Applied Mathematics
Project Report

Face Recognition using PCA

Authors:
Mohamed Ali
Elizaveta Genke

Supervisor:
Désiré Sidibé

January 15, 2018

1 Introduction

Principal component analysis (PCA) is a powerful tool, that can be applied in wide range of problems, including the Face Recognition problem. Particularly, for this problem it allows to reduce dimensionality and therefore decrease computation time.

2 Methodology and implementation

The algorithm was implemented in Python 3.6 (using Jupyter Notebook). The workflow of our program can be divided into two steps: normalization and PCA. After these steps we test our systems.

2.1 Normalization

For normalization we use Skimage library function AffineTransform which allows us to estimate 2D affine transformation (we need to normalize faces in order to work with them in recognition system). After we apply this transformation to all images using warp function from the same library. Also we convert our images to grey.

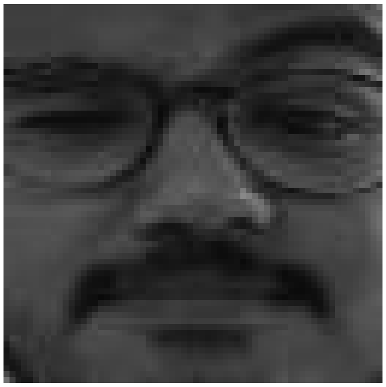


Figure 1: *Example of normalized image*

2.2 PCA

After we get all pictures normalized, we start working on recognition part of the algorithm. First, we divide pictures into two folders: 'Train set' and 'Test set'. In the folder 'Train set' we have front image and two side views and we put other pictures to the folder 'Test set'.

Every image from 'Train set' we represent as a vector by performing concatenation operation. Then we form an array from them, so every image is a row of this array. For this matrix we compute covariance matrix using formula:

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

Then we perform eigenvalue decomposition over the covariance matrix and take ¡how many?¿ principal components (eigenvectors corresponding to the largest eigenvalues).

Then we project every image to PCA space by formula:

$$\phi_i = X_i \Phi \tag{2}$$

and after we test our system using these feature vectors.

3 Experiments

After we loaded all images (and .txt files with positions of 5 features) we plot the positions in one graph.

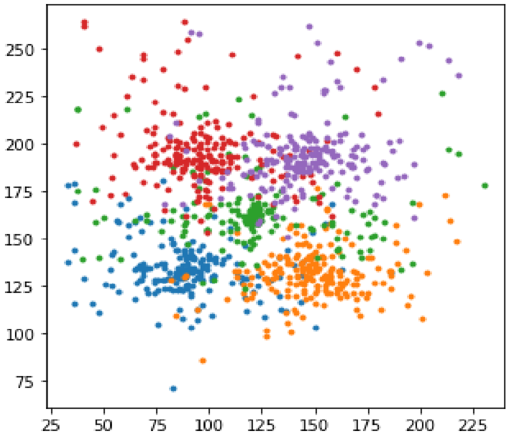


Figure 2: *Positions of features*

From this we see that most of the points is located in almost one place but we have a lot of outliers. So we perform normalization and after normalization we show the picture of mean guy of the class and plot points of predefined positions of featutes.

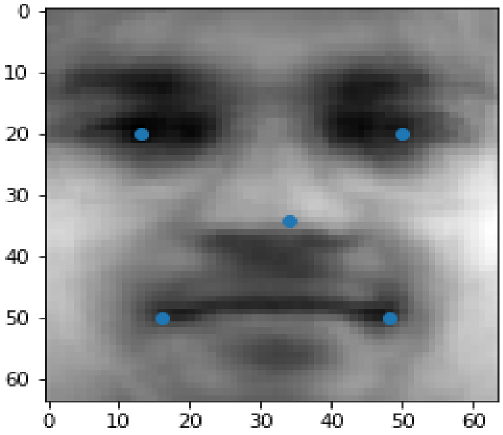


Figure 3: *Mean guy with predefined positions of features shown*

We can see that points are located correctly, that means that our normalization gives fairly good results.

After we put all train images in one matrix we perform PCA. The picture of covariance matrix is shown below.

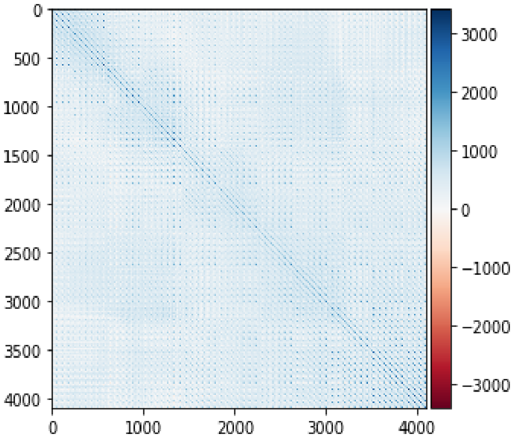
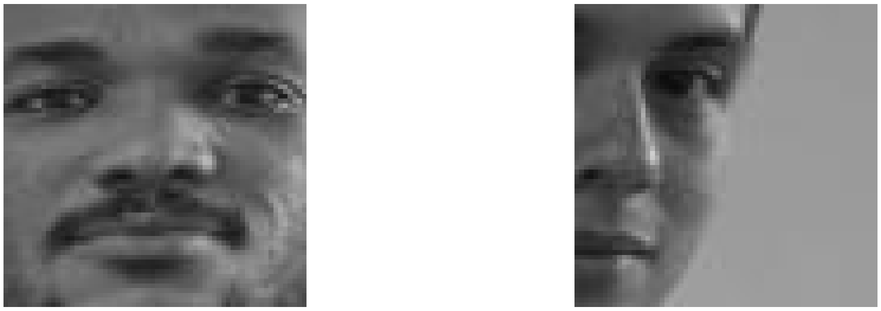


Figure 4: *Covariance matrix*

As expected, the highest values are located on the diagonal of the matrix.

4 Results

The results of normalization as we mentioned above are fairly good, but we have faced some problems, some pictures are not transformed correctly. For some we checked the positions and found them wrong, but after our correction it didn't increased overall performance of the system significantly. Probably, better option was to do iterative process of normalization as described in the assignment.

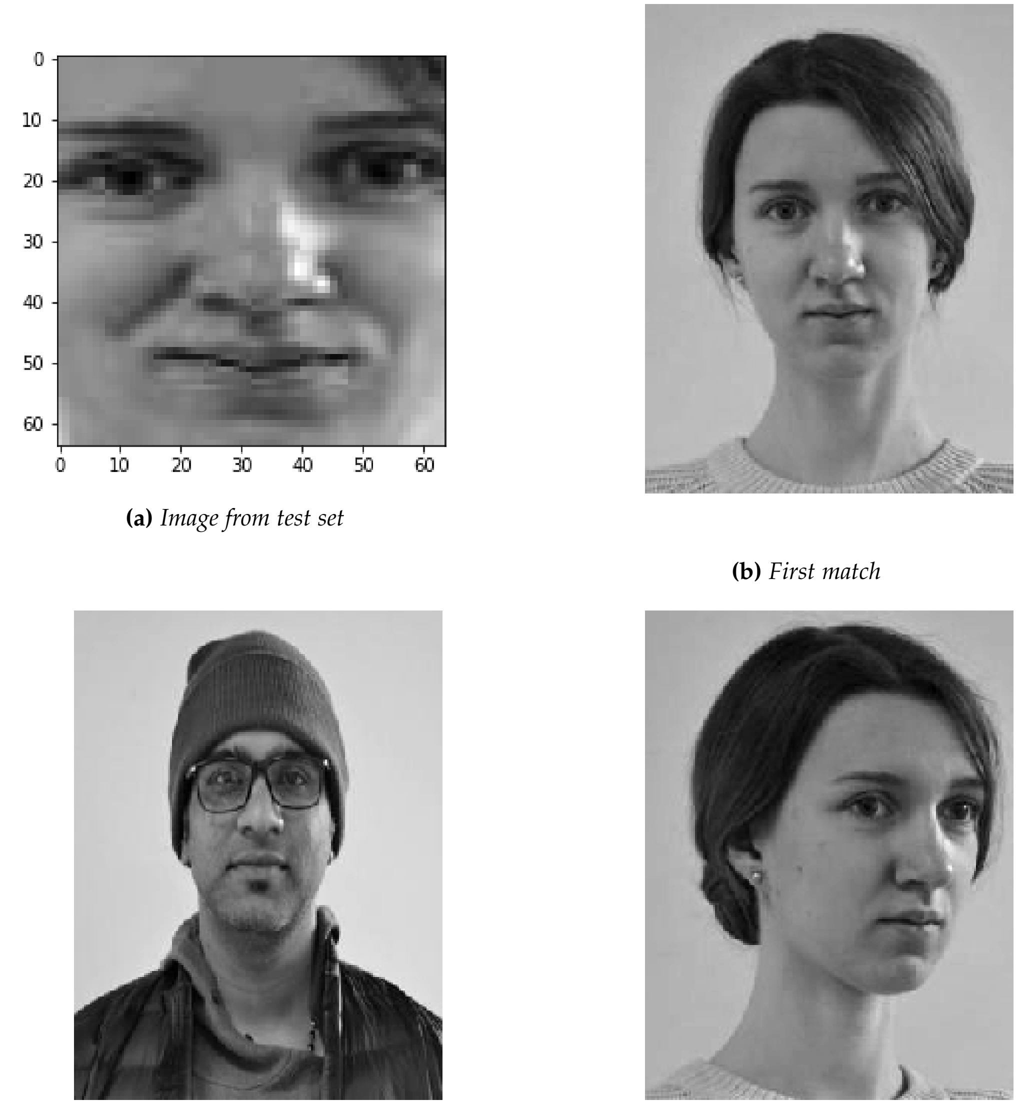


(a) *Example of good transformation*

(b) *Example of bad transformation*

Figure 5: *Normalization results*

The accuracy of our system is 60 percents (we check if image is in the first 5 matches). If we check only the first match the accuracy is 53,75 percents. Most probably, this comes from not perfect transformation.



(a) *Image from test set*

(b) *First match*

(c) *Second match*

(d) *Third match*

Figure 6: *Example of test results*

We see that the first and the third matches are right and the second is incorrect.

5 Conclusions

This project gave us an opportunity to apply our knowledge in practice. We worked with PCA in application to face recognition problem and achieved fairly good results. Following the steps for applying the algorithm and tweaking different parameters show us what is the most important points in that method. As consequence our Python coding skills improved.