Face Emotion Recognition: CNN with Low Computational Expense, and Threshold and Voting System Optimization

Zijun Wang[1]

Research School of Computer Science, Australian National University, Australia
u7100365@anu.edu.au

Abstract. This paper is a report of a new research on face emotion recognition. Compared with many previous studies, which mainly concentrate on datasets collected under lab environment, this research deals with data in a more realistic situation. Data augmentation is applied to make the dataset more sufficient and robust. Due to the research condition limitation, large network is not achievable because of the computational expense. Instead, a GPU-free CNN is constructed, with threshold and voting system being applied to improve the accuracy. As a result, the overall accuracy has reached 46.35%, which is at the same level of the result from SVM in Abhinav's paper [1]. Meanwhile, this research has the advantages of the high efficiency of the single network version, the easiness for tuning the model, and the GPU-free research requirement.

Keywords: Face emotion recognition · CNN · Computer Vision.

1 Introduction

Face emotion recognition is an important subtopic under face recognition. It can be used for camera such as those on iphone which has been able to recognise smile, and it can also strongly support psychological studies, since AI often has different angle of view compared with human when grabbing features and summarize laws.

However, solving this problem is quite complex and challenging, because the surroundings can affect the appearance of face emotions to a large degree, and the same emotion performed by different people can vary much.

In this research, the problem under solving is classifying 7 emotions from the inputs which are captured in the real-world. A dataset [1] from Abhinav is used, because the data in it are collected from movies, which is largely similar to the realistic situations. Faces are directly detected using a ready-made face detector [2], and CNN is used for classification. A series of optimization methods are applied. Data augmentation is used because the number of data is quite small. Inspired from Milne's paper [3], threshold is used. Although it is a technique used for binary classification in that paper, while seems not to fit our problem, this research has successfully mixed this technique in with voting system, resulting in an improvement of accuracy and stability of the model.

2 Method

2.1 Data Pre-processing

According to Abhinav's paper [1], compared with other datasets such as JAFFE [4], which collects all data from lab, this dataset use screenshots from movies so that it contains a variety of backgrounds, light conditions, face positions and face angles.

However it also has its own shortages, as shown in Figure 1, screenshot from movies will make most part of the image be background. Therefore, face detection is needed in order to cut the area belongs to face, so that neural network will not be confused by the messy backgrounds and extract features that is not related to the faces. Considering that face detection can become another independent research topic, this research used a ready-made face detector [2], faces in each image will be detected and recorded by 4 pair of points, which form a rectangle, and then the image will be cut by this rectangle and save the prat in it as a new image. Face will be resized to 48*48 to fit the input layer of the network, and such a scale will both ensure efficiency in training and preserve most important information in the faces. After turning the new images into grayscale, histogram equalization is also applied here as a common normalization method for images, and because there are a number of faces that are in extremely dark environment.

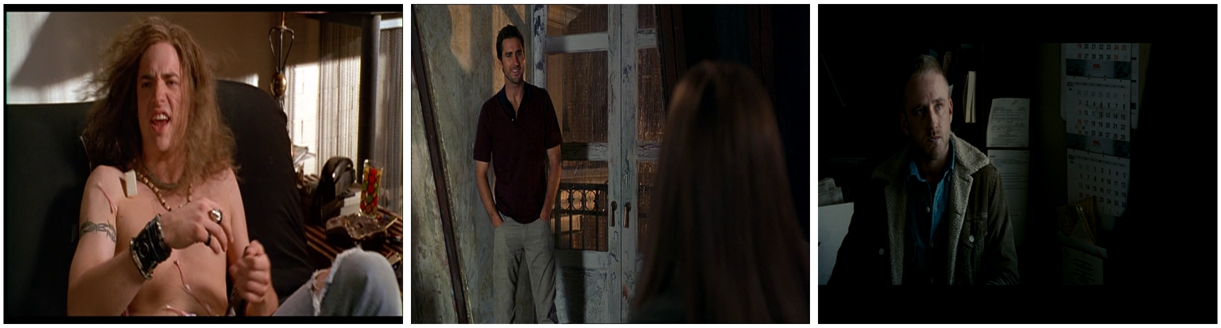

Fig. 1. Some pictures in the original dataset.

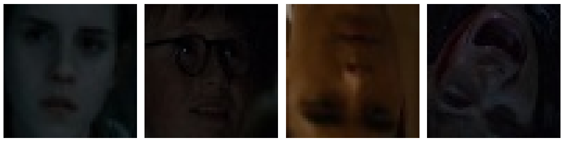

Fig. 2. Some faces that are in extremely dark environment.

This dataset records 7 kinds of emotions, which are labelled by integers 0 to 6 according to the folder the images are in. There are 675 images in total, with the second emotion missing 25 images and each of the others have a hundred, respectively.

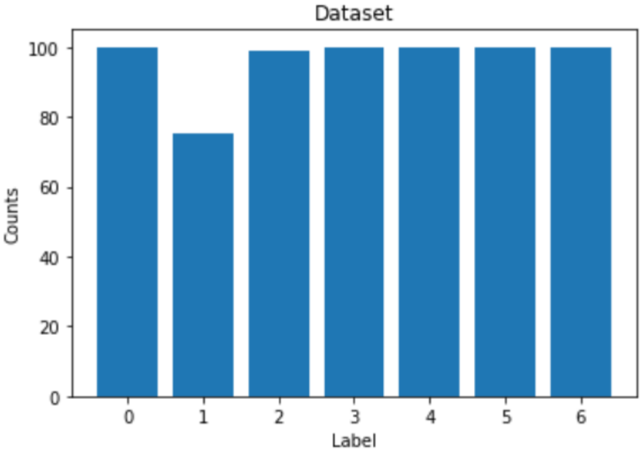

Fig. 3. The figure shows the distribution of original data.

After applying face detector and cut the areas that are recognized as faces, the number of images become further small, since some faces are not successfully detected because of lack of brightness or the large degree of side face. Obviously, such number of faces are not enough. Therefore, data augmentation is used to improve the amount and diversity of data. Each detected face is flipped horizontally, vertically, and horizontally and vertically. Therefore, the dataset becomes four times as large as before. Considering that faces are collected from movies, which means the faces that are bottom up are rare, such data augmentation strategy can also make this kind of face more commonly seen by the network, improving generalization and robustness. Since normalization has been done after face detection, which means the result images have been normalized, histogram equalization is not necessary here.

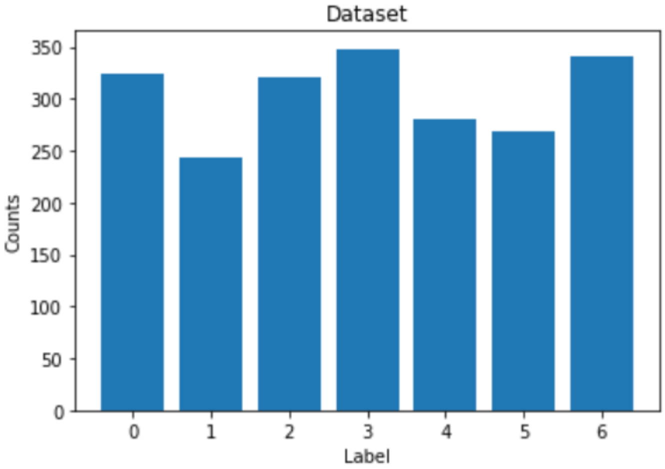

Fig. 4. The figure shows the distribution of data after face detection and data augmentation.

To read the data, the relative path of each image in the pre-processed dataset is saved into a .txt file, which will be stored into a data-frame so that it can be easily split into training and testing set by a proportion of 8:2. Later, each image will be read in via the path.

2.2 CNN

Since we want to use pure images as input, CNN is a good choice regarding its wide applications and good performance on image recognition [6]. However, there is no GPU on hand when this research is carrying out. Instead of trying to find one, an idea occurred me. Why not try to construct a light and small convolutional neural network and see how far it can go? Obviously, when the network is small, it also needs short time for training and predicting, which means it will be easier to tune and it can be applied to applications that need the network to react in an extremely short period of time, such as real-time face emotion recognition.

The structure and setting of the network are shown below:

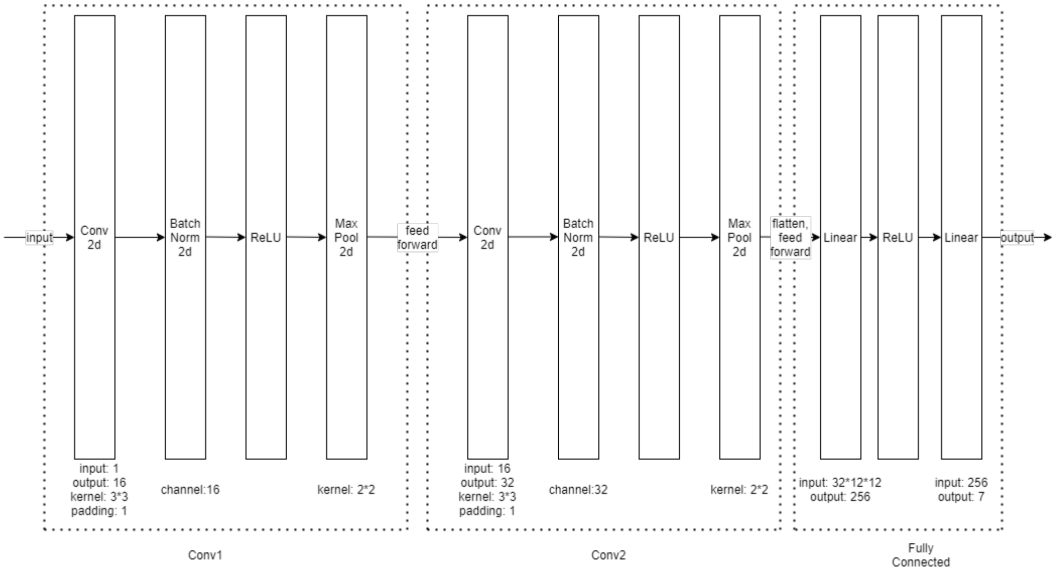

Fig. 5. The figure shows the structure of the convolutional neural network.

Improving efficiency while preserving performance, such a network is considered as a top-level model. The convolution kernel is 3, considering that the image is not big either. One round of padding is added so that the pixels on the edge can be the same as the others. Batch normalization layers are applied as a common method for accelerating training and improve generalization. To avoid gradient explosion, ReLU is used as activation function. Max pooling is another way for improving generalization while keep the main features pure and sharp. Two fully connected layers are placed at the end of the network, with the same activation function as previous layers. Except the last layer which is used to predict the final class, the other layer is used for improving the nonlinear expression ability of the model. With the same total number of neurons, networks that are relatively deeper usually have better performance [7]. Two layers are appropriate here, since one layer only will lead to a 10% drop of testing accuracy, while add more layers will not lead to any imporvement but make the computational expense increase. Hidden neurons are also shrinked as small as possible, and 256 hidden neurons is the floor that can keep the testing accuracy.

Training settings are as follow:

- Optimizer: SGD
- Loss function: Cross Entropy
- Learning rate: 0.05
- Training epochs: 30
- Batch size: 64

In every epoch, training is performed batch by batch, with forward feeding first, calculating loss and then performing back-propagation to update parameters.

Some methods are not applied in this network. As it is a small network, which can hardly go overfitting, dropout layers do not appear in the model. In fact, if the first two fully connected layers are each followed by a dropout layer, this network will easily become unable to make any valid predictions. Initialization of the network is also not treated specifically, as batch normalization can largely reduce the dependence of model on parameter initialization.

In terms of evaluation, the overall accuracy is calculated so that the general performance can be seen directly, followed by class-wise recall, precision and specification to identify details and give supports for class-wise discussions.

2.3 Threshold and Voting System

In Milne's paper [3], authors used a threshold for the binary classification problem they faced, which inspired me whether this method can be applied to a multi-class classification problem. In this research, the final prediction is made by the 7 output neurons in the output layer of the network, the one with highest value will be regard as the class of prediction. To apply threshold onto this task, softmax is used for the original output of the network, so that they will be transferred to probabilities distributed within [0, 1], and the threshold will also be a number in this interval. For each prediction, if the output value after softmax is larger than the threshold, it is assigned to 1, or it will be assigned to 0.

However, this may lead to multiple 1 for the prediction for one image. Another method, voting system, occurred me, which is inspired by its applications on some deep learning researches [5]. To achieve this, 50 networks described in section 2.2 are trained using the same strategy as above. Then threshold method described in the last paragraph is used to them when testing. The predictions of each network are regarded as votes, and all these networks will make predictions on the same testing set. For each image in the testing set, votes from these networks will be gathered together, and the class received largest number of votes will be regarded as the final prediction. If it ends with a draw, obviously most networks in this system believe these classes are all possible emotions that the face in the image has, so we have to admit this case is still indistinguishable for these networks, but we have to select one of the classes that have the highest number of votes. In this research, the class with smallest number of label (from 0 to 6 as described in data pre-processing section) will be chosen, as python inbuilt function does. After testing, the best value for threshold is 0.5.

Table 1. Performance of voting system with different thresholds.

threshold	testing accuracy
0.3	44.71%
0.4	45.88%
0.5	46.35%
0.6	44.94%
0.7	43.76%

3 Result and Discussion

3.1 CNN

The Table 2 below shows the performance of the network constructed in section 2.2. Notice that this is the average performance, and the overall testing accuracy can fluctuate from 38% to 42%. Using the timer library in python, we can also see that training such a network on a training set with 1699 images only need 274 seconds, while it only spends 6.6 seconds to make predictions on testing set, which includes 425 images. Notice that the whole process is done on a business-oriented laptop with no assistance from GPU.

Table 2. Performance of the single CNN network.

Emotion	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Precision	0.42	0.37	0.58	0.45	0.30	0.30	0.42
Recall	0.46	0.14	0.33	0.55	0.27	0.48	0.52
Specificity	0.88	0.97	0.96	0.88	0.91	0.87	0.84
Overall Testing Accuracy	40.24%						

Large differences among class-wise recall are detected. To explain this, we need to look into how the networks are trained. Initially, the network will predict all inputs to one class (call it default class), resulting in high probabilities on this class. From training data, the network learns features of each class by moving the weights from those which lead to all-one-class predictions, resulting in decline of probabilities on the default class while improving the probabilities on the true class when predicting an input. During this process, features of some emotions are more easily to be extracted while the others are more difficult. Therefore, the final extracted features are not evenly distributed among all classes, resulting in the unbalanced class-wise recall.

However, there are also a puzzling problem that worth being mentioned. At first, the performance after training is: the network can reach about 99% training accuracy while it only have around 40% testing accuracy. Intuitively, this will be considered as overfitting. However, things like dropout or early stopping do not work, and there has been batch normalization and maxpool that can reduce overfitting. To find out the reason, the training and testing accuracy in each epoch is calculated and recorded, by switching the network between training mode first and then turning it to evaluation mode within each iteration. When overfitting happens, the testing accuracy should increase first, then starts decreasing while training accuracy keeps climbing. However, the Figure 6 shows that testing accuracy is increasing first, then starts flowing, while obvious and continuous decrease is not detected. Extending the max epochs will only have similar result. Therefore, we have to attribute this situation to the variety of dataset. You will find some faces with similar expressions are labelled under different emotions (Figure 7), while under each emotion class there are also many kinds of expressions (Figure 8). But the high accuracy on training set is a positive signal which tells us that the current network has been able to extract most features from the images it has seen, which means the complexity of network is sufficient.

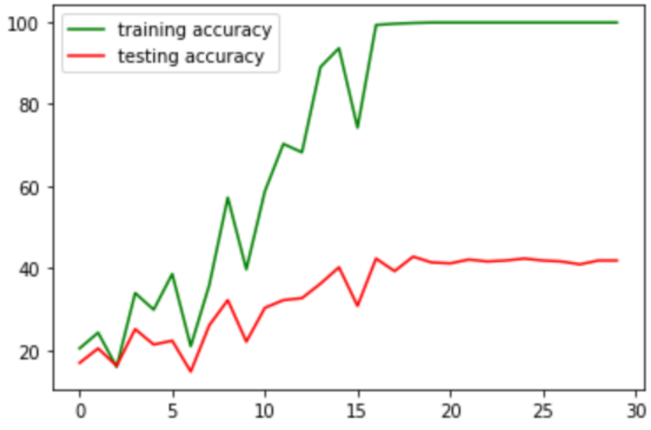

Fig. 6. The figure shows the changes of training and testing accuracy during training.

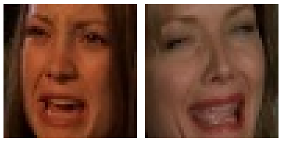

Fig. 7. The left one is labelled as 'Angry' while the right one is labelled as 'Sad'.

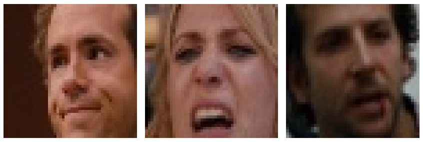

Fig. 8. Although these faces have largely different expressions, they are all labelled as 'Disgust'.

3.2 CNN with Threshold and Voting System

The Table 3 shows the results of the performance after introducing threshold and voting system in. The overall testing accuracy has a 6% increase, which leads to the general improvement of all class-wise recall, precision and specificity. In addition, the performance is also much more stable than the single network version, where the overall testing accuracy fluctuation is within 1%. In terms of efficiency, preparing the 50 networks is quite time consuming, but because they only need to be done once, after the debugging for single network is finished, such an increase of time consumption is still acceptable. When testing, the network can make predictions on the testing set, which includes 425 images within 24.5 seconds.

Table 3. Performance of the networks optimized by threshold and voting system.

Emotion	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Precision	0.48	0.43	0.59	0.54	0.34	0.33	0.49
Recall	0.49	0.3	0.53	0.56	0.32	0.41	0.55
Specificity	0.90	0.95	0.93	0.91	0.91	0.90	0.87
Overall Testing Accuracy	46.35%						

If we regard the single network model as a voting system with only one network in it, it is 6% lower on accuracy while it needs shorter time to make a prediction. Therefore, it is a trade-off between efficiency and accuracy, and the number of networks in the voting system can be adjusted regarding the demands. However, voting system is also not omnipotent, when the number of networks involved increasing, it will finally reach the upper limit on accuracy. Since it will be too computational expensive, the system with massive number of networks involved is not tested.

The classes that have low recalls have larger improvement here, so it seems that the this optimization method can give more help on predicting the emotions that are difficult to extract features and generalize. A probable explanation is that if a network is unsure what the face should belong to, its prediction will be low probability for many classes, while the correct predictions are gathered together, under the correct class, which means wrong votes are distributed among different classes while correct votes are not. Therefore, even if the number of wrong outputs is larger than the number of correct outputs, the final prediction on this condition will still be right.

However, if most networks recognize the input image as another emotion from the true emotion, i.e. this image is easy to be mistaken, the final prediction will still be wrong.

3.3 Comparison with Baseline

In Abhinav's paper [1], a detailed SPI baseline is introduced, but it is the result using features extracted by LPQ and PHOG instead of the images themselves. In this paper, we can find another seven expression class classification result, which is recognized by a SVM model, and it is 43.71% for LQ and 46.28% for PHOG.

Comparing the results shown in 3.1, you can see that the network in this research is not as good as the baseline, since this network is small and there is no GPU getting involved during training. However, this highly efficient network is also easy to tune the parameters, as the results can be seen in a short period of time. In addition, it can also be used in some situations that need the network to react quick but tolerate errors, such as real-time face emotion recognition, especially for the embedded development, where large running memory and GPU are difficult to get access to.

When the optimization method is applied, the model reaches 46.35% accuracy, which is almost the same as the baseline. Although it is not so efficient as the single network version, the advantage of GPU-free requirement is preserved, and it is relatively small compared with some networks that have over a hundred layers. Therefore, it can still be used on embedded development.

4 Conclusion and Future Work

To sum up, this research used a small and simple CNN as basic network structure, and reached 40.24% testing accuracy on SFEW dataset with such low computational expense that GPU is not required. Using threshold and voting system, the testing accuracy become 46.35%, while preserving the GPU-free and low running memory requirement. This model has the advantage of easy to debug and test and low hardware requirements, leading to a promising future on embedded development. By adjusting the number of networks involved in the model, the trade-off between accuracy and efficiency can also be balanced.

Future work will mostly focus on the following aspects. Firstly, the large gap between training and testing accuracy needs to be explained and if possible, solved. It is not overfitting considering the testing accuracy changing curve, and some counter-overfitting methods can make the performance sharply drops. Secondly, when doing face detection in data pre-processing, there are a few faces not detected by the detector. This may lead to some bias to the model, especially when these faces have some similar features that make the detector failed to detect them. Therefore, finding a way to analyse whether there is bias introduced or not is also necessary. Thirdly, a discussion of how to deal with the confusing faces such as those shown in Figure 7 and Figure 8 is necessary. Since the data is collected from movies, where is from a more realistic situation compared with the data collected from lab. Obviously, sometimes the emotion can be compound, such as the role is in a mood that is both angry and sad, and it is vividly expressed by the actor. Therefore, how should this kind of emotion as well as other ambiguous emotions being labelled should be an important topic that may even challenge the root of face emotion recognition.

References

1. Abhinav, D., Roland, G., Simon, L., Tom, G.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). pp. 2106–2112. (2011). https://doi.org/10.1109/ICCVW.2011.6130508
2. Library from third party. Download link: http://dlib.net/files/
3. Milne, L.K., Gedeon, T.D., Skidmore, A.K.: Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood. In: Proceedings 6th Australian Conference on Neural Networks. pp. 160–163, Sydney, 1995.
4. Lyons, M., Akamatsu, S., Kamachi M., Gyoba, J.,: Coding facial expressions with Gabor wavelets. In: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition. pp. 200–205. (1998). https://doi.org/10.1109/AFGR.1998.670949
5. Lei, C., Robert, Q., Haichun, L., Zenan, L., Tianhong, Z., Jijun, W.: Individual Recognition in Schizophrenia using Deep Learning Methods with Random Forest and Voting Classifiers: Insights from Resting State EEG Streams. https://arxiv.org/abs/1707.03467. Last accessed 17 Jan 2018
6. Heliang, Z., Jianlong, F., Tao, M., Jiebo, L.: Learning Multi-Attention Convolutional Neural Network for Fine-Grained Image Recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5209–5217, Venice, 2017.
7. Matus, T.: benefits of depth in neural networks. In: 29th Annual Conference on Learning Theory. pp. 1517–1539, New York, 2016.