FACIAL Recognition with Deep Convolutional Neural Network

Wang Jin, Xu Han

Abstract

We trained a sophisticated deep convolutional neural network based on AlexNet[1] to realize face recognition. This carefully built network tried to identify if the given picture belongs to someone or not. The architecture contains three convoluntional and corresponding pooling layers, along with two full connection layers. Around 23,000 images of faces are fed to train this network, and finally an accuracy of 97% was achieved.

1. Introduction

Bioinformatics, like fingerprints[2], speech[3] and signature dynamic[4] has long been applied on human identification. Also, Chellappa et al.(1996)[5] introduced face recognition to recognize people's identity.

To deal with face image, CNN has been introduced by Steve Lawrence in 1997[6]. Recently neural networks, especially deep convoluntionalal neural network has been verified very efficient in computer vision. Many great progress has been achieved recent several years. In 1980s[7], Kunihiko Fukushima proposed a neocognitro, which is the processor of CNN. LeCun st al. (1990)[8] manually developed a convoluntional neural network to identify the handwriting digits. In 2012, Alex st al[1]. built an deep convoluntional neural network with GPU on Imagenet dataset for the ourpose of image classification and make great improvement on accuracy. Many other networks inspired by AlexNet, ZFNet[9], VGGNet[10], GoogleNet[11] , ResNet[12] and DenseNet[13], have been created and increasingly improve the accuracy of image classification accuracy. Since then, deep learning has been the most popular method to process facial recognition.

With these methods and efficient models, face identification has been more accurate than fives before. Many companies, like Apple and Alibaba, are dedicated to the development of facial recognition. Apple uses Face Recognition in their brand new devices-Iphone X as well as Alibaba uses Face Recognition in their core application-Alipay. In another word, Face Recognition is changing our life step by step. Also many other application of face image procession have been developed. Sometimes facial emotion of fear could be the signal for people to identify the criminal psychopathy[14].

Since other CNN models are developed based on AlexNet, so in this paper, we built a deep convoluntional neural networks besed on AlexNet. Chapter 2 introduces the dataset used in this paper, Chapter 3 will discuss the details of the CNN architecture, Chapter 4 is the discussion of the results and Chapter 5 will mention what kind of work still need to be done to get a more reliable result and what we need to pay attention in real life application.

2. Dataet and Data Argumendation

2.1 The Dataset

This dataset consists of two part: pictures of the interested person(here we use Mr. Wang Jin's face) and pictures of other people's faces, i.e., noises, and each component contains 11,500 images. In the following training process, the interested peoson's face will be labled with '1' while the noises will be labled with '0'. All the image data here are RGB pictures instead of gray pictures to make this CNN model more reliable.

For the first part, we captured faces using prepared vedios and computer front camera. Python libraries, OpenCV[15] and dlib[16] are introduced here to capture face images and to complete the following processing part, i.e., diversify the lightness, contrast and resize them. Also, to make the faces representative, this person here tried to make different kinds of dacial expressions. The size of original captured data is 250*250 pixes, as shown below(Figure 1).

Comparing to camera capturing, vedio capturing could be more efficient. Also it is convenient to get different background photos by using this method. Record videos from the cell phone allows us to try many different graphic effects and improve the diversity of the dataset.

Figure 1: Original faces of the interested person(Mr.Wang Jin)

Figure 2： Original noise data from Umass Amherst

For the second part, i.e. the noise dataset is composed of data set from University of Massachusetts, Amherst[17][18] and University of Science, Technology of China[19][20] and images captured using the same method as shown above. Actually the dataset from University of Massachusetts, Amherst is representative except too little faces of eastern Asian. That is important for our problem because the interested person is from easten Asia, and faces from different area in the world are quite different. So for the sake of training a robust model, which can distinguish faces from very close face images, this paper also introduced faces of dataset developed by Shangfei Wang et al. in University of Science, Technology of China. The sample of dataset are shown above as Figure 2 and Figure 3 is sample of processed data from USTC.

Figure 3: Processed East Asian face image from USTC

2.2 Data Argumentation

After reading data and just before the training process, firstly we used data argumentation to make learning easier and more accurate.

Firstly we resize the images for the sake of the limitation of traing time and computation ablity (here a NVIDIA GeForce GTX 1070 was used). After trying different kind of sizes, a trade-off between trainig time and accuracy is gotten, and 64*64 pixels is adopted in the following network training process.

The second data argumendation method is normalization. The number on imgage matrix after reading is between 0 and 255. Normalization to between 0 and 1 is used here, this can help make the model robust and also the result converge faster.

3. The Architecture

The network here is a deep convoluntionalal neural nerwork, which is based on architecture of AlexNet. It is composed of a input layer, convoluntional layers, corresponding layers, full connection layers and output layers. Also in the process of building networks, a series of layers structures, activation functions, cost functions, gradient decient optimizers are tried, they have different properties and performance in this problem. Finally we built a network to pursue the best performance.

3.1 CNN Architecture

3.1.1 Convoluntional and Pooling Layers

Compared with other mainstream pipelines, like HAAR and SIFT, CNN models have been proved to have much higher modeling capacity[21]. The size of input image is 64*64 pixels and the height is 3 since they are RGB pictures instead of gray pictures. The filter size of the first convolution layer is 3×3 with stride is 1, with the the output channel is 32.

For the second layer, i.e., after the frst convolution operation, we use pooling layer to fetch the “signals” of these features and control the overfitting. We use max-pooling to downsample. And the batch size is 2×2 with stride is 2, meaning this is non overlapping pooling. Pooling operation can reduce parameters needed and then calculation, this helps to prevent overfitting.

Similar to AlexNet, in this paper, there are also two pair of convoluntional and pooling layers. For these layers, finally we use the nearly same structure, 3×3 filter size and 1 stride for convoluntional layers, and max-pooling with 2×2 batch size.

But in the processs of building the architecture, we tried different kind of convoluntional window size because the size of convoluntion determines how much feature the filter can capture in one time. Althrough larger filter size can capture more bigger feature, smaller filter size can capture more local feature, at the same time, reduce calculation complexity. Here we tried six combination of convoluntional kernels, 2×2 , 3×3, 5×5, 7×7 for all three convoluntional layers, and also two different combinations of (3,5,5) and (3,5,7) for the edges of convoluntional kernel. The results are shown below(Figure 4), from which we can find networks with smaller kernel size may have good accuracy and plateausing time. So finally for each convoluntional layer, we adopted 3×3 for the filter size.

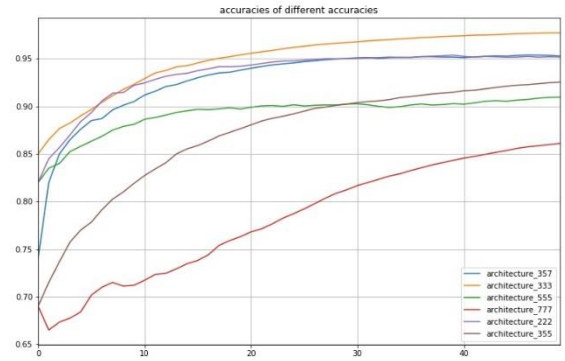

Figure 4: Accuracy of different filter size

3.1.2 Full Connection and Output Layers

After three pairs of convolutional layers and pooling layers, we use the normal full connections to connect the output layers. The tunnel size for the full connection layer are 512.

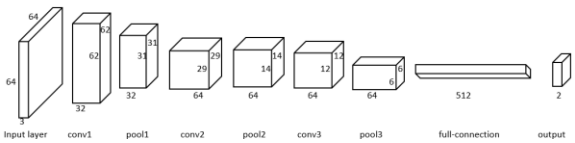

Figure 5: Final archetecture

For the output layers, there are two nodes which repectively represent the ‘yes’ or ‘no’ of classification results. And the result are labeled with 1 and 0 in the label matrix. Finally the architecture of this carefully built is below, as shown of figure 5.

3.2 Activation Function

For the network, we tested several activation functions, i.e., Rectified Linear Unit nonlinearity(ReLU), Exponential Linear Unit function(ELU), sigmoid, TanH and Softplus function. Fom the result below, it is easy to find that the last three are not suitable for this problem for their accuracy curve is alomost a constant and keep in a low level through the whole process. And both ELU and ReLU have good plateausing property and high accuracy, as can be shown on figure 6. Finally ReLU function are used in the network.

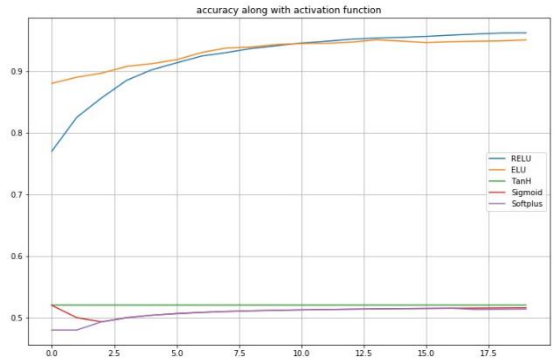

Figure6: Accuracy with different activation function

3.2 Cost Function

Cost functions are functions to evaluate the performance of classifiers, and different cost have different application scenarios and features.

In this face classification problem, five cost functions, cross_entropy, coscine_distance, hinge, mean squared error and sigmoid cross entropy are tried and compared, their impacts and results are shown.

From the figure 7, we can find except the coscine distance, all the loss functions left have good performance, short convergence time and high accuracy, especially the cross entropy and sigmoid cross entropy. Finally we use cross entropy in our network.

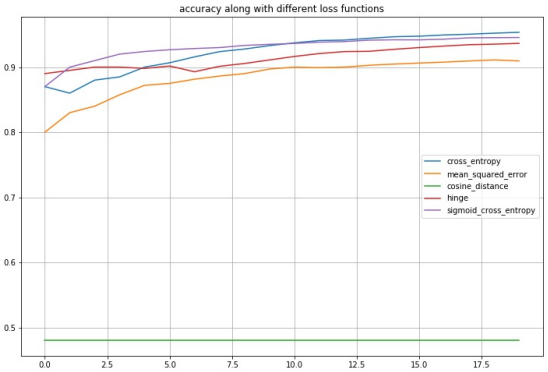

Figure7: Accuracy with different cost functions

3.4 Gradient Estimation

Gradient Descent is one of the most important method in machine learning. Many optimizer are developed to accelerate this process and keep it in the right direction. Here we tried optimizers of Stochastic Gradient Descent (SGD), Adam[22], Momentum[23], AdaGrad[24], RMSProp[25], and Adadelta[26]. From plot on figure 8, it is obvious to find Adam and RMSprop are good fits for this problem. For the sake of shorter plateausing time, we adopted Adam as the optimizer for training.

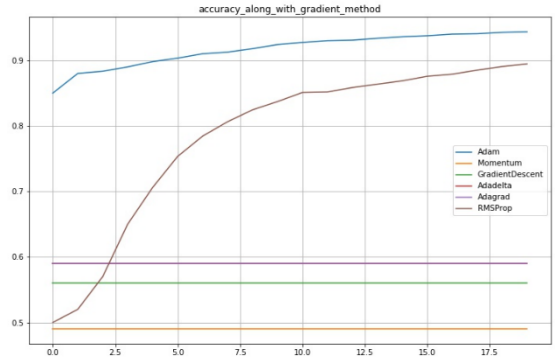

Figure 8: Accuracy with different optimizer

3.4 Other details

To avoid over fitting, drop out method is used here. Drop out means to drop some part of

networks(edges and nodes) to reduce the reliance on some particular edges and nodes, so that the network can still have good performance on validation dataset.

To meet the requirement of machine memory and at the same time push the accuracy to plateasu, batch are used when feed data in the training process. The final value is 200 images after the tradeoff.

To make a balance of convergence time and not drop into local optimum, a good learning rate is needed to be carefully designed. Here we use 0.005 after many tries.

To accelerate the learning process, the initialization method of weights and bias should be pay attention to, this network use normal distribution to creat the random values.

4. Results and discussion

4.1 Results

With the CNN networks described above, with the limitation of the machine, after 100 epoches of training, the final recognition accuracy could be 97.28%, which means evenly 2.72 images(of the interested person or the noise) out of 100 will be wrongly classified. This accuracy is good and show how powerful CNN in the area of image processing.

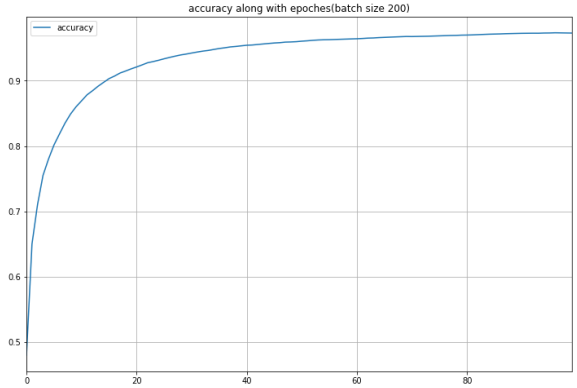

Figure 9: final accuracy on validation data

4.2 Discussion

There are still many works for this project to modify. Firstly, we can also try the newly designed networs like DenseNet and ResNet, to see if the recognition performance will be better. Secondly, to classify more accurately, three-dimention detection should also be included.

Acknowledgement

Many thanks to Professor Nicholas Brown, for the help of guidance and help in getting data. Also thanks to UMass Amherst and USTC for allow using there prepared image data. Thanks to Mr. Zhongjie He for providing his face image data in this project. Thanks to his and Mr.Seathirfwang's help in coding. Thanks to Google for the opensource deep learning framework Tensorflow [27].

References

[1] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems* (pp. 1097-1105).

[2] Blue, J.L., Candela, G.T., Grother, P.J., Chellappa, R. and Wilson, C.L., 1994. Evaluation of pattern classifiers for fingerprint and OCR applications. *Pattern Recognition*, *27*(4), pp.485-501.

[3] Burton, D., 1987. Text-dependent speaker verification using vector quantization source coding. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *35*(2), pp.133-143.

[4] Qi, Y. and Hunt, B.R., 1994. Signature verification using global and grid features. *Pattern Recognition*, *27*(12), pp.1621-1629.

[5] Mihalache, S. and Stoica, M.Z., 2014. Facial Recognition. *EIRP Proceedings*, *9*.

[6] Lawrence, S., Giles, C.L., Tsoi, A.C. and Back, A.D., 1997. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, *8*(1), pp.98-113.

[7] Fukushima, K. and Miyake, S., 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual

pattern recognition. In *Competition and cooperation in neural nets* (pp. 267-285). Springer, Berlin, Heidelberg.

[8] LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. and Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396-404).

[9] Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

[10] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[11] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*(pp. 1-9).

[12] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[13] Huang, G., Liu, Z., Weinberger, K.Q. and van der Maaten, L., 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*.

[14] Stanković, M., Nešić, M., Obrenović, J., Stojanović, D. and Milošević, V., 2015. Recognition of facial expressions of emotions in criminal and non-criminal psychopaths: Valence-specific hypothesis. *Personality and Individual Differences*, *82*, pp.242-247.

[15] Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, *25*(11), pp.120-123.

[16] King, D.E., 2009. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, *10*(Jul), pp.1755-1758.

[17] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. *University of Massachusetts, Amherst, Technical Report 07-49*, October, 2007.

[18] Gary B. Huang and Erik Learned-Miller. Labeled Faces in the Wild: Updates and New Reporting Procedures. *University of Massachusetts, Amherst, Technical Report UM-CS-2014-003*, May, 2014.

[19] Shangfei Wang, Zhilei Liu, Siliang Lv, Yanpeng Lv, Guobing Wu, Peng Peng, Fei Chen, Xufa Wang, A Natural Visible and Infrared Facial Expression Database for Expression Recognition and Emotion Inference, *IEEE Transactions on Multimedia*, Vol. 12, No. 7, Page(s): 682-691, NOVEMBER 2010.

[20] Shangfei Wang, Zhilei Liu, Zhaoyu Wang, Guobing Wu, Peijia Shen, Shan He, Xufa Wang,Analyses of a Multimodal Spontaneous Facial Expression Database, *IEEE Transactions on Affective Computing*, Vol. 4, No. 1, Page(s): 34-46, April 2013.

[21] Lu, L., Zheng, Y., Carneiro, G. and Yang, L. eds., 2017. *Deep learning and convolutional neural networks for medical image computing: precision medicine, high performance and large-scale datasets*. Springer.

[22] Kingma, D. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[23] Sutskever, I., Martens, J., Dahl, G. and Hinton, G., 2013, February. On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139-1147).

[24] Duchi, J., Hazan, E. and Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(Jul), pp.2121-2159.

[25] Tieleman, T. and Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, *4*(2), pp.26-31.

[26] Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

[27] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.