Face Biometrics Lab - Report

Hoyo Bravo, Guillermo

Task 1: Based on the provided code, run the "FaceRecognition.m" file and complete the following points.

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

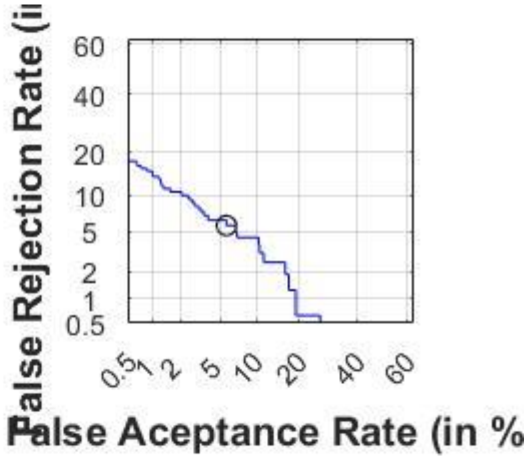
ATT Image 1
Here is shown the original image.

DCT Image 1
This image shows the processed image by Discrete Cosine Trasformation (DCT)

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

Here is shown the result for the standard parameters given (train = 6, test = 4)

Generated DET curve:



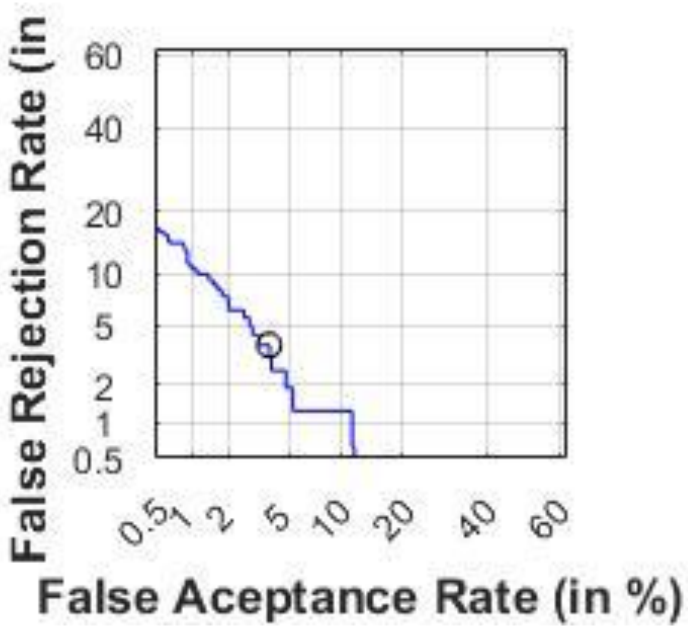
EER = 5.6891

The inner circle represents the exact error. This is the point where both, the false rejection rate and the false acceptance rate have the same value

1.c) 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.

Best Result:

Coeff = 5 – EER = 3.7500



Other coefficient values and their results are shown below, order from best to worst performance, showing the difference with the optimal value shown above:

Coeff 6 = EER = 3.9744

Coeff 7 = EER = 5

Coeff 4 = EER = 5.1282

Coeff 3 = EER = 6.6346

Coeff = 20 -- EER = 8.1250

Coeff 50 -- EER = 21.8750

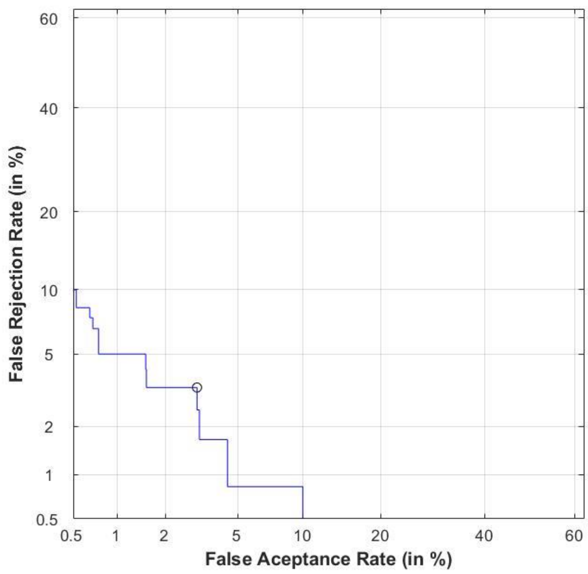
Coeff 1 -- EER = 100

1.d) 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.

Best Parameter Configuration Value:

Train 7 -- EER = 3.3333

Generated DET curve:



Other values:

Train 6 – EER = 3.7500

Train 5 -- EER = 5

Train 4 -- EER = 5.8333

Train 3 -- EER = 6.4286

Train 2 – EER = 9.3750

Train 1 -- EER = 13.0556

This action reduces in a great way the general error previously obtained. Training with more images means the script/program works better in exchange of some computational power. In this precise example the computational power is small, so increasing the training images number leads to better performance, obtaining a better image/subject identification.

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).

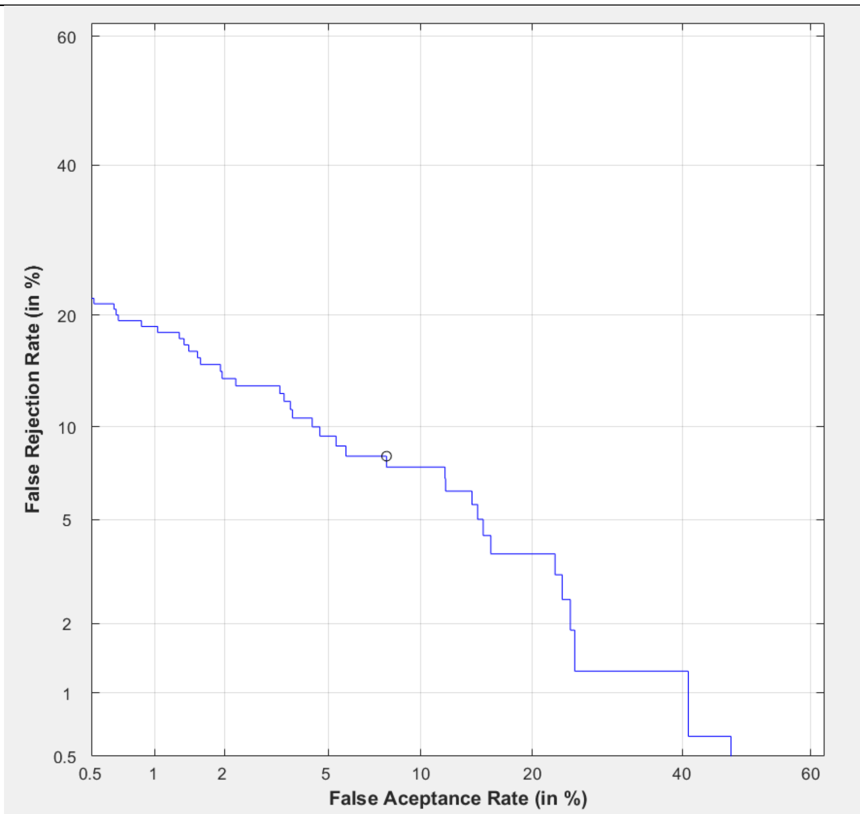
For the **test phase**, you should follow:

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

Generated DET curve:



EER = 8.1250

In comparation with the previous DTC method shows us that PCA is worst using all the PCA Components than DCT with the standard parameter values.

2.b) 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.

For the value of 31 we get the optimal value of the error.

Npca 31 -- EER = 3.9103

Other studied Values are the following ones:

Ncap 32 -- EER = 4.3750

Ncap 30 -- EER = 4.3750

Ncap 15 -- EER = 4.3750

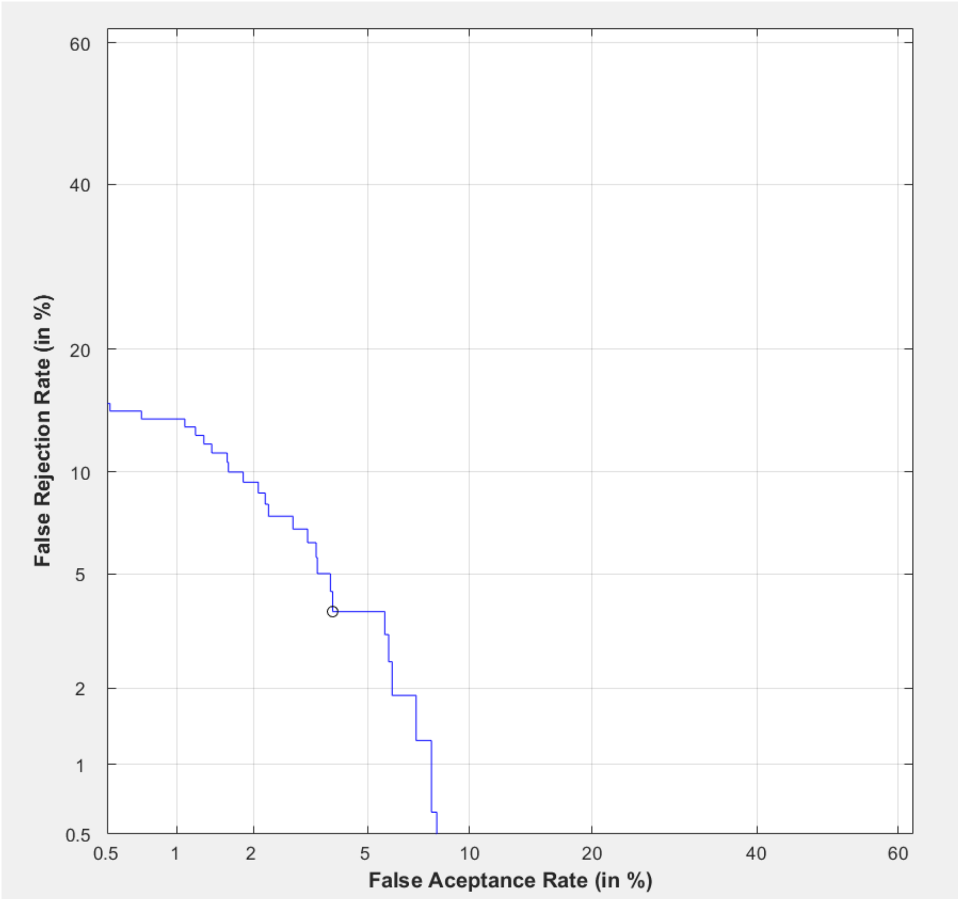
Ncap 100 – ERR = 8.125

Ncap 200 – ERR = 8.125

As in all other algorithms we can get high error by overfiting, this is what happens when ussing all PCA components.

2.c) 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 Task1.

As it is said at exercise 2b, Npca 31 -- EER = 3.9103 is the optimal value and error. The following image is the DET curve generated for this value.



Comparing this result to DTC result. The final observation is that PCA gets a slightly worst result, just about a 0,2%.

PCA (Principal Component Analysis) uses dimensionality reduction and linear algebra concepts to recognize faces (projecting face images on small feature spaces). Projection vector is a set of eigenvector of training samples.
This generate some limitations: PCA projection is optimal for reconstruction of face, but may not be optimal for discrimination (it is only based on simple linear algebra). Unseen users, variabilityin the pose, illumination, etc.

With all the previous exercises done correctly you can obtain a mark up to 7 points out of 10.

Extra work: If you want to obtain a mark up to **10 points out of 10** you should complete the following Task.

Task 3: 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.a) 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).