ECS797 Machine Learning for Visual Analysis

Lab 2: Face Recognition using Eigenfaces

Name : Tanmaiyii Rao
ID : 140361229

1. Introduction

This coursework is on learning models of human faces for recognition. In particular, it is based on Turk and Pentland's paper to build programs that use eigenfaces for face recognition.

2. Getting started

The files have been downloaded and the lab2.m file has been modified to compute the eigenfaces.

3. Completing lab2.m file

1. The training and test images have been read by running the two separate .m files.

2. The mean image and the covariance matrix of the training image set has been constructed by running the code in section 2 of lab2.m. Once you run the code, you find out that the covariance matrix is a 644x644 matrix.

3. The Eigenvalues are a 200x1 matrix and the mean image is a 1x644 in dimensions.

4. After computing the covariance mean and eigenvalues, the mean image has been visualised.

5. The first 20 eigenfaces have been displayed by implementing the following code below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display of the 20 first eigenfaces : Write your code here

% Normalise mean to add back to the Eigenface after normalisation
Means = (Means/255.);

EigenFace = zeros(1, 644);
num_eigenfaces = 20;
%Computing Eigenfaces - using V from SVD (Singular Value Decomposition)
eigs = S*V';
%normalization eigenfaces for better visualisation
eigs = 255 *(eigs - min(eigs(:))) ./ (max(eigs(:)) - min(eigs(:)));
% denormalise eigenvector by adding the mean back.
eigenf = eigs + Means;

for k = 1:num_eigenfaces
   EigenFace = eigs(k,:);
   EigenFace = reshape(EigenFace, [28,23]);
   subplot (5,4,k);
   imshow(uint8(EigenFace));
end
```

The visualisation of the eigenfaces

As can be seen, after using singular value decomposition (SVD) to get eigenvectors of the images. SVD is performed directly on the CenteredVectors and not on the covariance matrix. U is a diagonal matrix of singular values and V are the righ singular matrix. U and V are the eigenvectors of CenteredVectors CenteredVectors^T and CenteredVectors^T CenteredVectors respectively. Thus, we need the matrix V and each column in V is an eigenvector.

6. The training and testing images have been projected onto the first 20 eigenfaces. The function has been provided in the code. The LocationsTest and LocationsTrain are 70x200 and 200x200 matrices respectively. The threshold has been set to 20 for the projection of the images onto the first 20 eigenfaces.

7. The distances from the project training images to the projected test images have been computed by running the code given in section 7 of lab2.m. The distances matrix is a 70x200 matrix that contains the distance to the train image for every test image.

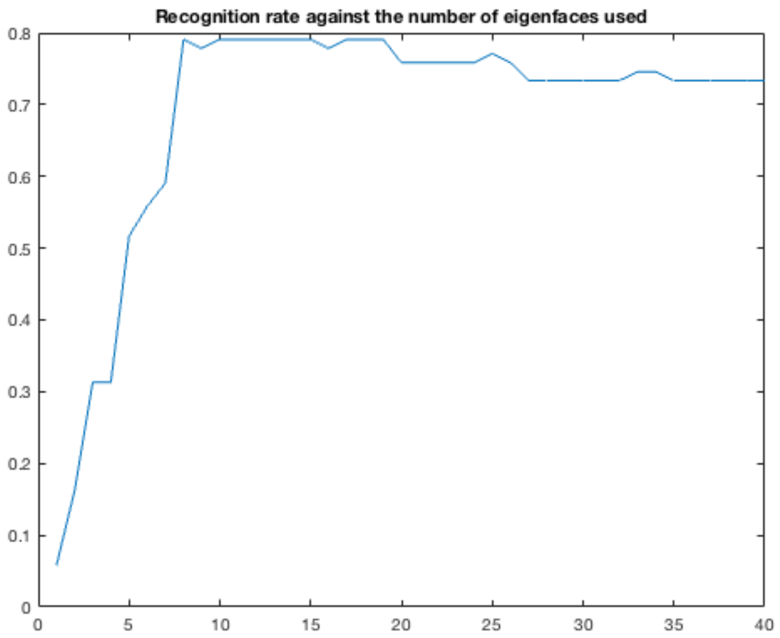
8. The top 6 best matched training images for each test image has been displayed below. The code was already given in lab2.m.

9. The recognition rate has been computes using the 20 eigenfaces. The code is given below. The recognition rate is calculated and is found to be 82.8571%

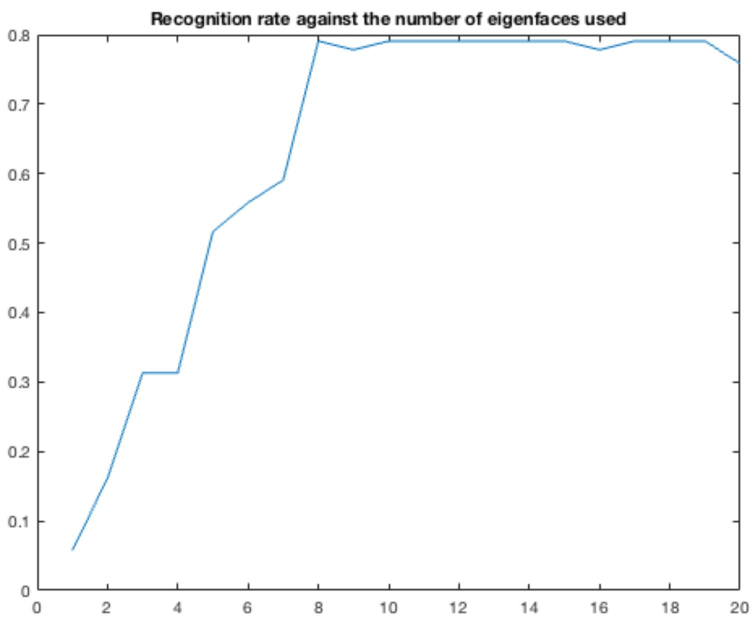
```
%% 9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%recognition rate compared to the number of test images: Write your code here to compute the recognition rate
recognition_rate = zeros(1, length(Imagestest(:,1)));

for i = 1: length(Imagestest(:,1))
    %the recognition rate has been computed by identifying the indexes of the 20 eigenfaces;
    % if the indices of train does not match with Identity in test then rate is 0.
    if ceil(Indices(i,1)/5) == Identity(i)
        rec_rate(i) = 1;
    else
        rec_rate(i) = 0;
    end
end
% The total recognition rate for the whole test set that has 70 images
recognition_rate = sum(rec_rate)/70 *100



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

10. The effect of using different number of eigenfaces for recognition has been investigated and plotted. The code has been provided in lab2.m. The threshold t is the number of eigenfaces.

Number of eigenfaces t = 40



Number of eigenfaces t = 20



Number of eigenfaces t = 10

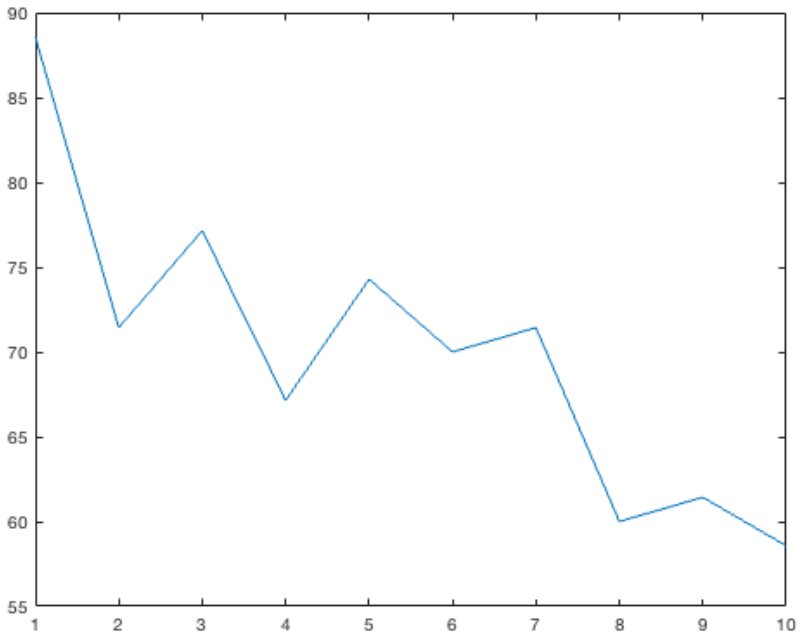
11. The effect of K in K-Nearest-Neighbour (KNN) classifier has been evaluated here. The average recognition rate has been plotted against K. The following code has been implemented.

```
%% 11
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%To investigate the effect of using KNN % Plotting the average recognition rate against K.

train_lab = [];
for i = 1:40
    train_lab = horzcat(train_lab, repmat(i,1,5));
end
%%
```

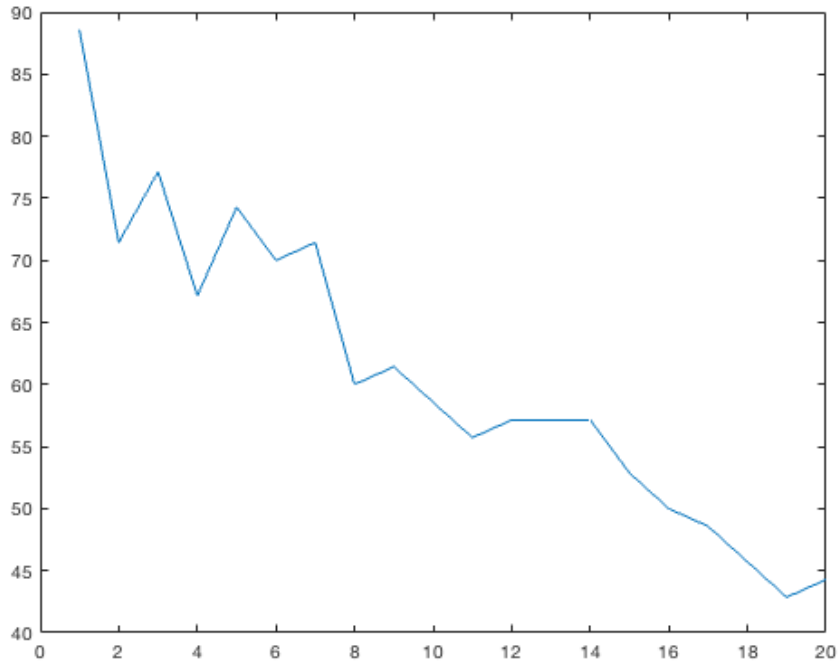
```
%%
% can set the value of K (nearest neighbours)
K=1:10;
rec_rate = zeros(1,10);

% the identity of the test image was found using KNN
% The obtained identity was evaluated.
% The recognition rate is calculated
for k = 1:10
    knn = fitcknn(Imagestrain, train_lab, 'NumNeighbors', k);
    knn_prediction = predict(knn, Imagestest);
    knn_rec_rate = zeros(1, length(Imagestest(:,1)));

    for i = 1:length(Imagestest(:,1))
        if ceil(Indices(i,1)/5) == knn_prediction(i)
            knn_rec_rate(i) = 1;
        else
            knn_rec_rate(i) = 0;
        end
    end

    rec_rate(k) = (sum(knn_rec_rate)/70)*100;
end
figure
plot(K, rec_rate);
Xlabel('K'); ylabel('Recognition rate')
```

The graphs have been plotted below: K is on the X axis and recognition rate on the Y axis.

K = 10



K = 20



Only lab2.m has been submitted with the report as these were
the files that have been modified. Otherwise it takes too much space and I am unable to upload the entire source files.