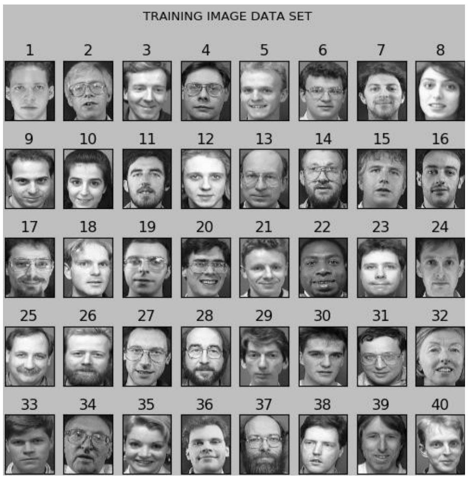
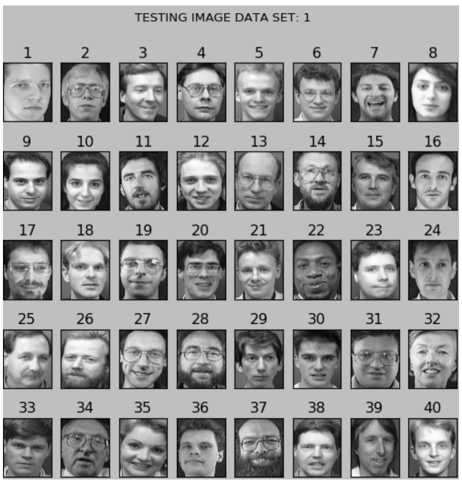
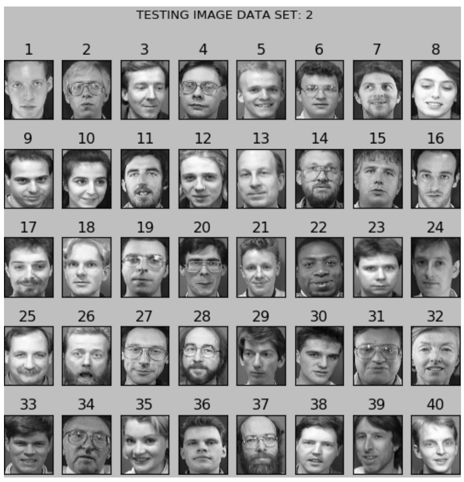
Student Number:	18998712
Module	: RW364 Computer Vision
Task	: Assignmnet05

<u>Question 1:</u>
Given The Following Training Images:



And Also given the following two test Sets Images set a.) and set b.) respectively:





We are tasked to implement a basic SVD approach to face recognition. Using the given Training and Test images.

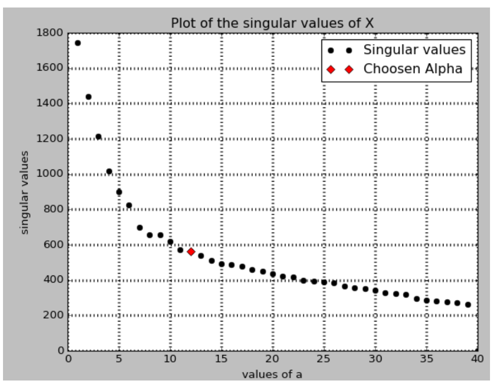
1a.) For this problem I was required to use the training image set of problem1 images to obtain **<u>a</u>** and **Uα,** Plot the singular values of **X** in order to pick a value for **α. Display the average vector a and** the first few **eigenfaces** (columns of **Uα**), as images. By doing the following:

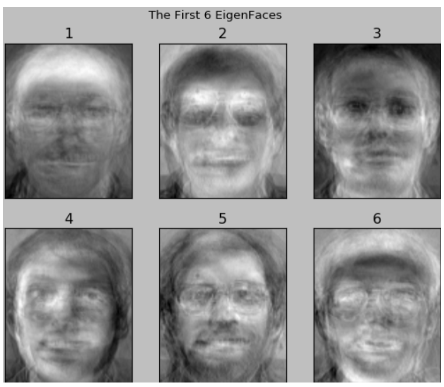
- reshape the vector image to be displayed as a **p × q** matrix and
- to scale the values of an **eigenface** to **[0, 255].**

To Achieve This I had to:
Reshape my training image dataset from **p**×**q** image to one long vector of length **m** = **pq**. To make the images occupy an m-dimensional space.
I stacked individual training images in to a large Matrix where each column represented a unique test image. And computed the average vector(eigenface) **<u>a</u>**.

and also computed the **X** matrix for the testing images, After this i use the computed X to obtain both the **Uα** matrices and the **S** vector.

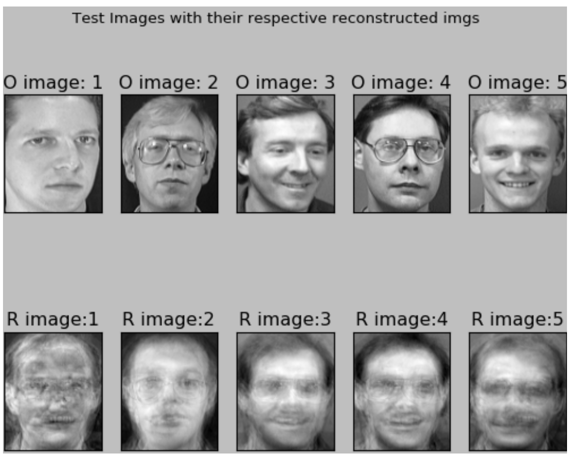
Bellow is a set of three images **a.)** The **average eigenface**, **b.)** Plot of the Singular Values of **X** and **c.)** Image representing the first **6 eigenfaces (**the first **6 columns** of **Uα)**

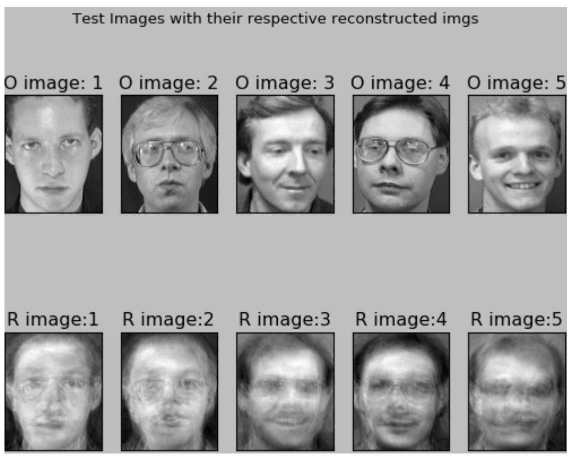




1.b) In this question we are required to make use of the **a** and **Uα** determined in the previous question to encode all the test images to **eigenface representations** called **y** and to Reconstruct the test images from their encodings.

The following approach was taken to achieve the results of the following images. Where image **a.)** depicts the first set of the Test images and image **b.)** depicts the second set of the test images. Both with their respective face reconstructions from their test image encodings as the second row.

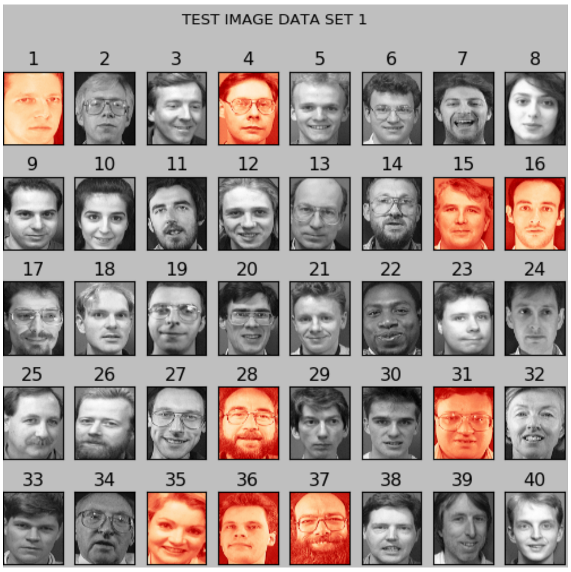


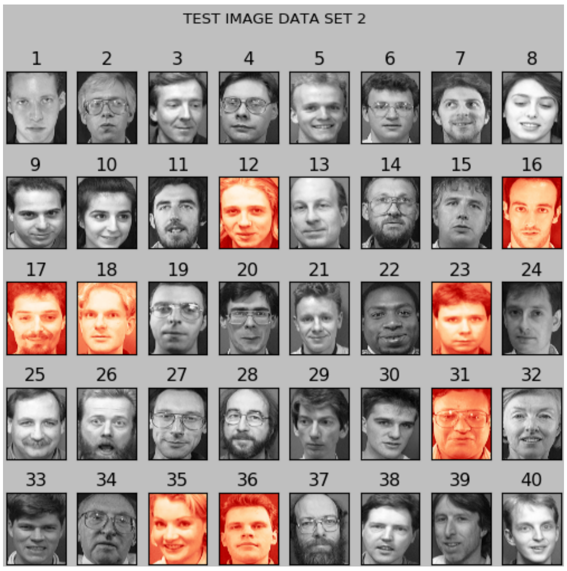


1c.) In this question we are expected to calculate the **eigenface** representation for each image in the test set, find the nearest neighbour in the **training set,** and classify accordingly.

In the following images I show the examples of test image and images chosen by the system as the closest match, the overall accuracy of your system (percentage of test images that where classified correctly):

- For the image at the left hand side**(Test Set 1)** the accuracy was**: 78%**
- For the image at the left hand side**(Test Set 2)** the accuracy was**: 80%**

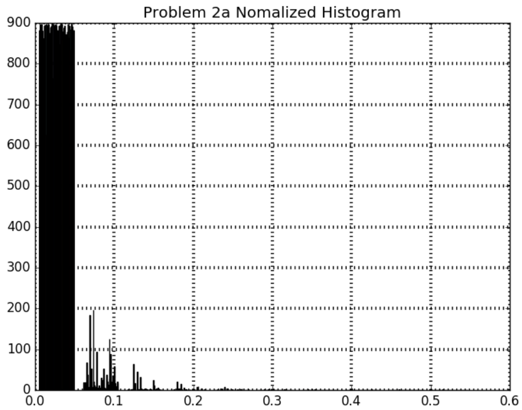


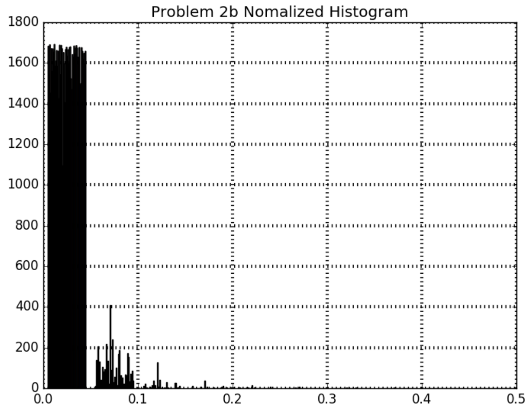


<u>Question 2:</u>

2a.) To build a vocabulary of visual words from the training data, I firstly stacked all the descriptor vectors across the training images into a matrix **Xtrain**. I then made use of the the **KMeans** algorithm by **sklearn** to build the vocabulary and chose the number of clusters to be **50** and number of maximum iterations to be **2**(for memory reasons). By doing this my aim was to set the number of clusters, as well as the number of centroids to generate. This method of clustering groups together a set of objects the same cluster are more similar to each other than objects in other clusters. I then constructed a vector of **kmeans** labels and used this vector to construct a normalized histogram. The histogram for the training set has dimensions The same process was followed for test images resulting in a matrix **Xtest** that has a histogram similar to the one for training dataset.

The following Histograms where achieved from the above stated procedure:





2b.) To train a linear, one-vs-one support vector machine for classification I made use of the SVC function by sklearn and set the kernel to linear, made C(the penalty parameter of the error term) 32 and also set the gamma parameter to auto. I then used fit to fit the SVC model according to the given training data and calculated the predictions by using predict which performs classification on samples in X.

To obtain a good overall test accuracy I did the following:

- adjusted the value of C and noticed that for lower values of **C** i.e **10** and fewer clusters i.e **20**, the accuracy was low i.e. approximately **10% – 20%.**
- compared to when I chose high values of **C** i.e. **32** and more cluster i.e. **50** the accuracy then increased to **63.5%**. Making it better compared to smaller penalty parameter of the error term and a smaller cluster.

The following was the confusion matrix obtain by applying my data to **sklearn**.

$$\textbf{Confusion Matrix} = \begin{matrix} 135 & 16 & 25 & 4 & 35 & 5 & 5 & 27 & 8 \\ 0 & 210 & 0 & 0 & 9 & 0 & 5 & 4 & 0 \\ 16 & 2 & 75 & 4 & 8 & 5 & 9 & 28 & 13 \\ 0 & 0 & 2 & 56 & 2 & 23 & 10 & 11 & 6 \\ 14 & 29 & 1 & 0 & 188 & 0 & 11 & 22 & 9 \\ 2 & 0 & 1 & 25 & 3 & 76 & 2 & 1 & 5 \\ 0 & 4 & 1 & 9 & 6 & 4 & 172 & 13 & 6 \\ 0 & 2 & 4 & 10 & 15 & 2 & 34 & 117 & 8 \\ 0 & 2 & 1 & 4 & 2 & 1 & 2 & 1 & 128 \end{matrix}$$