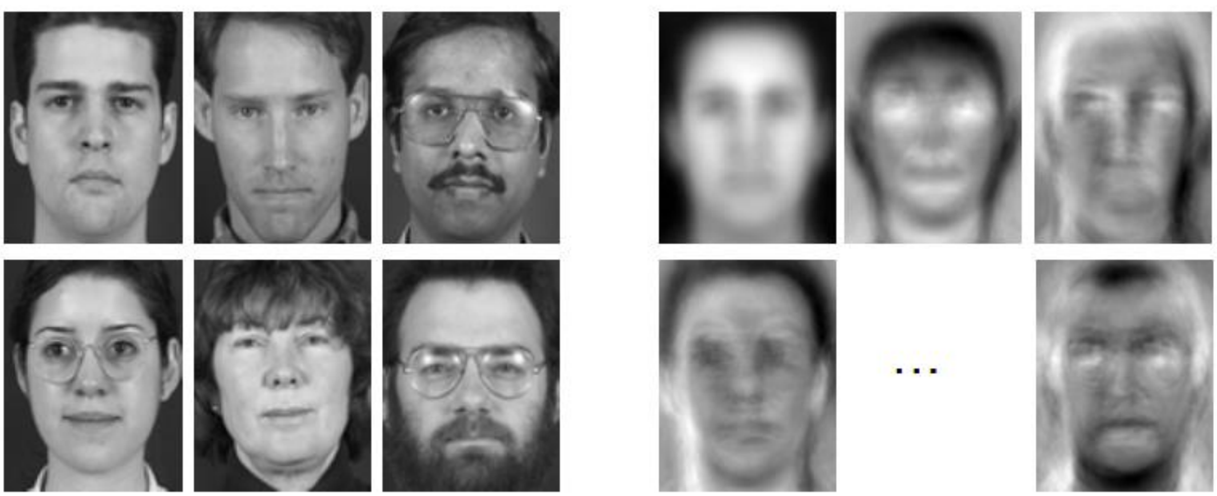
BioM Lab 2: Principal Component Analysis

Introduction

Most faces are dissimilar… but not that much. Especially if their images are compared with all other possible images than can be taken or generated. In this case, different faces just correspond to mimimum variations with respect to a general trend containing facial features such as eyes, nose, mouth, eyebrows…

This is the assumption of the Principal Component Analysis (PCA) for face images. The hyperspace defined by considering each pixel as a dimension, is reduced to a much smaller one that tries to still characterise the great small differences faces present. This lab exercise explores this technique and its application to face recognition… and inpainting.



Tools and kick off

Completing the quiz

If you have a GDrive account, you can complete this report online by generating a copy (File > Copy…). If you prefer to complete it locally with your word processor, you can export it to *.odp, .rtf, .doc*… (File > Download as…). You will be asked to deliver a PDF version of it at the end of the session on an Atenea task.

Matlab

This lab exercise will require the usage of *Matlab* to load the data and run the provided scripts. You can access the computers in lab D5-004 with your username *biotec11x*, where *x* corresponds to the ID shown on your desk. Ask the professor for your password.
Once you have logged in, copy to your home directory the contents of the *lab3* folder. Finally, launch *Matlab* and set your working directory to your copy of *lab3*.

Background

Read the PCA slides of the course and, if necessary, complete your understanding through web search or asking the teacher.

Dataset

1. A small learning dataset of images can be read from image files and stored in a Matlab .mat file for its use in further steps of the exercise. Generate it with the provided script:

```
read_faces
```

2. Describe this dataset: how many images does this dataset contain ? How many different people appear there ? What variations can you observe ?

Answer: 12 images, 4 different persons. There are three different images of each person, where the person in question does different poses.

3. If each pixel is considered as a dimension on an hyperspace of images, what is the dimension of such space ?

Answer: The size of each image is 480x640 pixels. All 12 images has the same size, so such a space has the dimension 480x640x12=3686400.

4. Run the script *face1.m* that will you this dataset to estimate a collection of eigenfaces for PCA.

```
face1
```

5. Array X contains in each of its rows the pixel values for each subject in the training dataset. What are the dimensions of this array ? Are these values consistent with your previous answers ?

> Answer: The dimension is 12x307200, where 3686400/12=307200. So these values are consistent with the previous answers.

6. Look at the contents of the resulting figure. What is presented there ?

> Answer:The average face (mean(X)) and the 11 calculated eigenfaces.

7. Read the code in face1.m and interpret it according to the contents of the slides.

8. [Optional-Medium] The provided dataset is much larger than the default faced considered. In particular, it contains 90 images from 15 people and 6 poses. Copy and modify the script read_faces.m to work with a larger and completely different dataset. Mention the name of your new script below. Test that all exercises work properly with them.

> Answer: read_faces_mod.m

PCA for coding

PCA allows representing most part of the information in a very condensed fashion. Once the average face and the n-1 eigenfaces are known, it is possible to compute the coordinates (or main components) of the training dataset. Instead of storing each individual pixel, it is enough to save its n-1 coefficients. In addition, if we allow a lossy coding, we can also choose the first q < n-1 eigenfaces

9. Launch script *face2.m* by just typing:

`face2`

10. Figure 1 contains the results of decoding the images from the training dataset using only a few eigenfaces. Study the source code to determine how many of them are being considered by default.

Answer:The value that q is equal to, here it is 8.

11. Modify the value of q in the script to visually assess the impact of changing the amount of considered eigenfaces.

12. Figure 2 plots the Root Mean Square Error (RMSE) obtained by individually comparing each decoded pixel from the original one.

13. Figure 3 displays the results of decoding two faces which were not in the original dataset. What is the result ?

Answer: The result is bad, it is not possible to regonize the decoded image.

14. [Optional-Avanced] Build your own dataset with a minimum of four people and a minimum of three poses or facial expresions. Obtain reasonable performance results with the Coding application. Explain how to run the experiment in the Answer box below.

IMPORTANT NOTICE: Unless you state the contrary, these data will be released in a Creative Commons License Attribution No-Commercial (CC BY-NC) and it might be used in future editions of this course.

Answer: In read_faces.m set chemin='datamat'; instead of chemin='faces'; The dataset contains 4 persons and 3 poses for each person.

PCA for inpainting

15. An unexpected application of eigenfaces are its potential to restore images which are degradated, where some of the pixels are noisy or just lost. In general, this family of techniques are called *inpainting*.

16. Launch script *face3.m* by just typing:

```
face3
```

17. Observe the results in Figure 1. Test some modifications on the location of the black bars by modifying the variables line_min and line_max. Is there any area where the impat of the occlusion is greater in terms of distortion ?

Answer: When removing the sweater in the first image, the distortion is greater than in the most of the other cases. That is when line_min=400 and line_max=480.

18. Study the source in face3.m. Copy the lines where: (a) the principal components are computed and (b) the image is reconstructed

Answer: The same for `face_incomplete_2`.

(a)

```
face_incomplete_1_dessin = face_incomplete_1;
face_incomplete_1=double(face_incomplete_1)/255;
face_incomplete_1=face_incomplete_1(:);
face_incomplete_1=face_incomplete_1';
composantes_principales_face_incomplete = (face_incomplete_1-
average_face)*W;
```

(b)

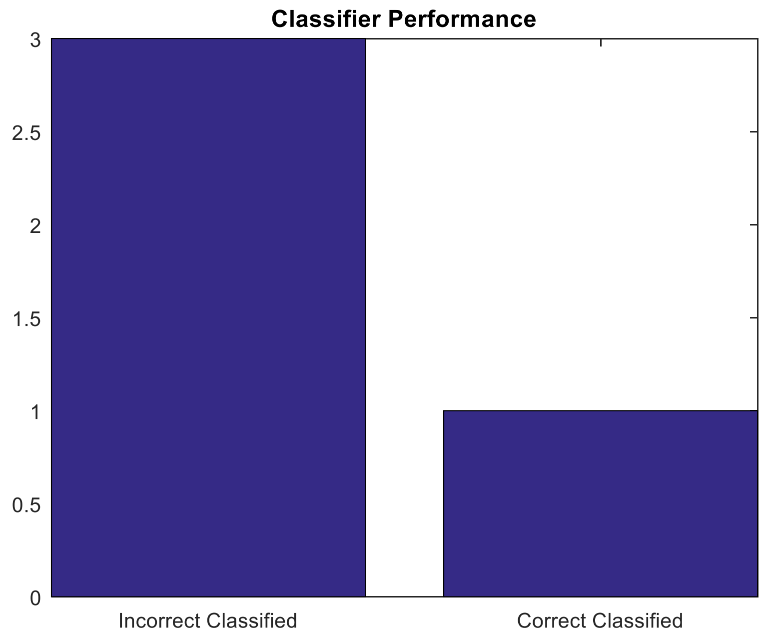
```
face_incomplete_reconstruite = average_face +
composantes_principales_face_incomplete*W';
```

PCA for identification

19. [Optional-Advanced] Follow the guidelines provided in the last page of the provided slides to develop an identification experiment based on the PCA representation of the face images using the provided dataset with the following configurations:
    - 3 poses for PCA estimation
    - 2 poses for training
    - 1 pose for test

    Use a K-Nearest Neighbours (k-NN) classifier.

Answer: The program is named PCArecog.m and k-value in k-NN is chosen in line 146. The program is written for maximum 4 different persons. The result is not good:



20. [Optional-Advanced] Instead of using a k-NN classifier, adopt a Naive Bayes Nearest-Neighbor (NBNN), as presented in the this paper:

O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In Computer Vision and Pattern Recognition (CVPR), 2008.

Answer: (comment on your implementation and provide classification results as a graph)

Scoring Rubric

The final grading of this exercise will follow this chart:

A	Excellent	Two advanced questions or more
B	Very good	One advanced question and one medium
C	Good	One medium question
D	Pass	No optional questions at all
F	Fail	Basics not completed

Notice: Additional penalties may apply if the working attitude in lab or the delivery of submissions does not exactly meet the provided instrucions.