ECE 513: Computer Assignment 1 Topic: Image Transforms and Applications

Danish Gufran

CSU ID: 833161673

Due Date: 03/23/2021

1 Introduction

The goal of this Computer Assignment is to implement Principal Component Analysis (PCA). Many sources confirm that PCA is the most successful technique that has been implemented for image recognition, image data reduction (compression), and image classification. PCA has widely been employed for prediction, redundancy removal, feature extraction, data compression. Since, PCA is a classical technique it is efficient to use it on linear models like digital signal and digital image processing. This assignment focuses on the digital image processing aspects and we can implement this technique for image transforms to reconstruct the trained image. The most exciting implementation of the PCA is the eigenspace projection. This is a facial pattern representation, where large 1-D vector pixels that are constructed from 2-D facial images. This data is stored as a compact principal-components of the feature space. Eigen space is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images(vectors). The details are described in the following sections. We use "Yalefaces" dataset which consist of 5850 faces, for this assignment we have extracted 165 faces from the original database, the obtained images are categorized as 15 people with 11 types of expressions for each person. The final portion for comparison and result analysis is to compare the image quality from the original and reconstructed images (SNR), the project also makes an attempt to determine minimum distance classifications between the images, where similar images are clustered and called based on the training images.

2 Theory

2.1 Principal Component Analysis

We convert a 2-D facial image to 1-D vector by concatenating each row or column into a vector. Let's us consider M vectors of size N representing a set of training images. Where M is the Row and N is the Column of the 2-D image matrix. The P's used in the below formula refer to pixel values of the image.

$$Xi = [p1 \ldots pN]^T \qquad (1)$$
$$\text{Where, } i = 1, \ldots, M$$

The images in the training set are of different people under same condition. For my training set I used consist of sad faces. These images are converted then converted from 2-D facial images into 1-D vector. Using this 1-D vector we calculate the mean faces using the formula. The images or faces are mean centered by subtracting the mean image from each image vector. Let m represent the mean image.

$$m = \frac{1}{M} \sum_{i=1}^{M} x_i \qquad (2)$$

Where X is the number of images

After this, we calculate the mean centered image. The zero mean face matrix is denoted at Z_m. This is calculated by subtracting the mean face matrix value with the training images.

$$Z_m - X_i - m \qquad (3)$$

After this, we calculate the covariance matrix C_m.

$$C_m = Z_m * Z_m^t \qquad (4)$$

Where, Z_m^t is the transpose of the zero matrix

The next step is to determine a set of eigen values which have the largest possible projection onto each centered image. This eigen value is denoted as e_i.

$$L_i = \frac{1}{M} \sum_{n=1}^{M} (e_i^T * Z_m)^2 \qquad (5)$$

Where, e_i and L_i are the eigenvectors and eigenvalues.

We then use this covariance matrix to calculate the eigen vectors and eigen values which consist of the Principal component features of an image. The eigen vectors of the covariance matrix define a space in which the dimensions are zero so that the covariance matrix is a diagonal matrix. The dominant principal component is calculated from the covariance matrix by using reduced dimensionality. We then use the highest eigen vectors to train the system. We then use this vector to calculate the eigen faces E of each images by multiplying the eigen vector to the actual image.

$$E = x_i * c_m \qquad (6)$$

Now, we can reconstruct the actual image from the eigen faces.

2.2 Reconstructing the Image

The Eigen vectors and Eigen values found using PCA is used to re- construct the image. We do this by find the PC of the Test image and then use this PC and multiply it with the eigen faces found from the training set. Then this matrix is reshaped and displayed as image.

2.3 Signal to Noise ratio

We calculate the SNR by multiplying the ratio of Reconstructed im- age with Actual image with 10log10.

$$SNR = 10\log_{10} * \frac{\text{Reconstructedimage}}{\text{ActualImage}}$$

The signal to noise ratio is the key factor to determine the quality of the reconstructed image with respect to the original or actual image. A high SNR value dictates that the image signal received or reconstructed is higher than the noise signal, this makes it a good image. While a lower SNR value dictates that the image signal received or reconstructed is lower than the noise signal, this makes it a poor image.

2.4 Image Recognition

For image recognition we start by training the dataset in different categories. In this project we have focused on sad faces as an example. Then we save the PC of all the image data into different matrix I_R. when the Test image is given the PC of these images is calculated then this is compared with dataset of Each Image and then the minimum distance of the image with the dataset is calculate. The minimum distance value is also utilized to calculate the difference between the trained the image and the tested image, this information can be used to find to detect similarities with different images or faces with respect to this project.

The recognition for the images can be calculated by the formula below.

$$Reco = \text{argmin}_k ||\text{Testimage} - I_r||$$

2.5 Algorithm for the project

General procedure to create MATLAB code for this project.

1. Load all training and testing data (images) onto separate folders
2. Call the training data address in MATLAB
3. Number all the images
4. Center the image data
5. Create image matrix
6. Reshape 2-D images into 1-D vectors
7. Calculate Eigenfaces and mean of the training data
8. Compute mean image vector
9. Display mean faces
10. Calculating A matrix (after subtraction of all image vectors from the mean image vector)
11. Calculate Eigenvector matrix
12. Display Eigen faces
13. Image reconstruction
14. Calculate and display the SNR value of the reconstructed image
15. Load test image
16. Extract PCA features from the test image
17. Calculating and compare the Euclidian distance of all projected trained images from the projected test image
18. Calculate the SNR value
19. Display the test case image with its SNR value
20. Perform minimum distance classification on the test data

3. Results

The training data category chosen are sad faces.

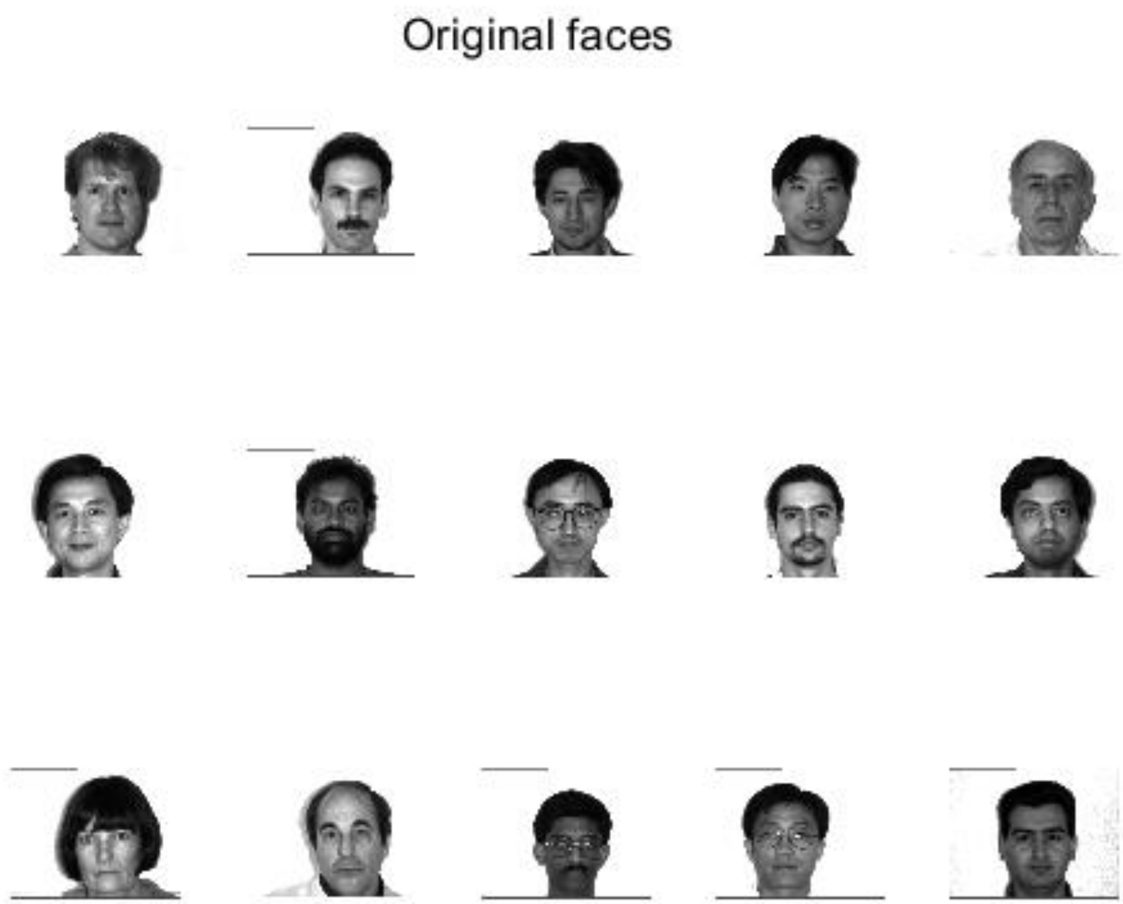

Fig. 1: Training Images

The next image is mean face from the training dataset

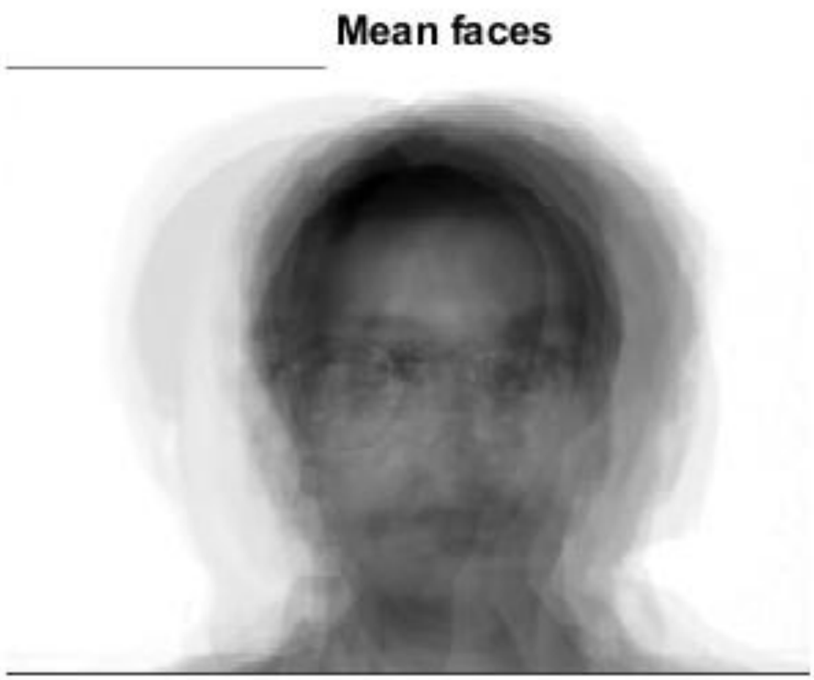

Fig. 2: Mean face image

The next image is zero mean face from the training dataset

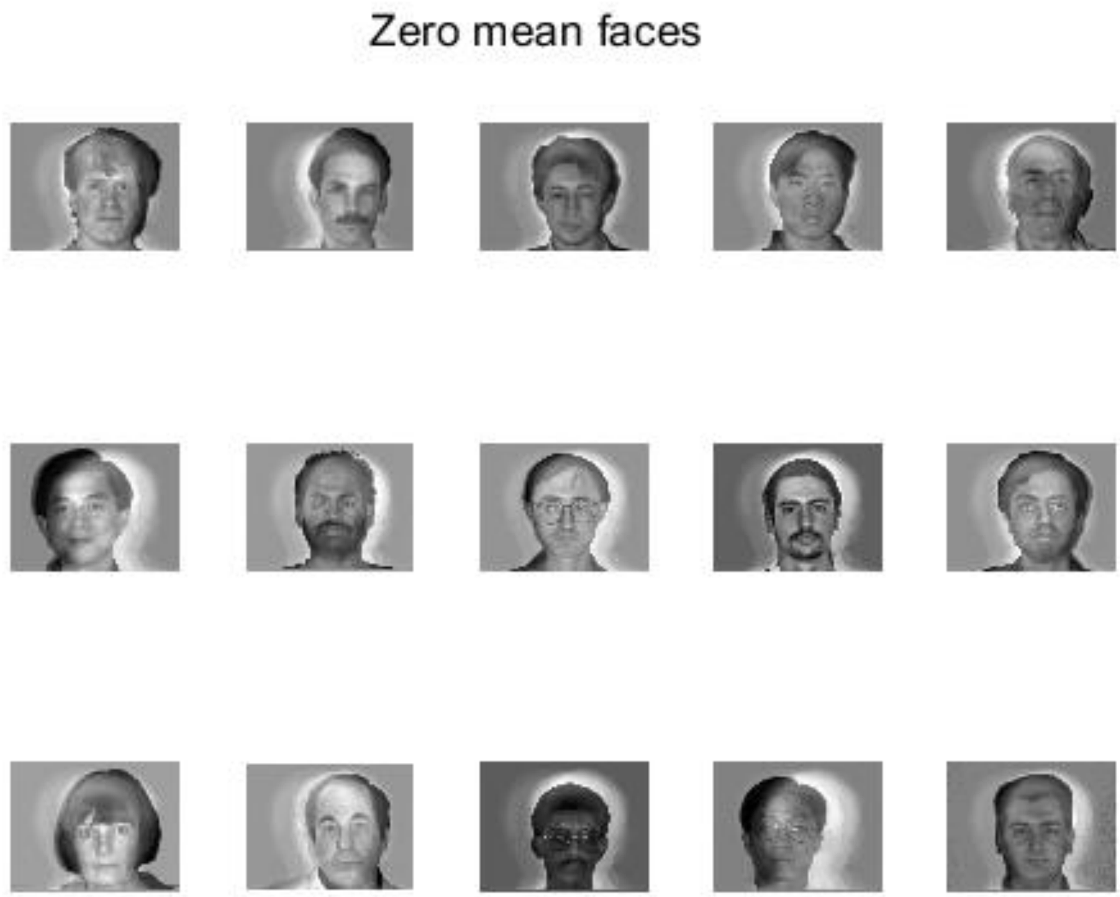

Fig.3 : Zero mean face image

The next image is the eigen image.

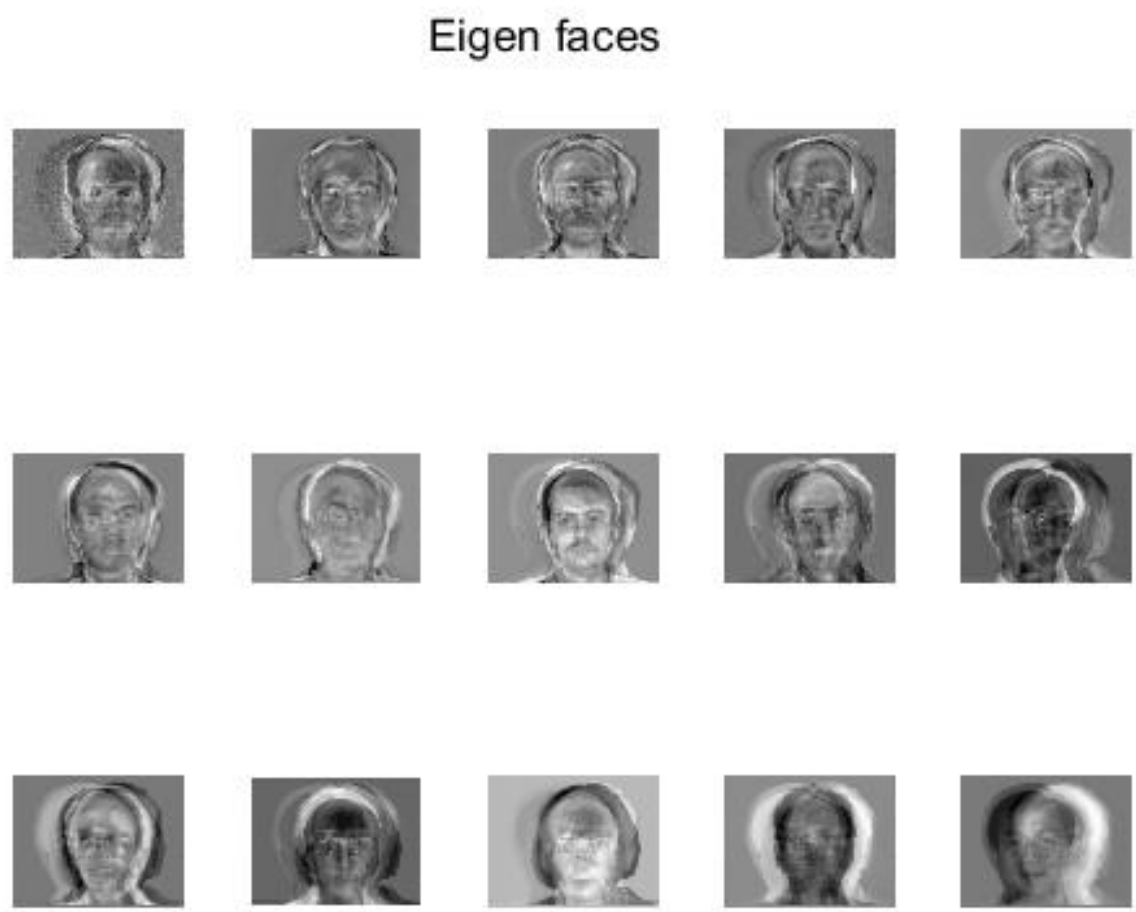

Fig.4 : Eigen images

The next image is the reconstructed image

Fig.5 : Reconstructed image with SNR

Testing individual images with improved dataset. The new original images now consist of 2/3 value of the dataset. The new images used for training are 110 images.

The loaded dataset images were 165 images, now we train 110 images and follow the same procedure to notice any difference in the quality of the reconstructed images. Since, the loading data is high (110 images). I have tested 4 images at random so check the image quality. Through this improvement we notice that there is a significant change in the SNR value of the improved dataset. This concludes that increasing the dataset will improve the working of the PCA.

The images for the 2/3 dataset are as follows.

3.1 Results from 2/3 Dataset

The training data category for 2/3 of the dataset.

Fig. 6: Training Images of the new dataset

The next image is mean face from the training dataset

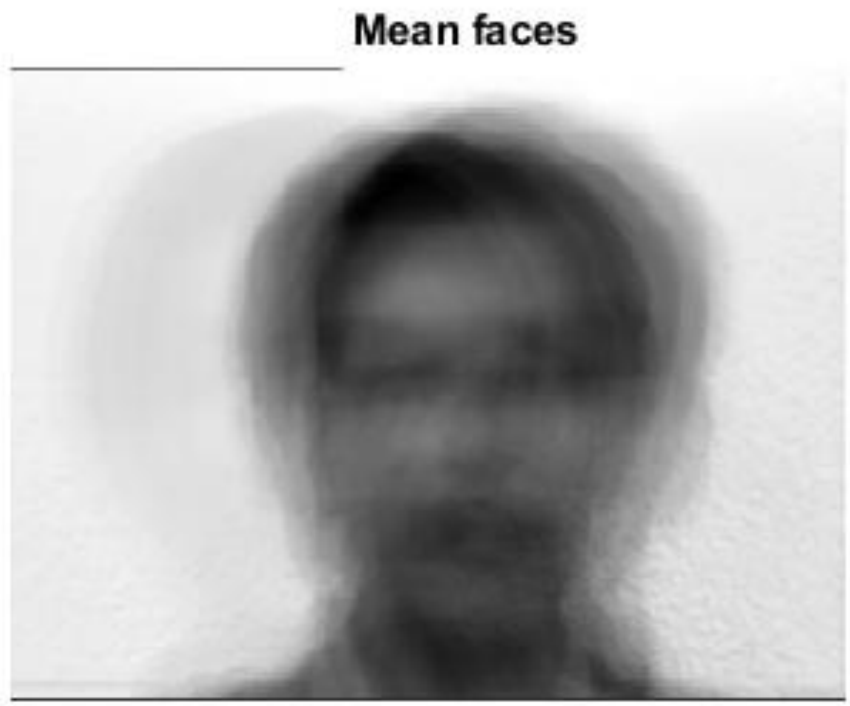

Fig. 7: Mean face image

The next image is zero mean face from the training dataset

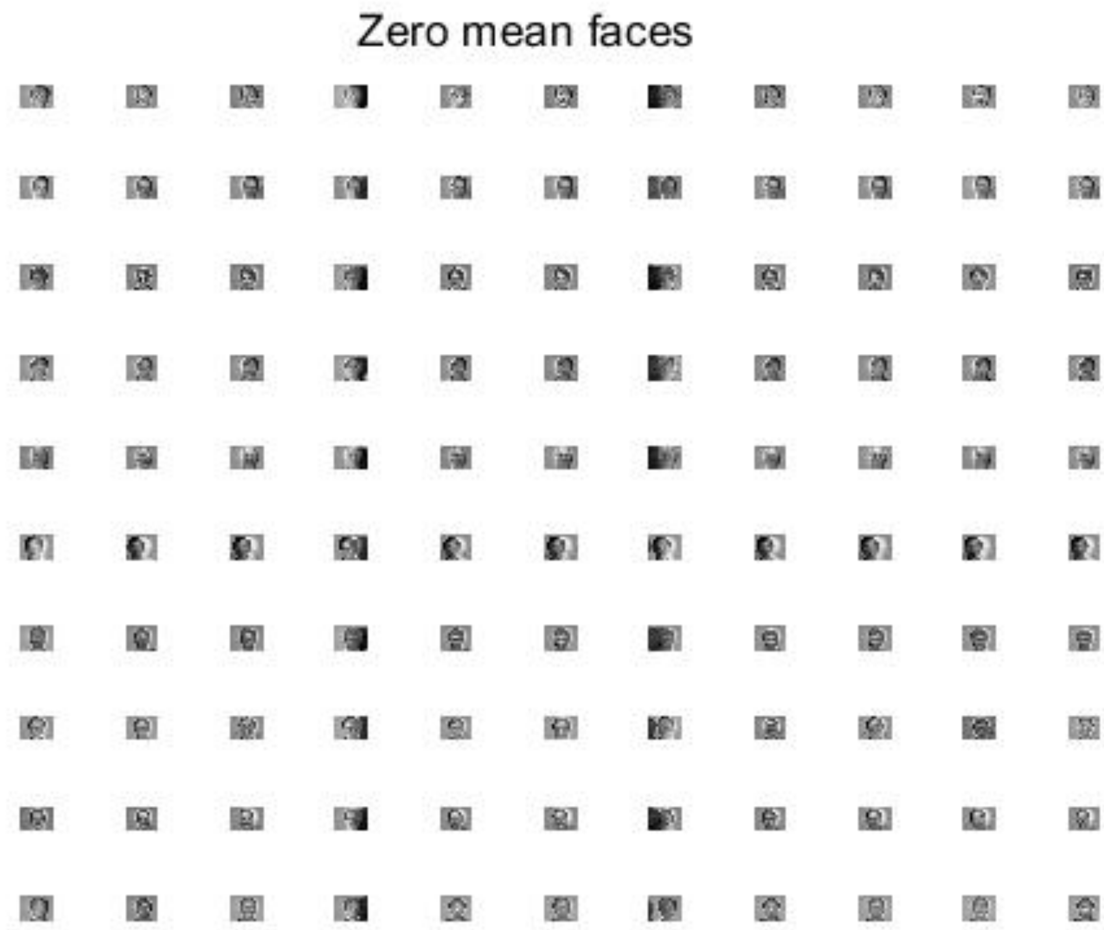

Fig.8 : Zero mean face image

Please note that the data received is for 110 images so the posted image here has lost its clarity, but this does not affect the SNR readings as the pixel values are still intact and this data is still usable for reconstruction.

Documenting this image onto a report has shrunk in size, but the image data and 1-D vector information is still reserved.

The next step is to test out the image by reconstructing the vector and calculating the SNR value.

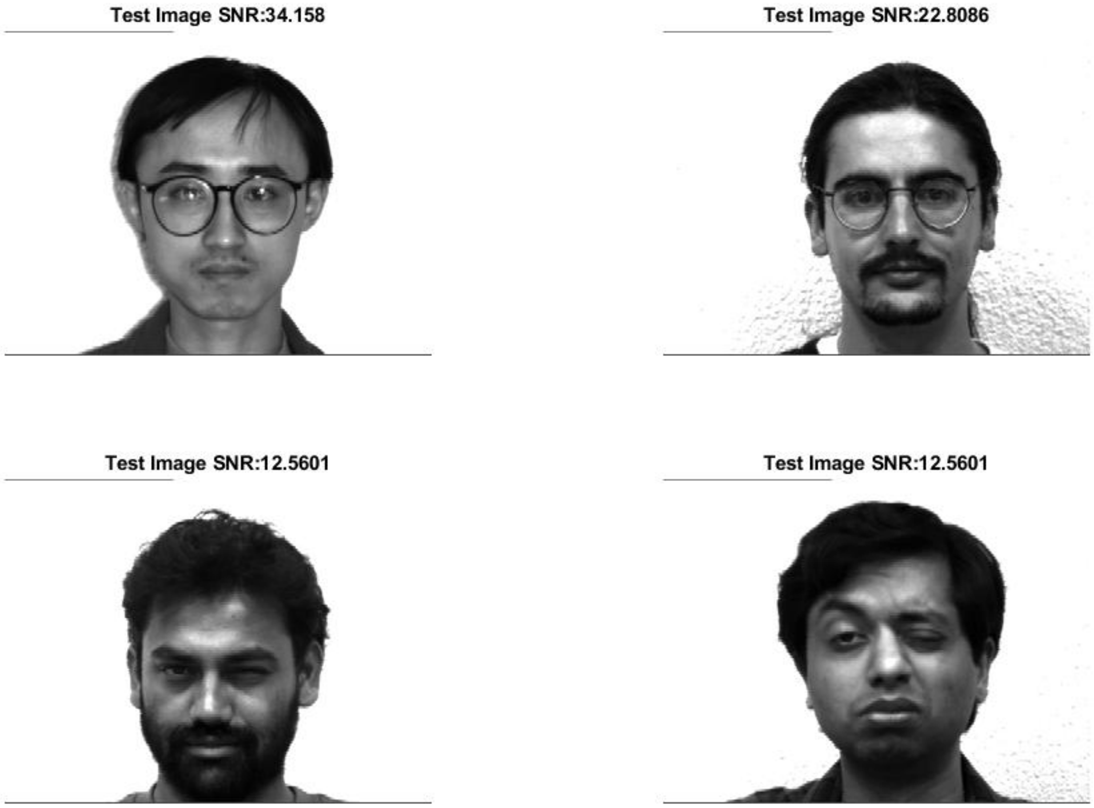

Fig.9 : Test images

Through comparison we can prove that the increased training data has an affect on the reconstruction quality. There is a clear understanding that the SNR value has bumped due to a surplus of training data.

3.3 Results of self-image

When I tested the system with my face, I got a very low SNR score but the image was reconstructed properly

Fig. 10: My picture (final test)

The reconstructed image was found to have a low SNR value as I only used 1 picture of mine to train and reconstruct and as a result a low SNR value (poor reconstruction) image was reconstructed.

Fig. 11: Reconstructed image

3.4 Minimum distance classification results

The minimum distance classification is used to generate similar images based on the training dataset. The procedure followed here is to train 2/3 of the images (110 images) and test one of the images to reconstruct a similar image. The results from this are as displayed.

Fig. 12 Called image

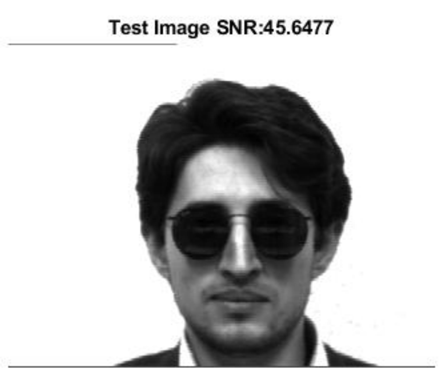

Fig. 13: Classified image.

Since, both the images (train and test) are available to the PCA model, we could extract the same image back with a significant increase in the SNR value. The next case of testing is done by completely removing the testing image from the training set. Now, we can truly test the minimum distance classification. As expected the model returned the closest image with a lower SNR value.

Fig. 14 new test for image classification

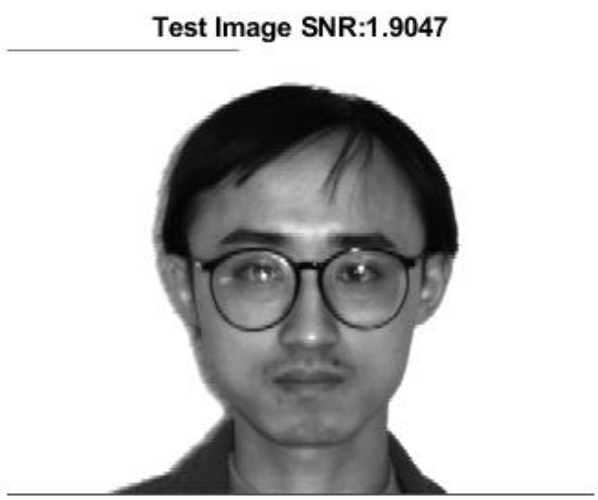

Fig.15 : classified image received

4. Conclusion

The conclusion of this project are as follows. It is observed that the accuracy of the PCA model primarily focuses on the Eigenface information for its construction. It can also be concluded that increasing the data images can have an impact on the reconstructed image SNR value (image quality), through testing we realized that the training model with 110 images reconstructed better quality images that the lower training data model. This was further verified when I tested my own picture, where I had trained only 1 picture of mine and when the same was reconstructed a poor-quality image was obtained. The later portion of this project also makes an attempt to perform minimum distance image classification. The end results show a nearly resembling image was reconstructed with a lower SNR value, but the goal of this classifier was to extract features and reconstruct them on a new image to find the closet match image and this matching was done successfully.

5. References

1. Dr. Mahmood R Azimi-Sadjadi Lecture Slides.
2. M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.
3. K. I. Diamantaras and S. Y. Kung, "Principal Component Neural Networks: Theory and Applications", John Wiley & Sons,Inc., 1996.
4. Face Recognition Using PCA and Minimum Distance Classifier by Shalmoly Mondal, Soumen Bag, Publisher: Springer Singapore