AMMAI HW1 Report

- **Problem Definition**

  This homework is to apply deep CNN models to cross-domain face verification problem. Three loss functions were adopted, and their effectiveness in cross-domain inferencing were further evaluated in terms of their AUC of ROC curves.

- **Model Architecture**

  Coping with this open-set problem, CNN models were trained to embed each face image of size (112,112) into a 512 length vector. Each vector is normalized to a specific L2-norm 64; in other words, they are all restricted to the surface of a 512-dimensional hypersphere with radius 64.

  Resnet50 of Keras was chosen to be the backbone and was initialized by weights pre-trained using ImageNet dataset. A fully connected layer with biases was employed after Resnet50; a L2 regularization cost is added to this dense layer.

  This encoder is trained as a classification problem. Each embedded 512-dimensional vector is transformed into softmax logits by a Kx512 matrix whose rows are all unit vectors. K is the number of classes in training set.

- **Dataset Description**

  - **Testing**

    Testing data provided in the homework description contains 20008 labeled images out of 701 identities. This dataset is obviously noisy, as shown in the following picture. However, since only 21796 pairs (half positive, half negative) are sampled for model evaluation and that hopefully no inconsistency was found at a glance, we shall ignore this issue.

  - **Training**

    A collection of faces of Asian celebrities was used as the training data and was downloaded from DeepGlint (see Reference). It contains 232076 face images of 4366 identities (~7.5GB). Due to time limit and memory constraint, only 700 identities were sampled as training data which actually contains 42100 images.

Let's take a closer look at this dataset. Many of them are quite distorted and contain unnecessary backgrounds; some of them are also side-faces and moreover, even covered by water-prints. These noises lead to the conclusion that alignment does boost performance. We will discuss more in the following.

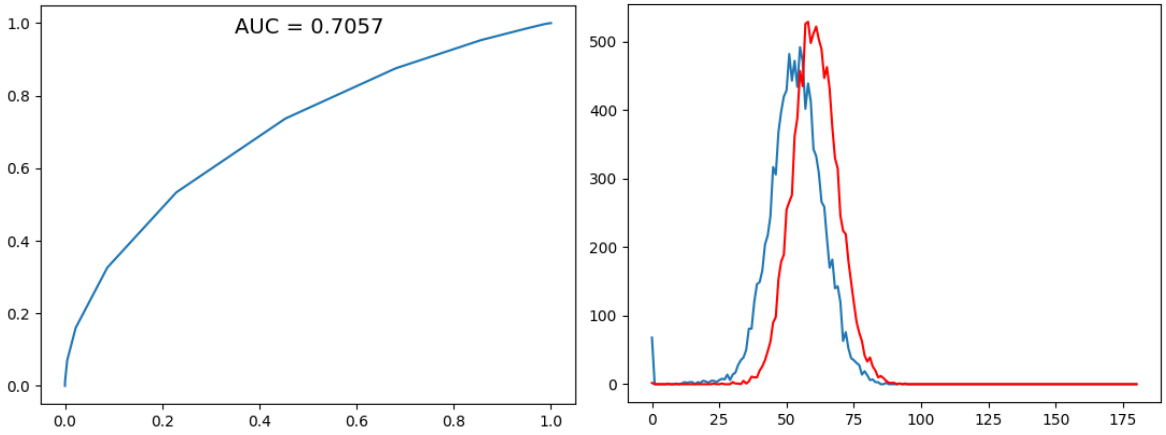
Some of the hard examples

- **Loss Functions**
  - **Vanilla Softmax**

    As the previous section explained, each face image is encoded to a 512-dimensional vector with length 64 and then further transformed into K real numbers, the logits for softmax function. Since our training set contains 700 identities, K equals to 700. Note that the transition matrix has 700 unit row vectors, so the logits can be viewed as s (radius of the hyper-sphere) times the cosine similarity of the embedded image vector and row vector.

$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\theta_{y_i}}}{e^{s\cos\theta_{y_i}}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

    The following shows the ROC curve evaluated on testing data. Although the model has gain over 95% of classification accuracy on the training data, it seems that the learned embedding is not general enough to discriminate unseen identities. The figure on the right illustrates the histogram of the angles corresponding to the cosine similarities of testing pairs. Obviously, the mediocre performance results from the lack of capability to separate positive and negative instances.



■ Softmax with Intra-Loss

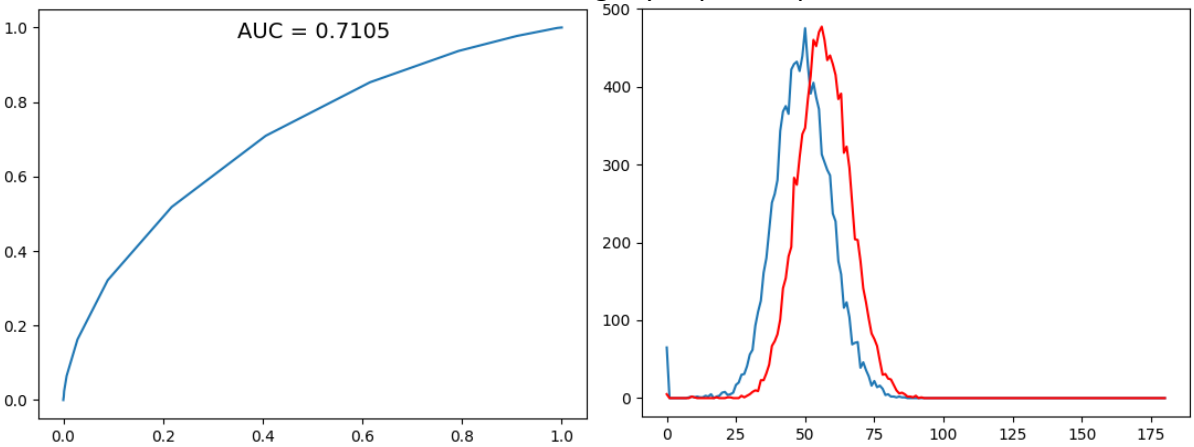
Heuristically, the embedded vectors should not only be separable but also discriminative on the hypersphere in order to possess more precise semantic meaning. That is, vectors within the same category shall be close on the hypersphere since they share similar semantic meanings.

However, it is hard to calculate center loss in each batch because within a small batch, say size of 100, we can expect that each appeared classes will only contain a few instances, some even have one only. Moreover, the center calculated in each batch may be inaccurate and fluctuates since it is computed over small number of points.

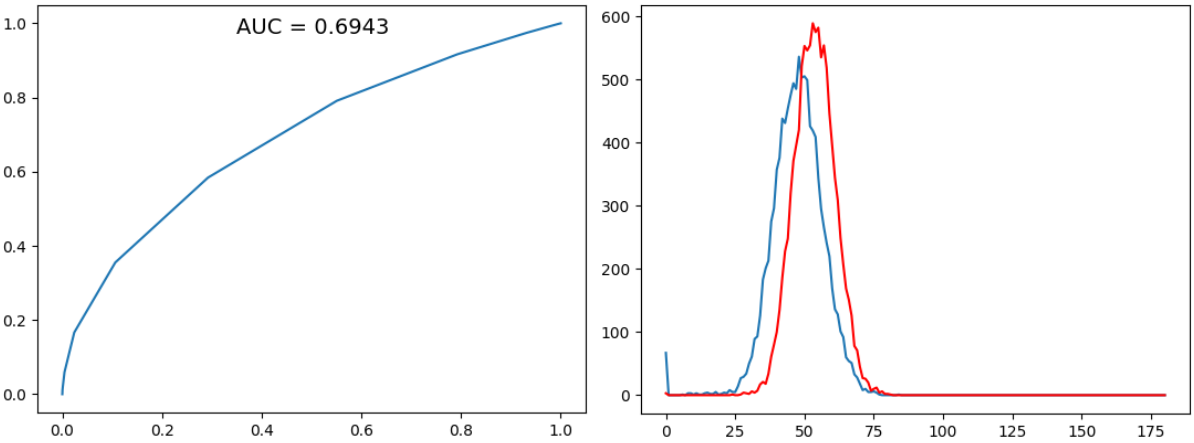
To formulate intra-class compactness, we use the row vectors of the transformation matrix as an approximation of the center. Hence, we only need to compute the mean of angles between each image vector and its corresponding row vector in every batch. This gives us the second function.

$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\theta_{y_i}}}{e^{s\cos\theta_{y_i}}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}+\frac{\alpha}{\pi N}\sum_{i=1}^{N}\theta_{y_i}$$

Again, we show the ROC curve and the angle histogram of testing pairs. We see that this new cost function does slightly improved performance.

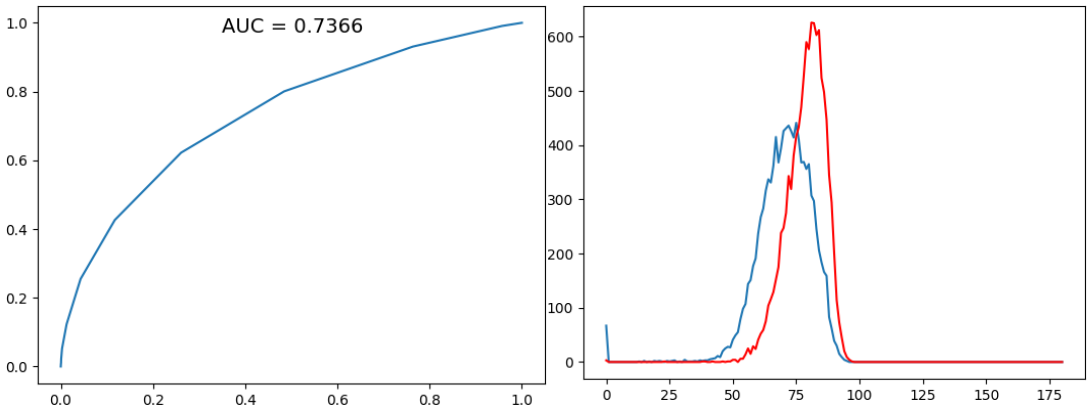


Note that if we change α from 1.0 to 0.1, the performance dropped since the added cost was not emphasized enough while disturbing classification.

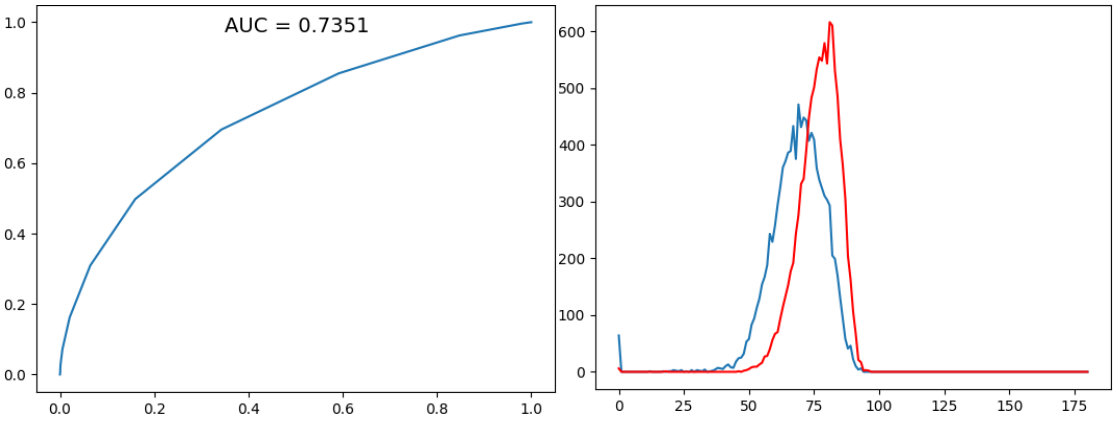


■ Softmax with Large Margin

Now we inspect the third loss function which leads to the best result. Before taking the logits into softmax function, we add a margin m to the angle (only the one corresponding to the real identity) between the row vector and image vector. This handicap forces the encoder to map vectors as close as possible to its clustering center (the corresponding row vector). Moreover, this angle should be at least m smaller than the second close cluster. This approach has boosted performance greatly, as shown in the following figures.

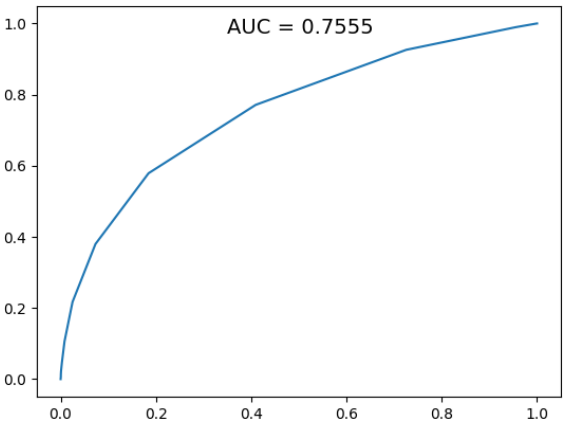


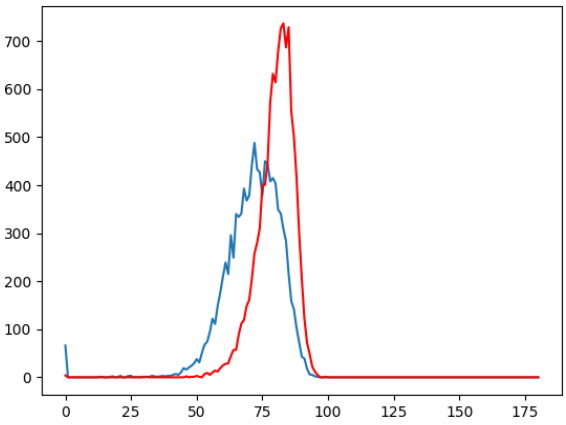
Note that larger margin does not necessary leads to better performance. When we changed the margin from 30° to 45°, the performance dropped. Moreover, when margin was set to 60°, the model even failed to perform classification and the accuracy on training data remained under 1% while model loss kept dropping whatever.



- **Further Discussion**
  - **Does Ethnicity Cares?**

    Intuitively, we expect better performance on models trained by Asian faces. Let's look at the result of the model trained on Westerners with the same large margin cost.



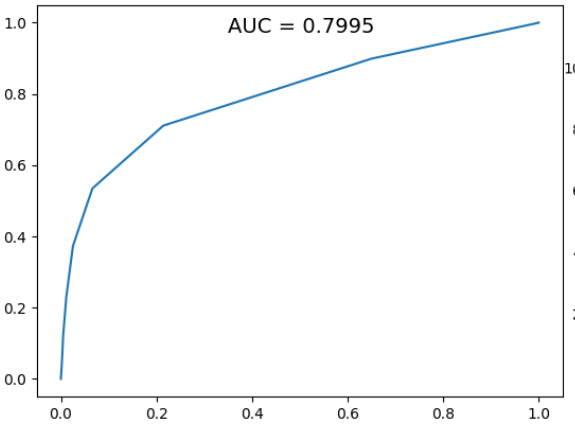


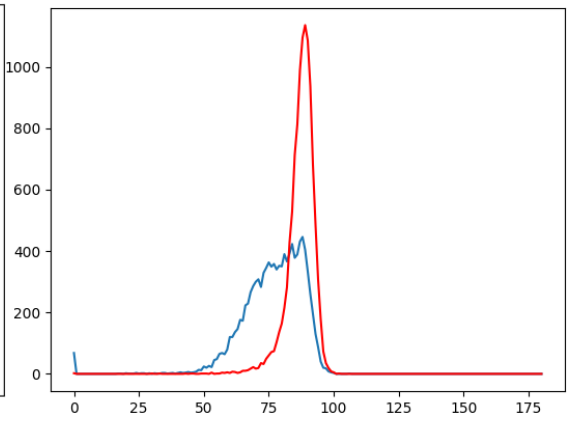
    This model is trained on ms_celeb_1M. Similarly, we only sample 700 identities, containing 43086 images. It is quite counter intuitive, since most faces in this dataset are of Caucasians, differing greatly from Asians.

  - **Importance of Segmenting and Alignment**

    The reason why we had better performance if we trained on ms_celeb_1M is that faces in it are well-segmented and aligned. As mentioned in the training data description, many images contain quite part of background, and plenty of them are distorted severely.

    After segmenting and aligning by dlib face detector, the new Asian training set contains 39677 faces out of 700 identities (dlib failed to detect faces in some images).





    This leads to the reasonable result that training on the same ethnicity does help improve performance.

- **References**
  - Asian Dataset from DeepGlint: http://trillionpairs.deepglint.com/overview
  - ArcFace: Additive Angular Margin Loss for Deep Face Recognition

    Deng at. el.
  - Code reference & ms_celeb_1M: https://github.com/peteryuX/arcface-tf2