Face Recognition using PCA

Submitted by
Meldrick REIMMER
Selma BOUDISSA
Danie SONIZARA

Supervised by **Dr Desire Sidibe**

Contents

List of Figures

Chapter 1

Introduction

1.1 Background of Study

Face recognition is one of the most useful applications of Image Analysis and Computer Vision. Many face recognition techniques have been developed over the past few decades, and they can be divided into two groups based on how they represent faces [1]:

1. Appearance-based, which applies holistic texture features to either whole face or specific regions in face image.
2. Feature-based, which uses geometric facial features and geometric relationships.

Techniques which are appearance-based and are popularly used are Principal Component Analysis and Linear Discriminant Analysis. Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observtions of possibly correlated variables into a set of values of linearity independent variables called **Principal Components**.

PCA is a dimension reduction method which projects n dimensional data onto (k) dimensional subspace where (k) is smaller than n, and the (k) dimensional subspace is defined by leading eigenvectors of the data's covariance matrix. In 1991, Turk and Pentland achieved face recognition by projecting face images onto a feature space called ”eigenface” through PCA [2]. Linear Discriminant Analysis (LDA), on the other hand, is a supervised learning algorithm. The aim of LDA is to find a linear combination of features which separate different classes of objects. In other words, LDA is supposed to find a subspace on which objects in different classes are far away from each other while requiring objects in the same class are close to each other.

In this project we will be implementing just the PCA method. In the next chapter you will see the methodology in which was done for the PCA and the result attained and few observations made. We also got the accuracy of the system in which we developed.

1.2 Aim

To Develop a face recognition system using the PCA.

1.3 Objectives

In order to achieve our aim we undertook the following steps:

- Create a database of pictures (Pictures of us and our class colleagues with same dimensions)
- Normalize the images to 64x64. Divide the images into two parts and name them (Train images and Test images).
- Perform face recognition using the PCA algorithm.
- Show results
- Perform accuracy of the system.

Chapter 2

Methodology

To be able to achieve the objectives of this face recognition systems we need to go according to 3 main steps:

1. Normalisation
2. Decomposition of the normalized images
3. Matching faces

2.1 Normalization

Once the face has been detected in an image, the normalization step is required . The main idea is to use a transformation to map certain facial features from a face image to predetermined locations in a fixed size window. In this project we will be using the following facial features:

1. Left eye center
2. Right eye center
3. Tip noise
4. Left mouth corner
5. Right mouth corner

The normalized size window will be 64x64.

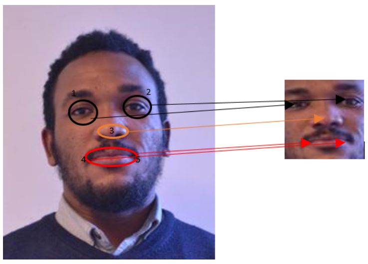

Figure 2.1: Process of normalization

Alogorithm Once we have extracted the locations of the facial features from all the database of images we need to compute the parameters of an affine transformation of the first image.

1. Initialize the vector F with the feature locations in the first image F1.
2. Compute the best transformation

3. The affine trasnformation is now $vecF = A * \boldsymbol{f} + b$
   The affine transformation can be defined by six parameters A = $\begin{bmatrix} a11 & a12 \\ a21 & a22 \end{bmatrix}$
   and b = $\begin{bmatrix} b1 \\ b2 \end{bmatrix}$.
   Since we have 10 equations, 2 for each features and 6 unknows we can used the SVD least mean square process to solve this over-determined system. So the new affina trasformation will be $F = f1 * Ab$

4. Computation of the best transformation on every images

5. Make the average of the aligned feature locations

2.2 Decompostion of the normalized image

Once we obtain the normalized image from the normalazation process we need to separate into two different folder.

2.2.1 Train set images

We will be using, in this folder, 3 different images for each person.

- One frontal view
- Two side views

We will need to comute the eigenfaces from the training set. If we have p training images, we can write our entire training data set as a pxd matrix D, where each row od D correspond to one image from the train set.

$$D = \begin{bmatrix} I1(1,1) & I1(1,2).. & I1(1,N)... & I1(M,1) & I1(M,2)... & I1(M,N) \\ I2(1,1) & I2(1,2).. & I2(1,N)... & I2(M,1) & I2(M,2)... & I2(M,N) \\ .. & & & & & \\ .. & & & & & \\ .. & & & & & \\ Ip(1,1) & Ip(1,2).. & Ip(1,N)... & Ip(M,1) & Ip(M,2)... & Ip(M,N) \end{bmatrix}$$

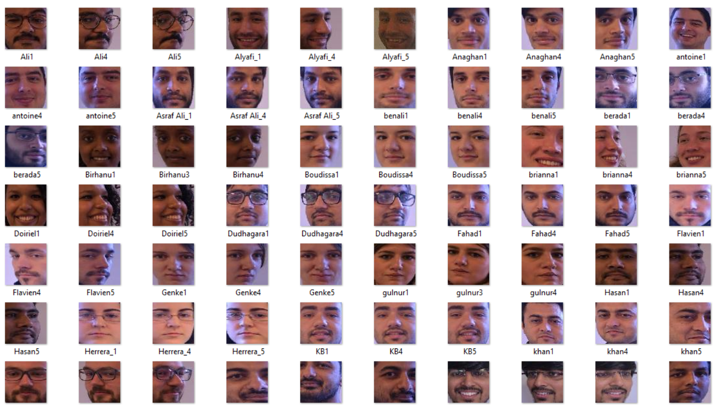

Figure 2.2: Train images folder

2.2.2 Test set images

For this folder we will be using the 2 images left per person.

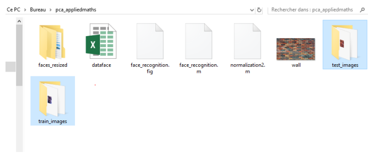

Figure 2.3: Folder with the face images

The data as been divided into two different folder

- train_images
- test_images

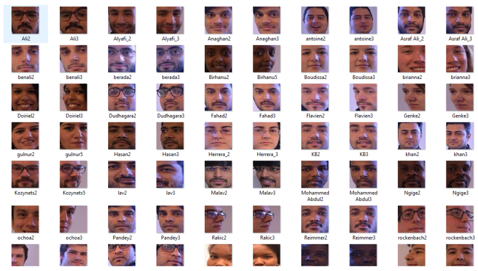

Figure 2.4: Test images folder

2.3 Face Recognition using PCA

Let X = $\{x_1, x_2, ..., x_n\}$be the matrix containing face images. Note that each image matrix has been converted into a vector. For example, a m*n matrix is converted into a vector with m*n rows. The subspace "eigenfaces" could be obtained by following the below steps:

1. Computer the mean μ
2. Compute the Covariance Matrix
3. Compute the eigenvectors and eigenvalues
4. Order the eigenvectors descending by their eigenvalues and select eigenvectors corresponding to largest eigenvalues

The above eigenvectors form the subspace W, and it is called as "eigenfaces". There is still a computational problem left. Each image is 320 x 240 meaning that there are 76800 dimensions. Once such faces are applied to calculate the covariance matrix , will end up with a 76800 x 76800 matrix which is really huge and almost could not be processed on normal laptops. Here we first define Y:

$$Y = X - U$$

where U =$\{\mu,\mu,...\mu\}$. By this way, the original data set is transformed into a data set with zero mean. In order to solve the problem just mentioned, the eigenvectors and eigenvalues of $Y^T Y$ is first calculated.

The last step is to normalize the eigenvectors just calculated. Finally, the (k) eigenvectors corresponding to k largest eigenvalues are retrieved and form the subspace W. Also, the column vectors of W are the so-called eigenfaces. Normally, the number of training samples is far less than the dimensions of each face. As a result, the above method is needed to avoid the computational problem mentioned before.

Chapter 3

Graphical User Interface

We implemented a GUI interface to provide a better way of showing the results attained after the implementation of the methods. Our GUI enables the you to be able to perform tasks in achieving the results needed. Below is the interface we developed.

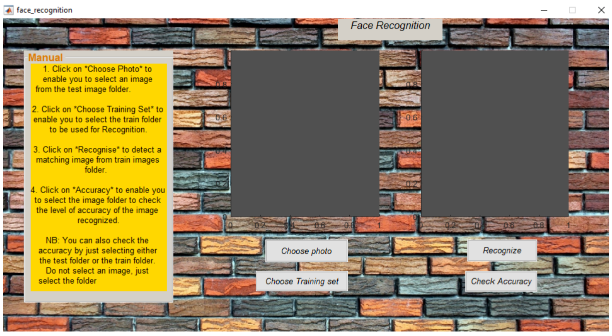

Figure 3.1: Graphical User Interface

The GUI contains four buttons, two axes for the outputs and and a panel. The 'Choose a photo' button enables you to load a picture from the images folder in which you will like to detect. The results is then shown on the left axes. The 'Choose training Set' button enables you to be able to select the train folder in which contains a set of images which will be used for the recognition process. The 'Recognize' button performs the recognition and outputs the results unto the second axes on the right. The 'Accuracy' button enables you to check the accuracy of the system in which the final out detected is true and comparable to the initial image. This is display on a pop-up message box on the GUI.

Chapter 4

Results

4.1 Recognition test

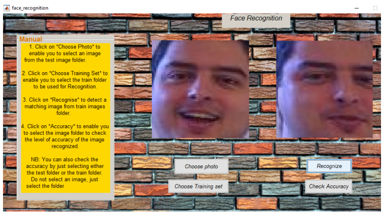

Figure 4.1: First Test

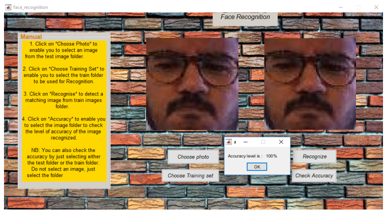

Figure 4.2: Second test

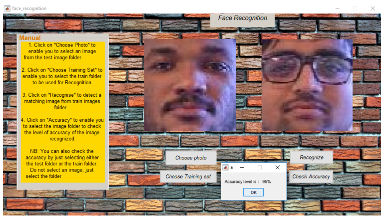

Figure 4.3: Third Test

Figure 4.4: Fourth test

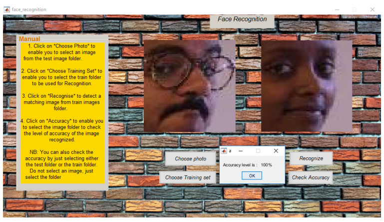

Figure 4.5: Fifth Test

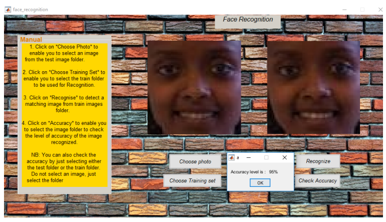

Figure 4.6: Sixth test

Explanation: We can see from the various tests made that we were able to detect images based on corresponding features. Some particular images had the similar features with other images and the accuracy found was 80%. This could be due to the face that they made similar gestures or they had the same dimensions. Other tests showed great results by identifying similar image and gave an accuracy of 100%. We realized that the recognition was based on similar gestures. This could be the smile made or the raising of the eye browse and could also be the wideness of the face. There could be many other reasons but in this case we know this was done due to the features corresponding to the sample images given. The level of accuracy given depends on the total number of database in which we have. Less database gives gives low accuracy level but the more the images in the test folder the higher that accuracy level.

Chapter 5

Conculsion and Perspectives

Face recognition is a big artifact in the technology world today. In as much as it has it great achievements it sometimes fails to perform is tasks. It is not able to give 100% results. It does this once in many. We saw it in our results, where it rather recognize people who had similar face gestures.

Performing this face recognition system help us to put in practice what we learnt in theory in our applied mathematic class. We have been using the SVD least mean square method and also the normalization process. Also we now understand why all this knowledge is important in different applications of nowdays life, for example video security, biometric passeport, enhance security measure to recognize a criminal catch in a video or images...

We could improve the face recognition system by also adding LDA and LPP approach in conjunction with the PCA. This we sure could give reasonable and satisfying results. We will pursue this and try to do implement this in our future studies.

Chapter 6

References

- **[1] Delac, K., Grgic, M., & Grgic, S. (2005). Independent comparative study of PCA, ICA, and LDA on the FERET data set. International Journal of Imaging Systems and Technology, 15(5), 252-260.**

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

- **Face Recognition(introduction part)**
  `https://en.wikipedia.org/wiki/Facial_recognition_system`

- **Face recognition article by David Guo**
  `http://www.sciencedirect.com/science/article/pii/S2212671612000431`