Intermediate Report

by Tahir Emre Semiz

Submission date: 25-May-2023 07:07PM (UTC+0300)
Submission ID: 2101704588
File name: Intermediate_Report.pdf (734.94K)
Word count: 1869
Character count: 11067

IMAGE PROCESSING

Face Detection and Recognation Project

Intermediate Report

Project ID:

PROJECT GROUP MEMBERS:

Alim Kadir KARA - 20190808013

Anıl Can ALPARSLAN - 20190808068

Tahir Emre SEMİZ - 20200808058

Contents

1.INTRODUCTION

Face detection and recognition are some of the image processing applications. Recently, the face detection and recognition branch has gained rapid popularity and importance. With this importance, it has developed in many areas both in itself and in the branches it is implemented. Faces are identified and verified in the digital environment with this application. There are multiple methods for this identification process. (It is stated in the continuation of the article.) According to the usage area and features of the work to be done, the advantages and disadvantages of the methods are compared. Thus, the more efficient method is selected and the work begins. By making comparisons between the images obtained in the studies and the dataset, the face recognition process is fully performed.

To perform face recognition, face detection must be done first. After identifying a face in a photo or video, the identity of the face is determined with the dataset. Of course, as in any business, some factors create difficulties in these transactions. Some of these challenges are:

Faces can appear in front of the camera from different angles. These face angles affect the performance of face recognition algorithms.

In poor lighting and high levels of lighting, algorithms cannot work successfully. In the case of instant lighting changes, inconsistencies are experienced in recognition and detection.

Some changes in facial features and accessories such as glasses and hats make it difficult for the algorithm to work properly.

The performance of the algorithms may be disrupted due to the instant facial mimics captured on the camera.

If a face is compared with the dataset years ago, there will be problems in the algorithm. because the algorithm may not catch the obvious changes in the face due to aging.

Some of the areas where face detection and recognition are currently widely used are:

In particular, departing and arriving passengers on international flights are inspected by face recognition systems at airports. Thus, criminals who try to escape or enter the country are arrested and security is ensured.

It is used as a security measure to unlock the phone on smartphones or in applications that require some admin login.

It has gone viral on social media in recent years. It is used to make filters in applications such as Instagram and Snapchat. this technology has made these algorithms effective in the entertainment industry.

2.SURVEY

Some of the commonly used methods are briefly used as follows:

Template Matching: It is not a very successful method for object discrimination. Starting from (1,1) coordinates and moving between all pixels, object detection is made.

Viola Jones algorithm: accuracy rate is high. It is preferred for real-time uses. gives clear results from the front profile. Navigation is done by looking for features related to the face in the picture. (ex. eye, nose, ear, mouth...)

Convolutional Neural Network (CNN): It is a deep learning algorithm. It takes images as input and processes them and classifies them.

Single Shot Detector (SSD): this algorithm performs worse than the CNN algorithm, but is much faster and easier. Because it works fast, it is mostly used for object detection.

Those who will make projects on this subject should investigate these methods in more detail and start their projects by considering their advantages and disadvantages. The above-mentioned face recognition difficulties have been decreasing over the years. With the development of hardware and software, for example, in 2014, the best face recognition algorithm had an error rate of 4.1 percent (ref. 5), but now this rate has decreased to 0.8 percent.

3. METHODS

We will use 2 methods in face recognition. One of them is eigenface and the other is fisherface.

3.1 Eigenface Method

The eigenface method we will use first is a face recognition method created using the Principal Component Analysis (PCA) algorithm. To briefly explain, this algorithm allows to reduce the size in the face space without losing critical information. In this way, when face recognition is performed, it does not need to be navigated in the entire face space, which increases efficiency.

The stages of this method are as follows:

3.1.1 Preprocessing

This step involves how to preprocess the face space. For this purpose, we will make the face space suitable for the PCA algorithm according to 3 normalizations.

- **Size Normalization**
  We resize all the images in a common size. (Such as 64x64 pixels or 256x256 pixels) Because in the PCA algorithm, it requires that the data be all in the same dimension.

- **Intensity Normalization**
  We normalize the intensity level to grayscale. This step is done because color information is not an important element in eigenface face recognition.

- **Geometric Normalization**
  This normalization is critical since there will be many variations (orientation and position of faces) in face space.

Figure 1 *(Ref. 1)*

Figure 2 *(Ref. 1)*

3.1.2 Creating the Eigenfaces

After preprocessing, we use PCA to find the "directions" (or eigenvectors) where the data changes the most. In this context, these eigenvectors are actually images, which we call eigenfaces.

- First of all, let's assume that there are m images in NxN dimensions in our face space.

- We create a matrix where each row corresponds to one face image from the training set. Each image is converted into a one-dimensional vector.

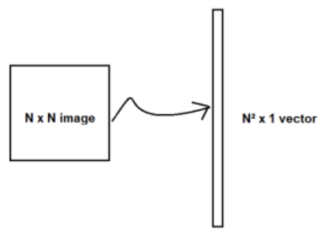


Figure 3 *(Ref. 1)*

$$X_1, X_2, X_3 \dots X_m$$

- Then, we calculate the average face of all these face vectors (Figure 5),Then we subtract each face image in the face space from the average face (Figure 6).

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

Figure 6 *(Ref. 1)*

$$a_i = x_i - \psi$$

Figure 5 *(Ref. 1)*

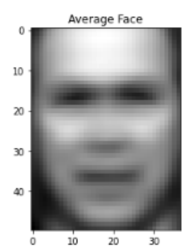


Figure 4 *(Ref. 1)*

- Then we take all the face vectors and get a matrix in the size of N^2 x m.

$$A=[a_1\ a_2\ a_3 \dots\ a_m]$$

- The covariance matrix of these difference vectors is calculated. This matrix will give a mathematical representation of the set of face images, showing how each pixel in the images relates to every other pixel.

$$Cov = A^T A$$

- Then, the eigenvectors of the covariance matrix and their corresponding eigenvalues are calculated using the formula (Figure 8).

$$A^T A\nu_i = \lambda_i \nu_i$$

$$AA^T A\nu_i = \lambda_i A\nu_i$$

$$C' u_i = \lambda_i u_i$$

Figure 7

- These eigenvectors, known as eigenfaces in this context, define the features that contribute most to the variation between faces.

Figure 8 *(Ref. 2)*

- The chosen eigenfaces are then used to transform the original dataset into a smaller, more manageable size. Each face in the training set is represented as a weighted sum of the chosen eigenfaces.

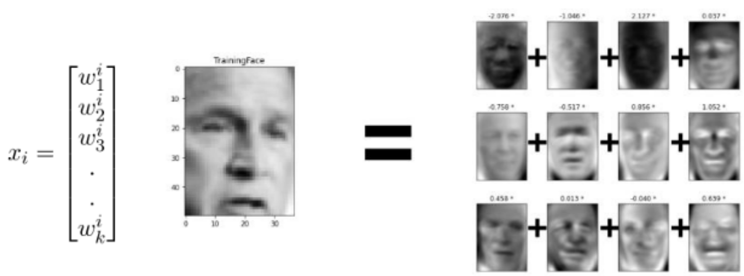


Figure 9 *(Ref. 2)*

3.1.3 Recognition

- First of all, we preprocess the image of the unknown face y that we have. Then we subtract the face from the average face.

$$\phi = y - \psi$$

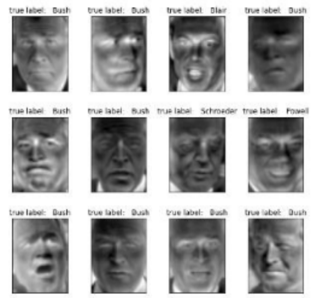


Figure 11 *(Ref. 2)*

- Then, we generate the vector of the coefficient. We then make him recognize the face by comparing the set of coefficients with those of the training set. We assume that the closest match (using a distance metric) is the correct identity.

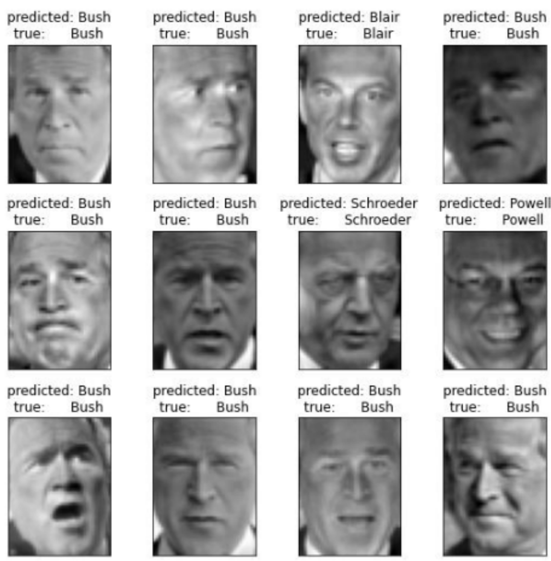


Figure 12 *(Ref. 3)*

3.2 Fisherface Method

The second fisherface method that we will use is a facial recognition method created using the Linear Discriminatory Analysis (FLDA) technique. To put it more simply, this technique deeply analyzes a lot of different faces and creates a unique identity for each face. And we call it "fisherface". At the same time, one of its most important features is that it can perform robustly even in low-light environments.

The stages of this method are as follows:

3.2.1 Preprocessing

This step involves how to preprocess the face space. For this purpose, we will make the face space suitable for the PCA algorithms according to 3 normalizations.

- **Size Normalization:** We resize all the images in a common size. (Such as 64x64 pixels or 256x256 pixels) Because in the PCA algorithm, it requires that the data be all in the same dimension.
- **Intensity Normalization:** We normalize the intensity level to grayscale. This step is done because color information is not an important element in fisherface recognition.
- **Geometric Normalization:** This normalization is critical since there will be many variations (orientation and position of faces) in face space.

3.2.2 Creating the Training Set

Here is a sample of photos photograph with everyone represented by a minimum of samples of face images with different positions and different expressions.

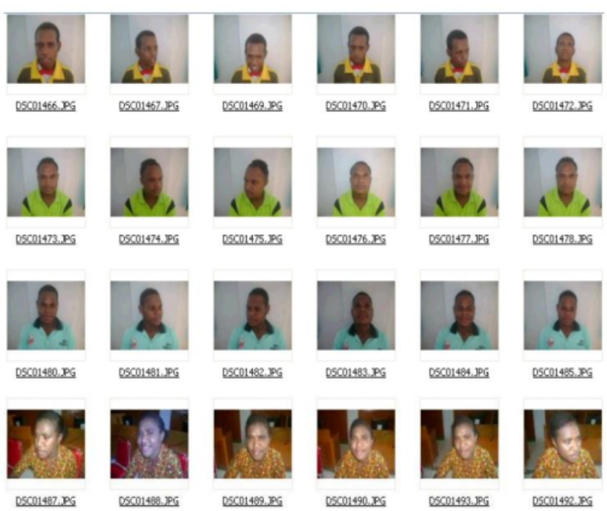


Figure 13 *(Ref. 4)*

Figure 14. Examples of some training images *(Ref. 4)*

3.2.3 Creating the Fisherface

Fisherface uses PCA and LDA. And it tries to maximize inter-class scattering and Decrement in-class scattering to a minimum.

3.2.3.1 PCA Algorithm

- Conversion training image $\Gamma 1, 2, \ldots m$ with size N x N into vector form with length size N ^ 2

Figure 15 *(Ref. 4)*

- We determine the mean value across all face images.

$$\vec{m} = \frac{1}{M}\begin{pmatrix} a_1 & +b_1 & +\cdots+h_1 \\ a_2 & +b_2 & +\cdots+h_2 \\ \vdots & \vdots & \vdots \\ a_{N^2} & +b_{N^2} & +\cdots+h_{N^2} \end{pmatrix},$$

or written as, $\vec{m} = \dfrac{\vec{a} + \vec{b} + \ldots + \vec{h}}{M}$

- We need to calculate Matrix A. To create this matrix, each image is first converted into a one-dimensional vector by concatenating each row (or each column) of the image. Then, these vectors are combined to form Matrix A. After that we are able to calculate Matrix A.

$$\vec{a}_m = \begin{pmatrix} a_1 & - & m_1 \\ a_2 & - & m_2 \\ \vdots & & \vdots \\ a_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{b}_m = \begin{pmatrix} b_1 & - & m_1 \\ b_2 & - & m_2 \\ \vdots & & \vdots \\ b_{N^2} & - & m_{N^2} \end{pmatrix},$$

$$\vec{c}_m = \begin{pmatrix} c_1 & - & m_1 \\ c_2 & - & m_2 \\ \vdots & & \vdots \\ c_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{d}_m = \begin{pmatrix} d_1 & - & m_1 \\ d_2 & - & m_2 \\ \vdots & & \vdots \\ d_{N^2} & - & m_{N^2} \end{pmatrix},$$

$$\vec{e}_m = \begin{pmatrix} e_1 & - & m_1 \\ e_2 & - & m_2 \\ \vdots & & \vdots \\ e_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{f}_m = \begin{pmatrix} f_1 & - & m_1 \\ f_2 & - & m_2 \\ \vdots & & \vdots \\ f_{N^2} & - & m_{N^2} \end{pmatrix},$$

$$\vec{g}_m = \begin{pmatrix} g_1 & - & m_1 \\ g_2 & - & m_2 \\ \vdots & & \vdots \\ g_{N^2} & - & m_{N^2} \end{pmatrix}, \quad \vec{h}_m = \begin{pmatrix} h_1 & - & m_1 \\ h_2 & - & m_2 \\ \vdots & & \vdots \\ h_{N^2} & - & m_{N^2} \end{pmatrix}$$

$\vec{A} = [\ \vec{a} - \vec{m}, \vec{b} - \vec{m}, \ldots, \vec{h} - \vec{m}\ ] \qquad = [\vec{a}_m, \vec{b}_m, \ldots, \vec{h}_m]$ or written as,

3.2.3.2 LDA Algorithm

- We calculate the average of each person / class. This allows us to determine the mean feature vectors that set one person or class apart from another.

$$\vec{x}=\frac{1}{2}\begin{pmatrix} a_1 & + & b_1 \\ a_2 & + & b_2 \\ \vdots & & \vdots \\ a_{N^s} & + & b_{N^s} \end{pmatrix},\quad \vec{y}=\frac{1}{2}\begin{pmatrix} c_1 & + & d_1 \\ c_2 & + & d_2 \\ \vdots & & \vdots \\ c_{N^s} & + & d_{N^s} \end{pmatrix},$$

$$\vec{z}=\frac{1}{2}\begin{pmatrix} e_1 & + & f_1 \\ e_2 & + & f_2 \\ \vdots & & \vdots \\ e_{N^s} & + & f_{N^s} \end{pmatrix},\quad \vec{w}=\frac{1}{2}\begin{pmatrix} g_1 & + & h_1 \\ g_2 & + & h_2 \\ \vdots & & \vdots \\ g_{N^s} & + & h_{N^s} \end{pmatrix}$$

- Then we need construct the scatter matrices S1, S2, S3, S4 in LDA to quantify the spread (or scatter) of the data within each class (the "within-class scatter") and between different classes (the "between-class scatter")

$$S_1 = \left(\vec{a}_m\, \vec{a}^{\,T}_{m} + \vec{b}_m\, \vec{b}^{\,T}_{m}\right),$$
$$S_2 = \left(\vec{c}_m\, \vec{c}^{\,T}_{m} + \vec{d}_m\, d^{T}_{m}\right)$$
$$S_3 = \left(\vec{e}_m\, \vec{e}^{\,T}_{m} + \vec{f}_m\, \vec{f}^{\,T}_{m}\right),$$
$$S_4 = \left(\vec{g}_m\, \vec{g}^{\,T}_{m} + \vec{h}_m\, \vec{h}^{\,T}_{m}\right)$$

and matrix within class scatter ($ScatW = S1 + S2 + S3 + S4$)

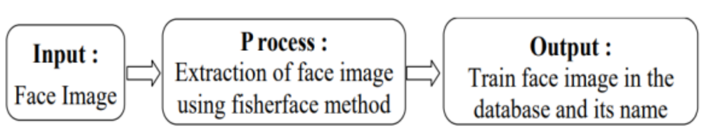
- The construct of also matrix between class scatter, (ScatB)

$$(ScatB\ =\ 2(\vec{x}-\vec{m})(\vec{x}-\vec{m})^T + 2(\vec{y}-\vec{m})(\vec{y}-\vec{m})^T + 2(\vec{z}-\vec{m})(\vec{z}-\vec{m})^T + 2(\vec{w}-\vec{m})(\vec{w}-\vec{m})^T$$

So, in here We need to calculate the multiplication of matrices transpose of pe, (PeT), with ScatB and ScatW until we obtain:

Sbb= PeT *ScatB *Pe
Sww= PeT *ScatW *Pe

Find eigenvector (VeSbb) and generalized eigenvalues (NeSww) of (Sbb, Sww) and we must sort it in such a way that it increases.



Each face is made up of a weighted mixture of Fisherfaces. Projecting the face image into the Fisherface space yields the weights. These weights combine to generate a feature vector that represents the face in Fisherface space.

3.2.4 Recognition

In summary, the Fisherface method works more smoothly and well against changes in lighting and facial expressions. Because it not only reduces dimensionality like PCA, but also maximizes class separability through LDA.

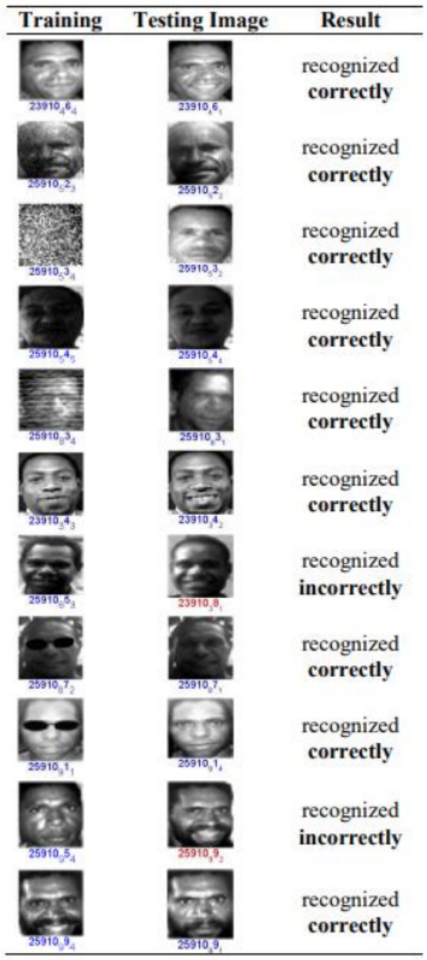


Figure 16 *(Ref. 4)*

4. REFERENCES

1. https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/

2. https://www.atmosera.com/blog/principal-component-analysis/

3. https://gamedevacademy.org/face-recognition-with-eigenfaces/

4. https://iopscience.iop.org/article/10.1088/1742-6596/1028/1/012119/pdf#:~:text=The%20Design%20System%20Face%20recognition,is%20recognized%20correctly%20or%20not.

5. https://www.csis.org/blogs/strategic-technologies-blog/how-accurate-are-facial-recognition-systems-and-why-does-it

Intermediate Report

ORIGINALITY REPORT

13%	10%	7%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	mafiadoc.com Internet Source	6%
2	Submitted to Institute of Technology, Sligo Student Paper	2%
3	www.mdpi.com Internet Source	1%
4	Submitted to University of Stirling Student Paper	1%
5	www.inderscience.com Internet Source	1%
6	Submitted to Sapientia University Student Paper	1%
7	Submitted to University College London Student Paper	1%
8	Submitted to University of Bristol Student Paper	1%
9	ir.jkuat.ac.ke Internet Source	1%

10	Submitted to The University of Manchester Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		