UNIVERSITÉ DE BOURGOGNE

University of Burgundy

# Applied Mathematics
# Face Recognition using PCA

*Submitted To:*
Dr.Désiré Sidibé

*Submitted By :*
SHAH Bhargav
PARMAR Hardiksinh

# Contents

# 1 Objective

Our main Objective of this project is familiarize ourselves with singular value decomposition(SVD) and principal component analysis(PCA) methods trough solving the problem of face recognition in MATLAB.

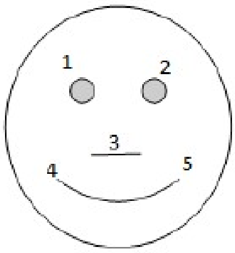
# 2 Introduction

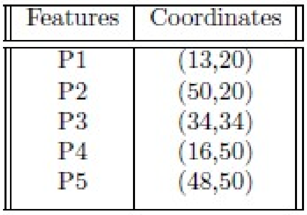
In this project, we implemented a face recognition system by using principal component analysis, which is known as PCA. PCA method provide a mathematical way to reduce the dimension of problem. PCA is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information in the large set. Principal component analysis is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components.
The main objective of our project is to recognize faces from the collected set of face data. In our case we have an input 2D image that should be compared with the database. For example, we have a face image with dimensions 300x300. It can be represented as a vector of size 3002, and we would have to compare vectors in R9000 to find the best match. However, it is difficult to compute distance in high dimensional spaces. We need to find the subspace of Rd where the data structure is preserved. For this purpose PCA tool is used. The aim of this project is to use SVD and PCA for solving FR problem and propose a software to demonstrate how the system works.

# 3 Methodology

## 3.1 Normalization

In the normalization process we need to find a transformation that better maps a facial feature into a given location determine by Fi into a 64×64 window. And Recognition which main objective is find the best match between a input image with a set normalize images. Before we can start recognition, images should be normalized to take into account scale, orientation and location variations. For the normalization step we have used an iterative scheme based on affine transformation. The transformation is needed to map specific facial features from a face image to predetermined locations in a fixed size window.





| Features | Coordinates |
|---|---|
| P1 | (13,20) |
| P2 | (50,20) |
| P3 | (34,34) |
| P4 | (16,50) |
| P5 | (48,50) |

Figure 1: The five facial features used

- Findtansformation.m this function compute the least square to find A and B

$$f_i^p = Af_i + b \tag{1}$$

predeterminde feature($f_i^p$) and $f_i$ are the 5 facial features of the input image. when the least square is compute we have 6 resulting values. the first correspond to A in order to have the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

- Applytransformation.m this function return $\overline{F}'.wegivethreeinputstothefunction : A, B, and\overline{F}$ and apply ($f_i^p$) to each feature. here we also get A and B for every images.

In the iteration process we perform the next sequence of steps:
1. Find the transformation matrix A and B for $\overline{F}$ using Findtransformation.
2. to find $\overline{F}$ to get $\overline{F}'$ using Applytransformation.m
3.At last, we get average set $\overline{F}$.

## 3.2 Recognition

We used PCA to transform our correlated data set into smaller uncorrelated data set. So PCA helped us on reducing the large dimensions of data set into smaller dimensions of it. After finding $\overline{F}$, we followed these steps:

We converted our 64 by 64 normalized images $X_i$ into row vector sized 1x4096 and constructed matrix $D_{pxd}$(p = number of images, d = 64x64) that each row corresponds to $X_i$. Then we defined k between 50 and 100 to $k$th largest eigen values.

Before computing the PCA, we computed the covariance matrix of D as follows:

$$\Sigma = \frac{1}{p-1} D^T D$$

Then we used matlab's "svd()" function to get eigenvectors from $\Sigma$ matrix. This process in our training is taking the most time. We put k eigenvectors corresponding to biggest k eigenvalues to $\Phi$.
We calculated PCA space by;

$$\phi_i = X_i.\Phi$$

PCA space represent the linear combination of eigenfaces of train images. Which means, for any train image, we just need know what is its linear combination of eigenfaces. After all, now we can store all our training images in the PCA space and easily search them to find closest matches.

# 4 Results

Too run this code. first you need to normalize the all data set which is button is provided into GUI you have to just need to change the path and click on normalization button and in few minute all image will be normalized after that you have to create the train and test data set from normalization images. after creating train and test data set you need to just click on Load button in GUI so all test images will load in list box after you need to click on Match button so you will find the match of the image.
We use No.4 and No.5 images of each person as test images and the rest are train images. For each test image, we calculate the distance of feature vector between test image and train images. Then we apply ascending sort on the train images by distance. Here We showed Input image form test images and output image from train images and we showed right and wrong results also.

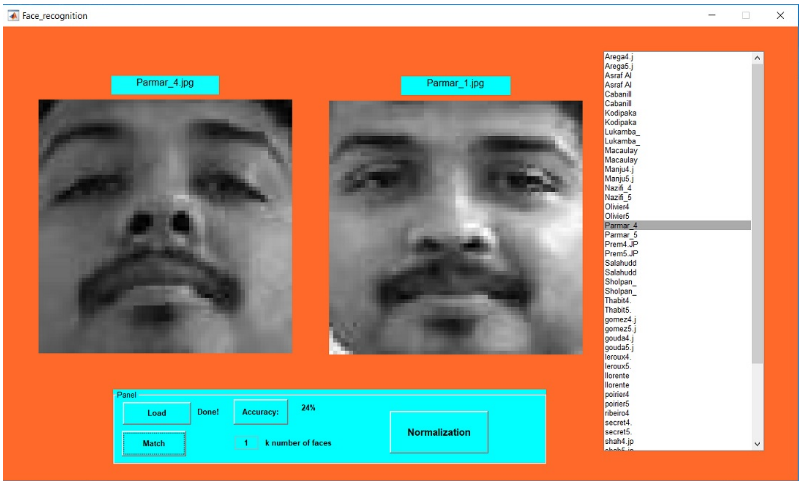


Figure 2: Figure 2

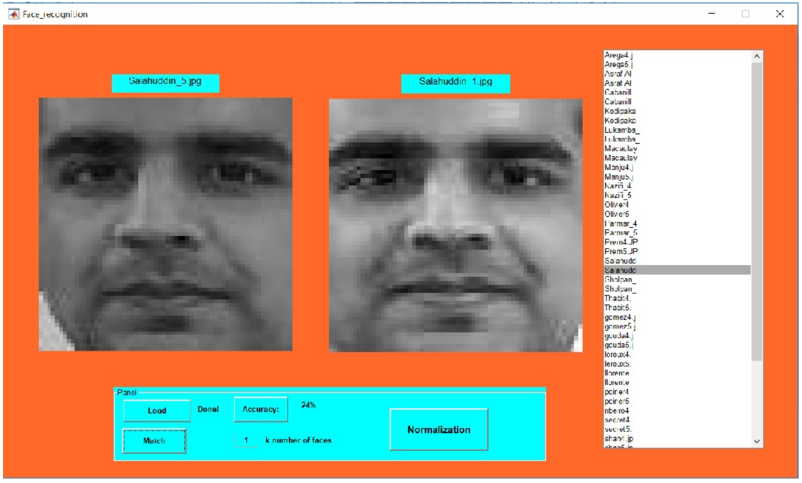


Figure 3: Figure 3

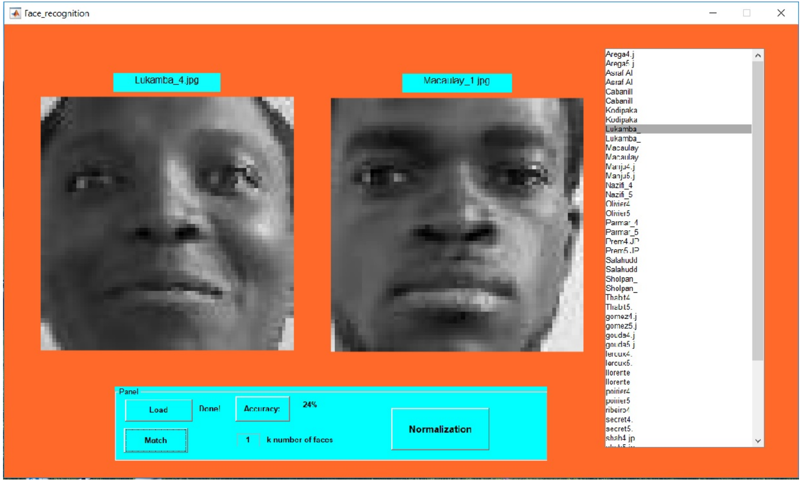


Figure 4: Figure 4

# 5 Summary

In the Summary i would like to say that we understand the concept of PCA provide: transform the data from correlated to independent.we understand the concept of SVD and normalization process too. PCA is kind of Fourier transform which present image as combination of some principal spectrum.face recognition represent a face by a combination of eigenfaces. and last not least it is powerful process to reduce the dimension of problem.