Indian Institute of Technology, Guwahati

EE657 : Pattern Recognition and Machine Learning

Assignment 2 :
Face Reconstruction and Recognition using Principle Component Analysis

Instructor: Dr. Suresh Sundaram

Name : Akshat Bordia
Roll No. : 11010848
Branch : Electronics and Electrical Engineering
Email ID : a.bordia@iitg.ernet.in

Contents

1. Problem Statement and Objective:

In this assignment, our objective is to do the task of **Face Reconstruction and Face Recognition** from given data set. Considering the huge dimensionality and limited number of data samples, we need to apply **Principle Component Analysis (KL Transform)** on the given data to reduce it to lesser dimensions and then proceed with other task in order to achieve an efficient computation. Following are the given aspects of the problem:

1. **Training Data: Gallery.zip:** The gallery folder `Gallery.zip' contains images from 40 individuals, each of them providing 5 images. The pixel intensities of the 200 face images will be used for computing the KL Transform.

2. **Test Data: Probe. Zip:** The test image folder `Probe.zip' contains 5 images of each of the 40 individuals. After applying PCA to data, we need to do face recognition task for this data set.

3. **Algorithm and Methods Involved:** In this assignment, first to reduce the dimensionality of given data set, we use **Principle Component Analysis (Karhunen Loeve Transform)**. For face recognition task in second half of assignment special case of **K-Nearest Neighbour Method** where K=1 was used

Few Data Samples:



2. Principle Component Analysis

In the practical applications of Machine Learning and Pattern Classification, we need an optimum classifier and suitable feature vector for minimum error classification. Apart from classifier design and feature extraction, a major issue in machine learning is the dimensionality of data and limited data sets. We often deal with limited image data set which have covariance matrix of 10^5 order. In this case, if our data sets contains only N observation (N<D: Dimensionality of Feature), then anyways we are going to have a covariance matrix which has rank <= N. Hence other than those N directions, has zero eigenvalues, which in turn represent the zero variance in those directions. Hence we can consider higher eigenvalues component, which has higher covariance and we can project our data onto those corresponding eigenvectors. This process is called Principle Component Analysis and hence significantly reduces the dimensionality of data making our computation less costly and efficient.

Mathematical Formulation and Derivation for PCA:

In order to reduce dimensionality, we are looking for projection of high dimensional data to a lower dimensional subspace with minimum loss of information. In PCA, this is achieved by maximizing the variance of projected data.

The projection of X on the direction of W is:

$$z = \boldsymbol{w}^T \boldsymbol{x}$$

If W_1 is a principle component, then variance of projected data must be maximum and to normalize W_1, we should find the W_1 which satisfies transpose $(W_1)*(W_1)=1$

If we project data on W_1, the variance of projected data will be given by following expression, where ∑ is covariance matrix of data in original subspace.

$$\mathrm{Var}(z_1) = \boldsymbol{w}_1^T \Sigma \boldsymbol{w}_1$$

We are looking for a W_1 such that $\mathrm{Var}(Z_1)$ is maximum and normalization criterion is satisfied. Using Lagrange Optimization Method :

$$\max_{\boldsymbol{w}_1} \boldsymbol{w}_1^T \Sigma \boldsymbol{w}_1 - \alpha(\boldsymbol{w}_1^T \boldsymbol{w}_1 - 1)$$

Taking the derivative with respect to W1 and equating it to zero, we get

$$\Sigma w_1 = \alpha w_1$$

This is eigenvector-eigenvalue equation of covariance matrix. The variance is maximized when we choose the eigenvector corresponding to highest eigenvalue. Hence first principal component is the eigenvector corresponding to highest eigenvalue of Covariance Matrix.

To find further Principle Components, we apply the same constraints as in first principle component and this principle component must be orthogonal to previous Principle Component. Thus it can be shown that we need to select eigenvectors corresponding to highest eigenvalues to do Principle Component Analysis and Project Data to reduced dimensions.

Apart from above mentioned Variance Maximization criterion, we can also formulate PCA based on **MINIMUM PROJECTION ERROR** criterion. This includes sequential process of finding the best directions to project the data in order to minimize the projection error and leads to same conclusion as in Variance Maximization Criterion.

PCA for High Dimensional Data:

In Principle Component Analysis of high dimensional data, we have the computation problems described in introduction of this section and all the eigenvalues of sample covariance matrix are not significant. Hence to reduce computation complexity and increase efficiency of our algorithm, we consider the following matrix:

X is an (N X D) dimensional matrix, whose nth row is given by : transpose (x_n- μ).

We can write covariance matrix in terms of X as:

$$\text{Cov} = \frac{1}{N} * X'X$$

We can write eigenvector equation for covariance matrix as:

$$\frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

$$\frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{u}_i) = \lambda_i(\mathbf{X}\mathbf{u}_i).$$

$\mathbf{v}_i = \mathbf{X}\mathbf{u}_i$,

$$\frac{1}{N}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

Thus we reach to above matrix X*X’, which is 200*200 case matrix in our case. This has same N eigenvalues as covariance matrix. Now we can compute eigenvalues with computation complexity of O (N^3) instead of O (D^3) in case of our covariance matrix. Hence this significantly reduces the computation complexity.

Finally we have:

$$\left(\frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)(\mathbf{X}^{\mathrm{T}}\mathbf{v}_i) = \lambda_i(\mathbf{X}^{\mathrm{T}}\mathbf{v}_i)$$

In original space:

$$\mathbf{u}_i = \frac{1}{(N\lambda_i)^{1/2}}\mathbf{X}^{\mathrm{T}}\mathbf{v}_i.$$

3. Analysis of Problems and Mathematical Formulation:

Task 1:

To do PCA for the high dimensional data (112*92=10304) give to us in gallery folder, we follow the following steps and find the required outputs for each part of question:

1. For high dimensional data, if we are given N data samples, rank of covariance matrix will be <= N. We can find these non-zero eigenvalues and corresponding eigenvectors by following equation (as described in previous section):

$$\mathbf{u}_i = \frac{1}{(N\lambda_i)^{1/2}}\mathbf{X}^{\mathrm{T}}\mathbf{v}_i.$$

2. **To display the Eigen faces, corresponding to top five eigenvalues,** we select the top Eigenvectors and reshape them to their image matrix and we can display them.
3. **To Plot graph depicting the variance captured,** we use following formula, where we vary **k** from 1 to 200 to calculate the corresponding fraction of variance captured for **k** eigenfaces:

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

We can plot the variance fraction v/s number of dimensions once we calculate the fraction using above formulation.

Hence we can find the number of dimension required for 85% and 95% variance capture.

4. **Reconstruction of Image :**
   If W is our projection matrix, which has top eigenvectors as its columns, then :

$$Y = W'(X - \mu)$$

$$\dot{x} = W * Y + \mu$$

Where $\dot{X}$ is reconstruction of data sample X using eigenvector matrix W. We can choose different W for different eigenfaces and accordingly reconstruct the input image.

5. **Mean Square Error :**
   Mean square error of reconstruction is given by

$$\text{Mean Square Error} = \sum \|\dot{x} - X\|^2$$

Hence we can calculate MSE for different reconstructions and plot it accordingly.

Task 2:

1. **Classification using 1- Nearest Neighbour Classifier:** The data is reduced to 25 dimension and we calculate the distance of test sample from all the images in training sample. We assign the test image to the category for which the distance is minimum. Hence the test images are classified using 1-NN algorithm.

2. We do the classification for various dimension projection and calculate the accuracy in case of each projection. Thus we can plot the accuracy of classification v/s no. of dimensions used for projection.

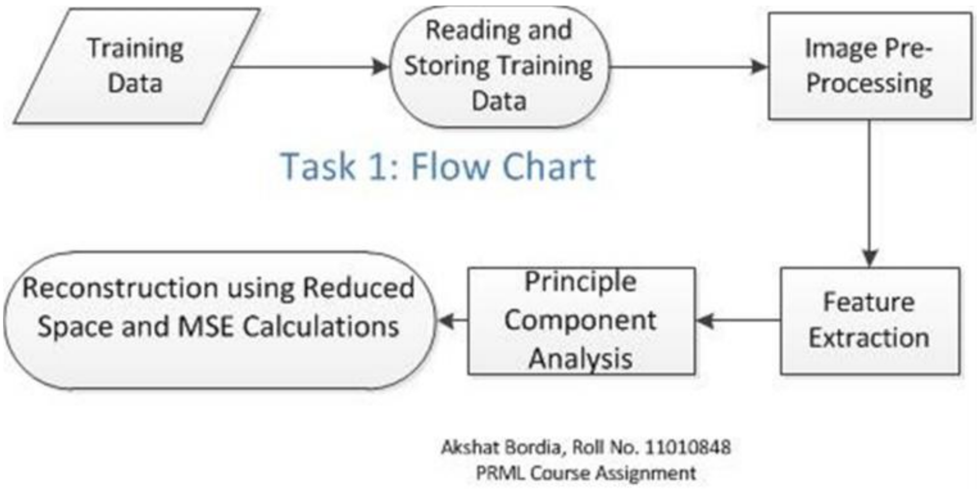
4 .Implementation and Description of Code:

Code for this assignment was implemented in MATLAB2012 (b) using commands of MATLAB Image Processing Toolbox (IPT).

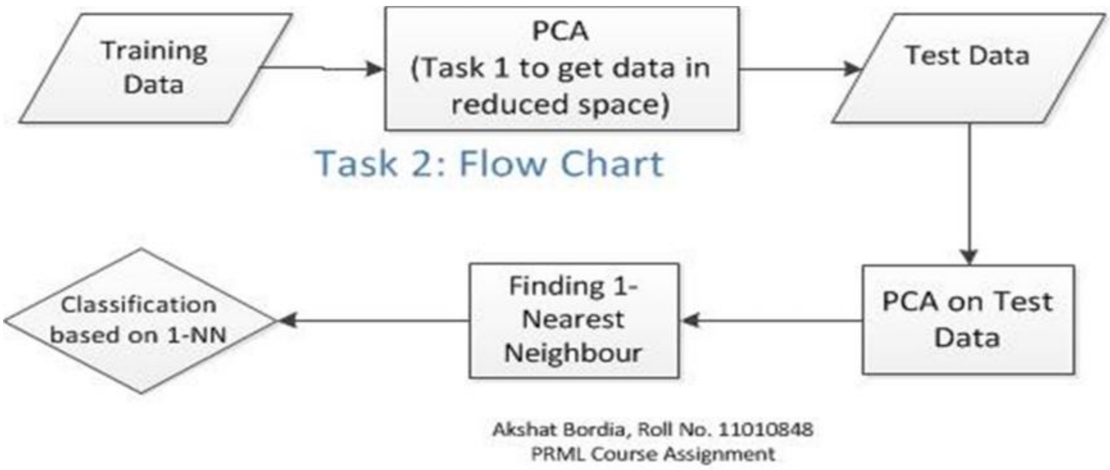
For efficient and fast implementation, special care was taken while implementing code and all large image cells were immediately cleared after their use to speed up. It significantly increased the speed. All parts (except Task 2: Second Part) are showing output within 5-8 seconds. Second Part of Task 2 is taking around 5-6 minutes as in this part, we have to project all 400 images onto 200 different spaces and calculate 200*200 distances every time. Hence 5-6 minutes for this code are reasonable enough to consume.

Flow of code can be explained by these block diagrams:

Task 1:



Task 2:



TO RUN Code:

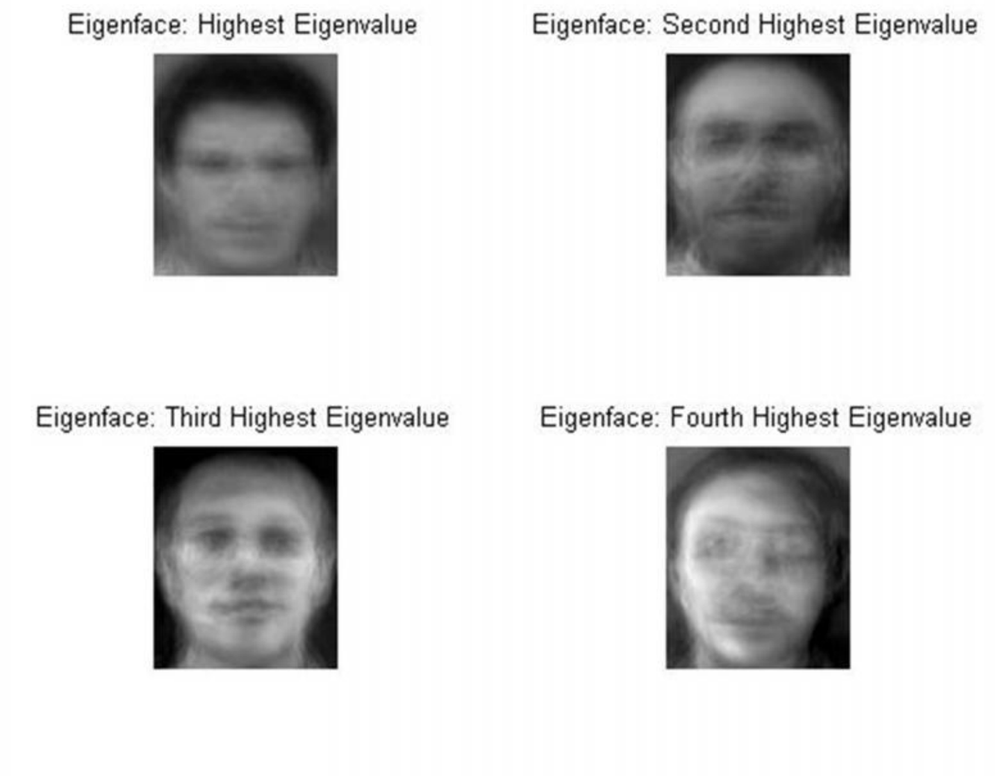
I have prepared separate files for Task A and Task B named: “AkshatBordia_11010848_Assignment2_TaskA” and “AkshatBordia_11010848_Assignment2_TaskB” respectively. One can run them by entering their names in MATLAB command prompt or by clicking on RUN Button in MATLAB Editor.

Each file will output the parts of that task in sequential order. Reconstructed faces, Eigenfaces and Plots are shown in pop up window after running the code.

5. Observation and Inferences: Task 1

Eigen faces Images:

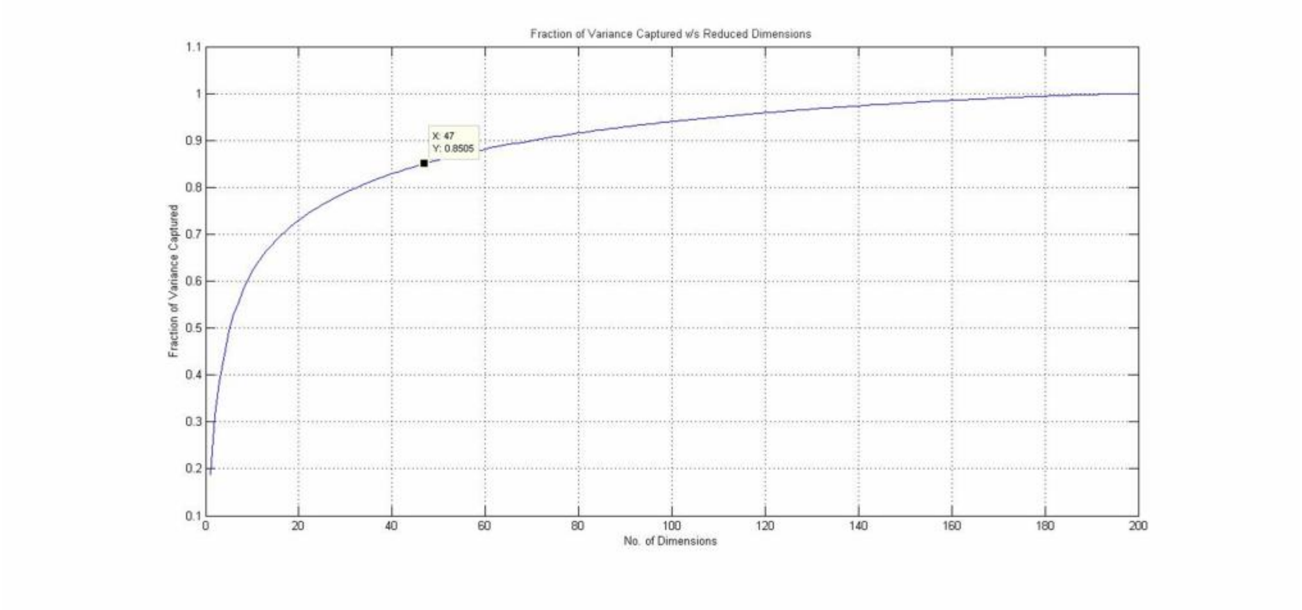
1. Display the Eigen face images corresponding to the top 5 Eigen values of the covariance Matrix.



2. **Plot a graph depicting the percentage of total variance of the original data retained in the reduced space versus the number of dimensions. From this graph, find the number of dimensions required for projecting the face vectors so that:**
   (a) At least 85% of the total variance of the original data is accounted for in the reduced space.

   (b) At least 95% of the total variance of the original data is accounted for in the reduced space

Fraction of Variance Captured v/s No. of Dimensions Plot:



- Plot for fraction of total variance captured v/s No. of Dimension was created using given formulation and “Plot” command in **MATLAB.**
- **Using Data Cursor Tool**¸ We found answers for (a) and (b) part.

- **No. of Dimensions required for 85% variance capture =47**
- **No. of Dimensions required for 95% variance capture =110**

3. Reconstruct the image ‘face input 1.pgm’ using the:

Using Top Eigenface

Using Top 4 Eigenfaces

Using Top 25 Eigenfaces

Using Top 150 Eigenfaces

Using ALL Eigenfaces

Normalized Mean Square Error in each case:

(i) Reconstruction Using Top Eigen face : 201.5895

(ii) Reconstruction Using Top 4 Eigen face : 109.2422

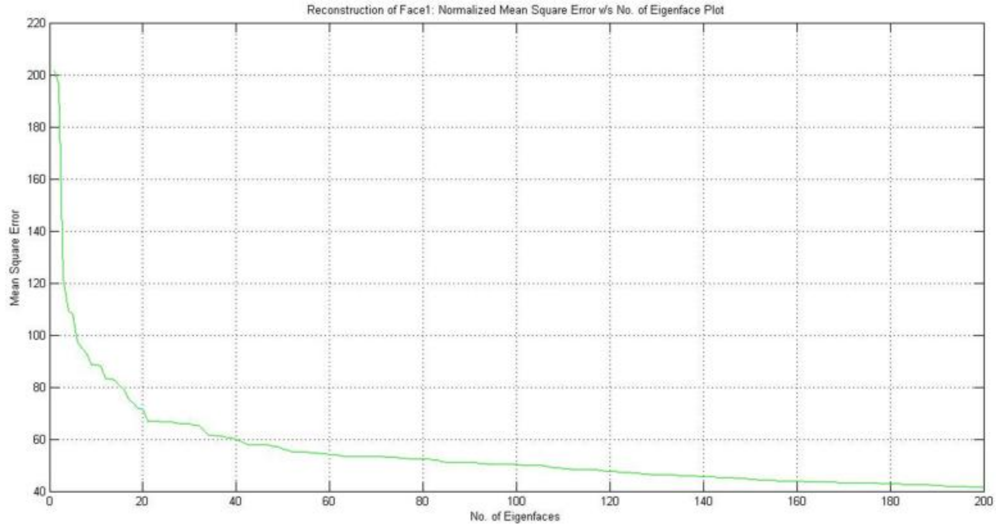
(iii) Reconstruction Using Top 15 Eigen face : 80.5952

(iv) Reconstruction Using Top 150 Eigen face : 55.9456

(v) Reconstruction Using Top 200 Eigen face : 41.5911

4. Depict graphically the mean squared error obtained for different number of Eigenfaces. Note that the Eigenfaces will vary from 1 to 200 (total number of training samples).

Normalized Mean Square Errors v/s No. of Dimensions used for reconstruction:



---

5. Repeat the parts (iii) and (IV) for the image ‘face_input_2.pgm’.

Here we observe (see plot below) that if we use more than 180 dimensions for reconstruction, the error reduces to zero. This essentialy implies that our face_input_2 is the image which is also contained in training set and hence can be represted completely in 200 eigenfaces. Hence we achieve a very good reconstruction for face 2 when we use 200 dimensions.

Reconstruction Using Top Eigenface

Reconstruction Using Top 4 Eigenfaces

Reconstruction Using Top 15 Eigenfaces

Reconstruction Using Top150 Eigenfaces

Reconstruction Using All Eigenfaces

Normalized Mean Square Error in each case:

(i) Reconstruction Using Top Eigen face : 202.5122

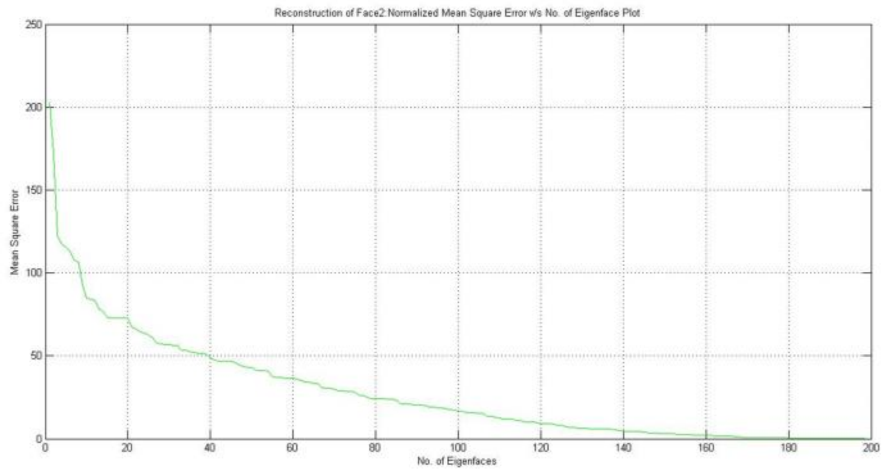
(ii) Reconstruction Using Top 4 Eigen face : 116.8187

(iii) Reconstruction Using Top 15 Eigen face : 72.9776

(iv) Reconstruction Using Top 150 Eigen face : 43.1665

(v) Reconstruction Using Top 200 Eigen face : 0

Normalized Mean Square Errors v/s No. of Dimensions used for reconstruction:



Task 2

The test image folder `Probe.zip' contains 5 images of each of the 40 individuals.

(a) Classify the test samples in this folder by a 1-nearest neighbour classifier (with Euclidean distance) in a reduced 25 dimensional subspace. Compute the classification accuracy.

The training images provided in gallery folder was reduced to a 25 dimensional subspace corresponding to eigenvector's of 25 highest eigenvalues and each test image in Probe folder was also reduced to same subspace. After converting this we evaluated the Euclidean distance of each test image from every training image in reduced subspace. 1-Nearest Neighbour classifier was used to classify the test images. All test images were assigned a number based on their shortest distance. This number was the number of corresponding image in training set.

Total Number of test images : 200
Total Number of Images Classified to same category: 174
Total Number of Images Classified which were not classified correctly: 26

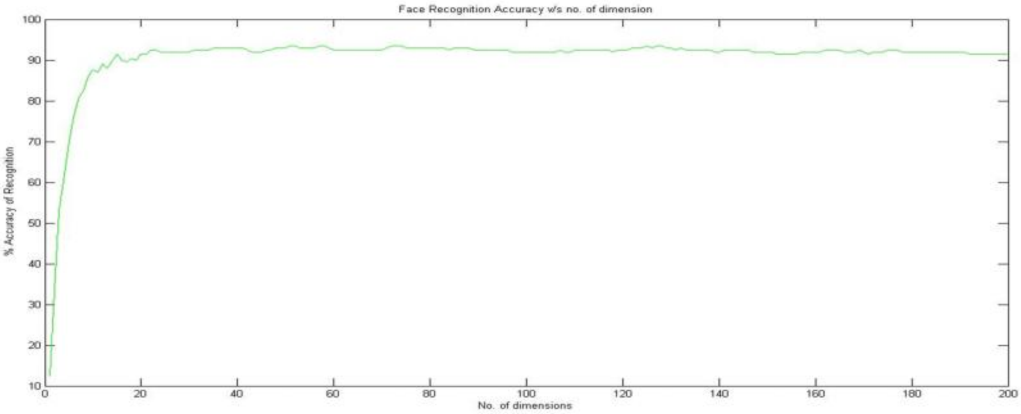
$$\% \, Accuracy = \frac{No. of \; Faces \; Classified \; Correctly}{Total \; no. of \; test \; faces} X \, 100$$

$$\% \, Accuracy = \frac{174X100}{200}$$

Accuracy = 87%

Hence an accuracy of **87%** achieved for face recognition after reducing data to 25 dimensional subspaces.

(b) Depict graphically the recognition accuracies obtained for different number of dimensions. For this part, you have to vary the dimensions from 1 to 200 (total number of training samples).



6. Bibliography:

1. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, *3*(1), 71-86.

2. Abdi, H. and Williams, L. J. (2010), Principal component analysis. WIREs Comp Stat, 2: 433–459. doi: 10.1002/wics.101

3. Duda, R. O., et al. (2012). Pattern classification, John Wiley & Sons.

4. Alpaydin, E. (2004). Introduction to machine learning. MIT press.