REPORT OF FACE RECOGNITION

Yixiao Yuan
School of Artificial Intelligence
Southeast University
School number: 58119201

August 17, 2021

ABSTRACT

We use Inception Resnet (V1) models to implement face recognition. We fine-tune pretrained model using center loss with our dataset. Finally, we achieve 98.8% accuracy on face recognition and 0.9 F1 score on face verification.

1 Introduction

We have a dataset of 246 face images from 41 students (6 images per student) in our machine learning class. The goal is to implement face recognition and face verification based on the dataset. We separate it into train set and test set. There are 4 images per student in the train set and 2 images per student in the test set. We fine-tune Inception Resnet models by implementation of center loss. After we get the deeply learned features, we use K- Nearest Neighbor to implement face recognition, and set a threshold of euclidean metric between two face images to verify whether it is a match to implement face verification.

2 Procedure

2.1 MTCNN: face detection prior

MTCNN is a tool for face detection prior to inference. We clone it from https://github.com/timesler/facenet-pytorch. The MTCNN module can crop the faces from original images and return a normalized tensor with 3 channels of RGB. It is worth notice that images in the dataset may be revolved when loaded because of EXIF, and in this suitation MTCNN can't detect any valid face. Therefore, we need to ignore information about orientation in EXIF when we load dataset. Besides, our model is were trained on 160x160 px images, so will perform best if applied to images resized to this shape. We check the result and 245 face images are cropped correctly.

2.2 Fintuning Resnet

2.2.1 Loss Function

To achieve the best performance in our dataset, we choose to fine-tune the pretrained model instead of using it directly. Our model is based on inception resnet [1], which helps to extract the features of face images. Because our dataset is small, we choose to fine-tune the model pretrained on VGGFace2 [2]. We choose center loss [3] instead of triplet loss in [2] as our loss function to update the parameters in the model, because center loss is easy to implement and it can improve the discriminative power of the deeply learned features which fits our small dataset.
Center loss is defined as follows

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2 \tag{1}$$

The $\boldsymbol{c}_{y_i} \in \mathbb{R}^d$ denotes the y_i th class center of deep features.
We adopt the joint supervision of softmax loss and centerloss to train the CNNs for discriminative feature learning. The formulation is given in Eq.2.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_S + \lambda\mathcal{L}_C \\ &= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}} + \frac{\lambda}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2\end{aligned} \tag{2}$$

It is worth noting that due to the existence of batchnorm layers we must use eval model to train to avoid our dataset changing the batchnorm layers.

2.2.2 Model Selection

The optimizer we use is SGD .We use different hyperparameter to train the model and test on test set. We use KNN to classify the face images. We compare the accuracy of classification using k from 1 to 4. The reason why we use criterion with mixed k is that our dataset is small and the difference between models fine-tuned on different hyperparameter is small,so if we evaluate our model on a single k, we may make a wrong choice because of variance.
Finally, we choose $\lambda = 0.01$ and train 10 epochs with batch size 32. See comparison in Table 1.

Table 1: Recognition accuracy on test set

k	Original model	Finetued model
1	98.8%	98.8%
2	96.3%	97.6%
3	97.6%	98.8%
4	93.8%	97.6%

2.3 Threshold Selection

After finetuing the model, we can get 512-dimensional embeddings of face images. We can implement face recognition with KNN easily, but how can we implement face verification? We just need to set a threshold of distance between two face images. If the distance is larger than the threshold, we assume it is a match. Otherwise, we assume it is not a match. We use the test set to create 41 positive pairs and 1640 negative pairs and evaluate on different threshold value.To select the appropriate thresholdm, we draw the PR curve on test set, as shown in Fig 1.

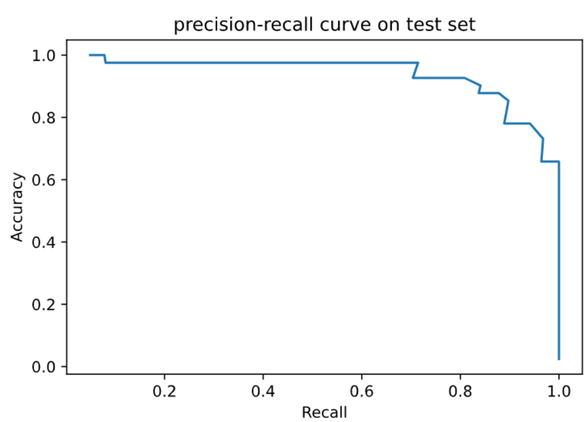

Figure 1: PR curve on test set

Because our test set is small, so the result of precision and recall is discrete. We also draw the F1 score on test set, as shown in Fig 2.

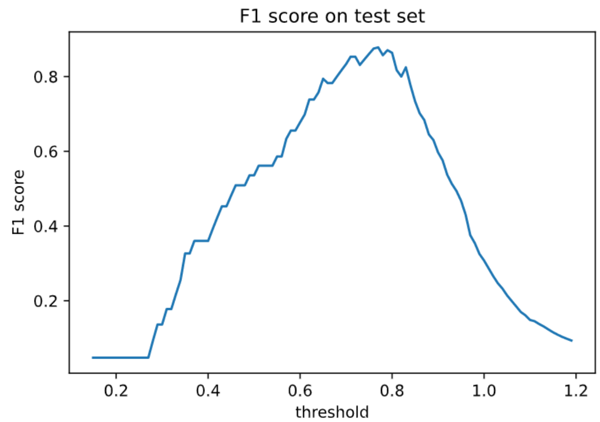

Figure 2: F1 score on test set

According to the curve, we can choose 0.77 as the threshold, which achieve 87.8% accuracy , 87.8% recall and the best F1 score of 0.878. But we have For a new face image and a person name, we will compare this image with all the images of the person in our dataset, and vote to determine whether it is a match. We also test the vote methods in the evaluation code provided by the task using our dataset, and the result is in Fig 3.

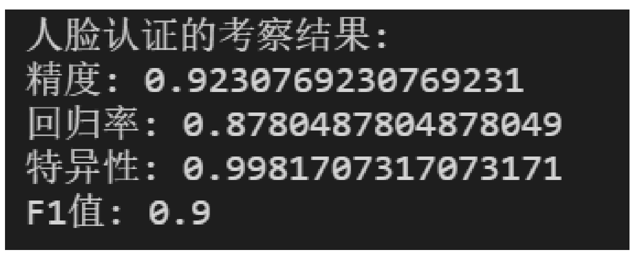

Figure 3: Verification Result

3 Conclusion

In this project, we have implemented face recognition and face verification on our dataset and achieve good performance. Although when I review the procedure, I wasted so much time on stupid problems. For example, I even did not know we must set model to eval mode before inputting. In the project, I followed a complete pipeline of deep learning and learned a lot.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[2] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[3] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.