Smile Detection

CS9840 Course Project Report

Renfeng Liu

January 27, 2015

1 Introduction

1.1 The Problem

Facial expression recognition is a very important area of research, which has lots of practical values for real world applications. Smile is one of of the most import facial expression and many studies have been focusing on simile detection. In this project, I aimed to do smile detection using machine learning methods. The goal of this project is to obtain a classifier which is capable to detect whether a person in an image is smiling or not. The input of the classifier is an image which contains a face in it and the output of the classifier is a boolean value—the people in the image is smiling (1) or not smiling (0).

1.2 The Methodology

In this project, two different machine learning approaches are used to learn the classifier, the first approach is to employ the traditional machine learning models like SVM and adaBoost using a variety of features extracted from the image as input. The accuracy of this approach highly depends on the salient of the extracted features, and the corresponding parameters of the learning model. Several feature extraction methods have been used in this project as described in detail in section 3.1.

The second approach is using the state-of-art Convolutional Neural Network(CNN) to train the classifier. In the past few years, CNN had won lots of machine learning competitions such as ILSVRC2012, ILSVRC2013, ILSVRC2014, etc,. The accuracy of this approach depends on the parameters which described the network structure and the size of the training dataset. It is not explicit to explain why specifical neural network works well while others dont Another drawback is that this approach is prone to overfit the training set which leads to bad generalization. The detailed network structure used in this project and my experiment results are described in section 4.

2 The Dataset

The dataset used in this project is from the Facial Expression Recognition 2013 (FER-2013) contest, created by Pierre Luc Carrier and Aaron Courville [1]. There are 35887 images in this dataset and each

[1] http://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data

image is labeled with one of seven types of facial expressions. The distribution of the dataset can be found in Table 1.

Expression	Number of images	Label
Anger	4953	0
Disgust	547	1
Fear	5121	2
Happiness	8989	3
Sadness	6077	4
Surprise	4002	5
Neutral	6198	6

Table 1: FER-2013 Dataset Distribution

2.1 Preprocessing

2.1.1 Filtering

In this project, the images from the category of Happiness is used as positive samples (smiling), and all other images are used as negative samples(non-smiling). Face detection using Haar Cascades is used to detect the face region in the images. In my experiments, only images on which faces can be successfully detected are included. As there are much more negative samples (non-smiling) in this training dataset, I removed some of them(several negative categories in original dataset) to make the dataset balanced. As a result, only 5330 images are retained in the dataset. The distribution of the retained dataset can be found in Table 2.

Expression	Number of images	Label
Smile	2611	1
Non-smile	2719	0

Table 2: Smile Detection Dataset Distribution

2.1.2 Face Normalization

We cropped the face area in the image using Haar Cascades facial detector and scaled the frontal face region to the size of 48*48 pixels, since we only need to focus on the frontal face region for smile detection.

2.1.3 Shuffling and Splitting

After filtering and normalizing the image dataset, I shuffled the dataset several times so that the samples distributed as randomly as possible. After that the dataset is splitted into two parts : 80% of the images are used as training dataset and the rest 20% are used as the testing dataset.

3 Smile Detection in Traditional Machine Learning

3.1 Feature extraction

For all classical learning method, the most important step is to select salient features for specific learning model. In the project, I applied the following feature extraction methods to obtain the feature vectors for smile detection.

3.1.1 Gabor Energy Filter

Gabor Energy Filter (GEF) is modeled from humans visual cortex. It is a linear filter used for edge detection and is very popular for facial expression detection. In J Whitehill et al. [9], there are five spatial frequencies(wavelengths of 1.17, 1.65, 2.33, 3.30, and 4.67 iris widths) and 8 orientations(22.5 degree interval) are proved to be the efficient way to extract features in the face expression recognition process.

In this project, I used the 16 orientations(11.25 degree interval) which could give the best result other than 8 in J Whitehill et al. [9]. Thus, there are a total of 5*16 =80 Gabor energy filters applied on each image. (The size of the filters are the same of the input images, so each filter would be applied only once in the image). I applied each Gabor energy filter to each pixel on the image and I concatenate all the resulting features of each voxels into a vector to form a feature vector of this image, which would be of the size of 80(filters) * 48*48 = 184,320.

3.1.2 Pixels Intensity Difference

C Shan et, al. [7] used the intensity difference of any two pixels on the image as the feature for smile detection. All permutations of the pixels are calculated and for a single grayscale image of size 24*24 pixels, there would be (24*24)*(24*24-1) = 331,220 combinations. The image has to be scale to the size of 24*24 pixels because for a 48*48 image, there would be (48*48)*(48*48-1) = 5,306,112 features for a single image, which would cost at least that many bytes of memory to store the feature vector for a single image, and its also very computationally expensive to compute that feature vector.

In this project, the training set images are scaled to size of 24*24 and for each image, the feature vector is of the size of (24*24)*(24*24-1) = 331,220 in my experiment.

3.1.3 Area of Interest

N Kumar et, al. 2008[4] broke down the facial image into several parts like mouth, chin, eyes, etc,.and they found that the mouth area plays the most important role in classifying whether a person is smiling or not in an image. However, they didnt describe in detail how to locate each region of the face in the image.

In my experiment, a naive method is used to locate the region of the mouth. As I have already cropped the frontal face region, each image contains only the detected faces area from the face detector and the mouth region is almostly fixed in the images. The center region of the lower half of the image is

assumed to contain the mouth area, which is decided by these two points (32, 12) to (46, 36).hat is, 14*24 = 336 pixels are remained for further process.

3.1.4 Histogram of Oriented Gradients(HOG)

YH Huang et al. 2009[2], used optical flow of the images as features to train the classifier to detect whether a people in an image is smiling or not. Inspired by their method, HOGs are computed as the features to train the classifier.

In this project, Sobel derivatives of each cell in both X and Y directions are calculated first, and then the direction of gradient at each pixel is computed and quantized to 16 integers. The image is divided into 9 sub-areas with size of 16*16 pixels, and the histogram of direction are calculated in each sub-area. After that, the result of each sub-area is concatenated together to form the feature vector. As a result there would be 16*9 = 144 features for each vector.

3.1.5 SIFT and SURF

SIFT is a famous feature extraction method in computer vision. In this project, SIFT features are also used in one of the experiments. Both SIFT and SURF algorithms will detect key points in the images, but for each image the algorithms will detect different number of key points. For some image, the algorithms even cannot detect any features.

In this project, 15 most important key points are selected to form the feature vector. If less than 15 key points are detected, zeros will be filled into the vector. So there would be 15*128 = 1920 features in the vector. SURF features are processed the same way as SIFT features.

3.1.6 Principal Component Analysis(PCA)

PCA is widely used for face recognition, so in this project PCA is also used. In the experiment, 20 most import eigenvectors are used as the feature vector. So there are 20*48 = 960 features in the feature vector.

3.1.7 Bag of Words

In the experiment, the mouth area, which picked out according to section 3.1.3, is first clustered into 100 categories, then the centroids of each categories are calculated. After that, the L2 distance to each centroids is calculated and the result is treated as the input feature vectors. Thus There would be 100 features in the feature vectors.

3.2 The Learning Algorithms

3.2.1 Support Vector Machine(SVM)

SVM is proven to be one of the most accurate supervised learning models. In this project, most of the features extracted by above described methods are used for training the SVM models. Different kernels

and C will be used to do the model selection.

3.2.2 Adaptive boosting(adaBoost)

AdaBoost gains its power by combining lots of weak classifier with learned weights into a strong one. It is often referred as the best out-of-the-box classifier. Pixels intensity differences are very weak classifiers for detecting whether a face is smiling or not. So in the second experiment, this model is used to train the classifier with the features of pixels intensity differences.

3.3 The experiments

In this project, the experiments mainly follow the work that has been done by other researchers. The first experiment is trying to repeat the work did by J Whitehill et, al.[9], and the second experiment follows the method provided by C Shan et, al. in 2012[7]. I have also tried some other methods to train the classifier which are described in following experiments.

3.3.1 Experiment 1

In the paper Developing a practical smile detector, J Whitehill et, al[9]. described a practical method to for smile detection, the method is described in Figure 1.

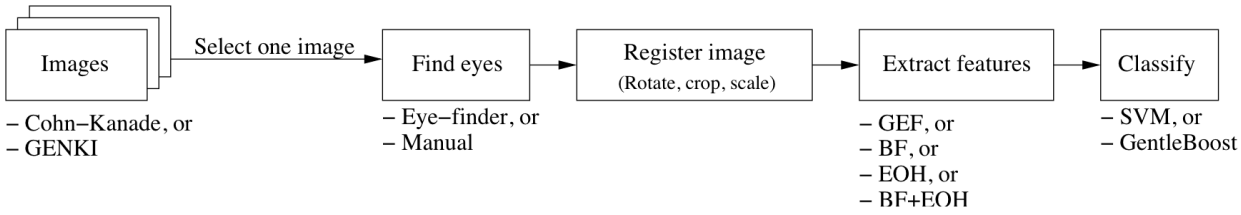

Figure 1: Developing a practical smile detector

There are several key steps in this method:

1. Find the locations of both eyes using their eye detector[1] or manually labeling the location of the eyes.
2. Use the locations of both eyes to do image registration by rotating, cropping, and scaling the face about the eyes to reach a canonical face.
3. Apply 40 GEFs as described in section 3.1.1 to get the feature vectors.
4. Using linear SVM model to train the classifier.

3.3.1.1 My experiment

In my experiment, I skipped the step of finding eyes as I dont have a eye detector to find the locations of eyes. Cropping and scaling is done by the location of face detected by the OpenCV face detector. The feature vectors are obtained by the method describe in section 3.1.1. The accuracy of the classifier, with a linear SVM kernel and C equal to 1000, is 81.6%.

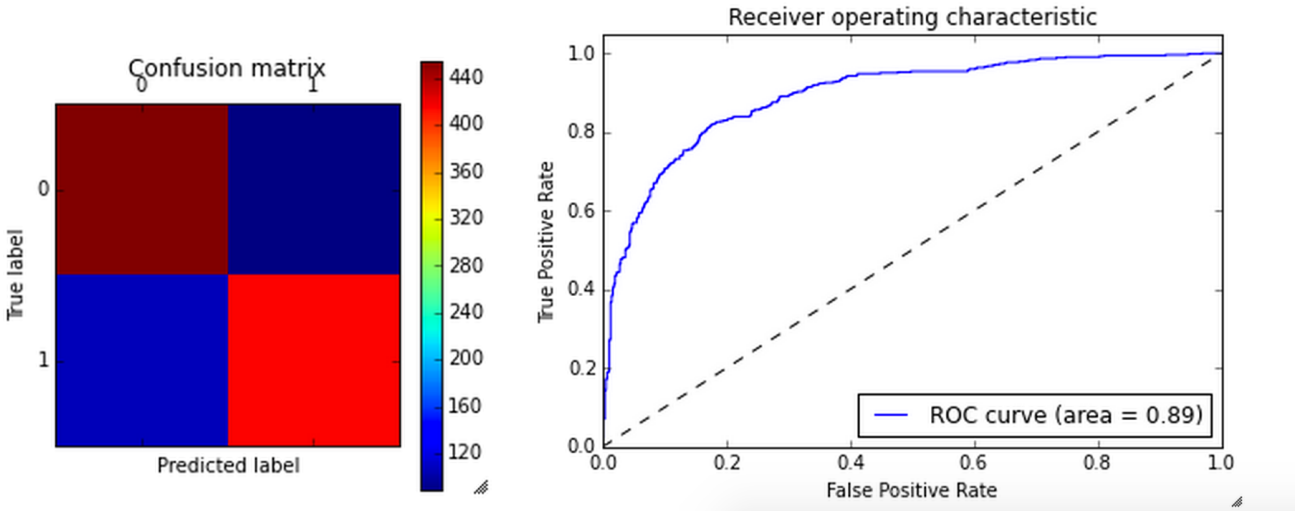

Figure 2: Confusion Matrix and ROC of Experiment 1

3.3.1.2 The Results

The detailed results of the classifier is listed in table 3. The confusion matrix and ROC curve are shown in figure 2

True Positive	415
True Negative	455
False Positive	89
False Negative	107
Accuracy	81.6%

Table 3: Result of Experiment 1

3.3.1.3 Optimization– Model Selection

The above results are computed from an optimised kernel and the C parameter. To get a better classifier, different models and parameters need to be compared to get the best classifier. In the Scikit-Learn toolkit[6], the grid_search module is designed for this purpose. In this experiment, following parameters and values are tried in the model selection process:

1. Kernel: linear, RBF;
2. C: 1, 10, 100, 1000;
3. Gamma: 0.001, 0.0001;

The mean scores of 5-fold cross validation are used to evaluating the models.Table 4 described the ranking of the parameters for the model.

Ranking	Accuracy	Kernel	G	Gamma
1	81.1%	Linear	1000	—
2	81.0%	Linear	1	—
3	81.0%	Linear	10	—

Table 4: Ranking of Parameters for SVM with 5-fold Cross Validation

3.3.1.4 Disccusion

In the paper[9], they used state-of-art method to get the locations of both eyes and used those information to do the image registration[1]. So in their training dataset, the images are much more better normalized, and the location of each regions almost fixed in the images after the registration. While in my experiments, only faces region is cropped and no other registration is applied. As the author of the paper pointed out, the step of locating the eyes was vita for higher accuracy.

3.3.2 Experiment 2

C Shan et, al. 2012[7] used the pixels intensity difference as feature vectors as described in section 3.1.2. They used adaBoost as the learning model and 500 most weighted features are selected from the classifier as the baseline to simplify the classify process.

3.3.2.1 My Experiments and Results In my experiment, adaBoost from Scikit-learn[9] is used as the library to train the classifier. The results by using 500 most weighted features are list in Table 5:

True Positive	352
True Negative	417
False Positive	127
False Negative	170
Accuracy	72.1%

Table 5: Result of Experiment 2

With 2000 pairs of most important features, the accuracy of this classifier could be increase to 74.6%.

3.3.2.2 Discussion Theres big gap between the results in my experiment and the results reported by C Shan et, al[7]. The reported accuracy is much higher than 90%. As they have showed in the result (see Figure 3), the classifier is very sensitive to the position of the pixels on the face. In that case, good image registration will play an important role for higher classification accuracy. In my experiment, faces are not normalized according the location of eyes, so different parts of face will distributed on different area on the image, which will decrease the accuracy of this learning algorithms.

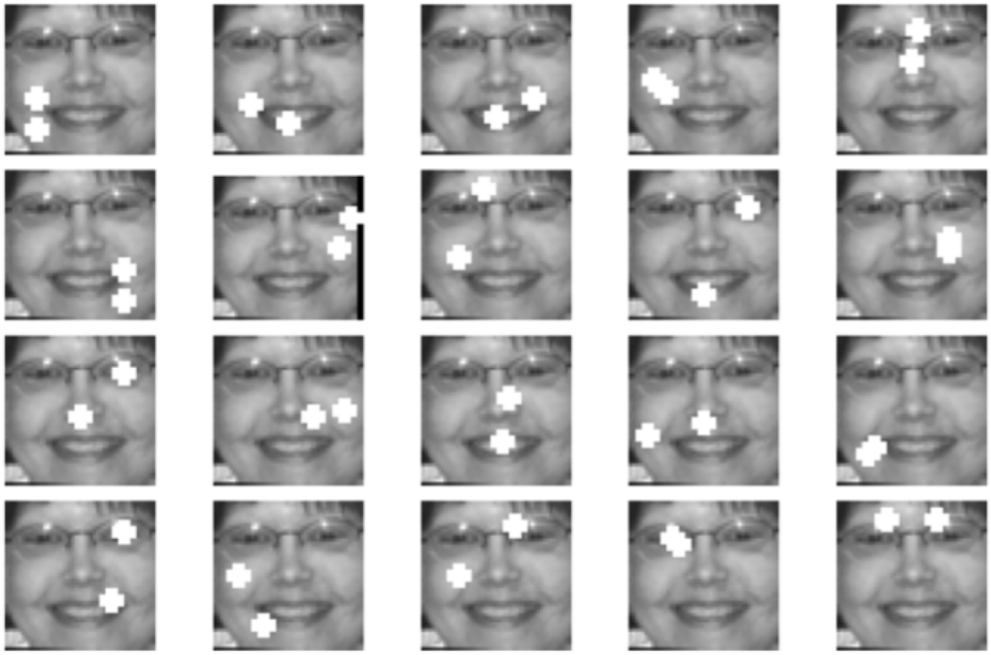

Figure 3: Top 20 pair of pixel being selected [7]

3.3.3 Experiment 3

In this experiment, a variety of other feature extraction methods were tried in order to discover more robust features to train the classifier. Feature extraction methods described from section 3.1.3 to section 3.1.7 were used, and linear SVM kernel with parameter C equal to 1000 are used to learn the classifier. Each feature extraction method was used once to train the classifier, and the results could be find in Table 6.

Feature extraction method	Learning Model	Accuracy
Area of Interest	SVM	73.9% —
HOG	SVM	73.5% —
SIFT	SVM	61.8% —
PCA	SVM	52.3% —
Bag of Words	kMeans + SVM	70.8% —

Table 6: Results of Experiment 3

4 Smile Detection with Convolutional Neural Network

Convolutional Neural Network(CNN) has been proved to be one of most accurate method for image classification in recent years. A lot of researchers have won lots of image classification competition in the past few years. Y Tang et, al.[8], the winner of FER2013 competition, used the CNN to train the

classifier and achieved an accuracy of 69.4% with 7 classes in the dataset.

Inspired by Y Tang et, al.[8], a simpler CNN which is similar to LeNet[5] (Figure 4) is designed and used. This network has two convolutional layers, two subsampling layers, two fully connected layers and one output layer.

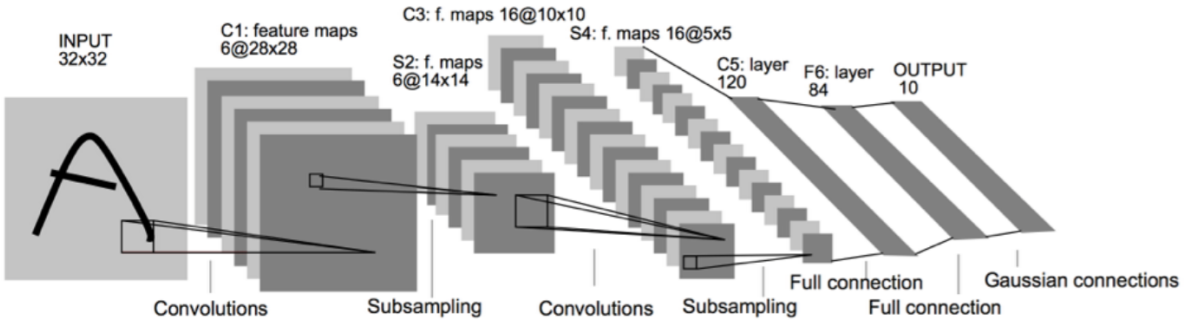

Figure 4: LeNet network structure: Yann LuCun 1998

The detailed network structure used in this project is described in Figure 5 and the description of each layer is in Table 7.

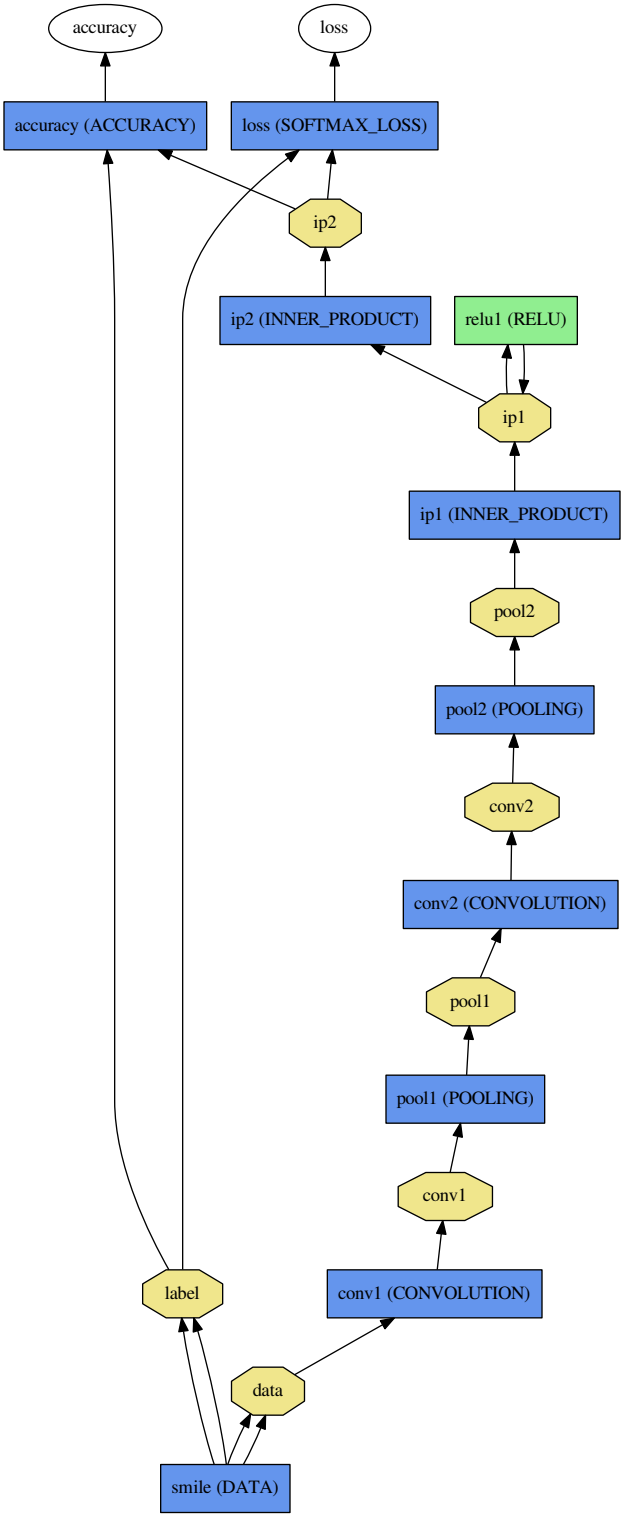

Figure 5: CNN structure for smile detection

Name of Layer	Description
Data	Input of the network with 48*48 nodes.
Convolutional layer 1	Apply 20 filters with size of 5*5 to each input image.
Pooling layer 1	Using Max Pooling with kernel size of 2*2 and stride of 2
Convolutional layer 2	Apply 50 filters with size of 5*5 to the output from pooling layer 1
Pooling layer 2	Using Max Pooling with kernel size of 2*2 and stride of 2.
Inner Product layer	Fully connected layer with input from pooling layer 2.
Relu layer	Fully connected layer and apply RELU function.
Output layer	Output 2 predicted labels

Table 7: Structure of CNN for Smile Detection

Figure 6 shows the results of an image processed by the first convolution layer. It could be found that the learned filters focused mainly on the mouth area which indeed is the most important area to tell if a person is smiling or not. For output of other deeper layers, its very difficult for human to understand what the algorithm has learnt. The output of the second convolution layer for the same image is shown in Figure 7, which is difficult to tell the content in those images.

Figure 6: An image applied 20 filters after the first convolution layer

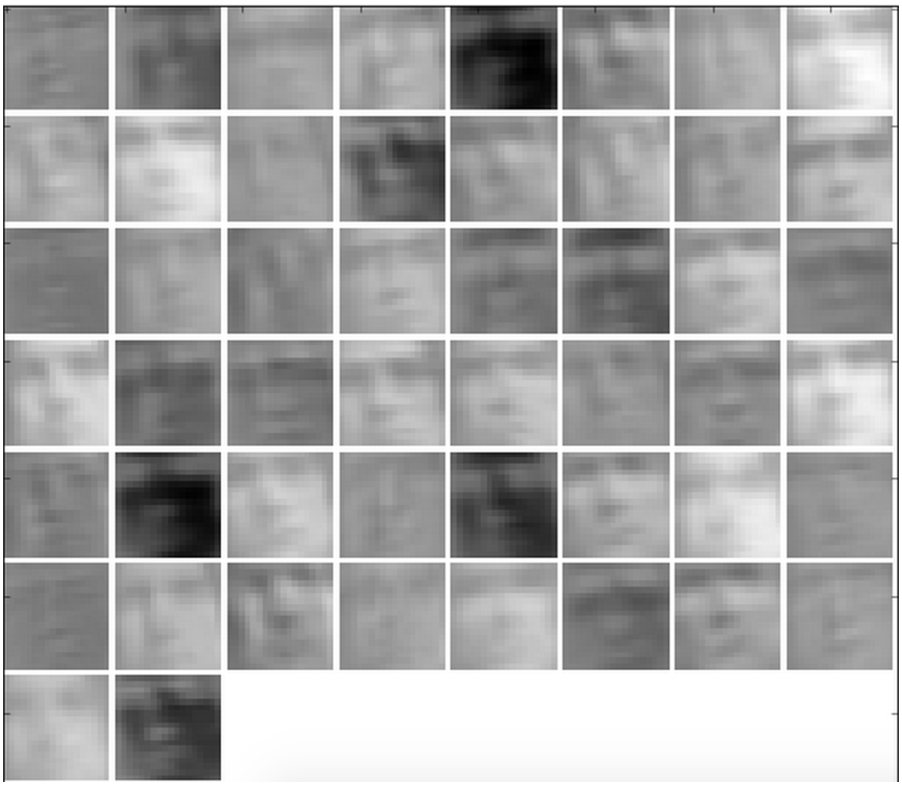

Figure 7: An image applied 50 filters after the second convolution layer

In this experiment, Caffe[3] deep learning framework was used to train the network and the results are listed in Table 8.

Number of iterations of gradient descent	Accuracy
500	83.8%
1000	85.4%
5000	86.5%
20000	87.1%

Table 8: Results of Smile Detection with CNN

5 Conclusions

In this project, the traditional machine learning methods and the convolutional neural network are used to learn the classifier for smile detection. With traditional machine learning method, different feature extraction methods are used and the corresponding performances are evaluated. Among all the feature extraction methods, the features got from Gabor energy filters had the highest accuracy. However, the accuracy of the methods used in experiment 1 and 2 depends highly on the preprocessing of image with image registration, as unnormalized images in training dataset would decrease the accuracy of the result. For the convolutional neural network, it got the highest accuracy among all the approaches.

However, it would be tricky to design an efficient network model, in this experiment, the simplest network structure(LeNet) is followed, and got the best result in all the experiments I have conducted. More complex and sophisticated network structure would definitely gain better result.

References

[1] I. Fasel, B. Fortenberry, and J. Movellan. A generative framework for real time object detection and classification. *Computer Vision and Image Understanding*, 98(1):182–210, 2005.

[2] Y.-H. Huang and C.-S. Fuh. Face detection and smile detection. In *Proceedings of IPPR Conference on Computer Vision, Graphics and Image Porcessing, Shitou, Taiwan, A5-6*, page 108, 2009.

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[4] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *Computer Vision–ECCV 2008*, pages 340–353. Springer, 2008.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[7] C. Shan. Smile detection by boosting pixel differences. *Image Processing, IEEE Transactions on*, 21(1):431–436, 2012.

[8] Y. Tang. Deep learning using linear support vector machines. In *Workshop on Challenges in Representation Learning, ICML*, 2013.

[9] J. Whitehill, G. Littlewort, I. Fasel, M. Bartlett, and J. Movellan. Developing a practical smile detector. In *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition*, 2008.