1. Approach:

In this assignment we employ a concept of eigenvectors to identify eigen faces representing the face images in the training set. Typically, the number of eigen faces correspond to the number of face images in the training set. However, the remarkable feature of eigen faces is that the information of the total training face images is encoded in an order from highest to lowest. The last eigenface might capture highest degree of the information of variation in all the images considered. The second last has second highest information captured and so on so forth. So the top few eigen faces might collectively represent most of a training image. Hence using these few number of eigen faces to represent any training face image can facilitate subsequent computation at an expense of significantly less loss of data consequently accuracy.

Before we determine the eigen faces we subtract the mean of the faces from all the images so the resulting images are made up of characteristics that are unique for the corresponding images. Then we find eigen faces which are eigen vectors in the face image data space.

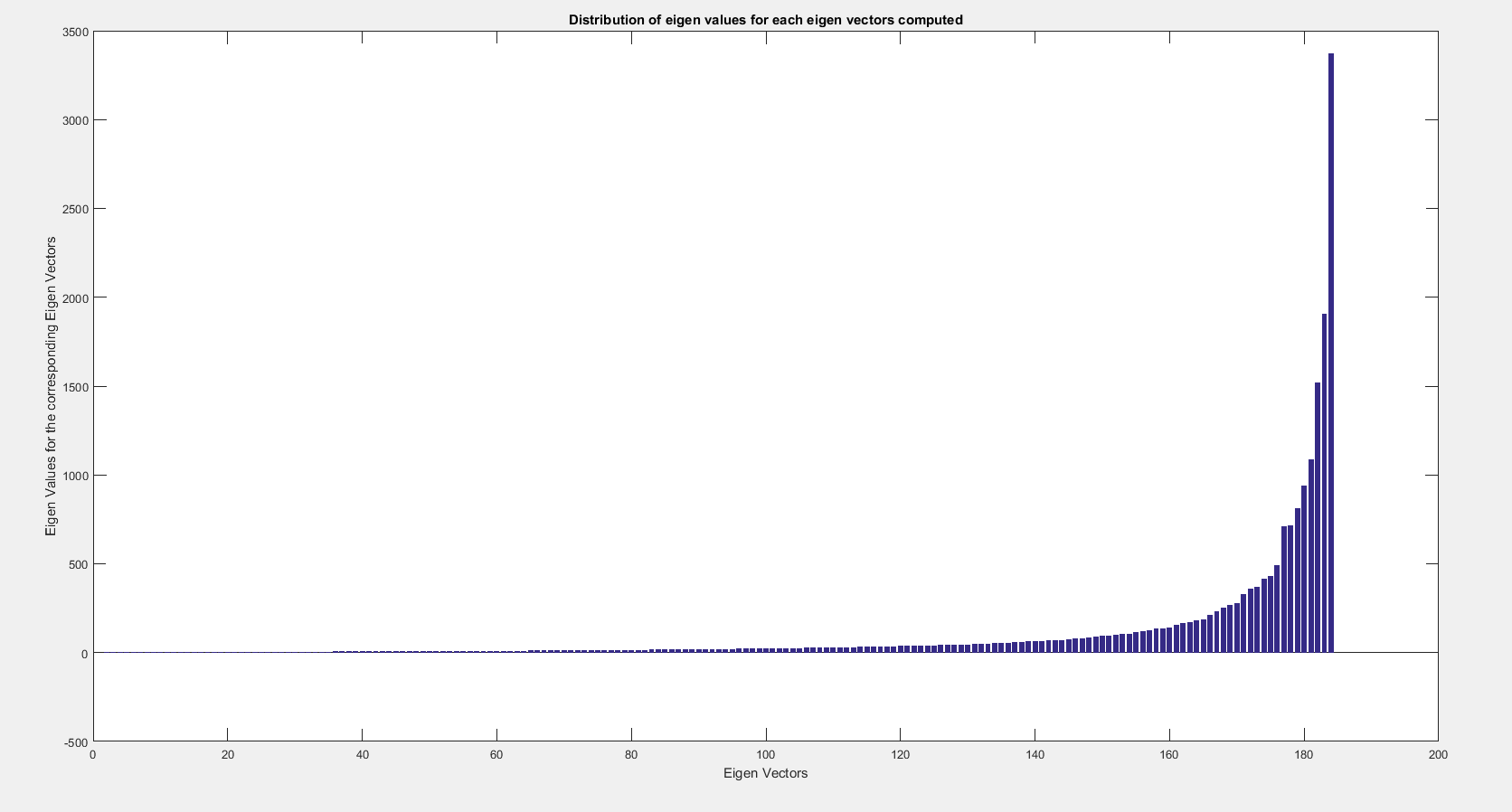
As we have established that the selected few eigen faces can greatly represent images we can find the matrix representation of weights when multiplied with these eigen faces and summed over can recover images sufficiently enough. Hence we get the matrix representation of weights by taking inverse of eigen faces and multiplied with the face matrix representation itself.

We follow the same principle stated above in determining corresponding weights of the corresponding eigen faces (same as above) this time for testing images. We are practically trying to represent the testing images as a weighted sum of the same eigen faces we found for the training dataset. Now we use different distance metrics to compute the distances between the weights of a certain testing image and the weights of all the input images. We pick an image whose weights have the smallest distance from the testing image as sharing similar characteristics and as closet in resemblance.

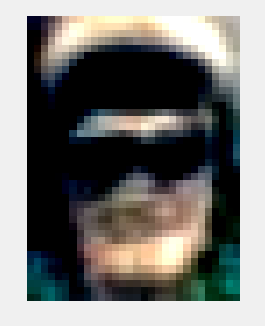
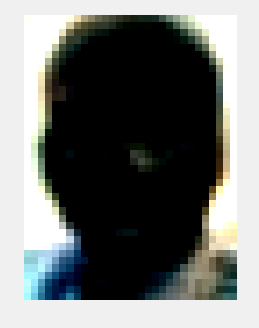
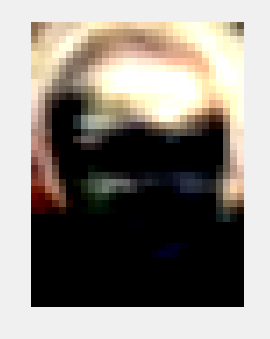
1. Implementation in Matlab:

The main body of code resides in the file eigenface.m. The distribution of the eigen values of eigen vectors is presented below. Eigen Values represent the degree of variance of the total images explained by the corresponding eigen vector. From the graph plotted below it is evident that the last few eigen vectors account for the majority of the variance in the training images. Hence, heuristically, choosing the value for the number of eigen vectors from last can help save significant computational cost otherwise incurred by considering the original data size of the training images. This action comes at the expense of performance but that is likely trivial.

So, we experiment with the value for the number of eigen vectors and record the performance at each step. K denotes the number of eigen vectors selected from last index.



Below are the eigen faces for the last few eigenfaces.



1. Results:

Below are the tables displaying recorded accuracies in recognizing testing faces parameterized on for each K value and the distance metric combination.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Norm1 | Norm2 (Euclidean) | Mahalanobis |
| K=10 | 0.20 | 0.20 | 0.26 |
| K=20 | 0.24 | 0.23 | 0.21 |
| K=40 | 0.30 | 0.23 | 0.20 |
| K= 60 | 0.32 | 0.23 | 0.17 |
| K = 80 | 0.31 | 0.23 | 0.13 |
| K=100 | 0.32 | 0.23 | 0.16 |

Table Table for RGB color space

|  |  |  |  |
| --- | --- | --- | --- |
|  | Norm1 | Norm2 (Euclidean) | Mahalanobis |
| K=10 | 0.22 | 0.17 | 0.29 |
| K=20 | 0.23 | 0.17 | 0.26 |
| K=40 | 0.26 | 0.19 | 0.26 |
| K= 60 | 0.28 | 0.19 | 0.22 |
| K = 80 | 0.27 | 0.19 | 0.16 |
| K=100 | 0.28 | 0.19 | 0.17 |

Table 2 Table for Gray color space

|  |  |  |  |
| --- | --- | --- | --- |
|  | Norm1 | Norm2 (Euclidean) | Mahalanobis |
| K=10 | 0.20 | 0.22 | 0.18 |
| K=20 | 0.29 | 0.29 | 0.27 |
| K=40 | 0.32 | 0.31 | 0.20 |
| K= 60 | 0.30 | 0.31 | 0.13 |
| K = 80 | 0.31 | 0.31 | 0.14 |
| K=100 | 0.31 | 0.31 | 0.12 |

Table 3 Table for HSV color space

|  |  |  |  |
| --- | --- | --- | --- |
|  | Norm1 | Norm2 (Euclidean) | Mahalanobis |
| K=10 | 0.19 | 0.13 | 0.30 |
| K=20 | 0.17 | 0.14 | 0.14 |
| K=40 | 0.20 | 0.14 | 0.24 |
| K= 60 | 0.20 | 0.14 | 0.21 |
| K = 80 | 0.21 | 0.14 | 0.15 |
| K=100 | 0.22 | 0.14 | 0.18 |

Table 4 Table for YCbCr color space

|  |  |  |  |
| --- | --- | --- | --- |
|  | Norm1 | Norm2 (Euclidean) | Mahalanobis |
| K=10 | 0.19 | 0.13 | 0.30 |
| K=20 | 0.17 | 0.14 | 0.14 |
| K=40 | 0.20 | 0.14 | 0.24 |
| K= 60 | 0.30 | 0.31 | 0.13 |
| K = 80 | 0.31 | 0.31 | 0.14 |
| K=100 | 0.22 | 0.14 | 0.18 |

Table 4 Table for HSVYCbCr color space

|  |  |  |  |
| --- | --- | --- | --- |
|  | Norm1 | Norm2 (Euclidean) | Mahalanobis |
| K=10 | 0.16 | 0.16 | 0.19 |
| K=20 | 0.20 | 0.17 | 0.22 |
| K=40 | 0.23 | 0.19 | 0.18 |
| K= 60 | 0.24 | 0.19 | 0.16 |
| K = 80 | 0.24 | 0.19 | 0.14 |
| K=100 | 0.25 | 0.19 | 0.12 |

Table 4 Table for Gradient color space