Audience Reaction Recognition

Shih-Ting Huang
Computer Science Department
Rochester Institute of Technology, NY
Email: sh3964@rit.edu

***Abstract* – This project proposed a model to understand audience's reaction by facial expression. The predictor is trained by convolutional neural network. The accuracy of the result reaches 69%.**

1. Problem Definition

The audience reaction is always an important factor to evaluate the performer/ performance. The project proposed a model to analyze audience reaction base on facial expression. The outcome of each input face consists three types: happy, sad, and neutral. This model can be applied to variety of fields. For entertainment business, the film company can use these data to understand which part of the script makes audience happy or sad. For education field, this model can provide information for teachers to understand how the students feel about lectures.

2. Data Collection

The training data and testing data are from Kaggle, the Challenges in Representation Learning: Facial Expression Recognition Challenge[1]. Each face image of this dataset is 48*48 pixels with grayscale. There are 5,091 happy face images, 4,830 sad face images, and 4, 964 neutral face images. 70% of the data (10,419 images) was used for training, other 30% of the data (4,466 images) was divided for valid data. For testing dataset (2,155 images), there are 895 happy face images, 653 sad face images, and 607 neutral face images.

For the audience test data, I used MATLAB for data collection. The preprocessing was to extract face by using face detecting tool, resize into 48*48 pixels, and transform images from RGB to grayscale.

3. Methodology

The current project consist two main parts: prediction model, and face detection. The flowchart of the project is shown in figure 1.

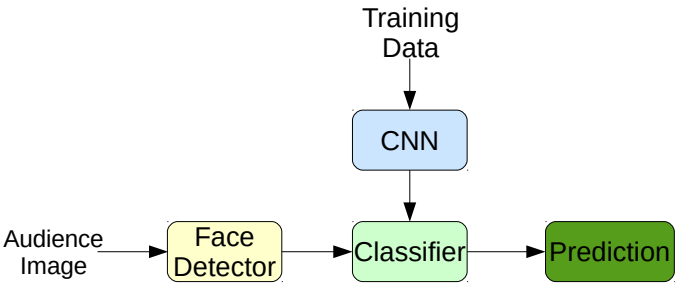

Figure 1. Project flowchart

In this project, the face detector was performed using Viola-Jones algorithm, build-in function in MATLAB.

In terms of the prediction model, since the training data was included with labels, a supervised learning was used to build up the prediction model. In this project, I used convolutional neural network (CNN) to train the classifier, which is commonly used in image predictor.

There are 5 common layer types in CNN: CONV layer, RELU layer, POOL layer, and Fully-Connected (FC) layer. In CONV layer, it computes the output of neurons that are connected to local regions in the input, each computes a dot product between their weights and a small region they are connected to in the

input volume. The depth depends on the filter number. RELU layer applies an activation function, such as the max(0,x) thresholding at zero. POOL layer performs a downsampling operation along the spatial dimensions (width, height). FC layer computes the class scores, resulting in volume of size correspond to a class score. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.[2]

The architecture of CNN, which was built for this project is shown in figure2.

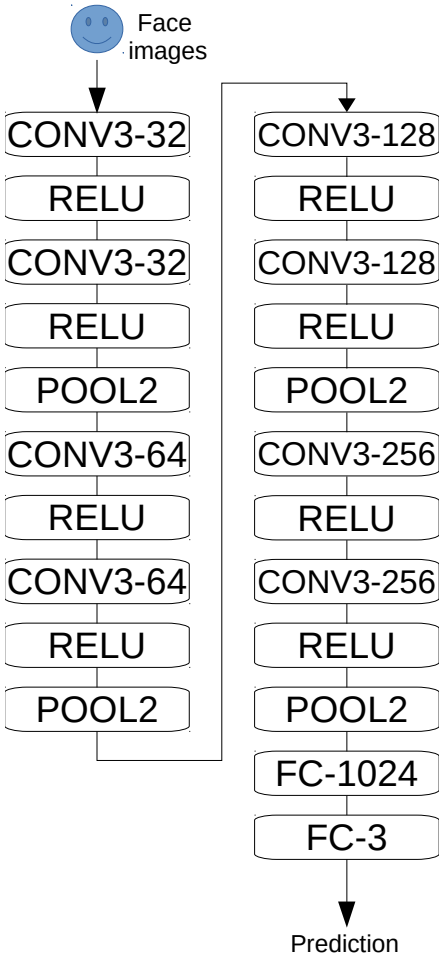

Figure 2: CNN architecture

4. Results and Discussion

The training curve is shown in Figure 3, and training score is shown in Table 1 and Figure 4.

Table 1. Scores

Train Score	Valid Score	Test Score
0.8764	0.6854	0.6909

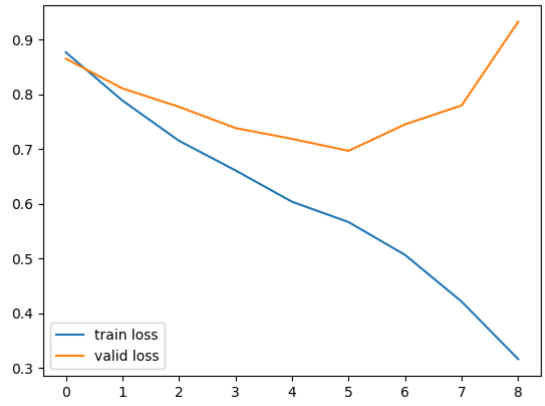

Figure 3: Training Curve

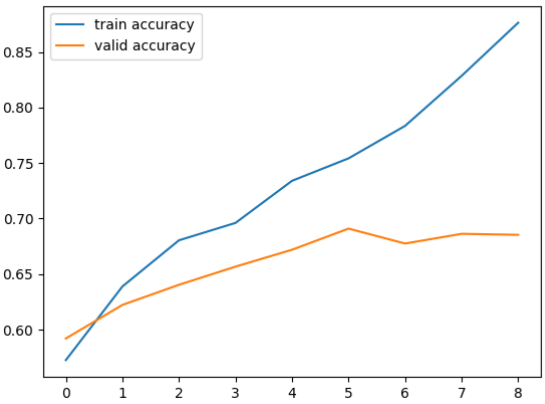

Figure 4: Accuracy curve

Overall, the accuracy was between 0.65 and 0.70, which was double accuracy than random guess (0.33). However, it was never greater than the 0.7 bar. I had tried adding more layers and CONV filter size and tuning the learning rate and batch size. None of above approaches improved the results to significantly reach a higher accuracy rate. Moreover, this program was performed on my laptop, which has

limited CPU and RAM. The study result may be able to reach higher prediction accuracy if a higher performance computer or resource is available to train the prediction model.

The following Figures 5-16 are results for audiences images prediction. The number of each type of face expression is shown in Table 2.

Table 2:

	Prediction			Mode
	Happy	Sad	Neural	
Img1	0	6	5	Sad
Img2	10	3	1	Happy
Img3	0	1	5	Neural
Img4	2	17	10	Sad
Img5	41	12	3	Happy
Img6	2	27	26	Sad

Figure 5: Image1 detected face

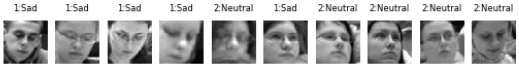

Figure 6: Image1 predicted result

Figure 7: Image2 detected face

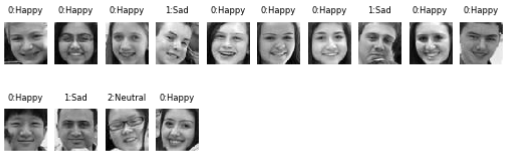

Figure 8: Image2 predicted result

Figure 9: Image3 detected face

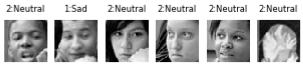

Figure 10: Image3 predicted result

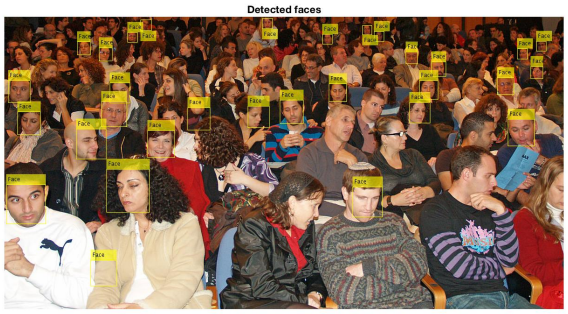

Figure 11: Image4 detected face

Figure 12: Image4 predicted result

Figure 13: Image5 detected face

Figure 14: Image6 predicted result

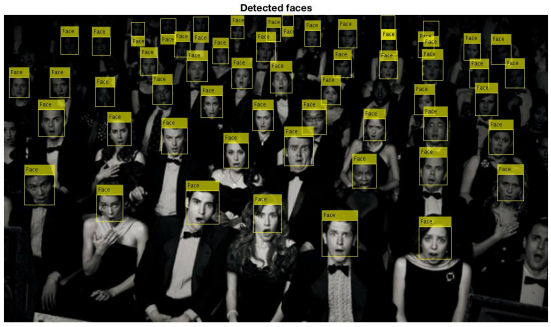

Figure 15: Image6 detected face

Figure 16: Image6 predicted result

The face detector form MATLAB has its limitation. First, not every face can be detected. Second, not every detected object was a face.

Furthermore, if the input face image was not front view, the angle of the face might affect the result of prediction. Adding more different-angle face images into training dataset could be one of the solution to improve the result of the prediction model.

5. Conclusion and Future Work

In conclusion, this project proposes an architecture to understand the audience's reaction base on facial expression. The prediction model can reach accuracy between 0.65 and 0.70.

For the future work, the program can add the a function to extract image from video for the preprocessing. Then, the prediction model can be applied to time series analysis. The face detector and the training data should be improved.

References

[1] Challenges in Representation Learning: Facial Expression Recognition Challenge. https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge

[2] CS231n: Convolutional Neural Networks for Visual Recognition. http://cs231n.github.io/