ARTIFICIAL VISION
PROJECT 1

Face Recognition

Lisbon, 2021.04.15.

15.

Contents

Introduction

During the first laboratory, we were introduced to two different methods used in the field of face recognition. These are namely Eigenfaces and Fisherfaces.

The goal of face recognition is the following: given a set of faces with known identities (this is called the training set) and a set of faces not identified, belonging to the same group of people (the test set), we would like to connect a person from the training set to the people in the test set, i.e. we want to identify them.

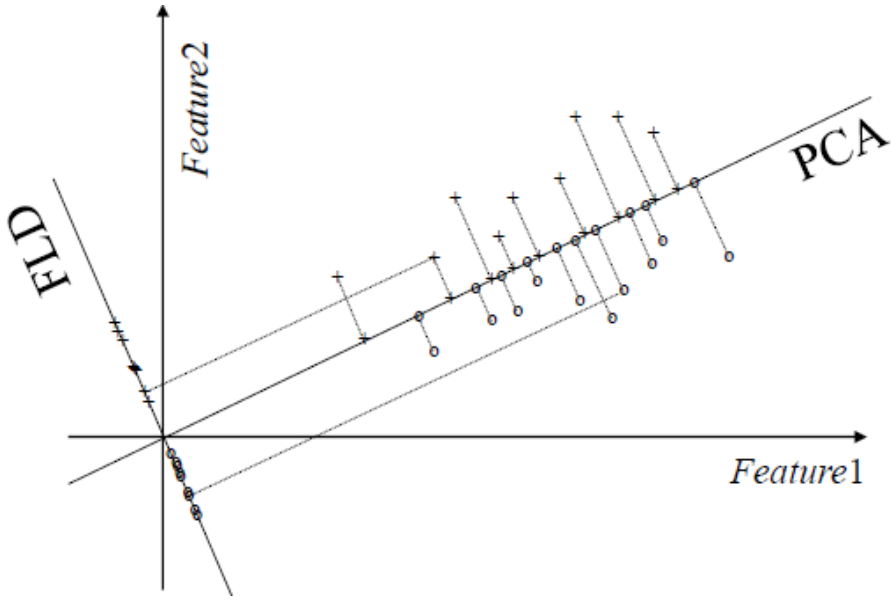
Both methods use normalised pictures of faces according to the MPEG-7 standard. These pictures are 56 pixels wide and 46 pixels high and can be created ’manually’ from every picture. This manual way means a Python script, using the openCV library, and we call it manual, because the user have to click on the eyes in the picture, then these points will be used by the script making the normalised face. The other method automatically makes the normalised faces. In our solution, we chose the manual solution.

The eigenfaces and the fisherfaces methods are based on projecting the faces - which are in a high dimensional space (56*45=2576) – to a lower dimensional subspace. The former method uses PCA (Principal Component Analysis), and the latter uses MDA (Multiple Discriminant Analysis).

In the following, we will show the advantages and disadvantages of the two method. For testing our code, we used the face database on the moodle page of the course, which is consists of 28 pictures (7 classes, 4 pictures each).

1 The eigenfaces method

The eigenfaces method is especially good in reconstructing the faces from the projected vectors. It maximises the scatter between all images. However, because of this, this method is not the best for the classification of the people in the images. For that reason, the intra scatter – i.e. between the pictures of one person – would have to be small, but the inter scatter – i.e. the scatter between the seperate classes – should be large. This is what the later introduced fisherfaces method is for.



The above figure shows the different behavior of the two methods. It can be seen that while the PCA (used by the eigenfaces) lowers the scatter between all vectors (both the ’x’-es and the ’o’-s), the FLD (used by the fisherfaces) organises the ’x’-es and ’o’-s into groups, inside which groups the scatter is low. However, between the two groups the scatter is quite large.

The algorithm used for implementing the eigenfaces method was the following.

- Prepare the training set consisting of N faces, $x_1 \ldots x_N$, properly aligned
- Determine the mean face μ
- Define matrix A (size of n×N) whose columns contain the "AC components" of the faces from the training set
- Compute the eigenvectors and eigenvalues of matrix R= $A^T A$ (size of N×N)

- Select m (maximum of N-1) eigenvectors from R, associated to the highest eigenvalues. Define matrix V (N×m), formed by the m eigenvectors of R
- Get matrix W(n×m) by the relation W = AV (not forgetting that the Wcolumns form an orthonormal basis)
- Develop a face classifier (e.g., nearest neighbor) based on models trained with the observations (feature vectors) of the projections in the face subspace, y= $W^T(x-\mu)$

1.1 Results

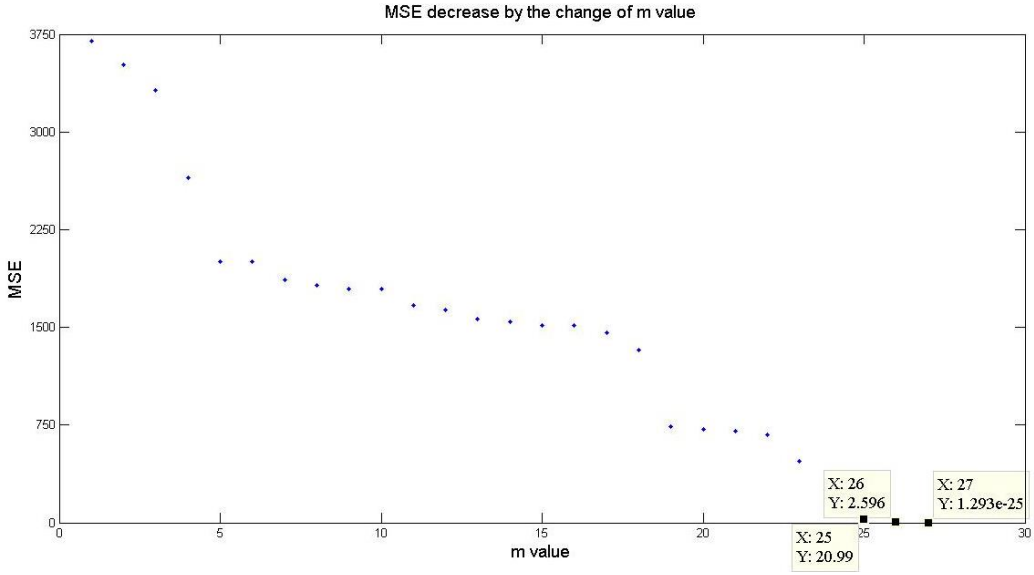


Figure 1. MSE decrease

The Figure 1. shows the mean squared error (MSE) decreasing by the change of the m-value, which responds to the „AC component" of the reconstructed face, while the meanface represent the „DC component". While the m-value increases the MSE drops down, especially after the 3rd and 19th component. The maximum value of m is the number of images -1. The error in this case is negligibly small.

A picturesque result is to view the reconstructed faces in a sequence by changing the m value. This result can be found for three different faces from the database here: http://1drv.ms/1EIMTZL

On the following pictures, we show how the reconstructed face gets better and better if we increase the ’m’ variable – the number of the AC components.

2 The Fisherfaces method

With the Fisherfaces method one can assign most similar face images from a database to a new face image. By this method the face reconstruction is not expected, so we will focuse on the facedetection.

The algorithm used for implementing the fisherfaces method is the following:

• Given the faces $x_1 \ldots x_N$, properly aligned and classified into class

c (i = 1,..,c), each with ni elements

• Determine the mean face μ, and the mean face of each class μ_i

• Determine the scatter matrix S_T (n x n)

– Determine m (maximum of N-c) non-zero eigenvectors, W_{pca} (see

algorithm eigenfaces)

• Determine the S_b (n x n) and S_w (n x n) matrices

• Compute

$$S_b = W^T{}_{pca} S_b W_{pca}$$

$$S_w = W^T{}_{pca} S_w W_{pca}$$

• Determine the c-1 "larger eigenvectors" from the matrix

$$S^{-1}{}_w S_b \leftarrow \text{m*m}$$

The classification is done in the function „faceengine". The basis for comprasion are the vectors that we can obtain by multiplying the AC faces with $W_{fld}^T W_{pca}^T$. Instead of store all of the pixels, in this case we store just few coefficients as we did with eigenface method. The „faceengine" gets an image and a k constant number as input. The number defines the amount of the most similar faces that should the „faceengine" gives as output. First the function computes the AC face of the given image, after makes the equivalent computation that we've done before with the database to obtain a new coefficient vector. After this step the function computes the Euclidien distance between the new vector and the vectors of the database. Finally the distances are sorted and the fuction gives as output the most relevant k images from the database.

We had a database with 28 faces in 7 groups. We've tested the algorithm with 7 face images, 1 by each group. There was only one error, in the rest of the cases the algorithm has given the right solution.