1 The Yale Face Database

In this assignment, we will have a look at some simple techniques for object recognition, in particular, we will try to recognize faces. The face data that we will use is derived from the Yale Face Database - to get more information on the database, have a look at the website

http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

where you can also get many more details on the image acquisition process. In short, the entire database consists of 5760 images of 10 individuals, each under 9 poses and 64 different lighting conditions. The availability of such standardized databases is important for scientific research as they provide a common testing ground for the efficacy of different algorithms.

Figure 1: The Yale Face Database B

For our purposes, we will need only the 640 images that correspond to frontal orientation of the faces. These faces are included in the file yaleBfaces.zip. You will find the faces divided into 5 different subsets. Subset 1 consists of images where the light source direction is almost frontal, so that almost all of the face is brightly illuminated. From subset 2 to 5, the light source is progressively moved toward the horizon, so that the effects of shadows increase and not all pixels are illuminated.
The faces in the subset 1 will be used as training images and subsets 2 to 4 will constitute the test set.

2 Recognition Using Eigenfaces

(a) Write a function `eigenTrain`, which takes as input the $50 \times 50 \times 70$ matrix of all the face images from subset 1. Vectorize each face image and store them in a matrix $A$ of size $2500 \times 70$. Perform PCA on the data represented by the matrix $A$ and retain the top 20 eigenvectors. Note you can use the matlab command `svds.m` to find the first $k$ eigenvectors. \[5\text{ points}\]

(b) Rearrange the top 9 eigenvectors you obtain in part (a) into 2D arrays of size $50 \times 50$ and display in a $3 \times 3$ format, using the Matlab function `subplot.m`. \[2\text{ points}\]

(c) Write a function called `eigenTest`, which takes as input an array of subset indices, an array of subspace dimensions, and the eigenvectors computed in part (a). Project each image of the subsets 1, 2, 3, 4 onto the space spanned by the first $k$ eigenvectors, for $k = 1$ to 20. Use the L2-distance metric to classify each test image based on its nearest neighbor. The output of `eigenTest` must be a $4 \times 20$ matrix $R$, where $R_{ik}$ is the fraction number of images of subset $i$ that were misclassified when the image was projected onto the first $k$ eigenvectors. Plot the error rate of each subset as a function of $k$, the number of eigenvectors. In your plot, show the four curves (one for each subset) on the same axis. \[10\text{ points}\]

(d) Repeat part (c) for subsets 1 to 4 and $k = 1$ to 20, but ignore the first three eigenvectors and use $k$ eigenvectors starting with the fourth eigenvector. How do you explain the difference in recognition performance from part (c)? \[5\text{ points}\]

(f) Can you explain the error rates you obtain for subset 1 in parts (c) and (d)? Explain any trends you observe in the variation of error rates as you move from subsets 2 to 4 and as you increase the number of eigenvectors. \[3\text{ points}\]

3 Recognition Using Fisherfaces

(a) Let the number of training images be $N$ (in this case, $N = 70$) and the desired number of bases be $c - 1$. Use PCA to project each training image into a space of dimension $N - c$. Now apply Fisher’s Linear Discriminant to obtain a $c - 1$ dimensional feature vector for the face images. Do this for $c = 10$. Your Matlab code must be contained in a function called `fisherTrain` that takes as input the $50 \times 50 \times 70$ matrix of all the face images from subset 1 and the number of bases, $c$, and returns the Fisher bases as output. \[8\text{ points}\]

(b) Rearrange the 9 Fisher bases you obtain in part (a) into 2D arrays of size $50 \times 50$ and display in a $3 \times 3$ format, using the Matlab function `subplot.m`. \[2\text{ points}\]
(c) Similar to question 2(c), perform recognition on the test set with Fisherfaces. Write a function `fisherTest` that takes as input an array of subset indices $S$, an array of subspace dimensions, and the Fisher bases computed in part (a). Project each image of the subsets 1, 2, 3, 4 onto the space spanned by the first $k$ Fisher bases, for $k = 1$ to 9. This function will return a $4 \times 9$ matrix containing the error rate for each subset and each $k$. Plot the four error curves for each subset as a function of $k$ (on the same axis), and explain any trends you observe in the variation of error rates as you move from subsets 1 to 4.  

[10 points]

(d) Explain, qualitatively, the reason for the difference in recognition rates achieved by the Eigenface and Fisherface methods for a comparable number of basis images.  

[5 points]