CS663, Assignment 4 **Yash Sanghvi, Pradyot Prakash**
Instructor: Prof. Avijit Rawade 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]**

Code

(i) Face Recognition(I) for ORL Database

```
tic;
N = 32; % Number of faces
N_training = 6; % Training Images per face

x = imread('database/s1/1.pgm');
d = prod(size(x)); % Dimension of one image

X = zeros(d,N*N_training);

disp('Collecting Training Data');
for i=1:N
    for j=1:N_training
```

```
13          x = imread(strcat('database/s',int2str(i),'/',int2str(j),'.
                pgm'));
14          %y = myLinearContrastStretching(x);
15          X(:,(i-1)*N_training + j) = x(:);
16      end
17  end
18
19  % Mean Vector
20  x_mean = sum(X')'/d;
21
22  for i=1:N
23      for j=1:N_training
24          X(:,(i-1)*N_training + j) = X(:,(i-1)*N_training + j)-x_mean
                ;
25      end
26  end
27
28  L = X'*X;
29  disp('Finding eigenvectors of X''X');
30  [V, D] = eig(L);
31  [V, D] = sortEigenVectors(V,D);
32  % Find the eigenvectors of C from V
33  V_C = X*V;
34
35  %Unit Normalise the eigenvectors
36  V_C = V_C ./ (repmat(sum(V_C.^2),d,1).^0.5);
37  X_eigenface = V_C'*X;
38
39
40  disp('Collecting Test data');
41      X_test = zeros(d,N*(10-N_training));
42      for i=1:N
43          for j=1:10-N_training
44              x = imread(strcat('database/s',int2str(i),'/',int2str(j+
                    N_training),'.pgm'));
45              %y = myLinearContrastStretching(x);
46              X_test(:,(i-1)*(10-N_training) + j) = x(:);
47          end
48      end
49
50  X_test = X_test - repmat(x_mean,1,size(X_test,2));
51  X_Projection = V_C'*X_test;
52
53
54  K = [1, 2, 3, 5, 10, 15, 20, 30, 50, 75, 100, 150, 170];
55  K_recognition_rate = zeros(1,size(K,2));
56
57  for a=1:size(K,2)
58      k = K(a);
```

```
    % Pick the top k eigenvectors
    X_eigenface_k = X_eigenface((N*N_training)-k+1:(N*N_training),:);
    X_Projection_k = X_Projection((N*N_training)-k+1:(N*N_training),:);


    % Get eigenface of each image
    count = 0;
    for i=1:size(X_test,2)
        X_Diff = repmat(X_Projection_k(:,i),1,N*N_training) - X_eigenface_k;
        [Min,Min_Image_Index] =  min(sum(X_Diff.^2));

        if floor((i-1)/(10-N_training)) == floor((Min_Image_Index-1)/(N_training))
            count = count+1;
        end

    end
    K_recognition_rate(a) = double(count)*100/size(X_test,2);

end
figure(1);

plot(K,K_recognition_rate);
title('Recognition Rate for different k (in %)');
xlabel('k');
ylabel('Recognition Rate');

toc;
```

(ii) Face Recogntion (II) for Yale Database

```
tic;
N = 38; % Number of faces
N_training = 40; % Training Images per face
% h = waitbar(0,'Collecting Training Data');

x = imread('database/YaleB01/yaleB01_P00A+000E+00.pgm');
d = prod(size(x(:)));

X = zeros(d,N*N_training);
listing = dir('database');
count = 0;
for i=1:N
    listing_image = dir(strcat('database/',listing(i+2).name));
    for j=1:N_training
        n  = listing_image(j+2).name;
        x = imread(strcat('database/',listing(i+2).name,'/',n));
```

```
17          X(:,(i-1)*N_training + j) = x(:);
18          count = count + 1;
19  %           waitbar(double(count)/double(N*N_training),h);
20      end
21      disp(strcat('Collected images from: ',int2str(i),' th Face'));
22  end
23  % close(h);
24
25  % Mean Vector
26  x_mean = sum(X')'/d;
27
28  for i=1:N
29      for j=1:N_training
30          X(:,(i-1)*N_training + j) = X(:,(i-1)*N_training + j)-x_mean
                ;
31      end
32  end
33
34  L = X'*X;
35  disp('Finding eigenvectors of X''X');
36  [V, D] = eig(L);
37
38  % Find the eigenvectors of C from V
39  disp('Find the eigenvectors of C from V');
40  V_C = X*V;
41
42  %Unit Normalise the eigenvectors
43  disp('Unit Normalization');
44  V_C = V_C ./ (repmat(sum(V_C.^2),d,1).^0.5);
45  X_eigenface = V_C'*X;
46
47
48  X_test = zeros(d,N*(60-N_training));
49
50  % h = waitbar(0,'Collecting Test data');
51  count = 0;
52  listing = dir('database');
53  for i=1:N
54      listing_image = dir(strcat('database/',listing(i+2).name));
55      for j=1:(60-N_training)
56          n  = listing_image(j+2+N_training).name;
57          x = imread(strcat('database/',listing(i+2).name,'/',n));
58          X_test(:,(i-1)*(60-N_training) + j) = x(:);
59          count = count + 1;
60      end
61      disp(strcat('Collected images from: ',int2str(i),' th Face'));
62  end
63  X_test = X_test - repmat(x_mean,1,size(X_test,2));
64  X_Projection = V_C'*X_test;
```

```
65
66 K = [1, 2, 3, 5, 10, 15, 20, 30, 50, 60, 65, 75, 100, 200, 300, 500,
       1000];
67 K_recognition_rate = zeros(1,size(K,2));
68
69 for a=1:size(K,2)
70     k =K(a);
71 %   Pick the top k eigenvectors' Components
72     X_eigenface_k = X_eigenface((N*N_training)-k-2:(N*N_training)
           -3,:);
73     X_Projection_k = X_Projection((N*N_training)-k-2:(N*N_training)
           -3,:);
74
75     count = 0;
76     for i=1:size(X_test,2)
77         X_Diff = repmat(X_Projection_k(:,i),1,N*N_training) -
               X_eigenface_k;
78         [Min,Min_Image_Index] =  min(sum(X_Diff.^2));
79         if floor((i-1)/20) == floor((Min_Image_Index-1)/40)
80             count = count+1;
81         end
82     end
83     count,k
84     K_recognition_rate(a) = double(count)*100/size(X_test,2);
85 end
86 figure(1);
87
88 plot(K,K_recognition_rate);
89 title('Recognition Rate for different k (in %)');
90 xlabel('k');
91 ylabel('Recognition Rate');
92
93 toc;
```

Implementation Details

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 last 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.

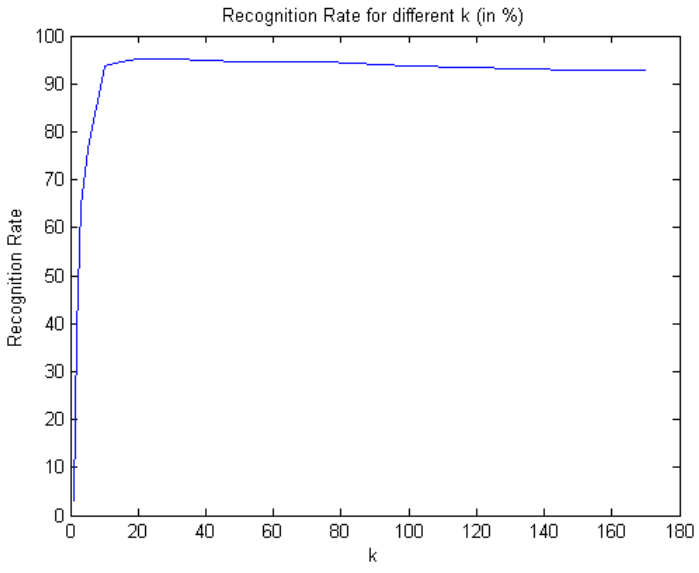
To test the image y_i, we used the following method (given the number of eigenvectors to use was k):

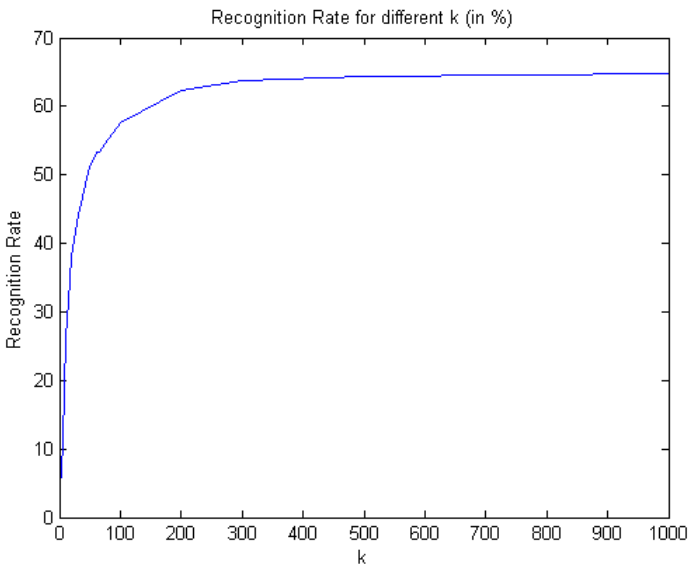
1. The test image y_i is taken and the mean obtained in the training phase is subtracted from it.

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 $floor((i-1)/4) = floor((j-1)/6)$

Result Images




Recognition Rate for different k (in %)
Recognition Rate
k
0
10
20
30
40
50
60
70
0
100
200
300
400
500
600
700
800
900
1000