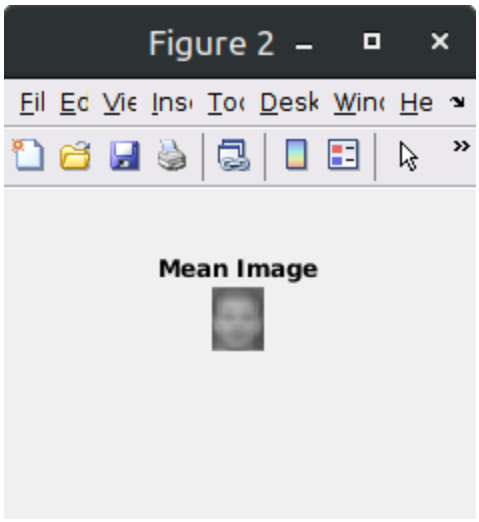
190573735 - Neeraj Vashistha

ECS797 Machine Learning for Visual Data Analysis
Lab 2: Face Recognition Using Eigenfaces

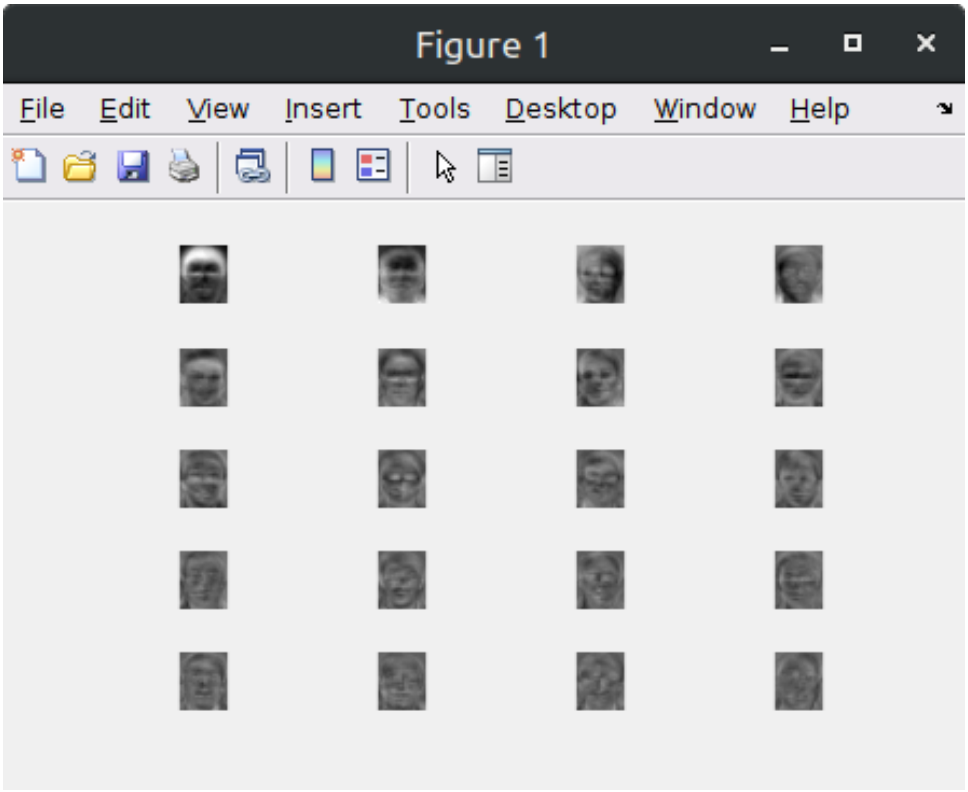
Section 3:

1. The dataset has been read with both the given code files.
2. A covariance matrix(644x644) has been constructed using the provided code in lab2.m.
3. The next section of code computes and stores Eigenvalues(200x1) and the Mean Image(1x644).
4. The obtained mean image:

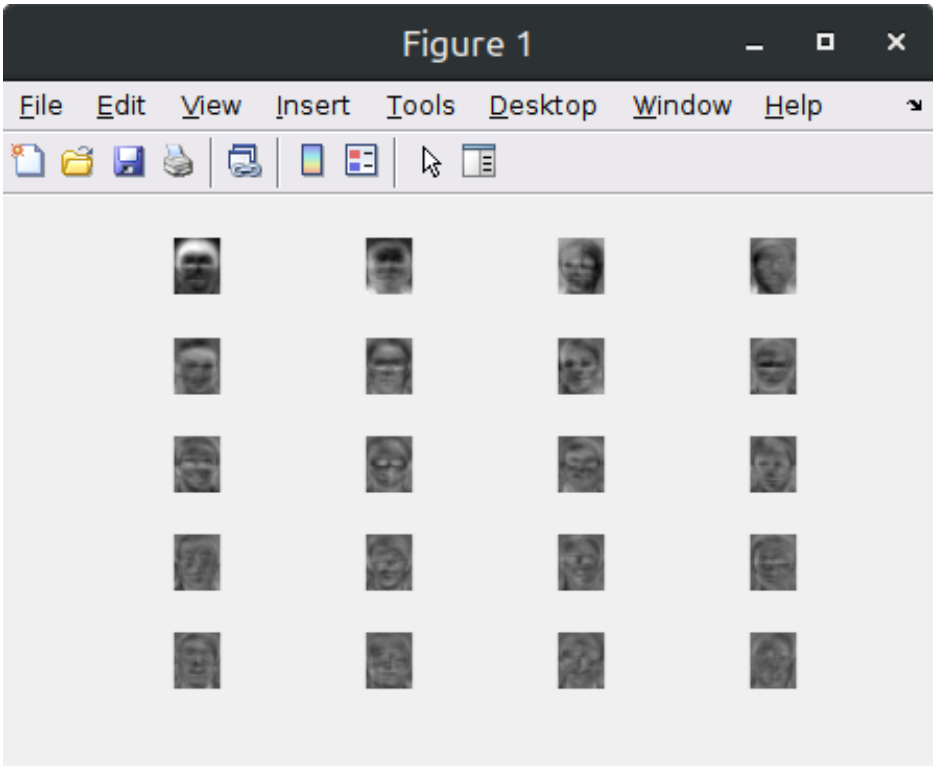


5. The first 20 Eigenfaces:

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

EigenFace = zeros(1, 644); % create a flat col matrix with 0s
EFMatrix = S*V'; % using SVD V compute EF matrix


%normalization eigenfaces for better visualisation
EFMatrix = 255 *(EFMatrix - min(EFMatrix(:))) ./ (max(EFMatrix(:)) - min(EFMatrix(:)));


for k = 1:20 % iterate over and plot plot
   EigenFace = EFMatrix(k,:);
   EigenFace = reshape(EigenFace, [28,23]);
   subplot (4,5,k);
   imshow(uint8(EigenFace));
end
```



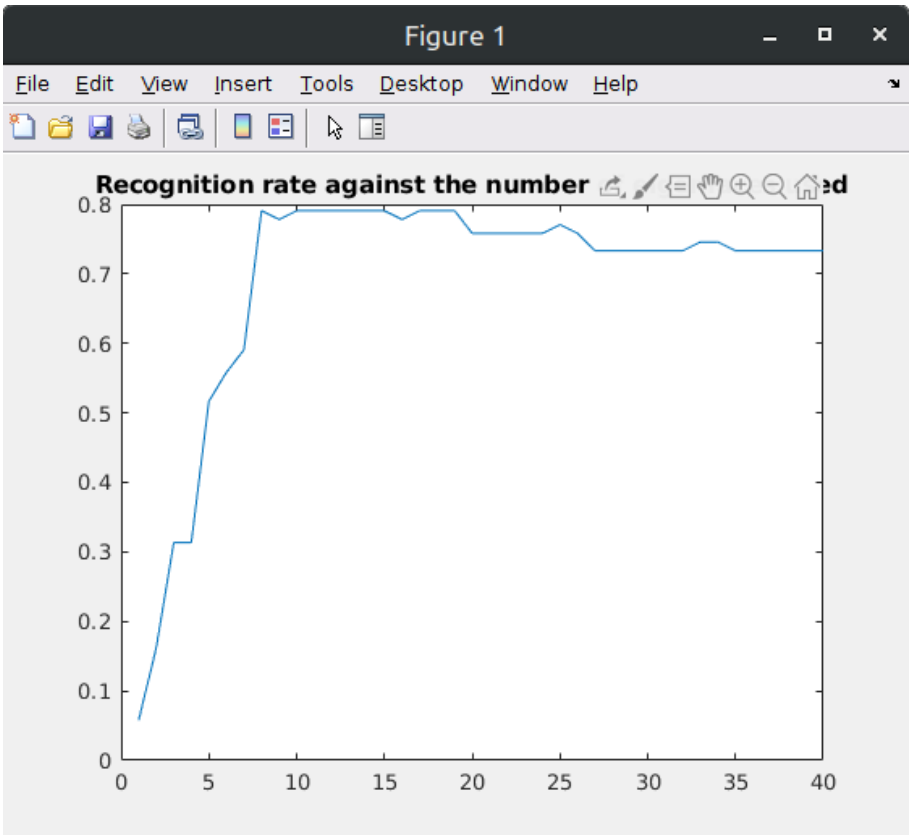
6. Both training and test images have been projected onto 20 Eigenfaces with the help of the given code.
7. The next section has computed the distance from projected test images to projected train images.
8. Displaying the top 6 best-matched images:



9. Computing recognition rate using 20 Eigenfaces is 82.8571

```
% RR is for 20 indices of eignefaces.
% if traon indices are not equal to test Identity then RR = 0

RR = zeros(1,length(Imagestest(:,1))); % RR is recogonition rate is of length of imagetest
for i = 1: length(Imagestest(:,1))
    if ceil(Indices(i,1)/5) == Identity(i)
        RR(i) = 1;
    else
        RR(i) = 0;
    end
end
% The total recognition rate for the whole test set that has 70 images
NETRR = sum(RR)/70 *100;
```

10. Plotting number of Eigenfaces against the average recognition rate:

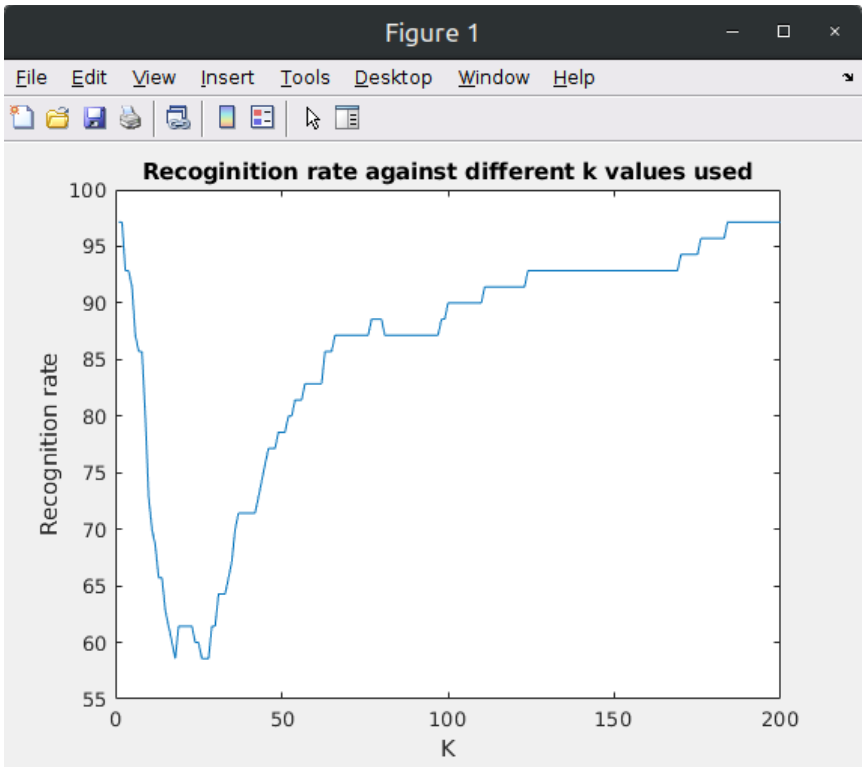


From the above graph we can understand that the number of k required is between 10 and 20. The recognition rate rises till k = 10 and stagnate around k = 10-to-20 and then it drops. This means that we only require at most 20 eigenfaces to completely represent the entire training set.

190573735 - Neeraj Vashistha

11. The plot for using different values of neighbours(k) against the corresponding average recognition rate for k-NN:

```
NETRR = zeros(1,200);
K=1:200; % value of k in K nearest neighbour


for k = 1:200
    KNNModel = fitcknn(Imagestrain, TrainingLabels, 'NumNeighbors', k,'BreakTies', 'nearest'); %fit knn model to training data
    KNNPredict = predict(KNNModel, Imagestest); % make pediction on test data
    KNN_RR = zeros(1, length(Imagestest(:,1))); % initalise recoginition rate

    for i = 1:length(Imagestest(:,1))
        %compare the predictions with identity of test image and predicted
        if ceil(Indices(i,1)/5) == KNNPredict(i)
            KNN_RR(i) = 1;
        else
            KNN_RR(i) = 0;
        end
    end

    NETRR(k) = ((sum(KNN_RR)/70)*100);
end
figure
plot(K, NETRR);
xlabel('K'); ylabel('Recognition rate')
title('Recoginition rate against different k values used');
```



From the above graph we can understand the value of k is around 20-25 when the model is neither underfitting nor overfitting on test labels and a stable recognition rate of 62% on 40 test images is obtained.