valid-set is 40.48% and in the test-set is 38.09%, which shows that our model is effective.

Keywords: Neural Network · Expression recognition · Deep Learning · ResNet.

1 Introduction

Facial expression recognition based on photos is an active research field, because facial expression recognition plays an important role in many fields, such as human-computer interaction. In the laboratory or some controlled environment, face recognition has achieved good results. But it is still difficult to recognize facial expressions in the real environment, because of the conditions such as image resolution, face position, illumination conditions and so on are uncertain, which make it difficult to recognize facial expression in real environment.

The data set we used is SFEW[2], which includes seven kinds of facial expressions. These expressions are extracted from movies, with a total of 675. They are divided into seven categories: Angry, Surprise, Sad, Neutral, Disgust, Happy, Fear. Due to different movie scenes, they include different lighting, different ages and different resolutions. These complex expressions well restore the real world situation. *JAFFE database*[6], *Multi-PIE database*[4] and *MMI database*[5] perform well on lad controlled datasets, But it doesn't perform well on this data set.

In order to complete the task of facial expression recognition in real environment, we propose a model, which uses Viola-Jones[3] face detection method and ResNet18[1]. Viola-Jones[3] face detection method is responsible for finding faces in real environment photos, and ResNet18[1] is responsible for classification of facial expressions. In addition, we expand the data set (data augmentation) and divide it into training set, test set and valid set. We have carried out experiments on sfew data sets, the accuracy of our model in the valid-set is 56.37% and in the test-set is 40.48%, which is significantly higher than that of random classification 38.09%, The validity of our model is proved. After the experiment, we analyze the results of the model. Finally, we propose a model based on decision tree[11] to explain the neural network.

In the second part of this paper, we provide the specific method to realize this model. In the third part, we got the results through experiments analyzed it. Finally, we conclusion our work and discuss the future work.

2 Method

2.1 Data pre-process

We use a total of 675 images in SFEW dataset [2], but these data can not be used directly, we need to preprocess the data.

Generate Integer Label

In order to mark the neural network understand, we need to convert text label to integer label. The conversion rules are listed in (**Table.1.**).

Face detection

Text Label	Integer Label
Angry	0
Disgust	1
Fear	2
Happy	3
Neutral	4
Sad	5
Surprise	6

Table 1. Result from decision tree

Fig. 1. Some photos in the dataset

I selected some photos from the dataset (**Fig.1.**), and we can see that in these photos, the position of the face is not fixed, which causes great difficulties for facial expression recognition. So we need to detect the face, cut the irrelevant part from the photo, leaving only the face. In this way, the neural network will be easier to complete the task of classification. Here we use Viola-Jones[3] face detection and use OpenCV pretrained model to complete this task. We only leave one face in the image, because other images will make it difficult for neural network to select.

Data Division

We randomly divide the data set into training set (70%), validation set (20%) and test set (10%). In this way, we can use validation set to select an excellent Hyper-parameter and we can test its performance on the test set.

Data Augmentation

We can see from **Fig.2.** that there are very few pictures in this dataset, in order to train the neural network better, we need to do data augmentation. We use three data augmentation methods for each image, so we will get the original four times of the data set. Data augmentation is only implemented on the training set.

Image flipping We flipped the image horizontally.

Add noise For a 0 to 255 RGB image, we add a random noise to each pixel of each channel, which is 0 to 30.

Histogram equalization We use histogram equalization of OpenCV to process the image.

2.2 ResNet

In the part of facial expression classification, I use ResNet[1] as a classifier. We use ResNet18[1]. ResNet[1] solves the problem that the classification performance can not be increased or even the accuracy will be decreased with the increase of network layers.

Structure

The **Fig.3.** is our network structure.

First Input the face image detected in the previous step, the image format is RGB, the size is 224×224.

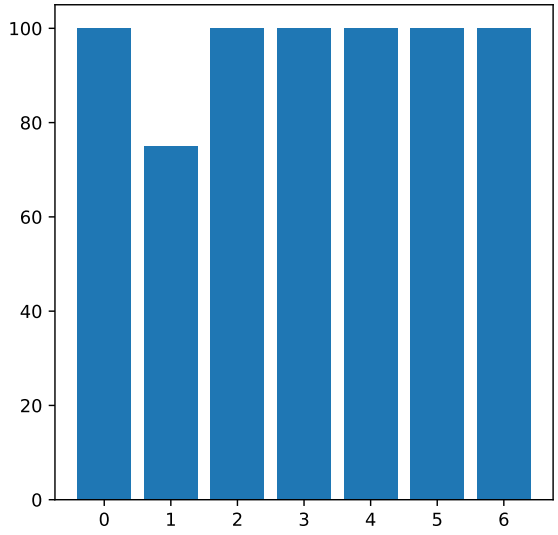

Fig. 2. The number of files in each folder in the dataset

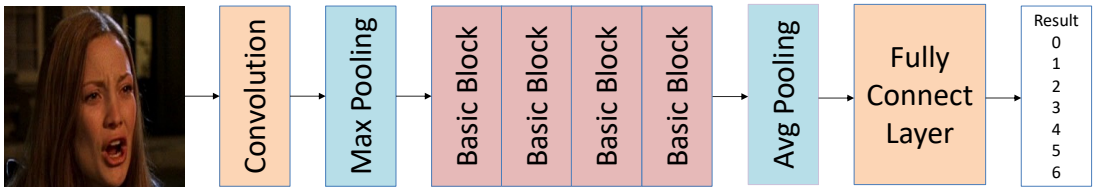

Fig. 3. Structure of ResNet18[1]

Second We convolute the image and get a tensor with the size of $64 * 112 * 112$.Convolution operation helps us to extract features from images, the formula of convolution operation is as follows:

$$C(x, y) = \sum_{k,l} F(k, l)G(x - k.y - l)$$

Third We put the results of the previous step to max pooling layer and we will get a tensor of size $64 \times 56 \times 56$. The pooling operation can compress features, improve the expressiveness of features, and reduce the dimension of features. The formula of max pooling is as follows:

$$S(x, y) = \max_{s,t}(F(s, t)G(x - s, y - t))$$

Forth We input the results of the previous layer into the 4 Basic Block one by one. Each Basic Block will output a tensor whose dimensions are $64 \times 56 \times 56$, $128 \times 28 \times 28$, $256 \times 14 \times 14$ and $512 \times 7 \times 7$. Finally, we get a tensor of size $512 \times 7 \times 7$. We can know from [1], basic block is the key to solve the problem of accuracy decline with the deepening of the network. Because it can compare the original input and the input after passing through the network, and only leave better results. So if the effect drops after passing through a layer of network, then this layer of network will not work It can make the number of layers after the optimal network do not work, then increasing the depth of the network will still make the network in the optimal state. The **Fig.4.** is the structure of basic block. The formula of ReLU is as follows:

$$F(x) = Max(0, x)$$

Sixth We put the results of the previous step to average pooling layer and we will get a vector of size 512. The formula of average pooling is as follows:

$$M(x, y) = \frac{1}{s + t} \sum_{s,t} (F(s, t)G(x - s, y - t))$$

Seventh Input the results of the previous layer into the full join layer, and then get a 7-category classification result.

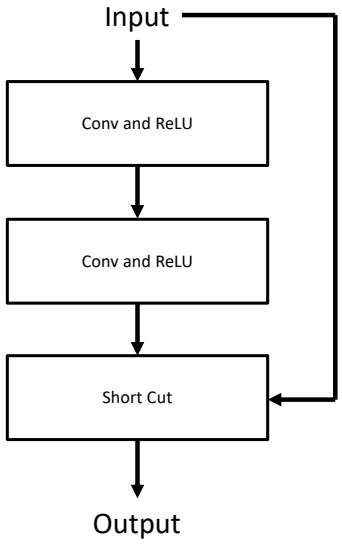

Fig. 4. Structure of Basic Block

Loss Function

The loss function is explained in [7]. Cross entropy describes the distance between two probability distributions. The smaller the distance is, the closer the two probabilities are. The larger the difference is. For two probability distributions P and Q, the cross entropy of P is expressed by Q:

$$H(p,q) = -\sum_{x} p(x) \log q(x)$$

Q is the predicted value and P is the expected value

Optimizer

We use Adam[8] as the optimizer, which is an extension of SGD and uses momentum and adaptive learning rate[9] to speed up training.

Hyper-parameters

learning rates After many experiments, I tested the accuracy of the test set and the verification set under different learning rates. Finally, I chose the learning rate as 0.001. In addition, I adjusted the learning rate to one tenth of the original in the 30th and 80th epochs.

batch size After many experiments, I tested the accuracy of the test set and the verification set under different learning rates. Finally, I chose the learning rate as 0.001.

epoch After my experiment, $epoch = 200$ is enough. Too low can't make neural network learn well, too high will waste time

2.3 Explanation Network

According to the method in [10], I will build an explanation mechanism to explain the neural network.

First We divide the image into nine regions, and then use the average pooling method to extract a feature value for each region, so we can transform an image into a 9-dimensional feature vector.

Second We process the output as binary data, and define the maximum True(ON) and the others as False(OFF). This result is consistent with the result of neural network classification.

Third According to the above processing, we get a 7-dimensional output data. For each dimension, we can find some important inputs. According to these important inputs, we can establish a satisfied rule set. I use the decision tree[11] to complete this task. We deal with each dimension separately. For each dimension, we build a decision tree[11]. For this decision tree[11], we can find the most important node (the meaning is that every decision is directed to more samples). We think that the column to make decisions on this leaf node is important inputs, and the rule to make decisions is satisfied rule set.

3 Result and Discussion

We trained on the SFEW dataset [2], and found the appropriate super parameters according to the verification set. Finally, we tested on the test set. Next, I will show, analyze, and discuss the test results and our model.

3.1 Predict Results

The accuracy of our model is 40.48% in valid set and 38.09% in test set. If we take random values, the accuracy is only 14.28%. So we can see that our model is effective. We show it in **Table.2.**.

	Our Model	Random
Valid	40.48%	14.28%
Test	38.09%	14.28

Table 2. Accuracy

In order to show the prediction results, we draw the confusion matrix ((**Fig.5.**)).

From the confusion matrix, neural network can basically complete the task of classification. But there are still some errors. For example, a large number of *Sad* are classified into *Anger* and *Surprise*. After my analysis of the data set, I think this is because the facial expressions in the data set are not very clear, I think this is because the expression in the movie is reflected by the video, and the picture loses the time information, so it will be difficult to distinguish the expression.

3.2 Explanation Results

We put the rules extracted from the decision tree[11] in **Table.3.**.

From the above results, we can explain the input and output of neural network. For example, 1st and part have a greater impact on *Sad* expression (result index = 5), and the feature of 1st is less than 160.5, which means that the possibility of sad expression is less. According to this method, we can make various explanations for this neural network of expression recognition, These explanations can help us better understand neural networks.

Result index	import input	Satisfied Rule Set	Number of samples	result
0	0, 3, 4, 5	5>20.5000 4>82.5000 0<105.5000 3>33.5000	146	OFF
1	4, 7	7<155.5000 4>51.5000	206	OFF
2	2, 6, 7	2<121.5000 6>7.5000 7>43.5000	59	OFF
3	4	4>101.5000	216	OFF
4	1, 2, 6, 7	7>28.0000 6<152.5000 1>76.0000 2<95.0000	13	OFF
5	8, 1	8<77.5000 8>6.0000 1<160.5000	115	OFF
6	0, 1, 3, 8	0>5.5000 3<156.5000 0<157.0000 1>43.5000 8>28.5000	40	OFF

Table 3. Result from decision tree

4 Conclusion and Future Work

4.1 Conclusion

We build a model based on resnet18[1] for facial expression recognition, and use Viola-Jones[3] face detection method and data augmentation. And tested on SFEW dataset [2], the accuracy of our model in the valid-set is 40.48% and in the test-set is 38.09%. The results show the effectiveness of the model.

4.2 Limitations and Future Work

There are too few images in SFEW dataset [2], so it is difficult to improve the accuracy of neural network. This model can be used in larger data sets in the future.

In this model, I use ResNet18[1], this is a shallow network in ResNet[1] family, we can use deeper network in the future.

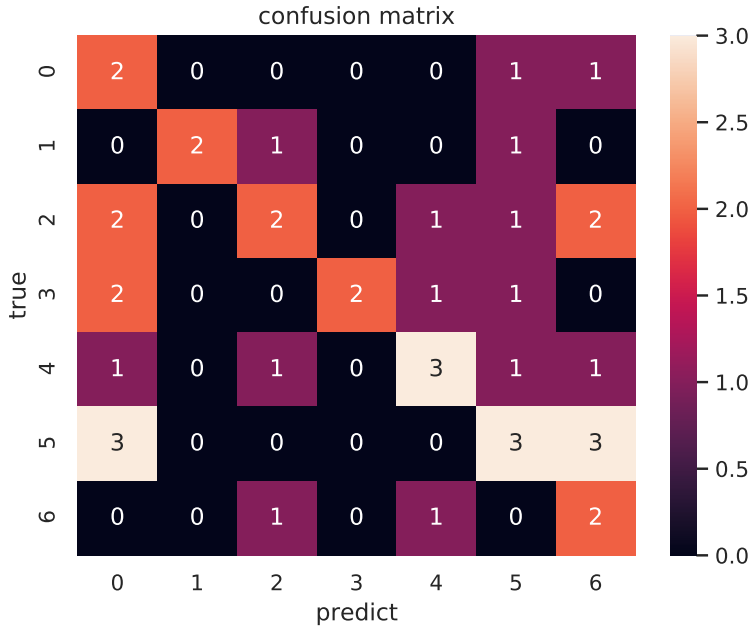

Fig. 5. Structure of Basic Block

Because of the limitation of hardware performance and time, I didn't use cross validation. In the future, we can use cross validation to choose better super parameters.

In the face detection part, I use the traditional machine learning method. In the future, we can use the method based on neural network, which may make the detection results better. At the same time, I give up the data that the number of faces is not 1. In the future, I can segment the image with multiple faces into multiple image.

References

1. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
2. Dhall, Abhinav, et al. "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark." 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, 2011.
3. Viola, Paul, and Michael J. Jones. "Robust real-time face detection." International journal of computer vision 57.2 (2004): 137-154.
4. R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. In Proceedings of the Eighth IEEE International Conference on Automatic Face and Gesture Recognition, FG'2008, pages 1–8, 2008.
5. M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Webbased database for facial expression analysis. In Proceedings of the IEEE International Conference on Multimedia and Expo, ICME'05, pages 317–321, 2005.
6. M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition and Workshops, FG'98, 1998. 1
7. Gu, Jiuxiang, et al. "Recent advances in convolutional neural networks." Pattern Recognition 77 (2018): 354-377.
8. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
9. Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv preprint arXiv:1609.04747 (2016).
10. Gedeon, T. D., and H. S. Turner. "Explaining student grades predicted by a neural network." Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan). Vol. 1. IEEE, 1993.
11. Safavian, S. Rasoul, and David Landgrebe. "A survey of decision tree classifier methodology." IEEE transactions on systems, man, and cybernetics 21.3 (1991): 660-674.