Machine Learning Assignment

JingLiu 18231917
MiaoLi 18230232

Part 1:

For the part 1 the triplet loss function is using the distance calculation and firstly create the embedding model, here shown as figure 1 below I add two layers for convolution2D and then add maxPooling model and using dropout to reduce the weights then flatten it, then add dense layer and dropout last using dense again to get the final embedding, here the embedding represent a vector of abstract features of one image.

On the complete model, here firstly input three image for anchor, negative and positive image, then using base model to get the embedding of the image and then calling the triplet loss to minimise the distance between positive and anchor and maximise the distance between negative and anchor. The final loss value of training and validation shown as figure 1.

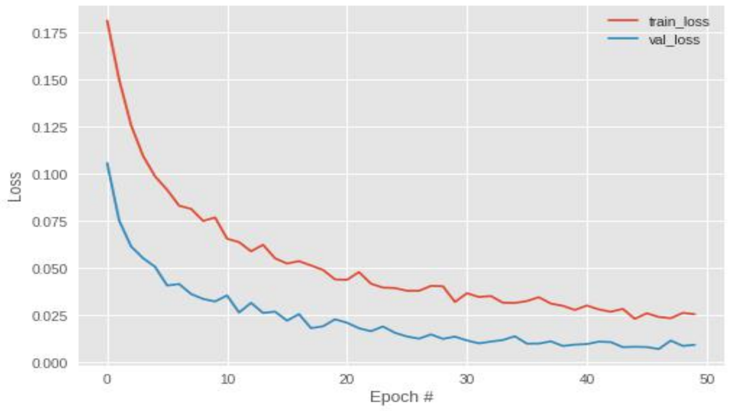


Figure 1

For the embedding figure, using the base model to predict the embedding values and using canvas to plot it, shown as figure 2.

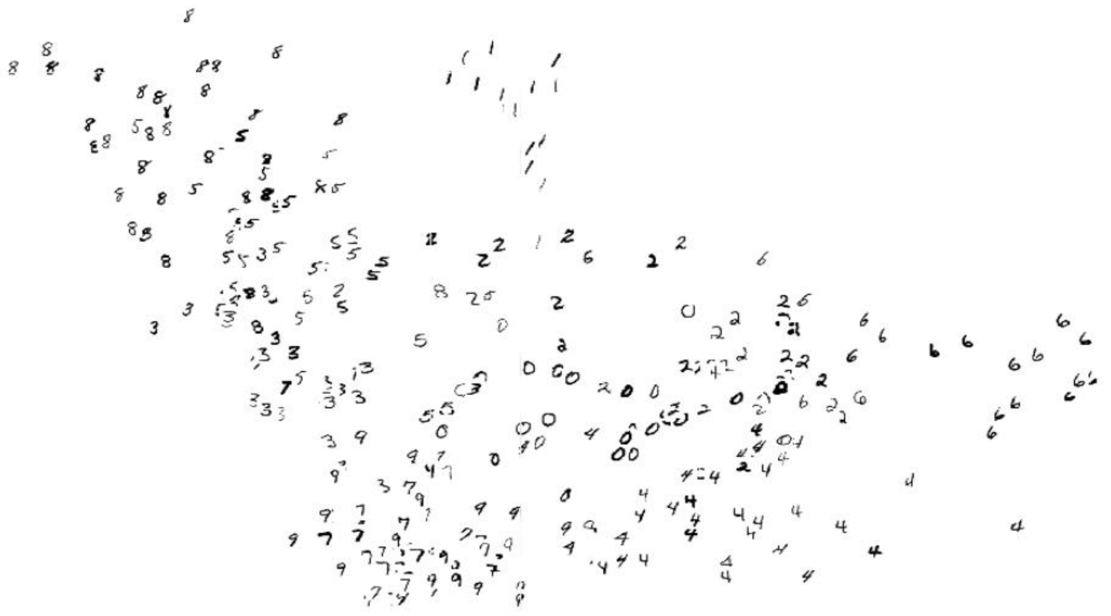
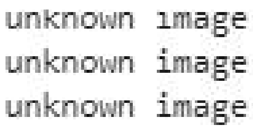


Figure 2

For the image recognize function, here I firstly randomly choose 10 images from 0-9, and call the base model to get the embedding model for each digit image and letter image, then calculate the distance with each digit and letter, find the minimal distance if it greater than 0.5 then return unknown otherwise return the digit label. The result shown as figure 3.



```
unknown image
unknown image
unknown image
```

Figure 3

Part2:

First I tried pre-training model VGG16 from Keras, but the accuracy is not very good. Therefore, I used model which is downloaded online. The file name is “vgg_face_weights”.

1, Use : `vgg_face_descriptor = Model(inputs=model.layers[0].input, outputs=model.layers[-2].output)` to get embedding.

2. Then use K-mean to make classification, I classify 3 clusters.

3. Image show: show 4 pictures for each cluster

The accuracy is better and the result is as figure 4:

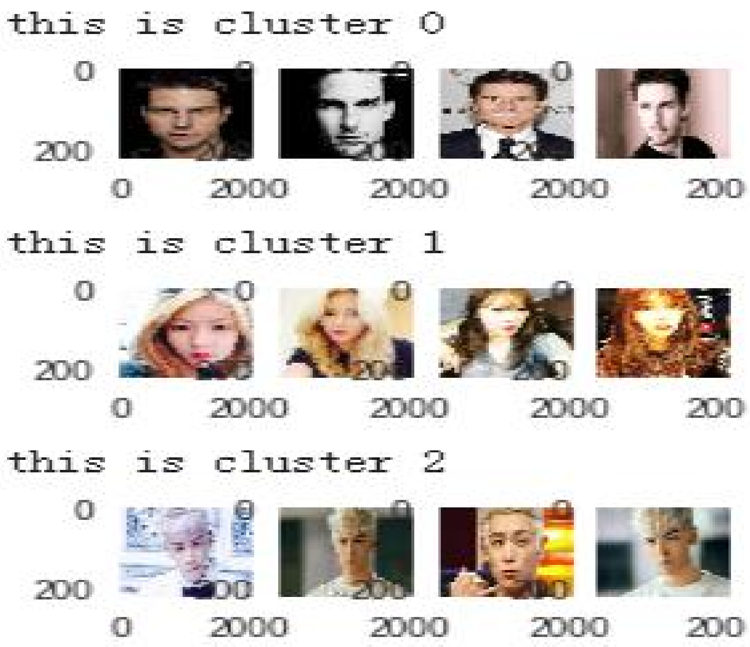


Figure 4

Reference

[1]. Vgg-model download:
https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/

[2]. Image processing:
https://keras.io/applications/

[3]. Triplet Loss
https://blog.csdn.net/u013082989/article/details/83537370

[4]. Image understanding - CNN Triplet loss
https://stackoverflow.com/questions/53572865/image-understanding-cnn-triplet-loss