CS663, Assignment 4 **Yash Sanghvi, Pradyot Prakash**
Instructor: Prof. Ajit Rajwade 13D070042, 130050008
Question 1: Face Recognition

Problem Statement

In this part, you will implement a mini face recognition system. Download the ORL face database.
It contains 40 sub-folders, one for each of the 40 subjects/persons. For each person, there are ten images in the appropriate folder. The images are of size 92 by 110 each. Each image is in the pgm format. You can view the images in this format, either through MATLAB or through image viewers like IrfanView on Windows, or xv/display/gimp on Unix. Though the face images are in different poses, expressions and facial accessories, they are all roughly aligned (the eyes are in roughly similar locations in all images). For the first part of the assignment, you will work with the images of the first 32 people. For each person, you will include the first six images in the training set and the remaining four images in the testing set (note: there are 10 images per person labeled 1.pgm to 10.pgm). You should create an eigen-space from the training set as described during the lectures without explicitly computing the covariance matrix (note, in this case, you do have N d where d = 92 110 is the size of the image, and N = 35 6 is the number of images in the training set). Record the recognition rate using squared difference between the eigencoefficients while testing on all the images in the test set, for k 1, 2, 3, 5, 10, 15, 20, 30, 50, 75, 100, 150, 170. Plot the rates in your report in the form of a graph.
Repeat the same experiment on the Yale Face database. This database contains 60 images each of 38 individuals (labeled from 1 to 39, with number 14 missing). Each image is in pgm format and has size 192 by 168. The images are taken under different lighting conditions but in the same pose. Take the first 40 images of every person for training and test on the remaining 20 images (by first 30 images, I mean the first 30 images that appear in a directory listing as produced by the dir function of MATLAB). Plot in your report the recognition rates for k 1, 2, 3, 5, 10, 15, 20, 30, 50, 60, 65, 75, 100, 200, 300, 500, 1000 based on squared difference between all the eigencoefficients and between all except the three eigencoefficients corresponding to the eigenvectors with the three largest eigenvalues. [**40 points**]

Implementation Details

ORL Database

The standard method for PCA was used to make the mini-face recognition system. The steps taken were as follows for the **training phase**:

1. Obtain the training images from $N = 32$ different faces, with $N_{training} = 6$ different images for each face. Each image is represented as a d dimensional column vector, where $d = sizeX * sizeY$, where sizeX = width of image (in pixels) and sizeY = height of image (in pixels). Let the training image column vectors be x_i and $i = 1, 2, 3...N * N_{training}$
2. Obtain the d dimensional mean vector $\overline{x}$ and subtract it from each vector: $x_i' = x_i - \overline{x}$
3. Obtain $X = [x_1|x_2|x_3|.....]$ and $L = X^T * X$. Note that it is a $(N * N_{training})$ X $(N * N_{training})$ vector. Find out its eigenvalues λ_i in increasing order and the corresponding eigenvector matrix V.

4. Obtain the eigenvector matrix, $X_{eigenspace}$ of Covariance matrix, C by $X_{eigenspace} = X * V$ and then unit normalising each column vector of the matrix $X_{eigenspace}$.

5. Choose the first k columns of $X_{eigenspace}$ (eigenvectors corresponding to the highest eigenvalues), form a matrix V_k, and obtain the eigenspace coefficient vector, α_i, for image x_i by $\alpha_i = V_k^T x_i$.
   Note: To reduce the time taken for script to run, all the $N * N_{training}$ eigen coefficients for each image were calculated and stored before hand and the top k eigencoefficients were chosen instead of always finding out top k eigencoefficients again whenever k was changed.

To test the image y_i, we used the following method (given the number of eigenvectors to use was k):

1. The test image y_i is taken and the mean obtained in the training phase is subtracted from it.

2. The projection of the vector y_i on the eigenspace V_k by $\beta_i = V_k^T y_i$. We obtain a vector on the k dimensional eigenspace awith coefficients as β_i

3. Find vector x_j such that $j = arg(min_i||\beta_i - \alpha_i||^2)$. If the images y_i and x_j belong to the same face, then the face has been identified correctly.
   Note: As we had 6 training images and 4 test images from the same image and they were obtained in the same sequence i.e. one by one from each face in increasing order of faces's seqauence, Image y_i and x_j belong to the same face if $floor((i-1)/4) = floor((j-1)/6)$

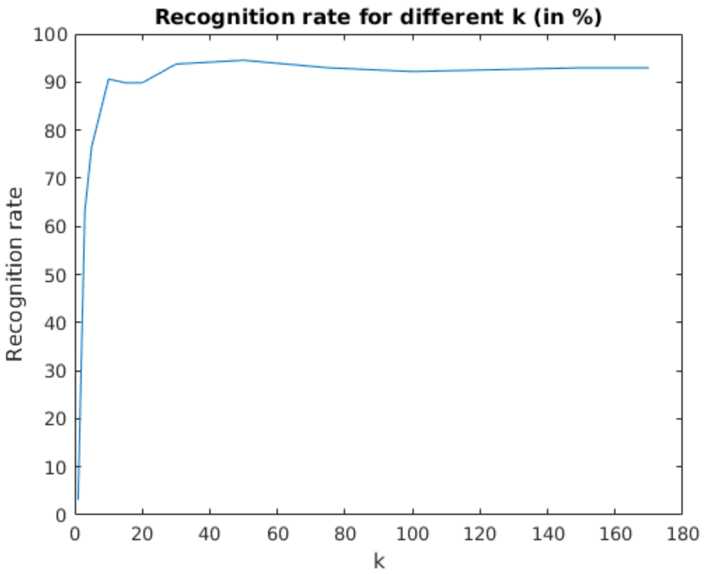
Comments for Yale Database: The face recognition for Yale Database was done in thee same manner except the following differences:

1. Firstly, the top three eigenvectors were dropped when we were considering the top k eigenvectors for PCA. It was discussed in the class that these eigenvectors correspond to the intensity and since out focus is on face recognition, we ignore those

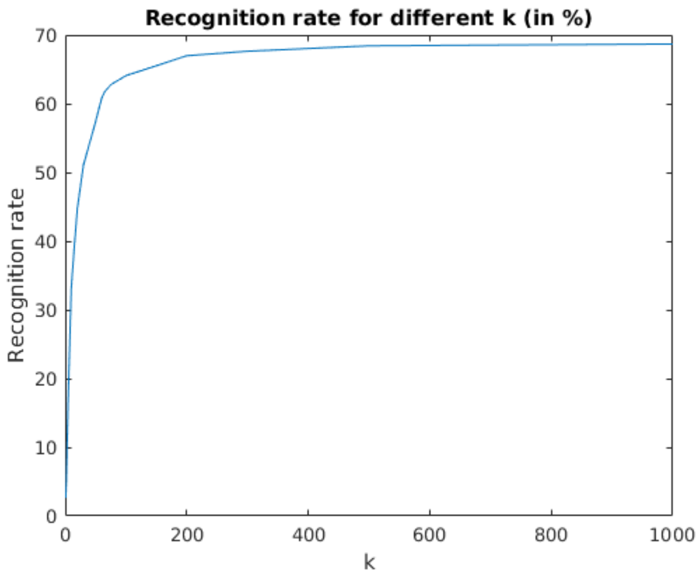
2. The k list i.e. the number of eigenvectors chosen was changed to 1, 2, 3, 5, 10, 15, 20, 30, 50, 60, 65, 75, 100, 200, 300, 500, 1000

3. Total number of faces changed to $N_{total} = 38$ and total number of training faces per face changed to $N_{training} = 40$. As a result to check if the test image belonged to correct face, we used the following relation: $floor(i-1)/20 = floor(j-1)/40$

Result Images



ORL Database



Yale Database