Image Matching Based on Siamese Neural Network and Convolution Neural Network (Based on Pytorch)

Xuanzhe Liu[1]

[1] Research School of Computer Science, Australian National University, Australia
u6920235@anu.edu.au

Abstract. In the experiment of Assignment 1, I used Bidirectional Neural Network [2] to match the features of face images, and finally got 69% accuracy. In this experiment, I used a new data set: it is no longer the coordinates and distances of the 14 feature points on the face, but the images of the face. It includes 36 pictures of human faces and is divided into 12 groups. Each group has two pictures of the same person and one picture of others. In technology, I use the deep learning method: Siamese Neural Network [3]. It consists of two convolutional neural networks, which are used to output the feature vectors of images and judge the similarity of two images. Through the adjustment of data set and hyper-parameters, the best accuracy is 63.8% which is lower than 69% of 2 layers BDNN [2] with different dataset and 75% of model in paper [1].

Keywords: Siamese Neural Network, Image Matching, Deep Learning, Convolution Neural Network

1 Introduction

Face recognition and feature matching are very popular in the neural network. They also have a very wide range of applications and practical value in life. This is the motivation why I chose the Facial-Features dataset.

The data set include 12 group of human face pictures, each group contains 3 pictures. For each set of 3 photos, A matches B, B does not match C, and C does not match A. In assignment 1, there are 14 markers on each picture, and the data I used is the distance between the 14 markers. In this experiment, I only used 24 images as data. The size, resolution and color of each image are different, so I adjusted the data, which I will explain in detail later.

I do not regard this problem as a simple classification problem. My goal is to output the feature vector of the image through the Siamese convolution neural network. After the neural network processing, the image of same person will get the vector with similar Euclidean distance. I will compare the Euclidean distances between the images to confirm if they match.

Different from the classic CNN and DNN, my model is Siamese Neural Network [3]. Traditional CNN can only receive one input, but Siamese neural network is composed of two neural networks which share weight. The two subnetworks receive an input respectively, map it to a high-dimensional or low dimensional feature space and output a feature vector, and then judge whether the two inputs are similar by comparing the similarity of the two vectors, such as Euclidean distance. Like CNN, it also uses back propagation algorithm to optimize parameters, but its loss function is called Contractual Loss, which can effectively deal with the relationship of paired data in Siamese neural network.

2 Methodology

2.1 Adjustment of data set

The original data set consisted of 12 groups with 3 images in each group. This means that there are a total of 12 labels 1 (meaning pictures are the same person) and 24 labels 0 (meaning pictures are not the same person). For neural networks, this is very insufficient data. It may result in over fitting or insufficient training. In addition, the number of labels for 0 and 1 is not balanced, which means that the model is not accurate enough. Therefore, I added 12 pairs of images labeled 1, which came from some rotated images in the original dataset. This not only increases the amount of data, but also makes the data more balanced.

Besides, in general, normalization is useful. In this dataset, the size, color and number of pixels of the images are different, so I resize them into 200 * 200 grayscale images.

2.2 Method for Implement Bidirectional Neural Network

The whole network is composed of two convolutional neural networks, and the overall structure is shown in Figure 2.2.1

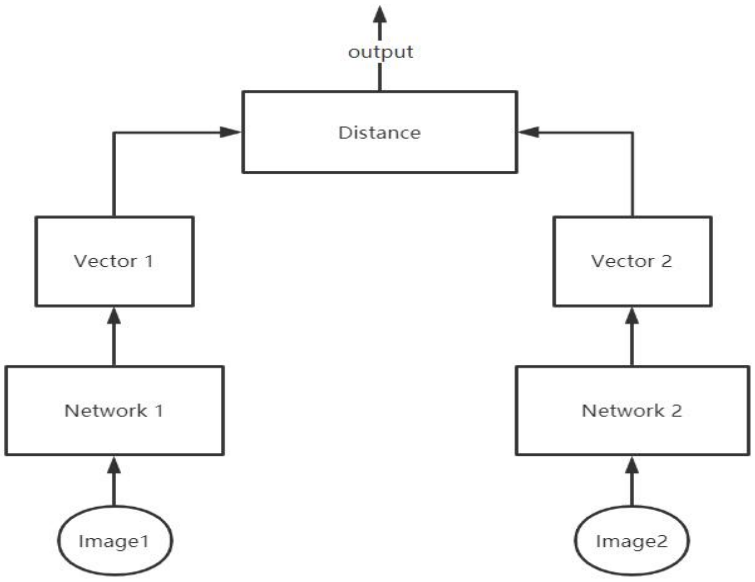

Figure 2.2.1 Implement a Bidirectional Neural Network

Initially, every convolutional neural network has three convolution layers and three fully connected layers, using ReLU as the activation function. The structure of each subnet is shown in Figure 2.2.2.

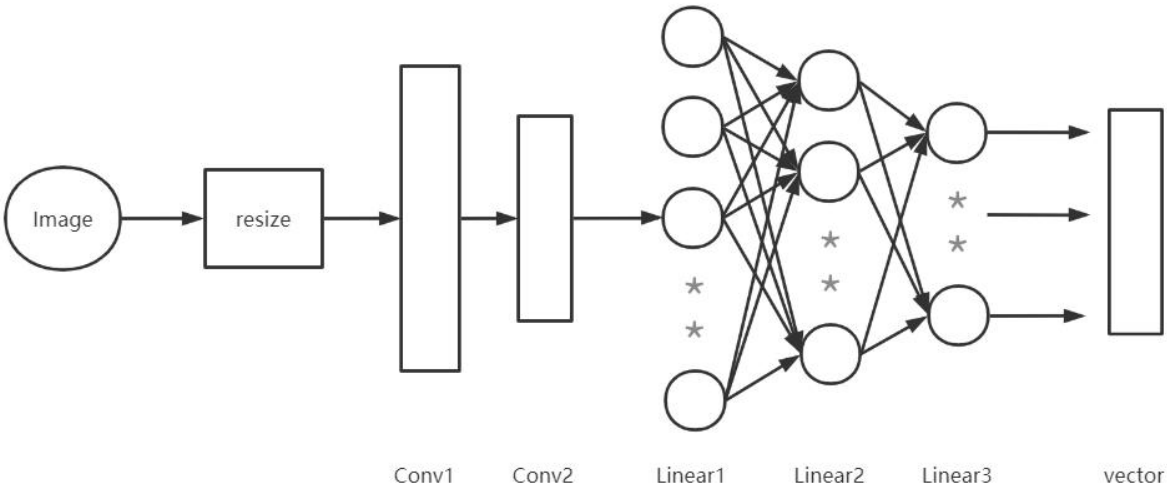

Figure 2.2.1 Implement a Bidirectional Neural Network

For the loss function, its expression is as follows.

$$L\big(W,(Y,X_1,X_2)\big)=\frac{1}{2N}\sum_{n=1}^{N}YD_W^2+(1-Y)\max\left(m-D_W,0\right)^2$$

$$D_W=\|X_1-X_2\|_2$$

Where D_W represents the Euclidean distance (two norm) of two sample features X_1 and X_2, P represents the feature dimension of the sample, Y is the label of whether the two samples match, Y = 1 represents the similarity or match of the two samples, Y = 0 represents the mismatch, m is a hyper parameter, which is the set threshold, and N is the number of samples.

This threshold m means that we only consider the dissimilar features whose Euclidean distance is between 0 and margin. When the distance higher than margin, the loss is regarded as 0 (that is, if the dissimilar features are far away, the loss should be very low; For similar features, we need to increase their loss.) [4]

This loss function can well express the matching degree of paired samples, and can also be used to train the model of feature extraction.

In general, we set the images to the same size and pixels, and feed the two images into two sub neural networks respectively. Each image will be processed by three layers of convolution to extract features, and then mapped to a low dimensional vector space through three layers of full connected linear layers to generate feature vectors. Next, the loss function calculates the distance between the two feature vectors and updates the network parameters. It is important to note that the weights of the two subnetworks make shared, that is, the same.

Finally, we feed the test images into the neural network and get the Euclidean distance of their feature vectors. Then set a threshold, if the distance is greater than it, it means they are the same person, otherwise they are not.

2.3 Adjustment of hyper-parameter and structure

The hyper-parameter of this model includes number of hidden layers, number of nodes in each hidden layer, training epoch, optimizer and learning rate. Through experiment, the optimal hyper-parameters are 2 convolution layers, 3 linear full connected layers, Adam optimizer, 10 epochs, 0.005 learning rate.

The number of parameters for each layer is shown in table 2.3.1

Layer	***Conv1***	***Conv2***	***Linear1***	***Linear2***	***Linear3***
Parameter	Kernel=3	Kernel=3	[48672, 500]	[500, 300]	[300, 10]
Active Function	ReLU	ReLU	ReLU	ReLU	None

Table 2.3.1 The Parameter of Siamese Neural Networks

2.4 Evaluation and Test

I use the cross validation method to test, because the data is too little to divide the test set and training set. I divide the data set into 6 groups, each group has 5 data, each time I take 5 groups as the training set and 1 group as the testing machine. After 6 times of training and testing, take the average of accuracy as the final accuracy. I also took all the data as a test set and calculated the F1 score [3].

3 Results and Discussion

3.1 Result of Neural Network

The accuracy in different dataset with different model are shown in Form 3.1.1. The loss in Image dataset with training epochs is shown in Figure 3.1.1

Dataset	*Model*	*Accuracy*	*F1-score*
Distance	DNN	83.3%	0.67
Images	Siamese Neural Network	56%	0.43
Extended-Image	Siamese Neural Network	61%	0.56
Extended-Distance	BDNN	69.4%	0.44

Table 3.1.1 The Accuracy of different Neural Networks

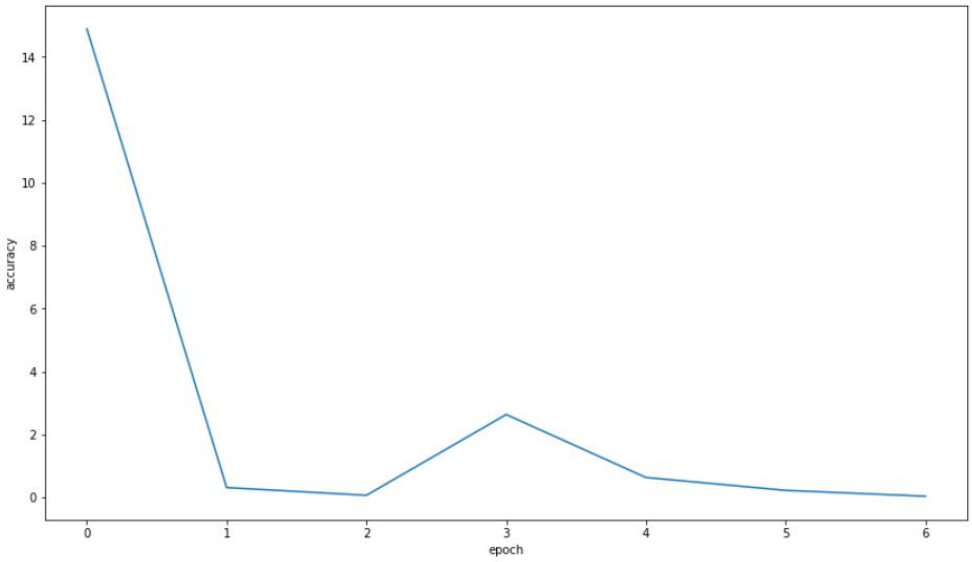

Figure 3.1.1 The Loss of Siamese Neural Network

3.2 Result of the Comparison with the Previous Paper

Through experiments, we can find that DNN has the best prediction ability. Siamese Neural Network is not as effective as DNN on FFMs-Distance datasets. The distance dataset effect is better than the Image datasets. However, F1-socre of Siamese Neural Network are higher than BDNN but it is still low. I think this is due to the lack of data and imbalance. Even if the data set is expanded to have the same number of labels 1 and 0, it is still possible that the predicted values are all 0 or 1.

In terms of data sets, the effect of Extended-Image is better than original dataset. However, we can see from Image 3.1.1 that loss decreases rapidly in the first epochs and then tends to remain unchanged, which is also caused by too little data.

Compared with the accuracy of the paper [1], the performance of my model is obviously not good enough. I think the main reason is that training neural networks with a large number of parameters requires a lot of data. Obviously, 60 pictures make it far from enough. The parameters in the network cannot be fully trained, but once the training rounds are too many, there may be over fitting.

Besides, there are also problems with the quality of the pictures. One picture of a certain person is only 100 * 100, while the other one is 300 * 400, which makes it difficult for even the same person to train similar vectors.

4 Conclusion and Future Work

Based on all the experimental data, the performance of Siamese Neural Network is not as good as that of traditional DNN, KNN or SVM [1]. Because Siamese Neural Network is more suitable for complex image matching problem with more data. In addition, there are some problems in the dataset, including but not limited to the data quantity too little and the sample imbalance. These factors together lead to lower accuracy and F1-score.

In the future, we must obtain more high-quality data to train our model. We can use manual labeling to get more data. We can use more data to train more complex models.

References

1. Caldwell, S. (2021) "*Human interpretability of AI-mediated comparisons of sparse natural person photos*," CSTR-2021 1, School of Computing Technical Report, Australian National University.
2. A. F. Nejad and T. D. Gedeon, "Bidirectional neural networks and class prototypes," *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1322-1327 vol.3, doi: 10.1109/ICNN.1995.487348.
3. S. Chopra, R. Hadsell and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 539-546 vol. 1, doi: 10.1109/CVPR.2005.202.
4. Blog.csdn.net. 2021. *(Siamese Network)*. [online] Available at: <https://blog.csdn.net/weixin_45250844/article/details/102765678> [Accessed 26 May 2021].