ECE 513: Computer Assignment 1
Topic: Image Transforms Applications

Abhishek Balasubramaniam

CSU ID: 832535894

Due Date: 03/23/2020

1 Introduction

The goal of this Computer Assignment is to implement Principal Component Analysis (PCA) for Image data reduction and facial pattern representation and classification. PCA is the most successful techniques that have been used in image recognition and compression. PCA is a statistical method under the used to reduce the large dimensionality of the data space to the smaller intrinsic dimensionality of feature space, which are needed to describe the data economically. The main idea of using PCA for facial pattern representation is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called Eigen space projection. 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 165 faces. This dataset has faces of 15 people with 11 type of expressions each.

2 Theory

2.1 Principle 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 images in the training set are of different people under same condition. For my training set I used consist of happy faces. These images are converted then

converted from 2-D facial images into 1-D vector[1]. Using this 1-D vector we calculate the mean faces using the formula.

$$m = \frac{1}{M} \sum_{i=1}^{M} x_i \tag{1}$$

In the above equation 1 M is the number of 1-D Vectors, x is the images.

Using this Mean faces matrix we calculate the Zero mean face matrix z_m by subtracting the mean faces matrix from each image matrix x_i from the training set.

$$z_m = x_i - m \tag{2}$$

From zero mean face matrix we can calculate the covariance matrix c_m using the following equation.

$$c_m = z_m * z_m^t \tag{3}$$

Where z_m is the zero mean face matrix and z_m^t is the transpose of the zero mean face matrix.

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.To calculate the dominant PC we have kept a threshold of 95% for each eigen value and the corresponding eigen vector is taken into consideration[2]. We then use this vectors to calculate the eigen faces E of each images by multiplying the eigen vector to the actual image.

$$E = x_i * c_m \tag{4}$$

]We the use this Eigen faces to reconstruct the actual image. The following this the algorithm to calculate Eigen faces using PCA.

Algorithm 1: Algorithm of Principal Component Analysis

Result: Finding Eigen faces
Load the Training dataset;
Convert the images to 2-D image matrix;
Converting 2-D Images Matrix to 1-D vectors;
Find Mean face matrix;
Find the Zero Mean matrix using mean face matrix;
Compute the co-variance matrix;
Compute the eigen values and eigen vectors;

2.2 Reconstructing the Image

The Eigen vectors and Eigen values found using PCA is used to reconstruct 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.

Algorithm 2: Algorithm of Principal Component Analysis

Result: Reconstructed Image
Load the test image;
Find the PC of the test image;
Multiply the PC found with the Eigen faces;
Reshape the output matrix;
Display it to the user;

2.3 Signal to Noise ratio

We calculate the SNR by multiplying the ratio of Reconstructed image with Actual image with 10log$_{10}$[3].

$$SNR = 10log_{10}\frac{Reconstructedimage}{ActualImage} \tag{5}$$

2.4 Recognition

For recognition we start by training the dataset in different categories example happy, sad, glasses, etc. Then we save the PC of all these dataset into different matrix P_K. 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 dataset with minimum of minimum distance is the corresponding classification[2].

$$Reco = argmin_k||Testimage - P_K|| \tag{6}$$

3 Result

The following is the Training dataset with Happy faces.

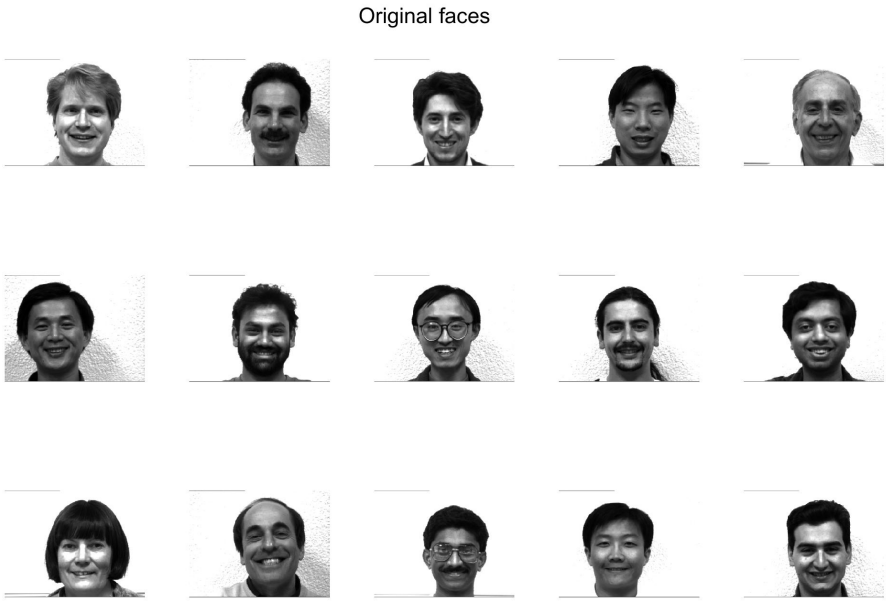

Figure 1: Training Images

The following is the mean face of the training dataset.

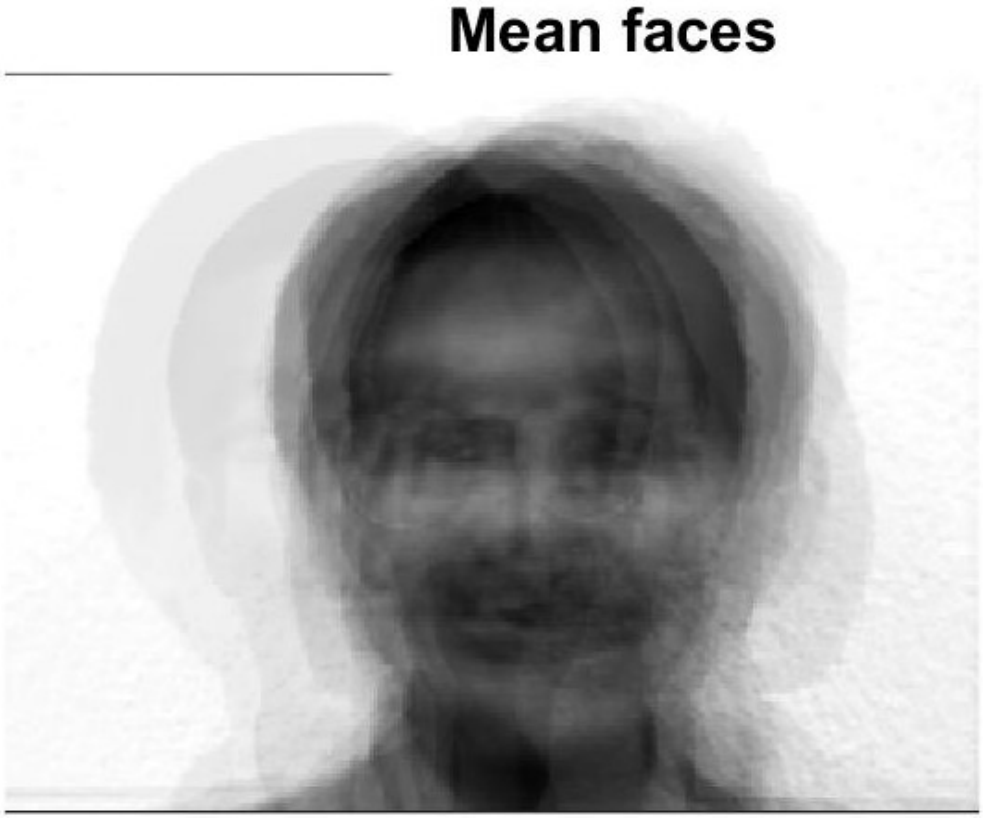

Figure 2: Mean face image

The following is the zero mean faces images.

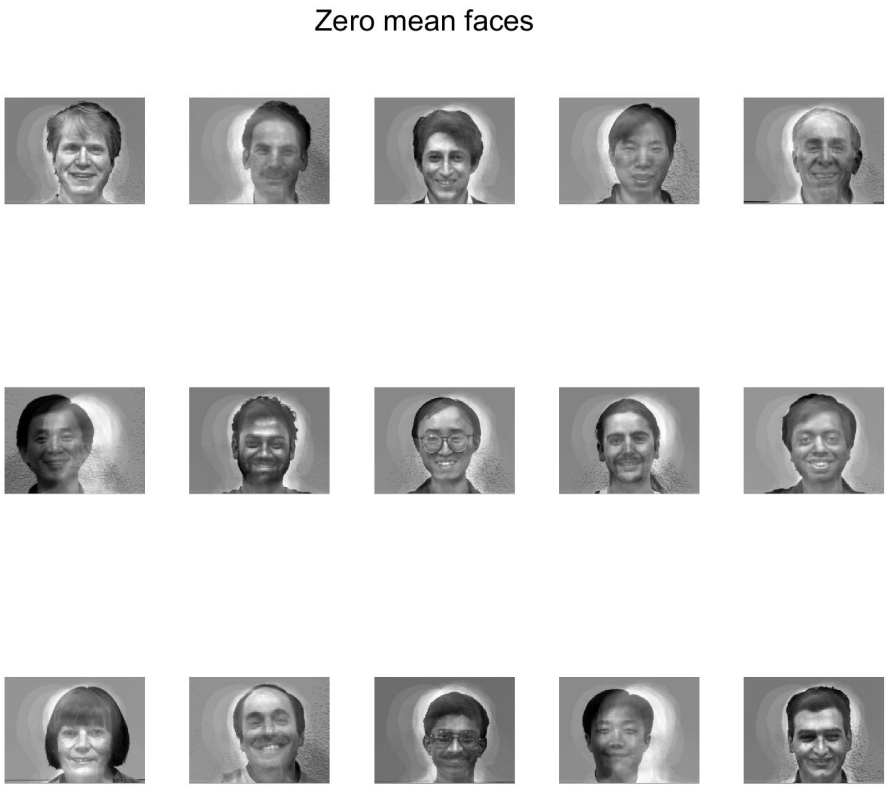

Figure 3: Zero mean faces image

The following is the reconstructed images.

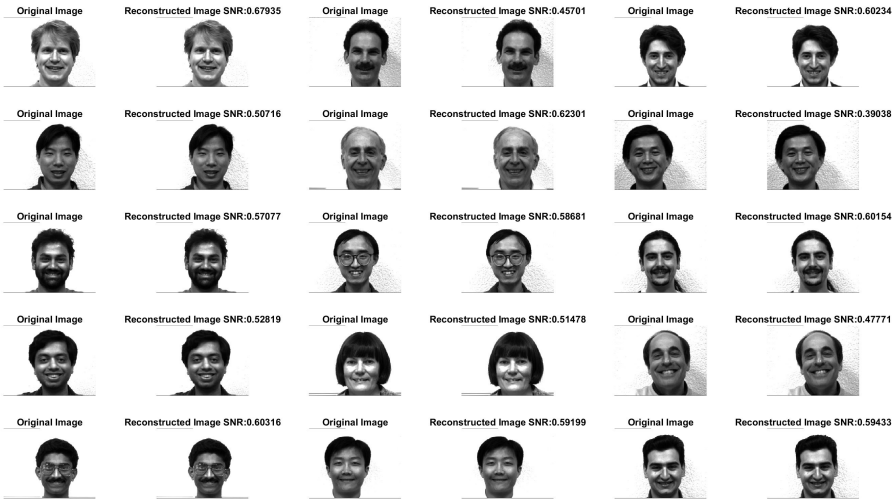

Figure 4: Reconstructed image with SNR values

The above images are found using 15 images in training dataset. when the images are trained with 2/3 of the dataset is used the reconstruction of the images get better as follows.

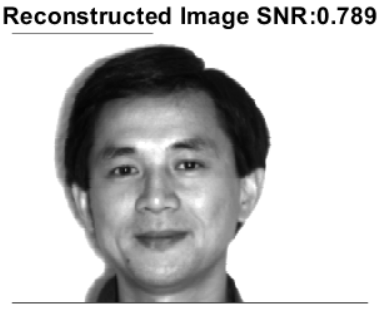

Figure 5: Reconstructed image with 2/3rd dataset

When I tested the system with my face i got a very low SNR score but the image was reconstructed properly when i inspect visually.

Figure 6: Original Image

The following is the Reconstructed image used

Figure 7: Reconstructed image with 2/3rd dataset

4 Conclusion

It can be observed that the accuracy of the PCA algorithm is mainly dependent on the eigenface that we extract from Algorithm 1. When I reconstructed my own image the snr value was very low but their was very low difference visually. The PCA algorithm depends on the eigen values so as long as the dataset is bigger their is a higher chance of getting a very good reconstructed image. Also i use the weights from the actual image to reconstruct the image so it gave me a better reconstruction. When I tried to change the weights and Reconstruct i got very less accurate images.

5 References

[1]. Kwang In Kim, Keechul Jung and Hang Joon Kim, "Face recognition using kernel principal component analysis," in IEEE Signal Processing Letters, vol. 9, no. 2, pp. 40-42, Feb. 2002.
[2]. Dr. Mahmood R Azimi-Sadjadi Lecture Slides.
[3]. https://www.image-engineering.de/library/technotes/706-noise-snr-vs-visual-noise