Facial Expression Recognition with Deep Convolutional Neural Networks

Gary (Yizhang) Lin (linyizh@berkeley.edu)
Xiaoxuan Shi (xiaoxuan_shi@berkeley.edu)

Abstract

In this project, we use deep convolutional neural networks to recognize seven types of human facial expressions. Inspired by VGG19, plus batch normalization and dropout layer, our architecture obtained a test accuracy of 58.6%, close to 65.5% of human readers in recognizing the expressions, far above the baseline of 14.3% of random guess and 35% of the commonest classifiers LDA, QDA, and logistic regression.

1. Introduction

Facial expression recognition (FER) is widely applicable in many ways. From the simple automatic tagging on social media where a mood icon is generated for photos uploaded by users, to audience interest and excitement capture for the purpose of event feedback evaluation, and to mood surveillance in security matters, facial expression recognition is playing an increasingly important role. In this project, our focus is to identify the expressions in images using deep convolutional neural networks (CNN).

As the project is a multi-class classification problem, several common classification models can serve as candidates. These include linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and logistic regression with soft-max function. In this report, we will first show the performance of these three common classifiers on our dataset and point out their invalidity or inadequacy in the specific context of classifying expressions based on image pixels. We then propose the use of CNN for CNN enables the learning of sophisticated features through deep neural networks in a reasonable computational time. We will demonstrate the best architecture we found in terms of testing accuracy, discuss the techniques we applied in improving the prediction performance of CNN, and analyze the strengths and weakness of our model. In the end, we conclude that deep CNN + batch normalization + dropout is the most effective model among all our attempts in classifying facial expressions and our model performs considerably well to be comparable to human in recognizing expressions in this dataset.

2. Related Work

FER has been an active research area since more than a decade ago. Various methods have been used to extract the appearance features of images, including Gabor, LBP, LGBP, HOG, and SIFT. However, these are only effective in specific small datasets, which poses challenges for FER with images in uncontrolled environments.

Under this setting, CNN were introduced in recent years to extract facial features. Multiple layers in CNN can extract higher and multi-level features of the entire face or local area, and have good classification performance of facial expression image features. Experience has shown that CNN is superior to other types of neural networks in image recognition.

For the FER2013 dataset we used in the project, various fine-tuning methods, such as grouping and fixing certain layers, different layers using different data sets, were applied to it to train a model to be used in prediction on EmotiW, another image dataset. Among all trials, a two-stage tuning strategy exhibited the best performance with validation accuracy of 48.5% and test accuracy of 55.6% (Ng et al., 2015).

3. Dataset

The data we use is the FER2013 dataset from "Facial Expression Recognition Challenge" on Kaggle (Goodfellow et al., 2013). The dataset consists of 48 by 48 pixel grayscale images of faces. There are two columns. The first one records the pixel values as strings, which can be converted into numeric matrices with preprocessing. The second is a label column indicating the expression present in the images (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). Our task is to classify the expression in each of the images, based on the pixel values, into one of the seven categories.

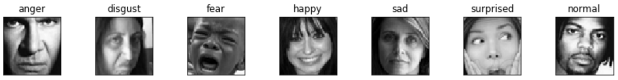


Figure 1. Example Images of Different Expressions.

We follow the training-validation-test split provided by the

Table 1. Classification accuracy for random guess, LDA, QDA, and logistic regression.

Classifier	Validation Accuracy	Test Accuracy
LDA	34.5%	35.1%
QDA	32.7%	32.9%
Logistic Regression w/ Soft-max	35.2%	35.1%

data source. The training set contains 28,709 samples. We use the public testing data during the Kaggle challenge as our validation set, and the private testing data released after the competition as test set for final evaluation of models. The validation and test set is of size 3,589 each. The number of most types of expressions in the training set is around 4,000 except Disgust. (Angry: 3995, Disgust: 436, Fear: 4097, Happy: 7215, Sad: 4830, Surprise: 3171, Neutral: 4965). Overall, the data is balanced.

4. Model and Design

4.1. LDA, QDA, and Logistic Regression

Before building a CNN, we first tried lighter models such as LDA, QDA, and logistic regression. The performance of these models serves as a comparison to that of CNN introduced later in this report. While the test accuracy of the models, around 35% (*Table 1*), is better than a random guess classifier, which has an expected accuracy of 1/7 (14.3%), it certainly does not meet our need. The low accuracy may result from the invalidity of the normality assumption in LDA and QDA and the insufficient non-linearity of the three models to capture complicated relationship between pixel values and the facial expression. This emphasizes the necessity of a CNN.

4.2. Deep CNN

Our final model is illustrated by *Figure 2*. The model was initially based on a simplified VGG architecture (Simonyan & Zisserman, 2014). VGG was attractive since it utilized increasing numbers of filters to extract different levels of information. However, VGG also required massive amounts of memory to train a single forward pass, which we could not accomplish given our limited compute resources. As a result, we implemented a ”light” design with simplified conv2D layers (32-32-32-64-64-64-128-128-128) filters and 2D max pooling layers between each block of conv2D layers.

We also experimented with batch normalization and dropout layer. We found that adding an dropout (rate = 0.5) layer resulted in the best outcomes, which will be discussed in detail in the next section. All of our models were trained using an Adam optimizer with randomized learning rate, beta values, and other hyperparameters. We found that, rather than comprehensively sweeping through all possible combinations, a randomized approach would yield a greater range of potential accuracies.

Type	**Filters**	**Kernel**	**Activation**
Input	-	-	-
Block #1			
Conv2D	32	(3*3)	Relu
Conv2D	32	(3*3)	Relu
Conv2D	32	(3*3)	Relu
BatchNormal	-	-	-
MaxPool2D	-	(2*2)	-
Block #2			
Conv2D	64	(3*3)	Relu
Conv2D	64	(3*3)	Relu
Conv2D	64	(3*3)	Relu
BatchNormal	-	-	-
MaxPool2D	-	(2*2)	-
Block #3			
Conv2D	128	(3*3)	Relu
Conv2D	128	(3*3)	Relu
Conv2D	128	(3*3)	Relu
BatchNormal	-	-	-
MaxPool2D	-	(2*2)	-
Top			
Dense	64	(3*3)	Relu
Dropout	32	(3*3)	-
Output	7	(3*3)	Softmax

Figure 2. Architecture of the final deep CNN model. The **Type** column specifies each layer while other columns indicate the setting of that layer. Input enters the first row and goes downwards through all the layers to reach output.

5. Results

The final validation accuracy of our CNN model was 57.6%, and the test accuracy was 58.6%. This is 3% higher than model without batch normalization, and 8% higher than model without dropout layer. This accuracy is 2% higher than the best performance of fine-tuning models mentioned in section 2, and approximately four times that of a pure random guess baseline (14.3%).

While not as accurate as the most cited research paper on this dataset, our simplified VGG model requires a greatly shorter time of training. For example, our model required around 200 seconds per epoch, while VGG19 of Keras applications (Pramerdorfer & Kampel, 2016), the model with highest accuracy of this Kaggle competition, required 1200 seconds per epoch.

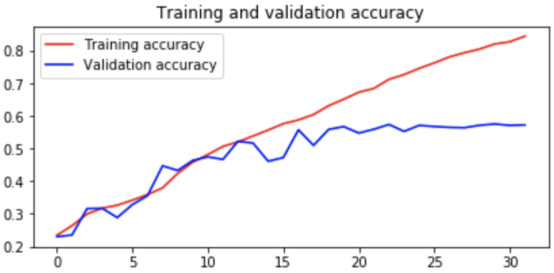


Figure 3. Train and Validation Accuracy vs. Epoch.

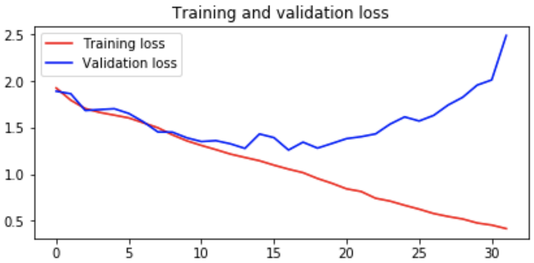


Figure 4. Train and Validation Loss vs. Epoch.

Given this improvement, we were able to tune our hyperparameters much more effectively and efficiently, as well as iterating over more epochs in a shorter period of time. Another benefit of our model is its low memory requirement, requiring less trainable parameters for all convolutional layers. Our model balances well among time, memory and accuracy and is using less than 1% of the previously mentioned best model's convolutional parameters to hit 70+% of its accuracy.

Comparing the accuracy curve in *Figure 3* to previous models we tried without batch normalization or dropout layer, there are some difference: (1) This validation accuracy curve shows small shocks around an average accuracy, while it's more stable in previous model. (2) This train accuracy curve is more close to linear line, which means it hasn't reach maximum test accuracy within 32 epochs. But for previous models, train accuracy goes to a platform within 32 epochs (even 98% without dropout). We can draw the conclusion that batch normalization layer and dropout layer have magical impact in reducing overfitting. (3) At the beginning, the curve for train and validation dataset almost eclipse, which tells us the primary model has similar effect in learning the two dataset. However, in previous models, the validation accuracy is always higher than train accuracy until it reaches the platform. Those differences are also shown in loss within different epochs. Even though the test accuracy for our final model is increased to 58.6% from 51% without dropout, the cross entropy loss doesn't change too much, with values around 2.0.

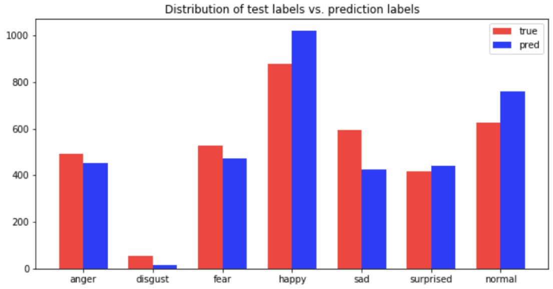
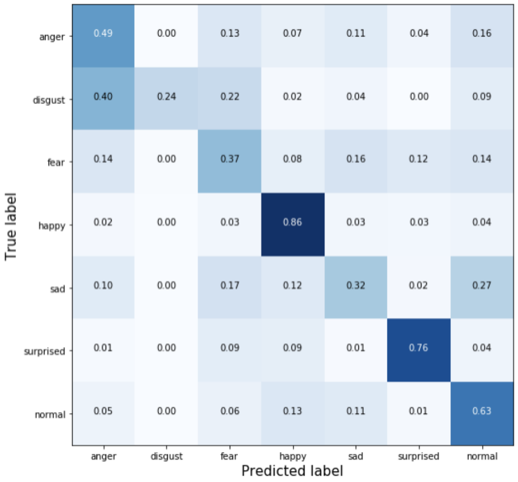


Figure 5. The distribution of true and predicted labels.

From *Figure 5*, we can intuitively have a guess the model would perform badly in predicting disgust in view of its scarcity in the training data (only 436 Disgust out of 28709 training examples). And the situation in the test data is almost the same as train data. Considering the seven classes, only happy is positive emotion, while the others are either negative or neutral. No wonder it has a dominant number of samples. Also in our prediction, most pictures are classified into happy and neutral emotions. So if we classify human emotions by attitude like "positive", "negative", "neutral", we expect our model to give a greater performance.



	anger	disgust	fear	happy	sad	surprised	normal
anger	0.49	0.00	0.13	0.07	0.11	0.04	0.16
disgust	0.40	0.24	0.22	0.02	0.04	0.00	0.09
fear	0.14	0.00	0.37	0.08	0.16	0.12	0.14
happy	0.02	0.00	0.03	0.86	0.03	0.03	0.04
sad	0.10	0.00	0.17	0.12	0.32	0.02	0.27
surprised	0.01	0.00	0.09	0.09	0.01	0.76	0.04
normal	0.05	0.00	0.06	0.13	0.11	0.01	0.63

Figure 6. Normalized Confusion Matrix.

Moreover, the confusion matrix (*Figure 6*) shows conclusion corresponding to the analysis above. The recall for happy, surprised and normal are 0.86, 0.76, 0.63, which are considerably satisfying results since human sensitivity to facial expression is only around 65.5% (Goodfellow et al., 2013). In other words, this model beats average human level of predicting happy, surprised and normal faces. However, the recall for other classes are less than 0.5. More obviously, more disgust faces are classified into angry than the correct label. Besides the small number of samples in train data, we can imagine human may have the same muscle movement (i.e. frowning) when feel disgusted or angry. So it's a reasonable suggestion to combine these two classes for future FER methods.

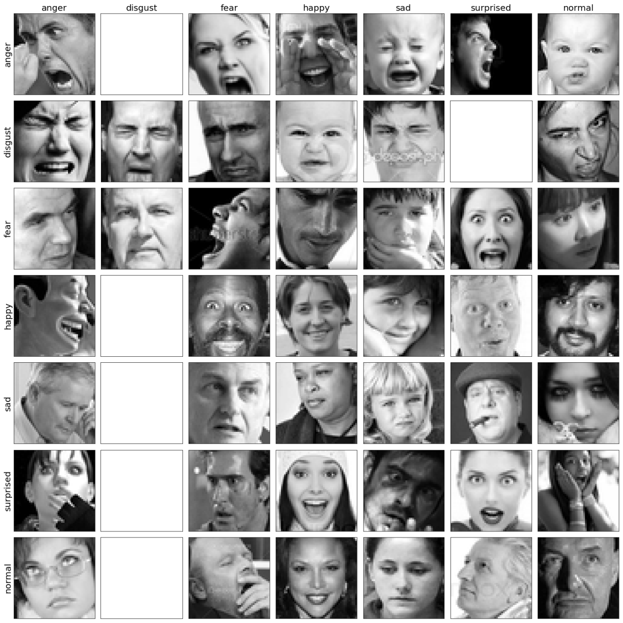


Figure 7. Image examples corresponding to confusion matrix.

To find more about our misclassification, we plot some image examples (*Figure 7*) corresponding to confusion matrix. The diagonal faces are correctly classified ones while the others are not. The x-axis denotes predicted label and y-axis true label. The blank block means there are no samples from y are misclassified into x, as where the values are zero in confusion matrix. The blank blocks show the fact that it's difficult to predict disgust, or to say it's difficult for CNN to capture the characteristics of a disgusted face. Probably that's because of lack of information, with such a small number of samples. No classes except fear are misclassfied into disgust. And no disgust faces are misclassified into surprised. In a broad view of FER, the class ”disgust” can be combined into others since unbalanced data is highly likely to have low recall.

Another problem we can see from *Figure 7.* is ambiguous boundary between true labels. For example , the baby face in first row, fifth column has true label as ”angry”. And the prediction is sad. As a human, we intuitively know the baby is crying sadly, so it seems our prediction even ”corrects” the originally ”wrong” true labels. Our model works good on predicting happy, surprised and normal, which are not negative attitudes. In *Figure 7.* there are some similarities within the prediction of them. In happy prediction columns, people are mostly facing directly to the camera with a smile. In surprised prediction columns, people are mostly opening their mouth with round eyes.

6. Conclusion

In this project, we designed a deep CNN model to obtain a high test accuracy on FER2013 within short training time. This validates reliability of deep CNN model on the expression classification problem. Although facial expressions are difficult to classify due to complicated muscle movements, our model can be compared to human eye with regard to average prediction accuracy (58.6% vs. 65.5%), and we still have considerable potential to improve our current model in the future.

References

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Grozea, C., Bergstra, J., Xie, J., Romaszko, L., Xu, B., Chuang, Z., and Bengio, Y. Challenges in representation learning: A report on three machine learning contests, 2013.

Ng, H.-W., Nguyen, V. D., Vonikakis, V., and Winkler, S. Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pp. 443–449, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450339124. doi: 10.1145/2818346.2830593. URL `https://doi.org/10.1145/2818346.2830593`.

Pramerdorfer, C. and Kampel, M. Facial expression recognition using convolutional neural networks: State of the art, 2016.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2014.