EN.520.612.01.FA20 Machine Learning for Signal Processing

*Laboratory 3 – PCA-based Face Recognition*

# Dataset

We will use the ORL database, available to download on AT&T’s web site. This database contains photographs showing the faces of 40 people. Each one of them was photographed 10 times. These photos are stored as grayscale images with112 x 92 pixels.

In our example, we construct a catalog called orlfaces, comprised of people named , …, , each one of them containing 10 photographs of the person. The data has already been split into a training and testing split, where for each person, we use the first 9 photographs for training and the last photograph for test.

1. Load the training data

% Your code goes here

1. Change each *(d1 , d2 ) = (112,92)* photograph into a vector

% Your code goes here

3. Using all the training photographs for the *N* people in the training dataset, construct a subspace *H* with dimensionality less than or equal to *N* such that this subspace has the maximum dispersion for the *N* projections. To extract this subspace, use Principal Component Analysis, as described below:

* Center the data
* Compute the correlation matrix
* Use either the SVD or eig functions to perform SVD and get the eigenvectors and eigenvalues for the correlation matrix.
* Normalize the eigenvectors by the corresponding eigenvalues.

% Your code goes here

1. Plot the eigenvalues

% Your code goes here

1. Plot the first 3 eigenfaces and the last eigenface (these will be the correctly reshaped eigenvectors)

% Your code goes here

1. Pick a face and reconstruct it using eigenvectors *k = 10, 20, 30, 40*. Plot all of these reconstructions and compare them. For each value of *k*, plot the original image, reconstructed image, and the difference b/w the original image and reconstruction in each case. Write your observations.

% Your code goes here

1. Load the testing data, and reshape it similar to the training data.

% Your code goes here

1. For each photograph in the testing dataset, you will implement a classifier to predict the identity of the person. To do this, follow these steps -

* Determine the projection of each test photo onto *H* with different dimensionalities

*d = 10, 20, 30, 40*

* Compare the distance of this projection to the projections of all images in the training data.
* For each test photo's projection, find the closest category of projection in the training data.

% Your code goes here

9. Show the closest image in the training dataset for the test example.

% Your code goes here

**Submission instructions:**

Submit your code (with functions, if any) in a zip compressed folder on Blackboard.