POLITECNICO DI TORINO

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Matematica

Computational linear algebra for large scale problems

HOMEWORK
Eigenfaces Analysis

Students:
Marco Bottino (s274110)
Maria Margherita Lovera (s278425)

Academic year
2020/2021

Contents

Chapter 1

Introduction

One of the most important applications of singular value decomposition (SVD) is the use of eigenfaces, in the field of human face recognition. A grey-scale image of a face can be descibed by a matrix of numbers corresponding to the values of the single pixels, which can be then reshaped into a one-dimensional vector.
Let's call X a matrix whose columns represents different human faces: by using SVD on X it can be found a new representation for human faces called *eigenface representation*.
Let the SVD of X be

$$X = U\Sigma V^T$$

Then the eigenvalue decomposition for XX^T is:

$$XX^T = U\Sigma\Sigma^T U^T = U\Lambda U^T$$

where Λ = diag (eigenvalues of XX^T).
The **eigenfaces** are therefore the columns of U associated with the nonzero singular values. They are mostly used as dimension reduction method, since most of the variance of any matrix X with n^2 elements can be described by considering only the first k eigenfaces, where $k << n^2$. Any face can therefore be represented by the components with respect to this new orthogonal basis.
We decided to show three different applications of this method (i.e. facial reconstruction, face recognition and classification of faces) by using two different datasets of faces for this project, with different properties:

- **Yale Faces B Dataset (YaleB)**
- **UTK Face Large Scale Face Dataset (UTK)**

Chapter 2

General analysis of datasets

Before starting our analysis we decided to explore a bit the datasets in our possess since we believe that their structure will influence our work in a quite significant way.

YaleB dataset

The YaleB datast contains 2410 pictures of 38 people. There are 64 photos for every person under different lighting conditions and each image has a dimension of 192x168 pixels.
In Figure 2.1 it is possible to observe 36 images of faces of different people and the average face of the dataset.

Figure 2.1: Images of 36 faces of different people (left) and average face (right) from dataset YaleB

By plotting the singular values on a logscale (see Figure 2.2), we have the singular values of the matrix with respect to the rank of truncation (x-axis, in fact we have 2410 because that is the number of columns we have) and the magnitude of the log of the singular value (y-axis).
It is interesting to notice the behavior. Indeed, while at first it slowly decreases, at a certain point it drops off to machine precision: this means that the first few singular values carry most of the variance in the faces and if we increase the number of singular values considered we have slower improvement (since by going on there is less information), but if we keep every single one of them we would have an almost inexistent error.

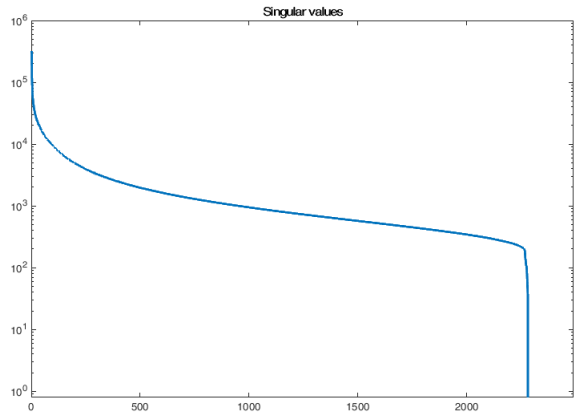

Figure 2.2: Singular values extracted from the training matrix of faces from YaleB

UTK dataset

The UTK dataset is a large-scale face dataset with long age span, which ranges from 0 to 116 years old. The faces are labelled and aligned and cover large variation in pose, facial expression, illumination, occlusion, resolution and other such. We worked on a sample of 2000 faces from different people from the dataset, where each image is 200x200 pixels. The distribution of the different categories is the following:

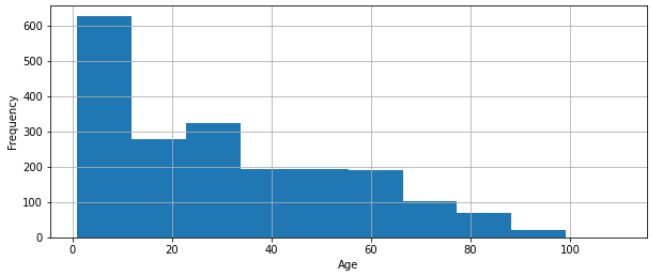

Figure 2.3: Age distribution

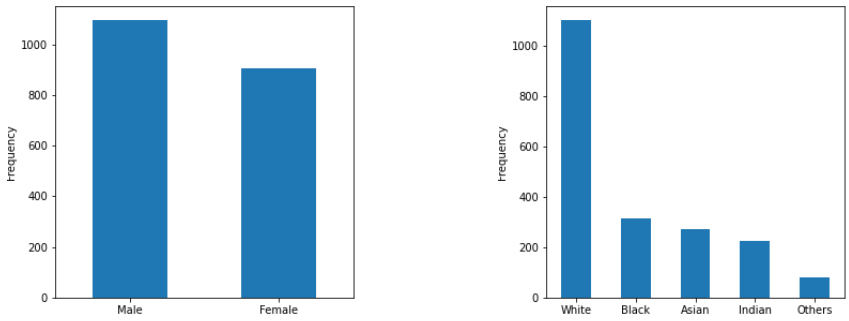

Figure 2.4: Gender and race distributions respectively

As we can see from Figure 2.3 the most frequent age range is that one related to very young people (between 0 and 16 years old). Concerning the gender (Figure 2.4 left), there are more male subjects than female and, regarding the ethnicity (Figure 2.4 right), in the dataset the white race is prevalent, followed by blacks, asians, indians and, finally, other races.

In Figure 2.5 we observe 36 images of faces of different people and the average face of the dataset.

Figure 2.5: Images of 36 faces of different people (left) and average face (right) from dataset UTK

We can already notice that this dataset is far more varied that the YaleB dataset: indeed, we can see from the 36 random faces printed that there are people of various age and race. As a consequence, the average face is far less defined with respect to that one of Figure 2.1.
Now, by considering the training matrix of faces composed of 1600 faces extracted from the UTK dataset, we can observe the same behavior in the singular values extraction as before (see Figure 2.6), motivated by the same reasons.

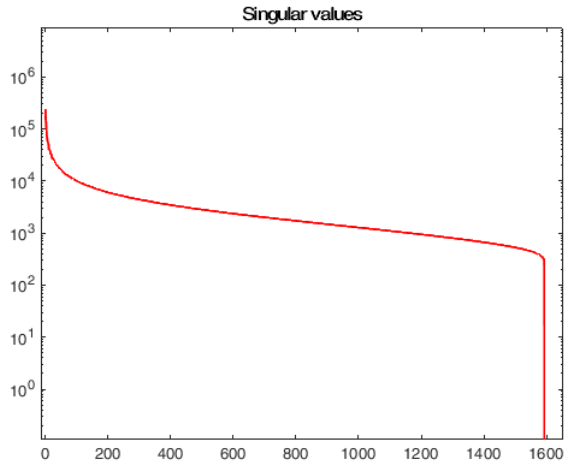

Figure 2.6: Singular values extracted from the training matrix of faces from UTK

Chapter 3

Face reconstruction

One of the main properties of the eigenfaces is that they can be used for image compression: indeed, what we can do is take a new image X and multiply it on the left by the new basis U_r^T (i.e. the first r columns of the basis U), obtaining a vector α with r elements, which represents the coefficients of the image on the new basis:

$$U_r^T X = \alpha$$

By multiplying again this vector by the matrix U_r we obtain an approximation $\tilde{X}$ of the original image:

$$\tilde{X} = U_r \alpha$$

The higher is r, the closer will the approximation be to the original image X.
In order to obtain the eigenfaces, we used the SVD: indeed, we decomposed the human faces into a set of orthogonal eigenfaces in order to capture the relevant features so to deal with the analysis in a lower dimensional space. The reconstruction was then made on faces not present in the data used to obtain the eigenfaces.
Before proceeding with the effective reconstructions on images of both datasets, we visualized the eigenfaces obained by reshaping the columns of matrix U and plotting the in a grid with the first 16 eigenfaces of both datasets, in order to see if we can already make some observations.

Figure 3.1: First 16 eigenfaces of datasets YaleB (left) and UTK (right) respectively

We noticed that in both datasets the first few eigenfaces did not contain any distinguishable characteristic features of individuals, while by going on they became more and more detailed. The most interesting fact about Figure 3.1 though is that it made quite clear the

difference between the two datasets: indeed, while the eigenfaces of YaleB dataset seem to be more detailed and all referred to middle aged people, those of the UTK dataset were less specific and more varied in terms of age. This is due to the big difference between our two working datasets in terms of variety and number of images of each individual. In both of the cases we can notice how every eigenface focus on some characteristic of the face: some have dark eyebrows, some pale noses, some prominent lips.

YaleB dataset

Concerning the YaleB dataset, we considered a training set containing all the pictures of the first 36 individuals over the total 38 and, as a first trial, we tried to reconstruct one of the two faces left out of the train set. Furthermore, we plotted the result of the reconstruction performed by using different numbers of eigenfaces (Figure 3.2).

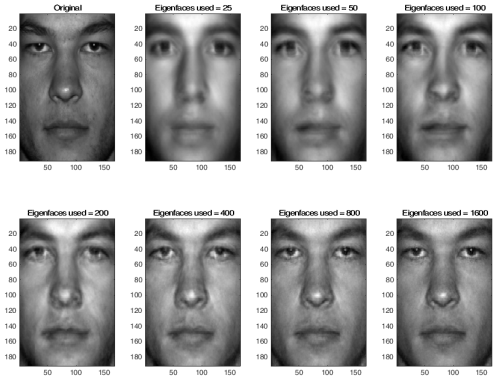

Figure 3.2: Reconstruction of a test face of dataset YaleB under different subspace dimensions

With 200 eigenfaces we alredy have a quite good face reconstruction, indeed we can almost recognize the person. But since we observed before that this dataset seem to be far less varied than the UTK one and, more importantly, for each face we have at least 60 photos in the dataset, we supposed that this very good reconstruction of the test face is mostly due to the dataset structure. So we tried to reconstruct our own faces with respect to the same training dataset, again by considering different numbers of eigenfaces (Figure 3.3).
As expected, in this case we have far less precision with respect to before. Indeed, for both Bottino's and Lovera's faces to be recognizable we needed at least 800 eigenfaces. Notice also that here with 200 eigenfaces we could not determine almost any particular facial features.

UTK dataset

Given the variety of the UTK dataset, we hoped to get better results in terms of reconstruction of our own faces. We considered a training set of 1600 faces out of the original 2000 and, in order to have a term of comparison, before reconstructing our faces we tried to reconstruct one of the 400 test faces left out (Figure 3.4).
With 200 eigenfaces we can already recognize the considered person. So we tried to reconstruct our own faces with respect to the same training dataset in order to uderstand if also in this case the dataset's structure affects a lot the analysis (Figure 3.5).

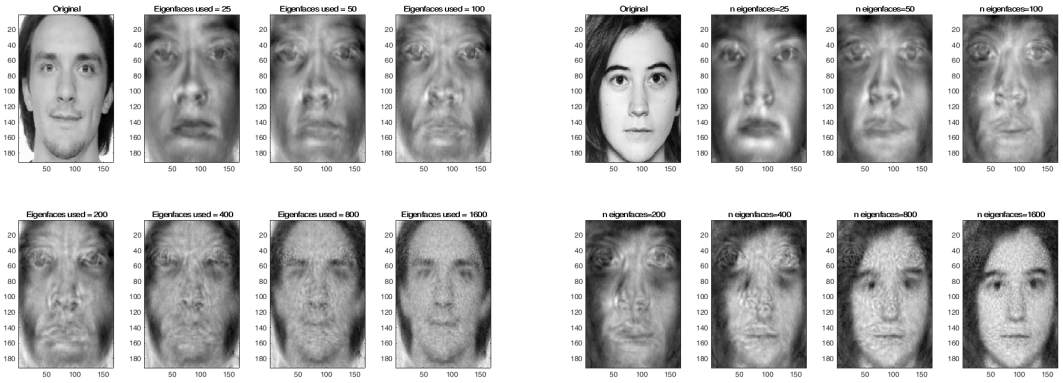

Figure 3.3: Reconstruction of students Bottino's and Lovera's faces under different subspace dimensions. Dataset used: YaleB

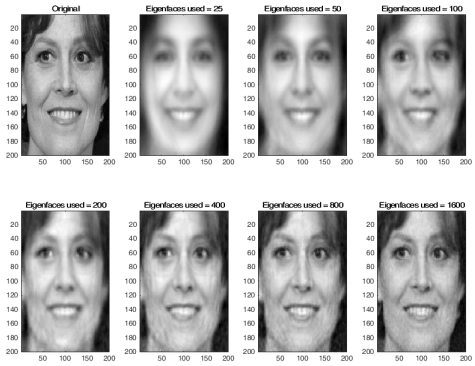

Figure 3.4: Reconstruction of a test face of dataset UTK under different subspace dimensions

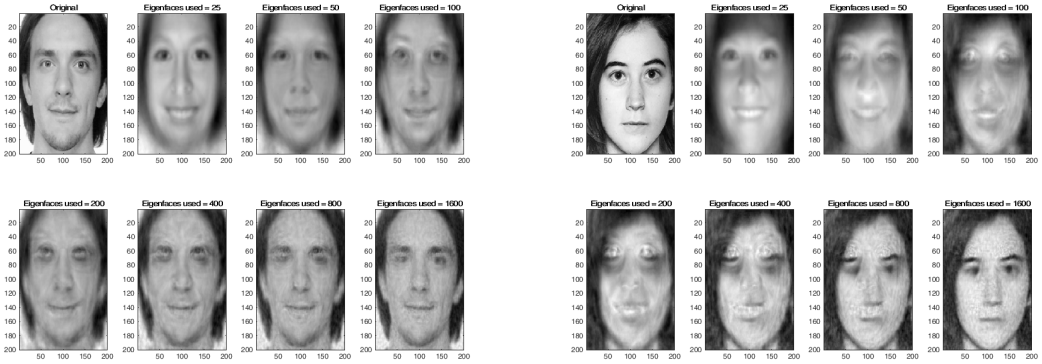

Figure 3.5: Reconstruction of students Bottino's and Lovera's faces under different subspace dimensions. Dataset used: UTK

Even if at least 800 eigenfaces are needed to recognize the face, with 200 eigenfaces there are some visible facial features of the individuals, while before there were none. This is

due to the differences between the training datasets used in the two analyses performed. The computational and memory saving though it's evident in both the cases: by using the eigenfaces evaluated we are able to reconstruct clearly a human face with just ~ 1000 values, whereas before we needed the values of more than 30000 pixels.

Chapter 4

Face recognition

Another important application of eigenfaces is face recognition. It can be performed with advanced classificators (we provide a toy example in Chapter 4), but the real advantage of this tool is its computational and time efficiency: by taking advantage of the low dimensionality of the images after being projected onto the eigenfaces space, one can simply evaluate the distance between the image of a face and all the images in our dataset and find the closest one to check if it corresponds to the same person.
By using the notation in Chapter 2, let's define

$$\alpha_r = U_r^T(X - Avg)$$

where r is the number of eigenfaces considered, X is our flattened image and Avg is the average of the images of our dataset. Then α_r is the centered vector of coordinates of a test face in the space of the first r eigenfaces: then the distance of that face from the i_{th} face in the dataset ($\tilde{\alpha}_i$) can be written as:

$$d_i = ||\alpha - \tilde{\alpha}_i||^2$$

- If $d_i < \epsilon_1$, then the i_{th} entry in the database is a candidate of recognition.
- If $\epsilon_1 < d_i < \epsilon_2 \ \forall i$, then X is the image of a face not present in the dataset.
- If $d_i > \epsilon_2 \ \forall i$, then X is not a face image.

The values of ϵ_1 and ϵ_2 vary according to r.

Finding the closest faces

We decided to focus on UTK dataset for this section, since it contains a larger variety of faces. We tried to evaluate the 7 closest faces to the one chosen, first using images from the dataset as test and then using our own faces. We tried different values of r, and also evaluated the closest faces according to the distance between the original pixels (which is a much heavier operation since we are comparing two vectors of 40000 elements). The results were very interesting, since not only they didn't vary for every $r > 400$, but they were also the same of the ones obtained by comparing the original vectors of pixels. By choosing $100 < r < 400$ the closest faces found started to be a little different, but only from the 4^{th} on, meaning that by projecting the faces into vectors of just 100 elements we were able to find efficiently at least the 3 closest faces.

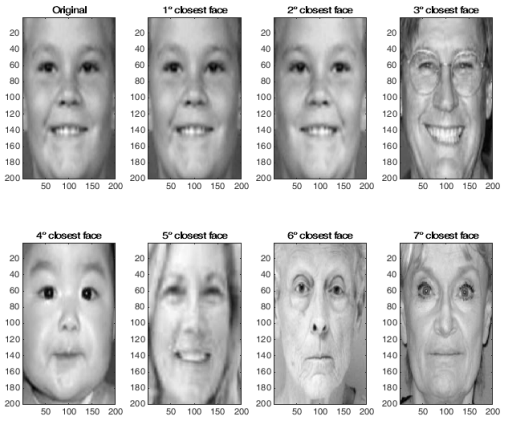

Figure 4.1: Closest faces in the dataset UTK to a face from the dataset

We report as an example (Figure 4.1) a face which contained a specular version of itself in the dataset. We can see that obviously the closest face found is the image itself ($d = 0$), and the specular face is the second. It's interesting to notice that this algorithm find similarities in the images, not in the persons in the common sense. The closest faces have different gender, ethnicity and even age, but all of them have some feature in common with the face chosen: some the shape of the face, some the color of the skin, some the presence of a smile.
Another interesting result was obtained by looking at the distances: even if the specular image of the child is almost identical to the original one, its distance was in the same order of magnitude to the ones of the other closest faces. We thought that this was a consequence of having projected our images on a different space, with a very different basis, but the results were analogous when evaluating the distances between the vectors of pixels.
We reported in Figure 4.2 also the closest faces found to our faces.

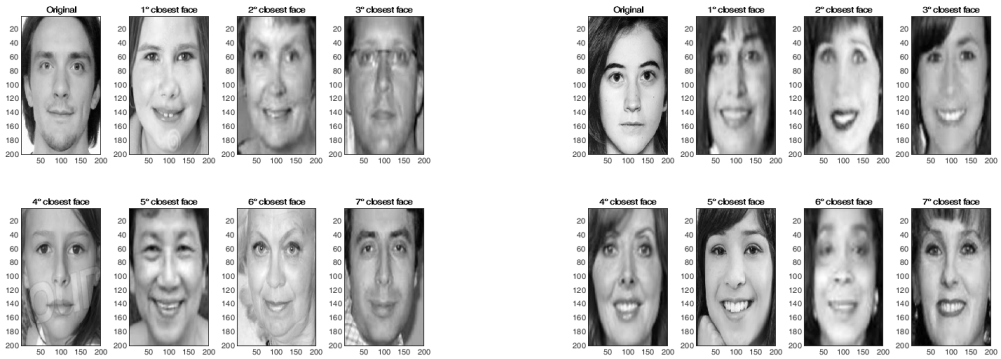

Figure 4.2: Closest faces in the dataset UTK to students Bottino and Lovera faces

Chapter 5

Classification of faces

Thanks to the faces reconstruction, we can also use the information present in the vector α (see Chapter 2) to classify people and recognize if a person is similar to another or if it is not. Furthermore, we can train a classification model: we chose the kNN as a simple example. We performed classification on both datasets, in order to analyze and discuss the differences. Concerning the YaleB dataset, we used the k-NN classifier more as a toy example since the structure of the dataset suggested an easy classification (remember that we have 64 pictures of every face present in the dataset).

YaleB dataset

In order to perform a toy example, we decided to narrow our dataset to the pictures of only two individuals and to build a classifier that, given as input a new image from one of this two people, would be able to recognise the right person.
Thus we choose randomly the subjects 5 and 6, we projected the flattened pictures of their faces onto the eigenfaces space (as explained in Chapter 2) and we classified the results with respect to the α coordinates (i.e. the eigenfaces scores). Then, to achieve visualization, we projected the results into the fifth and sixth eigenfaces directions (i.e. fifth and sixth principal components). We considered those two principal components because the first few principal components carry the generic features of human faces, while the others start to contain more specific distinct feature of each person. Since we wanted to distinguish the features of these pictures, we decided not to use the first few ones.
By looking at Figure 5.1 we observed that by considering person 5 and person 10, by taking all of the 64 images present in the dataset and by projecting those images on the fifth and sixth principal components the data happened to be strongly clustered: indeed, as expected, all the pictures of person 5 are well distinguished from those of person 10.
So we defined a k-NN classifier in order to see which would be the performance if we inserted a new image to classify. By considering different values for k (i.e. the number of neighbours) varying between 3 and 11, we obtained the confusion matrix of Figure 5.2.
The results obtained did not change for k varying from 3 to 11: this means that this classifier performed very well on the given data. We were expecting this result since, as stated before, the dataset has a very good structure in view of a classification algorithm.

UTK dataset

So we did a similar reasoning concerning the UTK dataset in order to see how much the situation would have changed. This time we did not consider two individuals (since in this

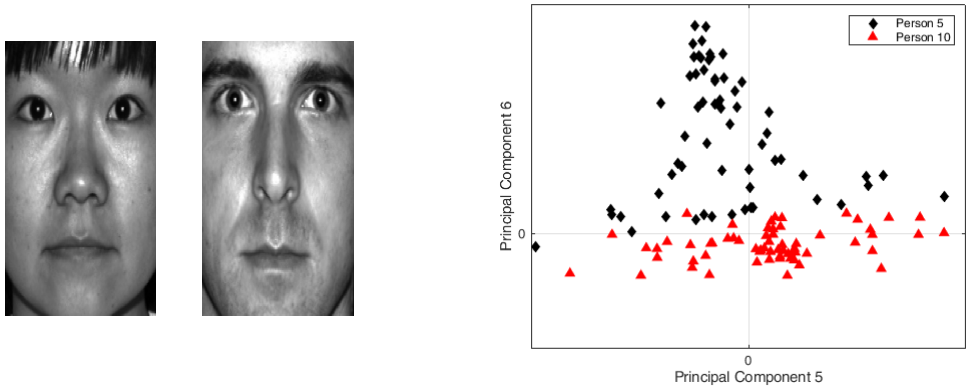

Figure 5.1: Pictures from two different subjects of YaleB (left, person 5 and 10 respectively) projected onto the principal components 5 and 6 (right)

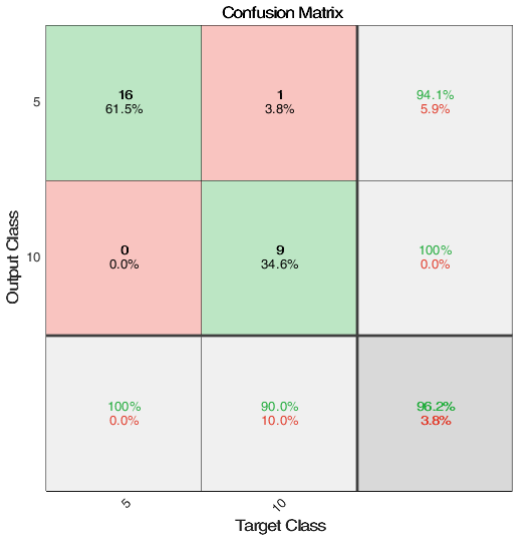

Figure 5.2: Results of the k-NN classification

dataset we do not have a lot of pictures for each person) but we decided to do a separate analysis according to categories: age, gender and ethnicity.
We observed then the average faces for each category out of curiosity (see Figures 5.3, 5.4 and 5.5). Notice that while gender and ethnicity were already labeled for the analysis, we had to create a division in classes for the age of the subjects. Finally, this was the result:

- age: 0 = 0-16 years old, 1 = 16-35 years old, 2 = 35-60 years old, 3 = over 60 years old
- gender: 0 = male, 1 = female
- ethnicity: 0 = white, 1 = black, 2 = asian, 3 = indian, 4 = others

Then for each category we considered the first 10 principal components and we looked for the 2 PCs that separated better the data according to the category taken in exam. We used

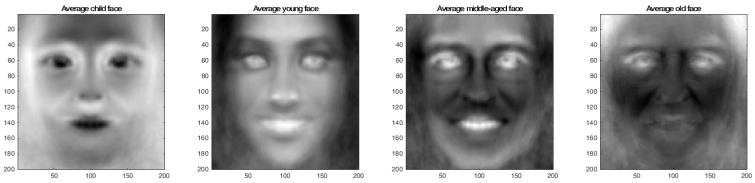

Figure 5.3: Average faces divided according to age (from the left: child, young, middle-aged, old)

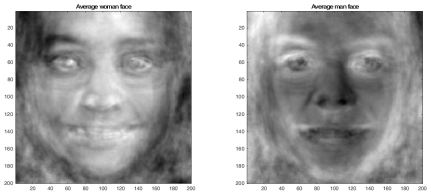

Figure 5.4: Average faces divided according to gender(from the left: woman, man)

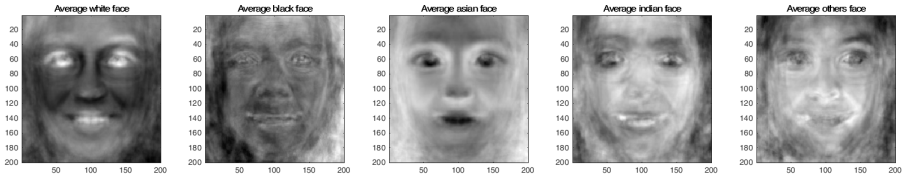

Figure 5.5: Average faces divided according to ethnicity (from the left: white, black, asian, indian, others)

as metric the average euclidean distance between the centroids of the different clusters. For the age category the best PCs found were the first and the third and the relative separation is represented below:

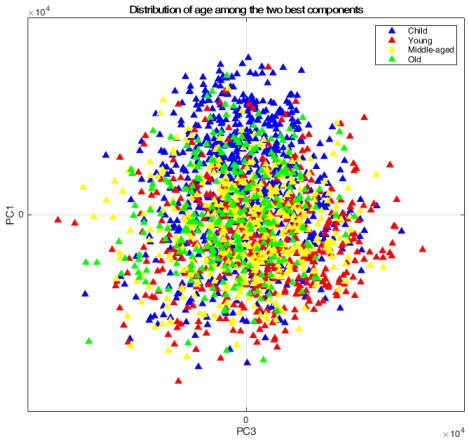

Figure 5.6: Scatter-plot of the pictures from UTK dataset divided according to the age and projected onto the best principal components

Notice that in this case the separation found was far worse than that one found for the

YaleB dataset: we expected this outcome, since the UTK dataset is more varied and it has far less pictures of each person's face. Furthermore, we expect to see the same results also for gender and ethnicity categories.
For the gender category the best PCs were the first and the seventh and, respectively, for the ethnicity categories the first and the second. Here there are the separations found:

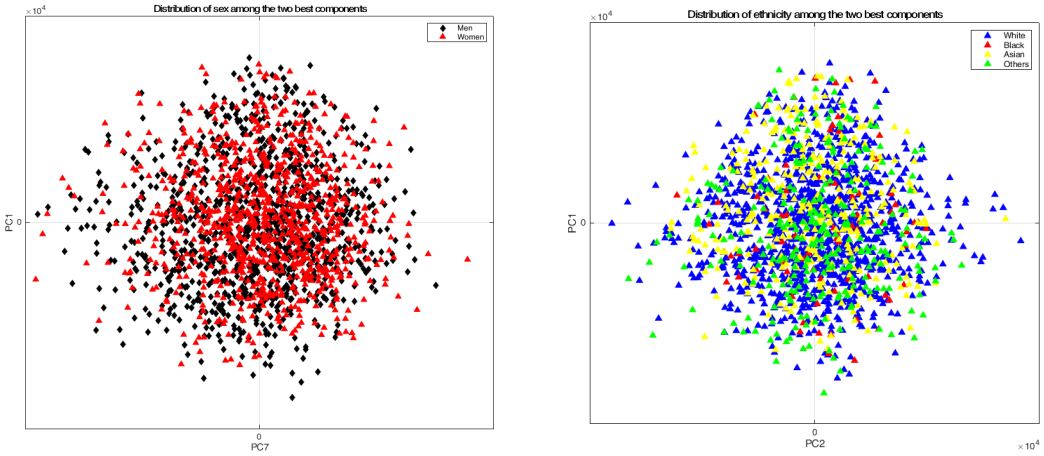

Figure 5.7: Scatter-plots of the pictures from UTK dataset divided according to gender and ethnicity respectively and projected onto the best principal components

As expected, also in these cases we found a very bad clustering: this made us think that the k-NN classifier would not perform well on this dataset. But in order to complete the analysis, we implemented the k-NN classifier by considering each category at a time and by looking at the best combination between the number of dimensions n considered (since the idea is to perform the classifier on the dataset projected on the first n eigenfaces) and the number of neighbors k chosen. We considered n varying from 3 to 20 and k from 1 to 15 (just considering odd numbers).
We found the following results:

Category	Best (n,k)	Error rate
Age	(3,7)	0.4585
Gender	(6,15)	0.4520
Ethnicity	(7, 11)	0.4715

Table 5.1: Best combination of parameters and k-NN classifier loss function for each category

By looking at the loss function's values for each category, we observed that we always correctly assign less than 56% of the images: this means that the k-NN classifier in this case perform quite badly.
For sure it is possible to build a good classifier that works well also with this dataset, but since this is not the aim of our homework we did not try different classification tecniques.

Chapter 6

Conclusions

The use of eigenfaces is a valid example of how most of the information of a complex dataset, like the pixels of an image, can be summarized by a far smaller number of features. This has advantages in images storage and in delivering low-dimensional inputs for many applications which result in better computational efficiency for any kind of algorithm. This is very important in fields like face recognition, where the desired output must be often obtained almost in real time.