Littoral Côte d'Opale University
Engineering School of ULCO

Final Year Thesis Project

Face Recognition Using Principal Component Analysis (PCA)

Author:
Oumaima Aabi

Supervisor:
Pr. Khalide Jbilou

Submitted in fulfillment of the requirements
for the degree of First year of master's degree
Engineering of Complex Systems
(MISC M1)

in the Department of Mathematics

June 10, 2021

Acknowledgements

All praise is due to Allah, Lord of the worlds. Peace and blessings be upon his Prophet Muhammad, blessings of Allah be upon him and his family and peace.

I would like to thank my parents for everything they did for me, for teaching me morals, for supporting me for my studies, for being on my side during happy moments and during difficult moments.

I would like to express my sincere gratitude and thanks to our supervisor Professor Khlide Jbilou for his constant guidance, encouragement and support throughout the course of this project.I was honored to be his student and have him supervise me for the final year project.

I sincerely thank the members of the Jury. I'm honored by their presence in the Jury.

I also express my sincere gratitude to all the Professors who helped me and encouraged me during this year in the engineering school of Engineering School of Littoral Côte d'Opale University.

Abstract

Face recognition has been a fast growing, challenging and interesting area in real-time applications. There are lots of algorithms effective at performing face recognition, such as for instance: Principal Component Analysis, Discrete Cosine Transform, 3D acceptance methods, Gabor Wavelets method etc.
This work has centered on Principal Component Analysis (PCA) method for face recognition in an efficient manner. We will approach the subject in the following way:

1. In Chapter 1, we will present a Literature review of Principal Component Analysis (PCA) method. We will see the relation between Principal Component Analysis (PCA) and the machine learning and specifically dimensionality Reduction.

2. In Chapter 2, we will study the mathematics behind the Principal Component Analysis (PCA) method. We will go through from the trainig phase to calculation and substraction the average face until the classification and identification of face.

3. In Chapter 3, we will jump to the implementation and some results of the training phase.

4. In Chapter 4, we will discover some of the challenges that limit the potential of a facial recognition system to go the extra mile.

5. In Chapter 5, we will end the thesis with a conclusion.

The document was typeset in LaTeX. In this project we will be using a dataset named **The Olivetti faces** dataset, provided by my supervisor. The system has been implemented by MATLAB R2017a, and the MATLAB source codes are attached to the appendix in the end of this report.
Please don't hesitate to contact me for any remark or error.
My e-mail: oumaima.aabi@etu.univ-littoral.fr
My advisor's e-mail: khalide.jbilou@univ-littoral.fr

Keywords: Dimensionality reduction, Principal Component Analysis (PCA), face recog-nition, Eigenfaces, and Eigenvectors

Contents

List of Figures

To my parents Latifa and Abdelkarim, my brother Ayoub, and my sisters Jihane, Ouassima and Ichraq.

Chapter 1

Literature review

1.1 Introduction

1.1.1 Biometrics

Biometrics is used in the process of authentication of a person by verifying or identifying that a user requesting a network resource is who he, she, or it claims to be, and vice versa. It uses the property that a human trait associated with a person itself like structure of finger, face details etc. By comparing the existing data with the incoming data we can verify the identity of a particular person.
There are many types of biometric system like *finger print recognition*, *face detection* and *recognition*, *iris recognition* etc., these traits are used for human identification in surveillance system, criminal identification.
Advantages of using these traits for identification are that they cannot be forgotten or lost. These are unique features of a human being which is being used widely.

1.1.2 Face Recognition

Face is a complex multidimensional structure and needs good computing techniques for recognition. The face is our primary and first focus of attention in social life playing an important role in identity of individual. We can recognize a number of faces learned throughout our lifespan and identify that faces at a glance even after years. There may be variations in faces due to aging and distractions like beard, glasses or change of hairstyles.

Face recognition is an integral part of biometrics. In biometrics basic traits of human is matched to the existing data and depending on result of matching identification of a human being is traced. Facial features are extracted and implemented through algorithms which are efficient and some modifications are done to improve the existing algorithm models.

Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification, security systems, identity verification etc. Face detection and recognition is used in many places nowadays, in websites hosting images and social networking sites. Face recognition and detection can be achieved using technologies related to computer science. Features extracted from a face are processed and compared with similarly processed faces present in the database. If a face is recognized it is known or the system may show a similar face existing in database else it is unknown. In surveillance system if a unknown face appears more than one time then it is stored in database for further recognition. These steps are very useful in criminal identification. In general, face recognition techniques can be divided into two groups based on the face representation they use appearance-based, which uses holistic texture features and is applied to either whole-face or specific regions in a face image and feature-based, which uses geometric facial features (mouth, eyes, brows, cheeks etc), and geometric relationships between them.

1.2 Principal Component Analysis

Principal component analysis (PCA) was invented in 1901 by *Karl Pearson*. PCA is a variable reduction procedure and useful when obtained data have some redundancy. This will result into reduction of variables into smaller number of variables which are called Principal Components which will account for the most of the variance in the observed variable.

PCA is a most widely used tool in exploratory data analysis and in machine learning for predictive models. Moreover, PCA is an unsupervised statistical technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit.

1.2.1 Dimensionality Reduction

Dimensionality reduction is a type of unsupervised learning where we want to take higherdimensional data, like images, and represent them in a lower-dimensional space. In general there's no general approaches for dimensionality reduction feature selection and feature extraction.

1. **Feature selection:** also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of the existing features without a transformation.

2. **Feature extraction:** is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. Feature extraction is transforming the existing features into a lower dimensional space.

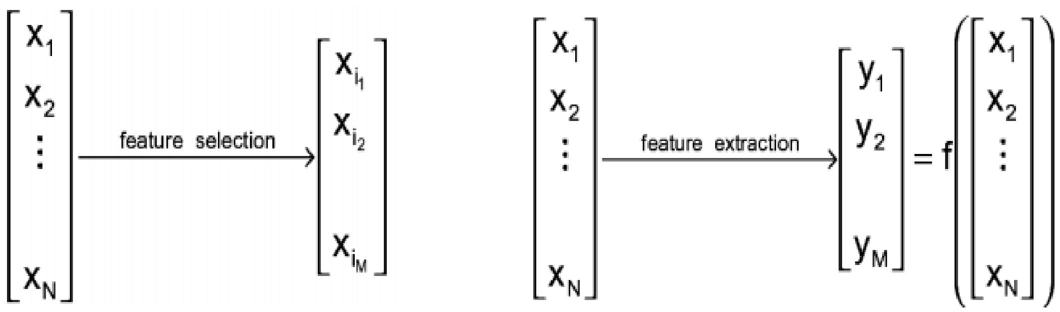

FIGURE 1.1: Selection vs. Extraction

Remark: Principle Component Analysis (PCA) is a common feature extraction method in data science.

The major advantage of PCA is using it in eigenface approach which helps in reducing the size of the database for recognition of a test images. The images ar stored as their feature vectors in the database which are found out projecting each and every trained image to the set of Eigen faces obtained. PCA is applied on Eigen face approach to reduce the dimensionality of a large data set.

1.3 Eigen Face Approach

It is adequate and efficient method to be used in face recognition due to its simplicity, speed and learning capability. Eigen faces are a set of Eigen vectors used in the Computer Vision problem of human face recognition. They refer to an appearance based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner.

The Eigen faces are Principal Components of a distribution of faces, or equivalently, the Eigen vectors of the covariance matrix of the set of the face images, where an image with N by N pixels is considered a point in N 2 dimensional space. Previous work on face recognition ignored the issue of face stimulus, assuming that predefined measurement were relevant and sufficient. This suggests that coding and decoding of face images may give information of face images emphasizing the significance of features.

These features may or may not be related to facial features such as eyes, nose, lips and hairs. We want to extract the relevant information in a face image, encode it efficiently and compare one face encoding with a database of faces encoded similarly. A simple approach to extracting the information content in an image of a face is to somehow capture the variation in a collection of face images.

We wish to find Principal Components of the distribution of faces, or the Eigen vectors of the covariance matrix of the set of face images. Each image location contributes to each Eigen vector, so that we can display the Eigen vector as a sort of face. Each face image can be represented exactly in terms of linear combination of the Eigen faces. The number of possible Eigen faces is equal to the number of face image in the training set. The faces can also be approximated by using best Eigen face, those that have the largest Eigen values, and which therefore account for most variance between the set of face images. The primary reason for using fewer Eigen faces is computational efficiency.

Chapter 2

Mathematics of Principle Component Analysis (PCA)

2.1 The training phase

Like any algorithm of recognition, the process of the Principle Component Analysis (PCA) is divided into two distinct phases; a phase of training and a phase of recognition (2.1).

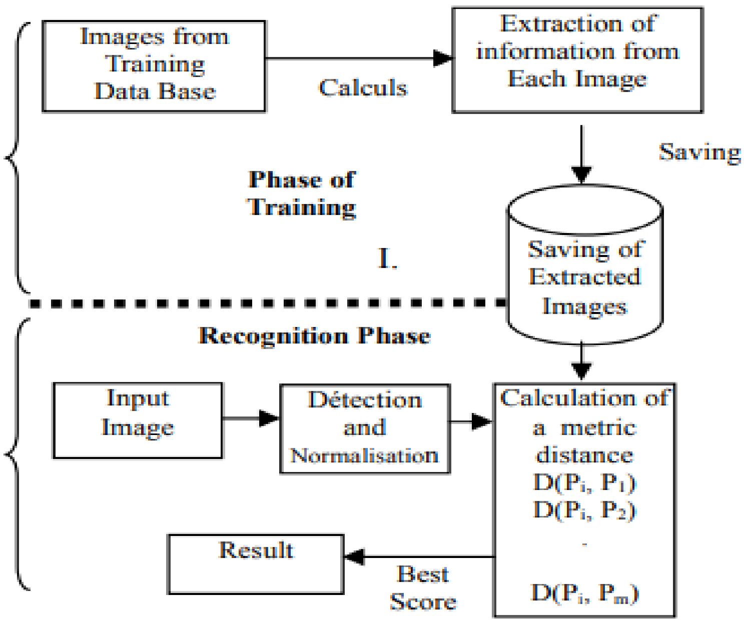

FIGURE 2.1: Face recognition process using PCA

2.1.1 Database

In this phase, suppose that the library of images (i.e. set of face images to which an input image will be attempted to be matched) consists of n numbers of images $[I_1, I_2, ..., I_n]$, and that each image is represented as an $M \times N$ matrix of pixels values.
The first step is to transform the M images into column vectors of length $(1\text{x}MN)$. Consider an arbitrary image from the total n-sized collection, call it I_1:

$$I_1 = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MN} \end{pmatrix}_{M \times N}$$

Now we wish to transform I1 into a column vector, call it Γ_1. This will be achieved by concatenating the columns of I_1:

$$\Gamma_1 = \begin{pmatrix} p_{11} \\ p_{21} \\ \vdots \\ p_{i1} \\ p_{12} \\ p_{22} \\ \vdots \\ p_{i2} \\ p_{13} \\ \vdots \\ p_{ij} \end{pmatrix}_{MN \times 1}$$

Example: An image is simply a matrix of size $N \times N$, that gonna be represented by vector of size N^2.

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

Each face image is represented by the vector Γ_i.

$$\Gamma_1 = \begin{bmatrix} 1 \\ -2 \\ 1 \\ -3 \end{bmatrix}, \Gamma_2 = \begin{bmatrix} 1 \\ 3 \\ -1 \\ 2 \end{bmatrix}, \Gamma_2 = \begin{bmatrix} 2 \\ 1 \\ -2 \\ 3 \end{bmatrix}, \ldots, \Gamma_M \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

After performing this transformation on all n images, we will obtain the following set S:

$$S = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$$

2.1.2 Calculation and subtraction of the average face

Since we are not interested in the commonalities between our n images, we would like to subtract the mean image, call it Ψ, from each Γ_x where $x \in \mathbb{N}, 1 \leq x \leq n$. We will computed Ψ as follows:

$$\Psi = \frac{1}{n} \sum_{i=1}^{n} \Gamma_i$$

Now taking the difference of each Γ_x, $x \in \mathbb{N}, 1 \leq x \leq n$, and Ψ, we obtain a new set of vectors that we can represent as matrix A:

$$A = [\Phi_1, \Phi_2, \ldots, \Phi_n]_{MN \times n}$$

where $\Phi_i = \Gamma_i - \Psi_i$, $i \in \mathbb{N}, 1 \leq i \leq n$.

Example:
Average face image is calculated by :

$$\begin{bmatrix} 1 \\ -2 \\ 1 \\ -3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \\ -1 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ -2 \\ 3 \end{bmatrix} + \ldots + \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} -1 \\ -1 \\ 2 \\ -3 \end{bmatrix}$$

$$\Psi = (\Gamma_1 + \Gamma_2 + \Gamma_3 + \ldots + \Gamma_M) \,/ n$$

Each face differs from the average by $\Phi_i = \Gamma_i - \Psi$ which is called mean centered image.

$$\Phi_1 = \begin{bmatrix} 2 \\ -1 \\ -1 \\ 0 \end{bmatrix}, \Phi_2 = \begin{bmatrix} 2 \\ 4 \\ -3 \\ 5 \end{bmatrix}, \Phi_3 = \begin{bmatrix} 3 \\ 2 \\ -4 \\ 6 \end{bmatrix}, \ldots, \Phi_M = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 4 \end{bmatrix}$$

2.1.3 The Covariance matrix

We now have a collection of data that will form our training set or library, and it defines the face space. This space is currently defined in $M \times N$ space, but we would like to reduce the dimensionality while preserving information of interest (the variance). To that end, we proceed by constructing the covariance matrix using a maximum-likelihood estimator for a population covariance matrix, call it C, of the random vectors Γ_x, $x \in \mathbb{N}$, $1 \leq x \leq n$:

$$C = \sum_{i=1}^{n} \Phi_i \Phi_i^T, \text{ an } MN \times MN \text{ matrix.}$$

Now we would like to show that $C = AA^T$. We proceed directly and observe that:

$$C = AA^T$$
$$\Longleftrightarrow \textstyle\sum_{i=1}^{n} \Phi_i \Phi_i^T = AA^T$$

$$\Longleftrightarrow \sum_{i=1}^{n} \Phi_i \Phi_i = \begin{pmatrix} \Phi_1 & \Phi_2 & \ldots & \Phi_n \end{pmatrix}_{MN \times n} \begin{pmatrix} \Phi_1^T \\ \Phi_2^T \\ \vdots \\ \Phi_n^T \end{pmatrix}_{n \times MN}$$

$$\Longleftrightarrow \sum_{i=1}^{n} \begin{pmatrix} \Phi_{i1} \\ \Phi_{i2} \\ \vdots \\ \Phi_{iMN} \end{pmatrix}_{MN \times 1} (\Phi_{i1} \quad \Phi_{i2} \quad \ldots \quad \Phi_{iMN})_{1 \times MN}$$

$$= \begin{pmatrix} \Phi_{11} & \Phi_{21} & \dots & \Phi_{n1} \\ \Phi_{12} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \Phi_{1MN} & \dots & \dots & \Phi_{nMN} \end{pmatrix}_{MN \times n} \begin{pmatrix} \Phi_{11} & \Phi_{12} & \dots & \Phi_{1MN} \\ \Phi_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \Phi_{MN1} & \dots & \dots & \Phi_{nMN} \end{pmatrix}_{n \times MN}$$

$$= \begin{pmatrix} \Phi_{11}^2 + \ldots + \Phi_{n1}^2 & \Phi_{11}\Phi_{12} + \ldots + \Phi_{n1}\Phi_{n2} & \dots & \Phi_{11}\Phi_{1n} + \ldots + \Phi_{n1}\Phi_{nMN} \\ \Phi_{12}\Phi_{11} + \ldots + \Phi_{n2}\Phi_{n1} & \ddots & \vdots & \\ \vdots & & \ddots & \vdots \\ \Phi_{1n}\Phi_{11} + \ldots + \Phi_{nMN}\phi_{n1} & \dots & \dots & \Phi_{1MN}^2 + \ldots + \Phi_{nMN}^2 \end{pmatrix}_{MN \times MN}$$

$$= \begin{pmatrix} \sum_{i=1}^n \Phi_{i1}^2 & \sum_{i=1}^n \Phi_{i1}\Phi_{i2} & \dots & \sum_{i=1}^n \Phi_{i1}\Phi_{iMN} \\ \sum_{i=1}^n \Phi_{i2}\Phi_{i1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \sum_{i=1}^n \Phi_{iMN}\Phi_{i1} & \dots & \dots & \sum_{i=1}^n \Phi_{iMN}^2 \end{pmatrix}_{MN \times MN}$$

$$= \sum_{i=1}^{n} \begin{pmatrix} \Phi_{i1} \\ \Phi_{i2} \\ \vdots \\ \Phi_{iMN} \end{pmatrix}_{MN \times 1} \begin{pmatrix} \Phi_{i1} & \Phi_{i2} & \dots & \Phi_{iMN} \end{pmatrix}_{1 \times MN}$$

where $\Phi_{ij} = \Gamma_{ij} - \Psi_j$.

Example:
Size of covariance matrix will be $N^2 \times N^2$ (4×4 in this case of our example).

$$A = \begin{bmatrix} 2 & 3 \\ -1 & -2 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}, A^T = \begin{bmatrix} 2 & -1 & -1 & 0 \\ 3 & -2 & 1 & 2 \end{bmatrix}$$

2.1.4 Calculation of the eigenvectors and eigenvalues

To find the principal components of this data set, we would now like to compute the eigenvectors and corresponding eigenvalues of this covariance matrix.
Recall that finding the largest eigenvalues and associated eigenvectors will provide a lower dimensionality space that can describe the variation between the training set images.
We are trying to find the principle components of the face space. But computing the eigenvectors of $C = AA'$ is a difficult task (C is an $MN \times MN$ real symmetric matrix). Recall that each face can be represented by a linear combination of all the eigenvectors of C. As long as enough information is preserved, we should be able to approximate each face (i.e. use only the "best" eigenvectors, or those with the greatest corresponding eigenvalues and thus account for the most variance within our set of images).

To go through this problem we propose the following solution. For an eigenvector v_i associated to an eigenvalue λ_i we have:

$$Cv_i = \lambda_i v_i$$

The matrix C has the form AA^T. Let us consider the matrix $L = A^T A$ having the eigenvectors u_i associated to eigenvalues e_i:

$$Lu_i = e_i u_i$$

Let,

$$A^T A u_i = e_i u_i$$

By multiplying by A the left of the two sides of the equality, we obtain:

$$AA^T A u_i = A e_i u_i$$

And since $C = AA^T$ we can simplify:

$$CAu_i = Ae_i u_i$$

$$C(Au_i) = e_i(Au_i)$$

According to the definition of the eigenvectors and eigenvalues of the matrix C, we have:

$$v_i = Au_i \ , \ \lambda_i = e_i$$

The matrix L is a matrix (MxM), the calculation of its eigenvectors and eigenvalues is much easier than with the matrix C. So we can find the eigenvalues of this huge matrix C by finding the eigenvalues of a much smaller matrix L. To find the eigenvectors of C, we just have to pre-multiply the eigenvectors of L by the matrix A.

Example:
Eigen vectors corresponding to this covariance matrix is needed to be calculated, but that will be a tedious task therefore.
For simplicity we calculate $A^T A$ which would be a 2×2 matrix in this case.

$$A^T A = \begin{bmatrix} 6 & 7 \\ 7 & 18 \end{bmatrix}$$

Consider the eigenvectors viof $A^T A$ such that $A^T AX_i = \lambda_i X_i$.
The eigenvectors v_i of $A^T A$ are X_1 and X_2 which are (2×1). Now multiplying the above equation with A both sides we get :

$$AA^T AX_i = A\lambda_i X_i$$
$$AA^T \left(AX_i\right) = \lambda_i \left(AX_i\right)$$

Eigen vectors corresponding to AA^T can now be easily calculated now with reduced dimensionality where AX_i is the Eigen vector and λ_i is the Eigen value.

2.1.5 Projection onto Eigenface Space

Then comes the stage of selection of the eigenvectors.
The eigenvectors found are then ordered according to their corresponding eigenvalues, in decreasing order. The larger an eigenvalue is, the more variance is captured by the eigenvector. variance captured by the eigenvector is.
This implies that most of the information is contained in the first eigenvectors. Part of the great efficiency of the PCA algorithm comes from the next step which consists in selecting only the k best eigenvectors (those with the k largest eigenvalues). From this, we define a vector space generated by these k eigenvectors, which we call the *eigenfaces*.
The original images can be reconstructed by linear combination of these eigenvectors. The graphical representations of these vectors are a bit like ghost images, each one highlighting a part of the face, we call them Eigenfaces (figure 2.2). We are now going to project our starting images on Ev. An image i is then transformed into its Eigenfaces components by a simple vector projection operation:

$$\Omega_k = v_k^T(\Gamma_k - \Psi) \quad ; \quad (K = 1, ..., M') \qquad \textbf{(1)}$$

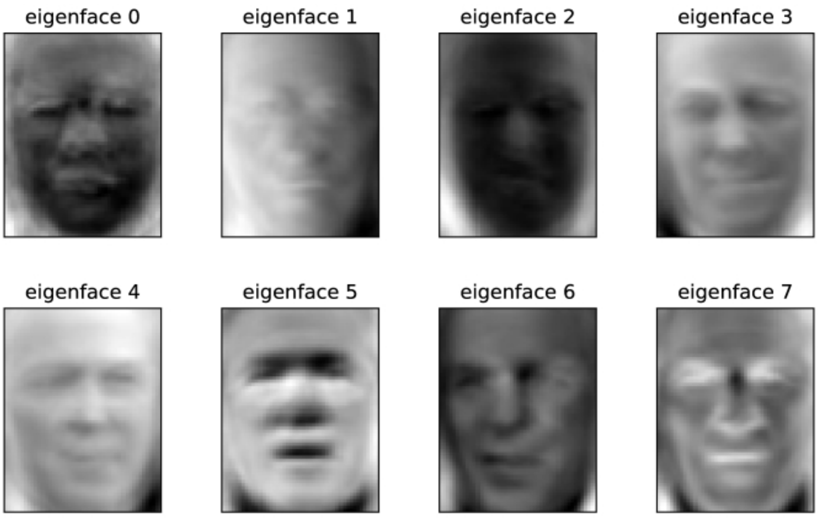

FIGURE 2.2: Average image and the first 8 eigenfaces

The W_k are called weights and form a vector Ω^T:

$$\Omega^T = [\Omega_1, \Omega_2, \Omega_3, ..., \Omega_{M'}]$$

Vectors Ω^T are saved in order to be used to classify a new image in the phase of recognition.

2.1.6 Classification and identification of face

The process of assignment of a new image i to a class resulting from the training phase proceeds on two stages: First, the image i is transformed into its components eigenfaces according to the formula **(1)**.

$$\Omega^T = [\Omega_1, \Omega_2, \Omega_3, ..., \Omega_{M'}]$$

Then the vector is used to find which of a number of predefined face classes, if any, best describes the face. The simplest method for determining which face class provides the best description of an input face image is to find the face class k that minimizes the *Euclidean distance*.

$$\epsilon_k = \|\Omega - \Omega_k\|$$

, where Ω_k is a vector describing the k_{th} face class.
A face is classified as belonging to class k when the minimum ϵ_k is below some chosen threshold θ Otherwise the face is classified as "unknown". The distance threshold, θ, is *half the largest distance between any two face images*, Mathematically can be expressed as:

$$\theta = \frac{1}{2} max_{j,k} \|\Omega - \Omega_k\|, \; where \; j,k = 1, ..., M'$$

Test: let ϵ be the distance between the original test image Γ and it's reconstructed image from the Eigenface:

$$\epsilon = \|\Omega - \Omega_k\|, \; where \; \Gamma_k = v_k^T \Omega_k + \Psi$$

Recognition process can formulated as :

- If $\epsilon \geq \theta$, then the input image is not even a face image and not recognized.

- If $\epsilon \leq \theta$ and $\epsilon_k \geq \theta$ for all k then the input image is a face image but it is an unknown image face.

- If $\epsilon \geq \theta$ and $\epsilon \leq \theta$ for all k then the input images are the individual face images associated with the class vector Γ_k

Figure 2.3 illustrates the projection and recognition by visualizing face space as a plane. In this case, there are two eigenfaces (u_1, u_2) and three known individuals ($\Omega_1, \Omega_2, \Omega$).

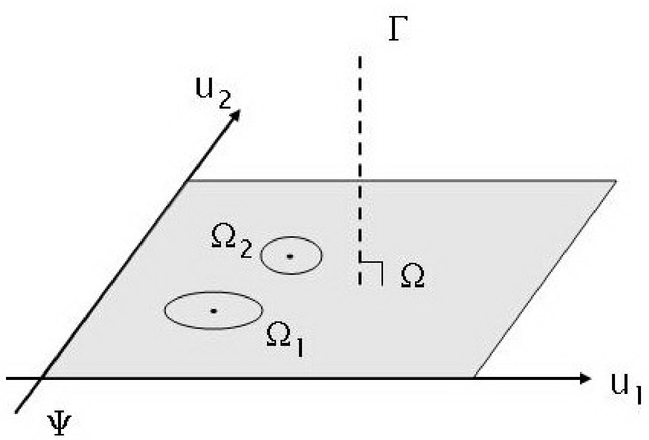

FIGURE 2.3: Visualization of a 2D face space, with the axes representing two Eigenfaces.

Chapter 3

Implementation and Results

We will be using a dataset named **The Olivetti** faces dataset. This dataset contains a set of face images taken between April 1992 and April 1994 at *ATT Laboratories Cambridge*.

Brief information about Olivetti Dataset:

- There are ten different image of each of 40 distinct people.
- There are 400 face images in the dataset
- Face images were taken at different times, variying ligthing, facial express and facial detail
- All face images have black background
- The images are gray level
- Size of each image is 64x64
- Image pixel values were scaled to [0, 1] interval
- Names of 40 people were encoded to an integer from 1 to 40.

This dataset will be our main reference for the rest of this study. Testing and Training subsets will be made from it.

3.1 The process of the recognition

The system has been implemented by MATLAB. We seperate the database Olivetti into two folders containing images for training (**Train**) and testing (**Test**), all images must be named as *1.png, 2.png, 3.png, ..., 400.png* in **Train** folder.
Please note that it's important to create MATLAB script files for training and testing and save them in Train folder, and set the path of Train folder as MATLAB's current directory for running the program.

When displaying the 10 pictures of each folder, we can see that the 10 images contains different facial expressions and lighting of the subjects, and we get:

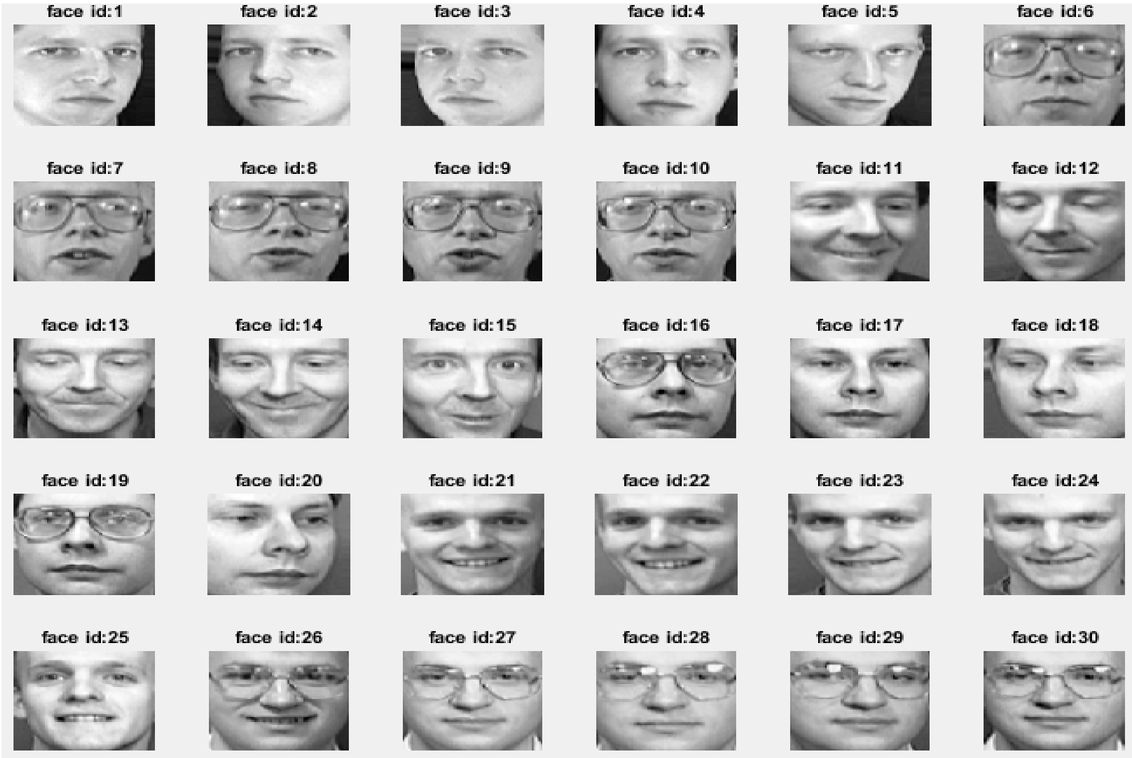

FIGURE 3.1: Original training images

3.1.1 Principle Component Analysis

The below illustration 3.2 shows a simple example on a synthetic two-dimensional data set. This drawing shows the original data points colored to distinguish points.The algorithm first proceeds by finding the direction of the maximum variance labeled "Component 1".

This refers to the direction in which most of the data is associated, or in other words, the properties that are most related to each other.

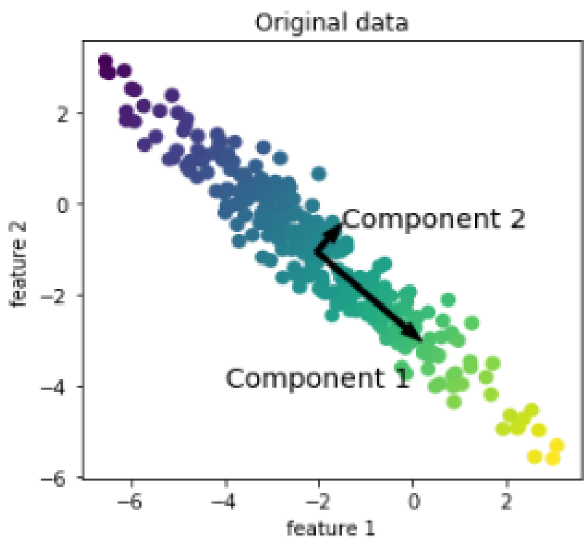

FIGURE 3.2

3.1.2 Finding Optimum Number of Principle Component

In the figure below 3.3, it can be seen that 90 and more PCA components represent the same data. Now let's make the classification process using 90 PCA components.

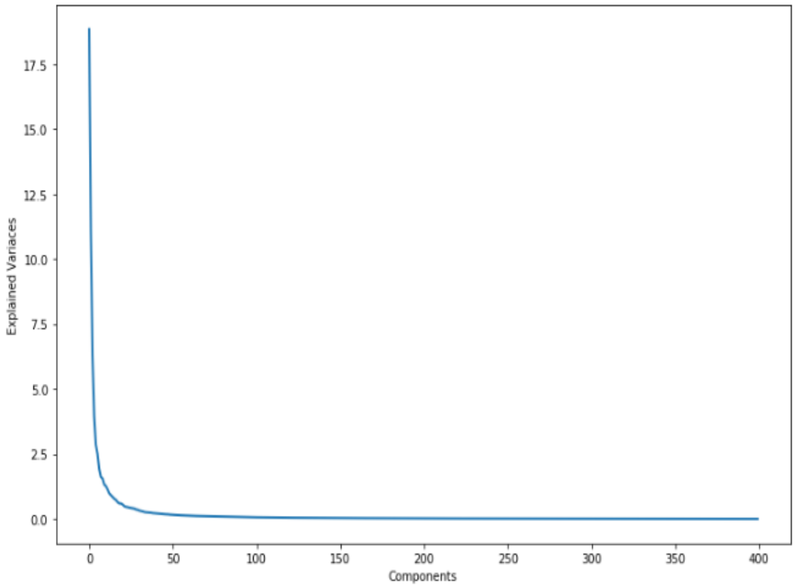

FIGURE 3.3: Explained variance

3.1.3 The average face

After computing the average face Ψ, it will appears like that :

FIGURE 3.4: The average face

3.2 Eigenvalues

Example of images from the training base is given in shows all 50 eigenvalues. Each eigenvalue corresponds to a single eigenvector and tells us how much images from training bases vary from the mean image in that direction.
It can be seen that about 10 of vectors have significant eigenvalues, while those for the remaining vectors are approximately equal to zero.

We do not have to take into account eigenvectors that correspond to small eigenvalues because they do not carry important information about the image.

On the figure 3.5, we showed the 16 first EigenVectors.

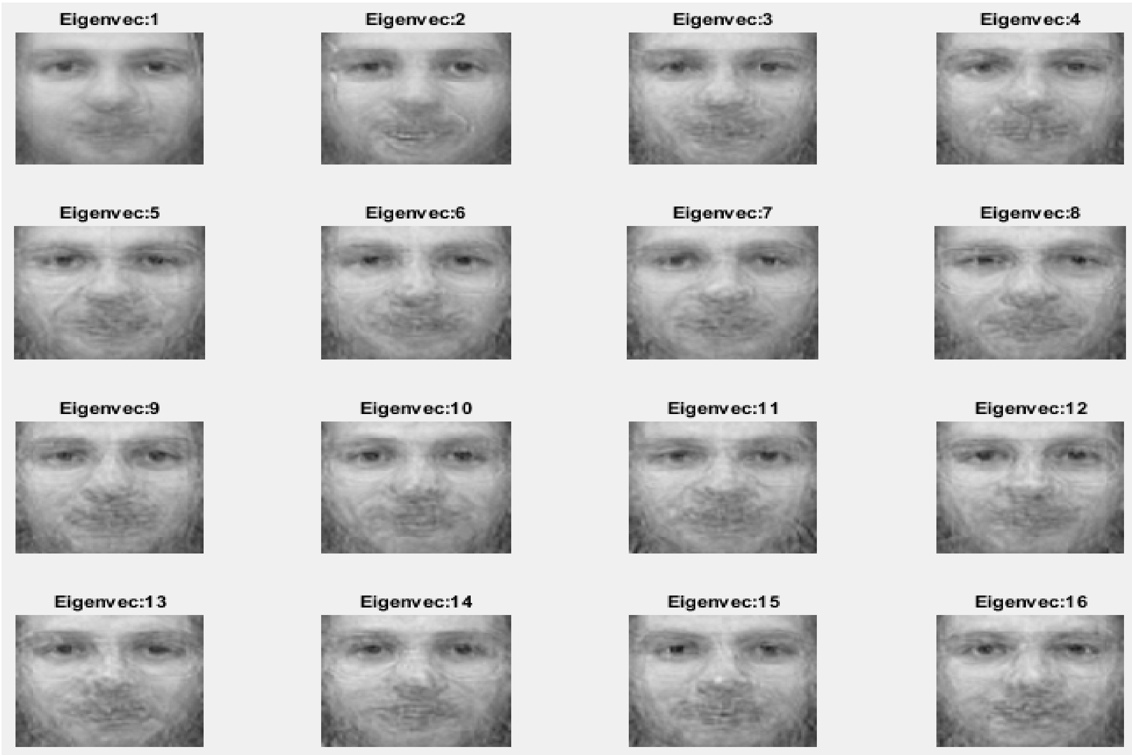

FIGURE 3.5: Eigenvalues

3.3 Testing phase

After we developing our program let's test it to see if it will recognize the selected image or not.

1. First we select an image as the test image, let's choose *image-10.png*. The output is in the figure 3.6.

   As you see, the goal has been reached successfully, it's the same person, but not the same test image.

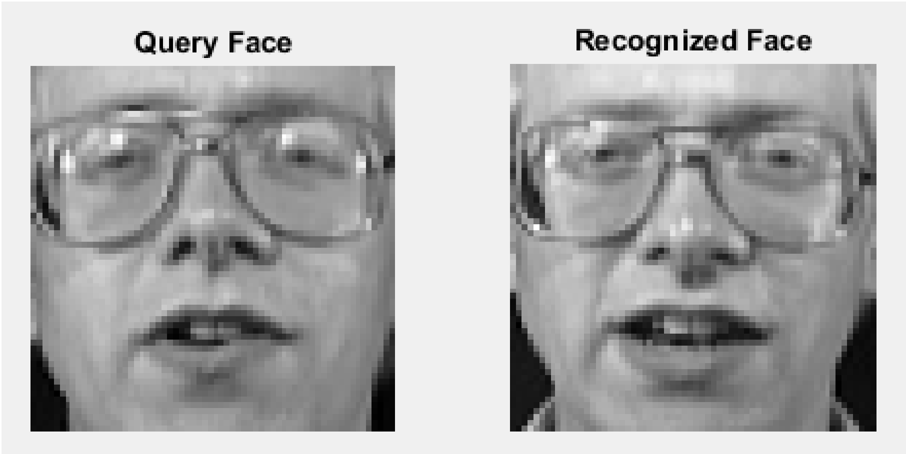

FIGURE 3.6: The first test

2. Let us try another image of another person. This time, we select **image-69.png** image and see what will be the result.

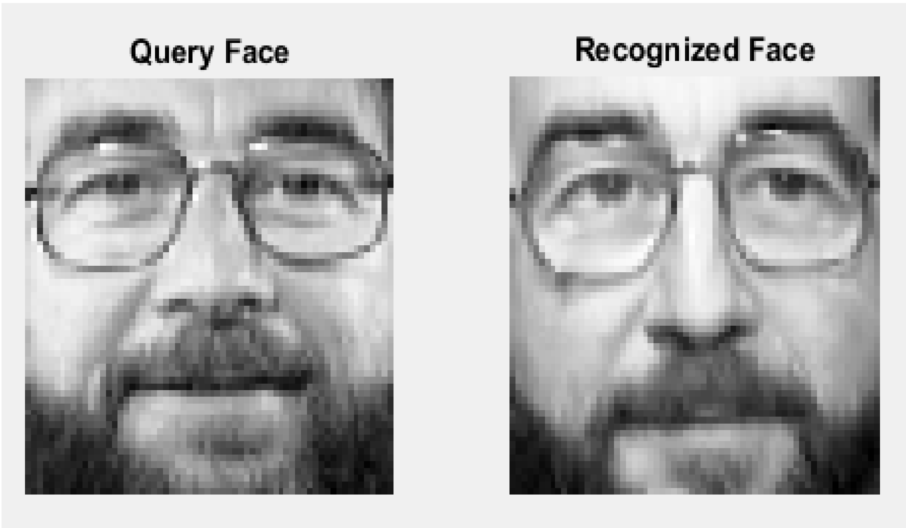

FIGURE 3.7: The second test

The program still get the correct recognition, as you can see it's the same person even with some distortion on the original image.

3. But what will happen if we choose an image does not exist in our training set? In this case, an alert will appear with the problem founded.

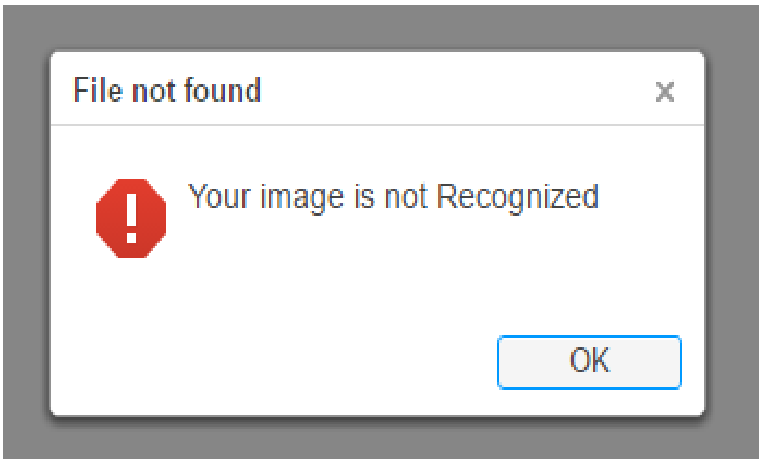

FIGURE 3.8: Image not recognized!

3.4 Result and Analysis

We have succeeded in developing a face recognition system. This system allows us to process an acquired face and compare it with the faces of the training set already defined using the *Eigenfaces* method.

In the experimental set-up, we did the performance only for 200 numbers of training images. But the experimental result shows that as the number of training images increases, efficiency of the system increases.

The accuracy of face recognition algorithm was measured by Euclidian distance between the test face and all train faces.
The results of the experiments ORL face database has been shown in next figures 3.9 to 3.11 respectively.

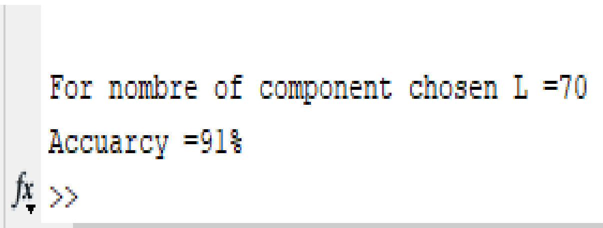

FIGURE 3.9: Accuracy

No. of Images for training n	Nombre of component chosen L	Accuracy (%)
200	40	85%
200	50	89%
200	70	91%

FIGURE 3.10: Classification Accuracy Table

According to the above results, PCA seems to have the best performances. However this efficiency cannot be generalized as it is performed on less number of train of images and conditions under which tested may be changed on other time.

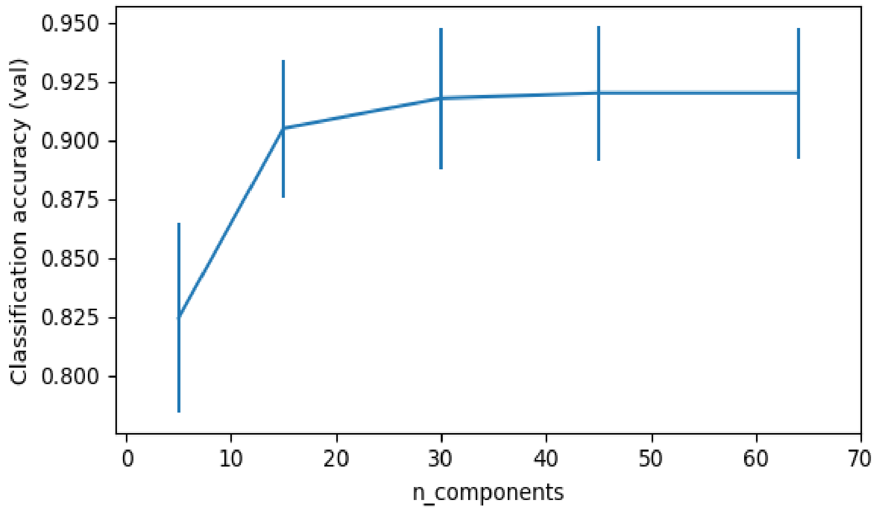

FIGURE 3.11: Accuracy Graph

<u>Conclusion:</u>

The results obtained show that our system is not 100% efficient because of lighting problems and identification in outdoor environments but we have achieved our face recognition goals.

Chapter 4

Challenges in Face Recognition System

Though face recognition have been a grown up research area, however, there still remain many problems that must be overcome to develop a robust face recognition system that works well under various circumstances such as illumination, pose, expressions, illumination and expressions,illumination and pose, and lastly llumination and expression and pose variations, as shown in figure 4.1.

The results reveal that all the recognition techniques were successful on large face databases recorded in well-controlled environments. But under uncontrolled environments their performance gets deteriorated mainly due to variations in illumination and head rotations. Such variations have proven to be one of the biggest problems of face recognition systems.

FIGURE 4.1: Challenges in Face Recognition System

4.1 Approaches to illumination variations

Illumination problem arises due to uneven lightning on faces as illustrated in figure 4.2. This uneven lightning brings variations in illumination which affects the classification greatly since the facial features that are being used for classification gets effected due to this variation.

The slight change in lighting conditions cause a significant challenge for automated face recognition and can have a significant impact on its results. If the illumination tends to vary, the same individual gets captured with the same sensor and with an almost identical facial expression and pose, the results that emerge may appear quite different.

Illumination changes the face appearance drastically. It has been found that the difference between two same faces with different illuminations is higher than two different faces taken under same illumination.

In the past few years, many approaches to cope up with illumination variations have been proposed. All the approaches towards illumination problem can be broadly categorized as: transformation of images with variable illumination to a canonical representation, extracting illumination invariant features, modeling of illumination variation and utilization of some 3-d face models whose facial shapes and albedos are obtained in advance.

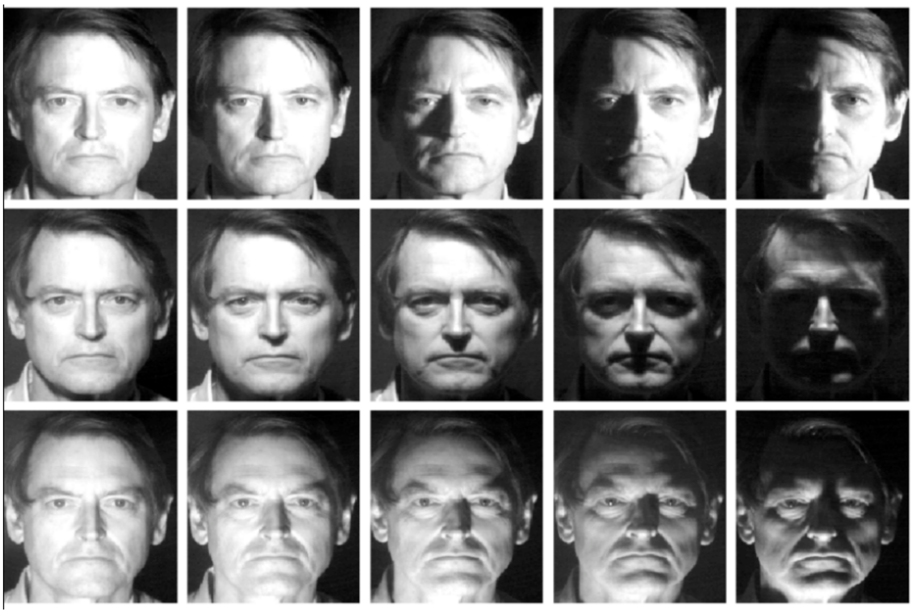

FIGURE 4.2: The Illumination Problem

4.2 Approaches to pose variations

The pose problem illustrated in figure 4.3 where the same face appears differently due to changes in viewing condition. Post-invariance recognition capability is crucial to a face recognition system because in general it is difficult, if not possible, to control the imaging direction when acquiring images of human faces

The pose of a face varies when the head movement and viewing angle of the person changes. The movements of head or differing POV of a camera can invariably cause changes in face appearance and generate intra-class variations making automated face recognition rates drop drastically. It becomes a challenge to identify the real face when the rotation angle goes higher. It may result in faulty recognition or no recognition if the database only has the frontal view of the face.

Till now many different methods have been proposed by various researchers to handle the rotation problem. Basically they can be divided into three classes: multiple images based approaches where multiple images per person are available; hybrid approaches in which multiple training images are available during training but only one database image per person is available during recognition and single image/shape approaches which require no training. Up to now, the second type of approach is the most popular one. The third approach does not seem to have received much attention.

FIGURE 4.3: Images of the same person (male) with different head poses

Chapter 5

Conclusion

In this thesis we implemented the face recognition system using Principal Component Analysis and Eigen face approach. The system successfully recognized the human faces and worked better in different conditions of face orientation.

First of all, the PCA algorithm is a global method using primarily the grayscale of the pixels in an image. Its simplicity to implement contrasts with a high sensitivity to changes in illumination, pose and facial expression. Nevertheless, the PCA does not require any a priori knowledge of the image and is most effective when coupled with the Euclidian distance measurement.

The principle according to which one can construct a vector subspace by retaining only the best eigenvectors, while keeping a lot of useful information, makes PCA an efficient algorithm and commonly used in dimensionality reduction where it can be used upstream of other algorithms.

Finally, the theoretical study of the PCA algorithm is very pedagogical and allows to acquire solid bases for 2D face recognition. It is an algorithm unavoidable algorithm!

Bibliography

[1] Prof. Khalide Jbilou, *"Face Recognition via Principal Component Analysis (PCA)"*, Lab Mathématiques, Université du Littoral Cote d'Opale, France.

[2] M.A. Turk and A.P. Pentland. *"Face recognition using eigenfaces"*. In Proc. of Computer Vision and Pattern Recognition, pages 586-591. IEEE, June 1991b. URL: https://www.cin.ufpe.br/ rps/Artigos/Face%20Recognition%20Using%20Eigenfaces.pdf

[3] A. Turk and A. P. Pentland. Eigenfaces for Recognition. *"Journal of Cognitive Neuroscience"* (1991). URL: http://www.vision.jhu.edu/teaching/vision08/Handouts/case$_s tudy_p ca1.pdf$

[4] Prof. Y. Vijaya Lata, *"Facial Recognition using Eigenfaces by PCA"*, Department of Computer Science and Engineering Gokaraju Rangaraju Institute of EnggTech, Jawaharlal Nehru Tech, University. URL: https://www.researchgate.net/publication/228748710$_F acial_r ecognition_u sing_e igenfaces_b y_P CA$

[5] Youssef Chahir, *"Face Recognition Using PCA and DCT"*, GREYC Laboratory CNRS UMR 6072, Departement of Computer Science, University of Caen Lower-Normandy, France. URL: https://chahir.users.greyc.fr/pdf/akrouf2009.pdf

[6] Kavita. R. Singh, Mukesh. A. Zaveri, Mukesh. M. Raghuwanshi *"Illumination and Pose Invariant Face Recognition: A Technical Review"*, International Journal of Computer Information Systems and Industrial Management Applications, India. URL: https://www.researchgate.net/publication/266178436$_I llumination_a nd_P ose_I nvariant_F ace_R ecognition_A T echnical_R eview$

[7] Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong *"MATHEMATICS FOR MACHINE LEARNING"*, Cambridge University, (2020). URL: https://mml-book.github.io/book/mml-book.pdf

[8] Kevin Huang, *"Principal Component Analysis in the Eigenface Technique for Facial Recognition"*, Senior Theses, Trinity College, Hartford, CT (2012). URL: https://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221context=theses

[9] Sezin KAYMAK. *Face Detection, Recognition and Reconstruction using Eigenfaces.*, EE574 Detection Estimation Theory. Class Presentation. Spring (2003). URL: https://docuri.com/download/kaymak$_5 9c1e913f581710b286d00e5_p df$