Francisco Javier Sáez Maldonado
franciscojavier.saez@estudiante.uam.es

Face Recognition Lab Report

Deep Learning for Biometric Signal Processing

2022-02-22

Task 1

Based on the provided code, run the "FaceRecognition.m" file and complete the following points.

1.1 Paste one image of the ATT Face Dataset and the corresponding image after using the 2D Discrete Cosine Transform (DCT)

Firstly, I randomly changed the pre-set image to load and show. The one selected was the one shown in Figure 1

(a) Original Image: *s07/3.pgm*.

(b) DCT applied to original image.

Figure 1: Examples of fingerprints.

1.2 Using the original configuration parameters (train = 6 images, test = 4 images, DCT coefficients = 10), plot the resulting DET image and indicate the resulting EER.

1.3 Find out the configuration of the DCT coefficients that achieves the best EER results (keeping train = 6 images, test = 4 images). Justify your result, including the resulting DET image and EER value.

1.4 Once selected the best configuration of the DCT coefficients (in previous point), analyze the influence of the number of training images in the system performance. Include the EER result achieved for each case (from $train = 1$ to $train = 7$). Justify the results.

Task 2

The goal of this task is to change the feature extraction module. Instead of using DCT coefficients as in Task 1, you must consider Principal Component Analysis (PCA) to extract more robust features. You can use the pca.m function available in Matlab. For the training phase, you should follow:

```
[coeff_PCA,MatrixTrainPCAFeats,latent] = pca(MatrixTrainFeats);
meanTrainMatrix=mean(MatrixTrainFeats);
```

It is important to remark that the PCA function must be applied once for all training users and samples (not one PCA per user as this would provide specific `coeff_PCA` parameters per user). For the test phase, you should follow:

> For each test, subtract the meanTrainMatrix, and multiply by the `coeff_PCA` transformation matrix in order to obtain the test features in the PCA domain.

For more information, check Matlab Help: https://es.mathworks.com/help/stats/pca.html

2.1 Using the parameters train = 6 and test = 4, paste the DET curve and indicate the EER when using all the PCA components.

2.2 A key aspect for the PCA is the number of components considered. Analyze and represent how the EER value changes in terms of the number of PCA components. Give your explanation about the results achieved.

2.3 Indicate the optimal number of PCA components and paste the resulting DET curve together with the EER achieved. Compare the results using PCA with the DCT features considered in Task 1.

Extra Task

The goal of this task is to improve the matching module. Instead of using a simple distance comparison, you must consider Support Vector Machines (SVM). In this case, you should train one specific SVM model per user using the training data (train = 6 images). Features extracted using the PCA module developed in Task 2 must be considered in this Task. You can use the fitcsvm function available in Matlab. For the training phase, you should follow:

```
SVMModel = fitcsvm(...)
```

For the test phase, you should follow:

```
[label,score]= predict(SVMModel,MatrixTestFeats);
```

to obtain the scores for each user model. For more information, check Matlab Help: https://es.mathworks.com/help/stats/fitcsvm.html?lang=en

3.1 Using the parameters train = 6 and test = 4, paste the DET curves and indicate the EERs in the following cases: 1) regarding the KernelFunction parameter of the SVM (using all PCA components), and 2) regarding the number of PCA components considered for the feature extraction module (using the KernelFunction polynomial and starting with 3 PCA components).