School of Electronic Engineering and Computer Science

ECS797 Machine Learning for Visual Data Analysis

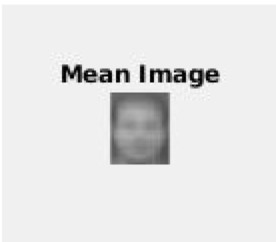
Lab 2: Face Recognition Using Eigenfaces

Name: Yogen Parekh

Student ID: 190306340

3. Complete the lab2.m file

1. The training and test images have been read from the dataset using the code given in two separate .m files.
2. The mean image and the covariance matrix has been constructed using the code given in the file. The resulting covariance matrix has 644X644 dimensions.
3. Using the code given in lab2.m, the Eigenfaces are computed and stored. The Eigenvalues have 200X1 dimensions and Mean image has 1X644 dimensions.
4. The mean image is displayed here:



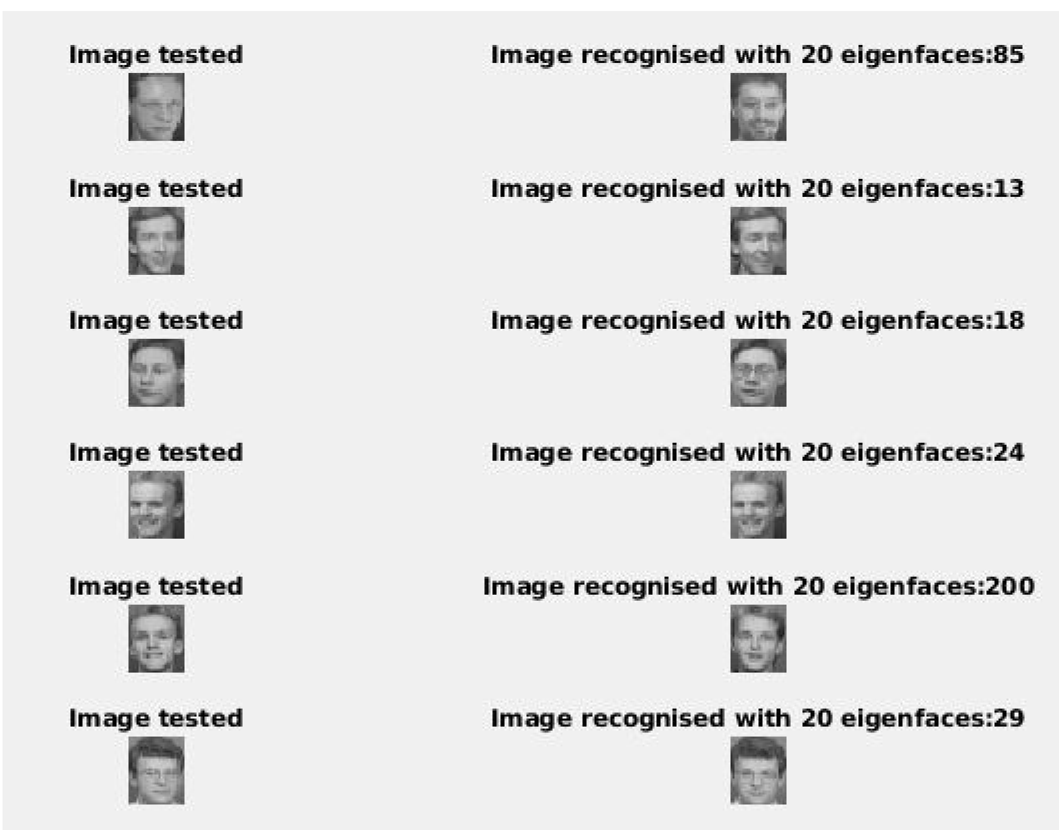
5. 20 Eigenfaces have been displayed here which were generated using the code shown below.

```
EigenFace = zeros(1, 644);
num = 20; % Number of Eigenfaces
% Computing Eigenfaces - using V from SVD
eigf = S*V';
%Normalizing eigenfaces for better visualisation
eigf = 255 *(eigf - min(eigf(:))) ./ (max(eigf(:)) - min(eigf(:)));

%Displaying Eigenfaces
for k = 1:num % iterate over and plot eigenfaces
   EigenFace = eigf(k,:);
   EigenFace = reshape(EigenFace, [28,23]);
   subplot (5,4,k);
   imshow(uint8(EigenFace));
end
```

6. The training and test images have been projected as required, onto 20 Eigenfaces using the code provided in the lab2.m file. The threshold has been set to 20.

7. Using the predefined code, distance between projected test images and the corresponding train images has been computed.

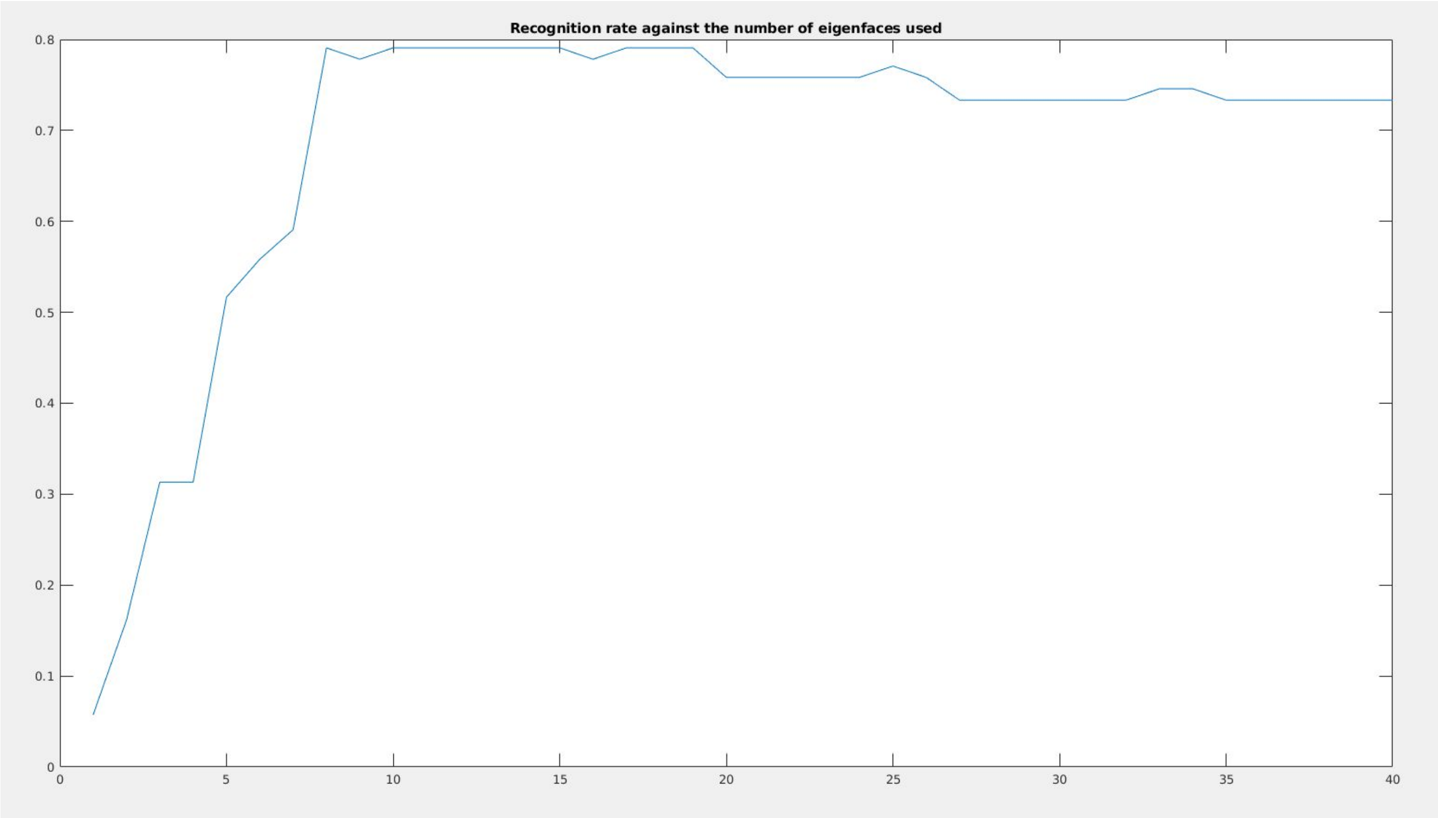
8. The top 6 best matched training images for the test images are:



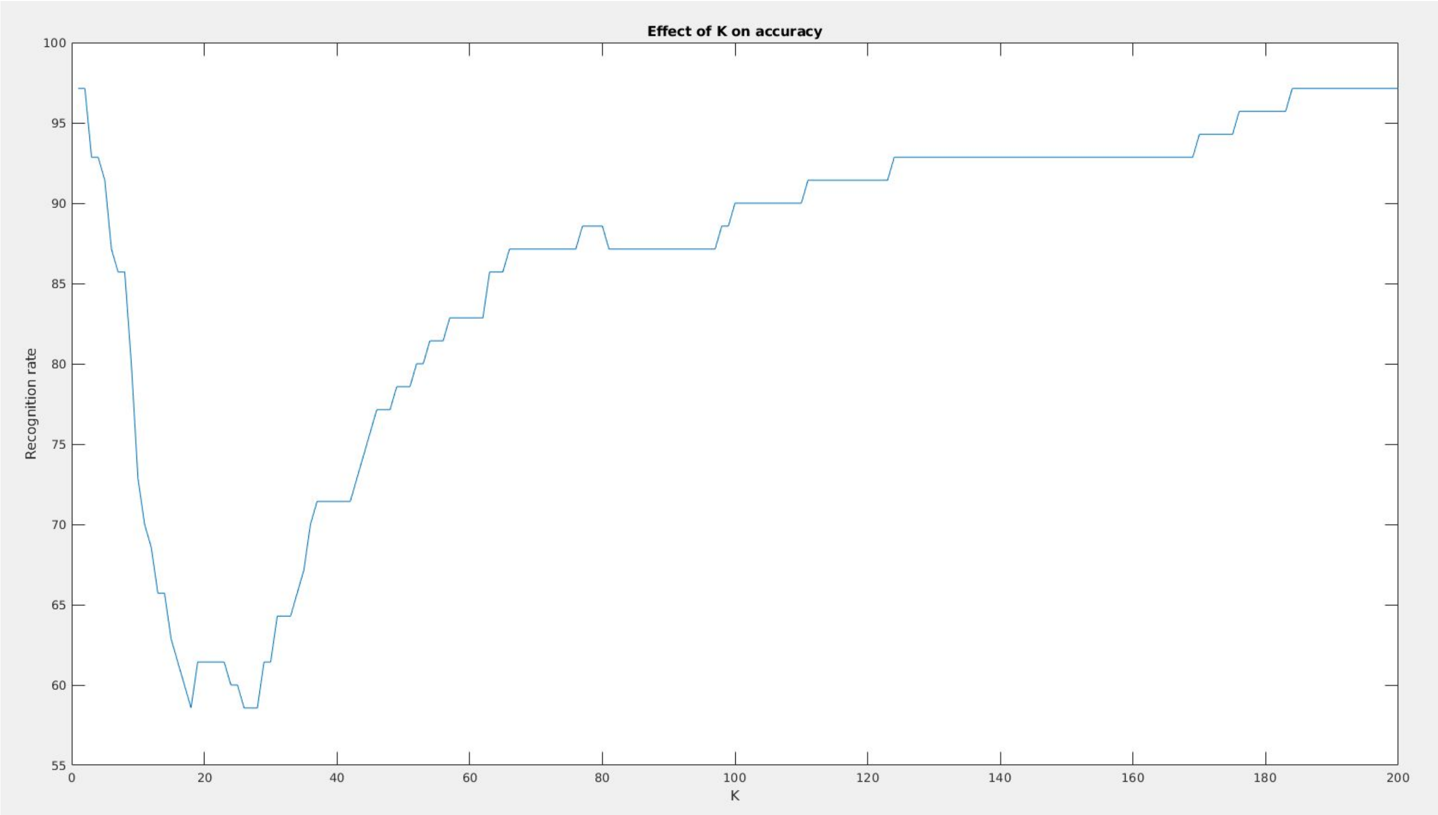
9. The recognition rate computed using 20 Eigenfaces and code displayed below is **82.8571**

```
%Initialising
len = length(Imagestest(:, 1));
rate = ones(1, 20);

for i = 1: len
    %Verifying the classification
    if ceil(Indices(i, 1)/5) == Identity(i)
        rate(i) = 1;
    else
        rate(i) = 0;
    end
end

%Cmputing overall recognition rate
RecoginitionRate = sum(rate) / 70 * 100;
```

10. The effect of using different numbers of eigenfaces is displayed by taking the number of eigenfaces = 0 to 40. It is clearly observable from the graph that the accuracy increases to upto 10 eigenfaces, decreasing underfitting and then attains a constant rate till 20 eigenfaces and starts a slow decrease in the accuracy. So, 20 can be considered as the optimal number of eigenfaces.



11. The effect of using different numbers of K while using k-NN algorithm is displayed here with the plots. This is a typical behaviour for a kNN model, which shows the different stages of underfitting, overfitting and a good learned model. This graph can be used to select an optimal number for k.




```
trainLabels = []; % get true training labels
for i = 1:40
    trainLabels = horzcat(trainLabels, repmat(i,1,5));
end

% can set the value of K (nearest neighbours)
K=1:200;
RecoginitionRate = zeros(1,200);


for k = 1:200
    % fit knn, refer to doc of fitcknn
    knn = fitcknn(Imagestrain,trainLabels,'NumNeighbors',k,'BreakTies','nearest');
    knn_prediction = predict(knn, Imagestest); % do predictions on
    KNN_class_result = zeros(1, length(Imagestest(:,1))); % empty zero initalisation
%     if knn class prediction is same as that of label then class is 1
    for i = 1:length(Imagestest(:,1))
        % compare if prediction == to indices/5(indices are 200 made as 1111122222 set of 5)
        if ceil(Indices(i,1)/5) == knn_prediction(i)
            KNN_class_result(i) = 1;
        else
            KNN_class_result(i) = 0;
        end
    end

    RecoginitionRate(k) = (sum(KNN_class_result)/70)*100;
end
figure
plot(K, RecoginitionRate);
title('Effect of K on accuracy');
xlabel('K'); ylabel('Recognition rate');
```