A Robust Strategy for Facial Expression Recognition

Kangdao Liu

Research School of Computer Science, Australian National University, Canberra Australia
u7094687@anu.edu.au

Abstract. Automatic facial expression recognition plays a great role in human computer interactions. Detecting human faces and recognizing their facial expressions in the real world are fairly challenging as there is a large variety of face images in terms of ages, races, looks, and the positions of the head, etc.. To accomplish the tasks, this paper proposes a robust deep learning strategy based on residual network for facial expression recognition which can cope with not only those image datasets taken in lab environment, but also in real life image datasets. This paper futher puts forward a mechanism to explain how our network makes decisions based on the extracted features of ResNet, which makes up for the deficiency that deep learning model is unable to explain the decisions made itself. The framework is evaluated by the semi-natural static facial expression dataset Static Facial Expressions in the Wild (SFEW) which includes about 700 images extracted from several movies. The results are that the test accuracy for the network is 54.90% and the accuracy of the explanation mechanism is 96.64%, indicating the excellent performance and great potential of this strategy.

Keywords: Facial Expression Recognition · Deep Learning · Residual Network · Explanation Mechanism.

1 Introduction

Detecting and recognizing facial expressions is of great interest and importance in the HCI research field. Despite the fact that facial expression recognition in laboratory controlled environment has been well solved [3–5], but it remains a challenging problem to do the work in realistic environments.

Hence, researchers in recent years paid more attentions on detecting and recognizing facial expressions in real life rather than the lab environment. Static Facial Expressions in the Wild (SFEW) dataset [1] was proposed then. Over 800 images extracted from 37 movies are included in the dataset. All the images in this dataset have corresponding expressions as Fig.1. From the paper[1], we know that many existing approaches can't work that well on this dataset.

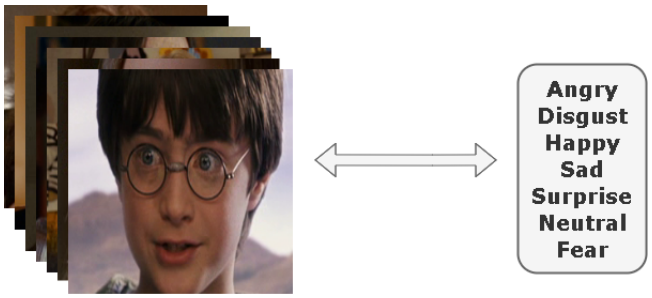

Fig. 1. Contents of SFEW

A more robust approach is needed to solve this problem. This paper proposes a framework which can recognize facial expression robustly and precisely based on ResNet[8] and build a explanation mechanism[2] on the fully-connected layer of our model which can interpret how the network makes decisions by the extracted features. In addition, in order to better solve the problem of 'faces in different images differ greatly', we use a face detection module based on dlib[9] to solve this problem smoothly. To solve the negative influence of the limited number of images in the dataset, we used the technique called data augmentation to increase the performance of the framework. We evaluate our model on dataset SFEW[1] and this framework shows competitive performance on the dataset.

This paper will be organized as followed. In next section, we will provide details of the methods used in the strategy. For section 3, we will analyze the results and performance of our framework and explain the practical significance of our explanation mechanism. Finally, we will discuss the limitations of our framework and talk about the future work.

2 Method

For this section, a face detection module base on dlib[9] will be firstly introduced, then the data augmentation strategy. The model based on ResNet-18[8] will be introduced with the method of building the explanation mechanism. Finally, the main methodology used to evaluate the model will be explained.

2.1 Face Detection Module

Dlib[9] is a modern C++ toolkit that contains machine learning algorithms and tools for creating complex software to solve real-world problems in C++. It is widely used in industrial and academic fields, including robotics, embedded devices, mobile phones. I download the module from [10] and use it to cut out the faces from all the images in SFEW[1]. However, some images in the dataset will lead our module to detect many unwanted expressions, as shown in Fig.2. The original label of this image is angry, so the old lady in the upper left corner with simile should not be detected. We will analyze the specific effects later.

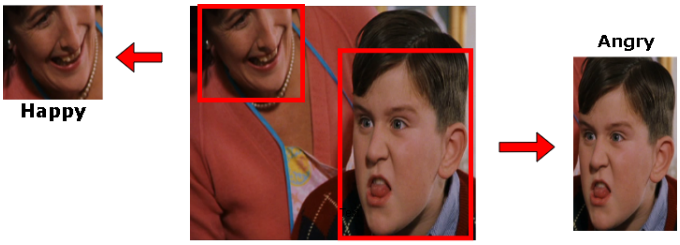

Fig. 2. Different expressions were detected in a single image

2.2 Data Augmentation Strategy

There are not enough images in our training set (about 700 image totally), which may lead to over-fitting of our model and fail to achieve the optimal effect. I used some basic operations from OpenCV library to process those images. Firstly, I flip all the images, and then both it and its flip version will be stretched or widened. One such image can generate at most five corresponding images. With this operation we can expand the training set by more than five times. When using these images for training, we will randomly crop a square image with a side length of 224 from it for training, which I think enhances the generalization ability of our model. An example of data augmentation is shown in Fig.3.

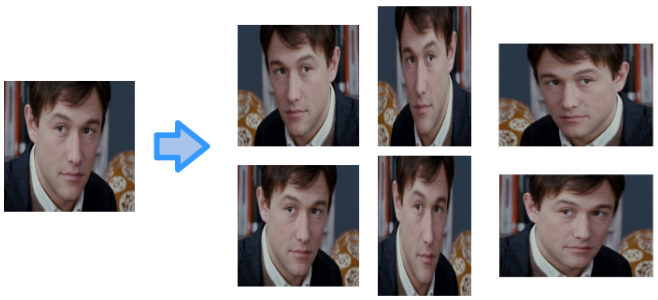

Fig. 3. Data augmentation

2.3 Structure of The Network

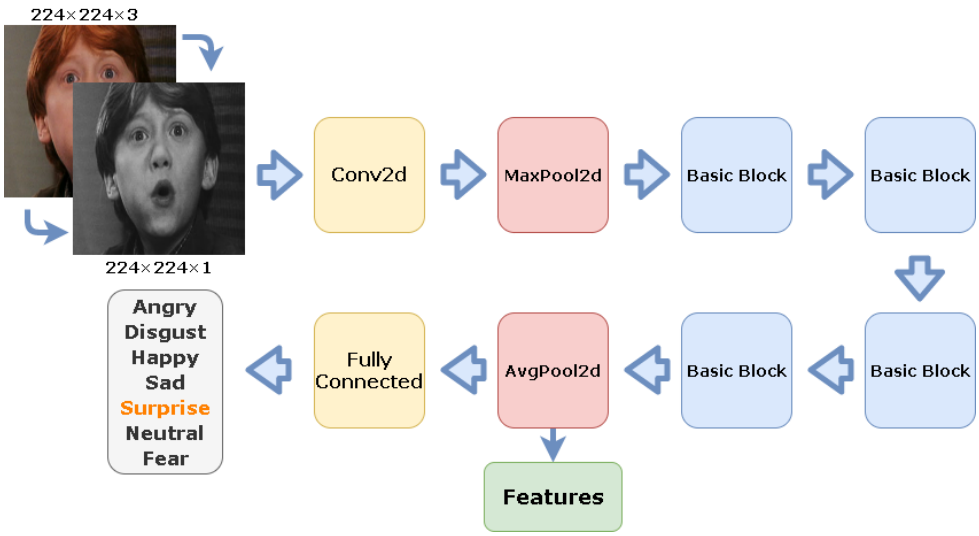

Fig. 4. Structure

We use ResNet-18[8] as the network in the strategy, stochastic gradient descent is used as the optimiser with the batch size equals to 256 to train the network which has a better performance than Adam[7] in this problem. After testing it many times, I found that there was little difference in the effect detected by color images and gray images. In order to improve the training speed and reduce the occupied gpu memory, we used gray images as the input instead. By doing this we can also reduce the effect of overall color style on the expression, since light images tend to be biased towards positive facial expressions and dark images towards negative ones.

I found that the performance of the model is getting better as the batch size increases, but the GPU memory does not support me to further expand the batch size and conduct further tests. I set learning rate to 0.05 and apply a 0.0005 weight decay rate and 0.9 momentum rate. Finally, the parameter size is 42.7 MB(44,776,569 bytes) totally.

Basic block above in Fig.4 is 'Resblock' which is a residual function f(x) is learned on the top and information is passed along the bottom unchanged. According to [10], I use four Resblocks totally here to build the network.

In order to save the features extracted by ResNet, I output the features to a CSV file before the fully connected layer for further testing. The output features of AveragePool layer is a 512 dimentions vector for each image which will be linearly mapped to 7 dimensions vector to determine which expression it is in the fully connected layer. We named the 512 features as feature 1, feature 2 to feature 512.

2.4 Explanation Mechanism

According to [2], the procedures of building the explanation mechanism is listed below.

1.Liken the input pattern to the characteristic input patterns, and present the most similar to the user.
2.In addition present inputs considered 'important' for the current network output, and their values in the characteristic pattern.
3.Produce a set of rules, and evaluate to confirm accuracy.
4.Give the network's next most likely output.

Due to the complexity of the network, we will not follow these steps from[2] directly. For this model, if we want to build an explanation mechanism, we need to figure out how the network makes its decisions, in other words, we need to extract some rules which can infer most of the behavior of the network. We here use a method based on decision tree to extract the rules from the network. The method is derived from [6]. By limiting the height of the decision tree, we can find the key features which influence mostly the decisions making of the network. We can see the figure below, these rules are generated by the decisions tree, patterns that satisfy these rules are more likely to be recognized as 'Angry', and the accuracy of this rule is 99.34% (99 percents of the patterns that satisfy these rules are labeled 'Angry'.). Hence, if we generate the rules for all the labels respectively by this method, we can get a basic understanding of how the network makes decisions.

Decision	**Rules**
Angry	**(Feature181 <= 1.024) & (Feature303 > 1.304) & (feature 429 > 0.838)**

2.5 Evaluation Strategy

We use confusion matrix as the tool to evaluate the proposed model. We mainly use 'recall' here as a yardstick to analyze the different performance of all labels. We calculated the proportion of correct and misclassified expressions under all given labels, so that we could analyze which kinds of expressions are easily to be misclassified by each other. Accuracy will be a comprehensive indicator to evaluate our model. In general, accuracy and recall are mainly used as two indicators to evaluate our model.

$$accuracy = \frac{TP + TN}{TP + FN + FP + FN}$$

$$recall = \frac{TP}{TP + FN}$$

TP, TN, FP, FN are true positive, true negative, false positive, and false negative classifications respectively. We mainly used accuracy and recall to evaluate the network, and used accuracy to evaluate the explanation mechanism.

3 Results and Discussion

After training and testing our model on SFEW[1]. The overall performance of the model is not bad, but different labeled pictures have different performance. In this section, we will analyze its performance and the significance and performance of the explanation mechanism.

3.1 Performance of The Network

We use the evaluate method mentioned in section 2.5 to evaluate the model, we can see the confusion matrix in Fig.3, it's easy to get recall for all labels in this confusion matrix. After calculating the confusion matrix, we let each element to divide the sum of each row, then we can get the recall for all the label on the diagonal of the matrix. According to the matrix, we can find the recall of 'Surprise' is high which is 100%. But the recall of 'Disgust' and 'Neutral' is fairly low (33% and 27% respectively).

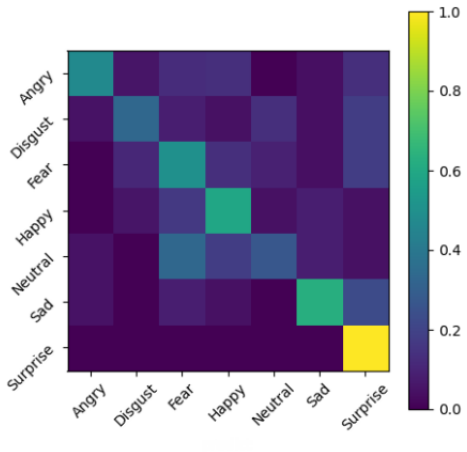

Fig. 5. Confusion matrix on test set

However, after browsing the misclassified images, I found there are serveral images are hard to be classified even by human beings. Like Fig.6 and Fig.7 below, the label of the left image is 'Disgust', and the label of the right image is 'Neutral', but our first impression of both of them was 'Sad'. Hence If we don't take front and back frames of this image in the film into consideration, sometimes it is difficult to recognize the expression by just one frame. The analysis of multiple front and back frames in the film will be discussed in the future work.

Fig. 6. 'Disgust' or 'Sad'

Fig. 7. 'Neutral' or 'Sad'

However, the overall accuracy is not bad which reached 54.90% on the test set. The model performs well on highly recognizable expressions such as 'Surprise' and 'Happy' and worse on lowly recognizable expressions like 'Digust' and 'Neutral'. If there are more images in the dataset, the model will have a broader and clearer understanding of lowly recognizable expressions, we think then the network will have a higher accuracy of recognizing them.

3.2 Performance of Explanation mechanism

I first divide the expressions into 'Angry' and 'non-Angry', and we use accuracy mentioned in section 2.5 to evaluate the mechanism. We can see the accuracy is 99.34%, which means our explanation mechanism makes the same decisions as the network on 99.34% of the data. After that, I divide the expressions into 'Disgust', 'non-Disgust' and repeat the same operations. The results are shown in the table below

Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
99.34%	100%	92.81%	95.42%	99.34%	96.73%	92.81%

Tab. 1. Accuracy of explanation mechanism

Statics in the table indicates that the explanation mechanism can explain the behavior of the network well. Then we can record some key decision conditions from the decision trees, and these conditions can be used as the rules to understand the network.

3.3 Extracted Rules

I put the extracted rules in this section, we can understand how the fully connected layer makes decisions by the rules below. These rules are all formed by key features in the decision trees. The left column is the decision and the right is the requirements to make such a decision.

Decision	Rules
Angry	**(Feature181 <= 1.024) & (Feature303 > 1.304) & (feature429 > 0.838)**
Disgust	**(Feature57<2.482) ||((Feature184 <= 0.486) & (feature 67<=0.908))**
Fear	**(Feature348 >= 0.326) || (Feature50 <= 1.177)**
Happy	**(Feature37 > 1.556) & ((Feature131 <= 0.189) || (feature298 <= 0.819))**
Neutral	**not((Feature32<=0.791) & (Feature458>0.678) & (feature 187<=1.325))**
Sad	**not ((Feature551 <= 1.196) & (Feature224>1.167))**
Surprise	**(Feature36 <= 0.414) & not (Feature271 <= 0.56)**

Note: not means NOT, & means AND and || means OR in the table above.

These rules can help us understand which features determine the final decision, and the rules can be a reference for follow-up researches, such as deleting nodes or exploring the extracted features of ResNet in depth. This explanation mechanism has important implications for understanding complex deep learning models like ResNet.

4 Limitation and Future Work

The first limitation is proposed in section 2.1, faces of some supporting characters or some side faces in the image may be regarded as faces by the detector. As the figure Fig.8, this head just works as background of the image, but it is recognized as a face. We can't tell if these misdetect images will have a negative impact on our network, but it's certainly a limitation in our face detection module which is based on dlib[9].

Fig. 8. An inaccurate detection

The second limitation is our model performs poorly on lowly recognizable expressions like 'Digust' and 'Neutral'. This limitation also leads to a overall low accuracy of our model.

We will analyze our future work on the basis of limitations. For the first limitation, I think a better face detector is in need to solve this problem, we can train a YOLO_v3[11] based face detector in the future. A precise non-maximum suppression was then used to pick the most accurate faces in some of the images. After all, face detection is the premise and basis of face recognition, we need to get it right before facial expression recognition.

Secondly, sometimes it's really hard to recognize a person's expression just from a image, especially for some lowly recognizable expressions like 'Digust' and 'Neutral'. In my opinion, if we want to robustly recognize expressions, it will be better to consider the image with its before and after situation together whether it's a reality photo or a screenshot of a movie. For example, the frames before and after this image in the movie can be used together to recognize the facial expression. Moreover, we can use a short video to do facial expression analysis instead. I believe this kind of framework will lead to a better performance than just use a single image.

5 Conclusion

In this paper, we proposed a strategy which can robustly recognize the facial expressions and validates our strategy on a semi-natural static facial expression dataset SFEW[1]. This strategy includes a face detection module based on dlib[9], a technique called data augmentation to enlarge the dataset, a network to recognize the facial expression based on Resnet-18[8] and a explanation mechanism which can explain how our network makes decisions based on the extracted features. We achieved that the test accuracy for the network is 54.90% and the accuracy of the explanation mechanism is 96.64%, which show the excellent performance and great potential of our framework.

References

1. Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011, November). Static facial expressions in tough conditions: Data, evaluation protocol and benchmark. In 1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV2011.
2. T. D. Gedeon and H. S. Turner, "Explaining student grades predicted by a neural network," Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), Nagoya, Japan, 1993, pp. 609-612 vol.1, doi: 10.1109/IJCNN.1993.713989.
3. Caifeng Shan, Shaogang Gong, McOwan, P.W.: Robust facial expression recognition using local binary patterns. In: IEEE International Conference on Image Processing 2005. vol. 2, pp. II–370 (2005)
4. Kotsia, I., Pitas, I.: Facial expression recognition in image sequences using geometric deformation features and support vector machines. IEEE Transactions on Image Processing 16(1), 172–187 (Jan 2007).
5. Moore, S., Bowden, R.: Local binary patterns for multi-view facial expression recognition. Computer vision and image understanding 115(4), 541–558 (2011)
6. Tameru Hailesilassie, R.: Rule Extraction Algorithm for Deep Neural Networks: A Review. (IJCSIS) International Journal of Computer Science and Information Security,Vol. 14, No. 7, July 2016011
7. Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009
10. http://dlib.net/files/
11. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.