FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Computational Methods for Detection, Estimation and Identification 2022/2023

Assignment 6 - Eigenfaces

Manuel Alberto Dionísio dos Santos
2019231352

1 Overview

This assignment aims to apply a popular method using PCA for facial recognition purposes named Eigenfaces. This method was first published in 1991, authored by Matthew Turk and Alex Pentland [6].

The method uses the singular value decomposition (SVD) of the covariance matrix of a set of training faces (normalized and extracted from the mean face) to estimate the principal components and express every face as a linear combination of these principal components (called Eigenfaces). An example of the resulting Eigenfaces can be seen on figure 1.

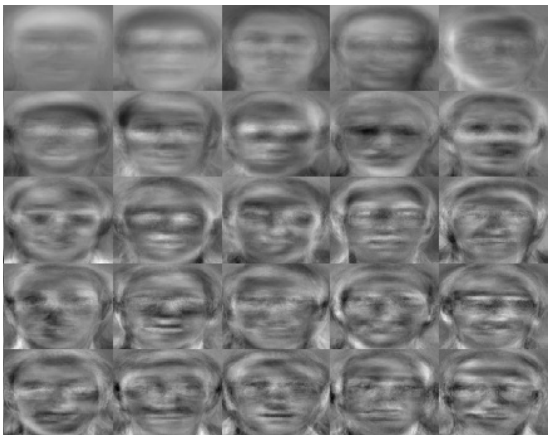

Figure 1: Example of Eigenfaces

The method is then able to express each new face as a linear combination of all vector in the basis, the Eigenfaces. As expected, there is an information loss in the projection of each new face to the Eigenfaces basis. However, it can be easily shown that the data reduction is the minimum, as the principal components selected are the ones that optimize the information retained.

In this assignment, the method is explored through some initial experimentation's and later through the implementation in Matlab of a face recognition system. As the main source of information for research and implementation of the method, the book entitled Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control [2] was used.

2 Methodology

To implement the Eigenfaces method, this steps must be followed:

1. **Data collection**: Collect a training set of **k** face images, typically grayscale, which will be used to derive the eigenfaces;
2. **Pre-processing**: Pre-process the images to standardize factors such as size, orientation, and illumination. This step ensures consistency across the dataset. Each image **(h by w)** is then reshaped into a single column vector **((h x w) by 1)**, and a new matrix **X** is built, composed of all **k** column vectors concatenated **((h x w) by k)**;
3. **Subtract average face**: In order to center the dataset, the average image has to be calculated and then subtracted from each original image;
4. **Compute SVD of X**: Using the Singular Value Decomposition (SVD), separate the principal components of $\mathbf{X} = \mathbf{USV^T}$. Each column of **U** has size **((h x w) by 1)** and can be reshaped into an **(h by w)** image. This image is called an Eigenface;
5. **Choose the principal components**: By analysing the obtained singular values and establishing a threshold **r**, one can then select the Eigenfaces with the most energy, $\mathbf{U_r}$. In practical applications, most faces can typically be identified using a projection on between 100 and 150 eigenfaces, so that most of the eigenvectors can be discarded. This dimensionality reduction is of extreme importance, since the obtained model is faster and more compact with less Eigenfaces, and slower, but more accurate, with more Eigenfaces.
6. **Face representation**: Using the obtained Eigenfaces, one can then represent any face in the training or testing datasets as a linear combination, $\tilde{\mathbf{x}}$, of the eigenfaces. This is done by projecting the image, **X**, onto the eigenface space, $\tilde{\mathbf{x}} = \mathbf{U_r U_r^T X}$.
7. **Face recognition**: It is even possible to use the obtained model to recognize an unseen test face image, **Y**. To do so, one must project it onto the eigenface space and calculate the resulting coefficients, $\alpha_\mathbf{Y} = \mathbf{U_r^T Y}$. Then is a matter of comparing these coefficients with the coefficients of the training set faces, $\alpha_\mathbf{X} = \mathbf{U_r^T X}$, using a distance metric (e.g., Euclidean or Mahalanobis distance). The closest match or matches indicate the identity or similarity of the input face.

3 Acquire and pre-process data

The first step in the Eigenfaces method is to prepare a set of face images. Ideally, they should have been taken under the same lighting conditions, and must be normalized to have the eyes and mouths aligned across all images. They must also all have the same pixel resolution **(h x w)**.

Each image is treated as one vector, simply by reshaping the original image, resulting in a single column with h x w elements.

For this implementation, it is assumed that all images of the training set are stored in a single matrix $\mathbf{X_train}$, where each column of the matrix is an image.

For this assignment, three different datasets were experimented on: Yale [7], AT&T [1] and Extended Yale Face Database B [3] [5] (from now on referred as YaleB).

In order to standardize all photographs (same pixel resolution, all in grayscale) of each dataset, a script named *processData.m* was created. For each dataset, the script gathers all the photos, grayscales them, normalizes them, reshapes each one into a column vector and concatenates all of the column vectors into an unique matrix.

This matrix, along with the original image resolution, the number of subjects per dataset and the number of photos per subject is then saved into a *.mat*. This file is then used in each main script as the main data source. This process makes it easier to access all the data.

Figure 2: All subjects of Yale dataset

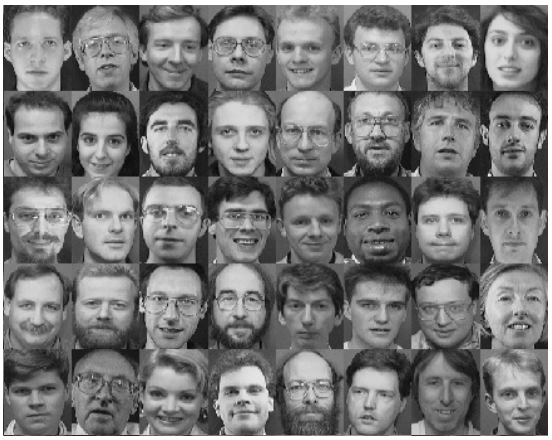

Figure 3: All subjects of AT&T dataset

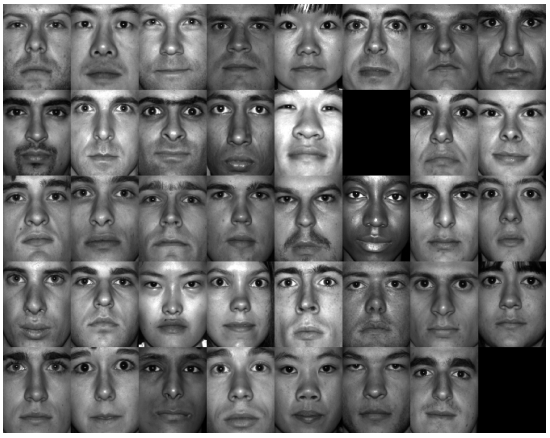

Figure 4: All subjects of YaleB dataset

After some initial tests, the YaleB proved to wield better results on the long run, since all the faces are aligned and cropped to the faces, and as so do not present as much background noise. As so, all the results from now on will refer to the YaleB dataset.

4 Division into training and test datasets

The YaleB dataset presents 38 subjects, from 1 to 39 (14 excluded) with around 64 photos each, in different lighting conditions. Each photo has a dimension of (192 by 168) pixels (h by w).

The training dataset will consist of 64 photos from each of the 35 subjects, from subject 1 to subject 16 (subject 14 excluded). It will be stored in the format of a (192x168 = 32256 by 35x64 = 3204) matrix.

The testing dataset will consist of 64 photos from each of subjects 36, 37 and 38. It will be stored in the format of a (192x168 = 32256 by 3x64 = 192) matrix.

This division is the one used for the remaining of the experiment, except when explicitly mentioned otherwise.

In the developed script is possible to test the effect of changing the ratio of training to testing datatsets. Is possible to affirm that with more training images, the model is computationally heavier but shows better results, as the face space is broader.

5 Average Face

The next step in the Eigenface method is to subtract the average face from the datasets. In order to do so, it is necessary to obtain said average face. To do so, the Matlab function *mean()* was used to obtain a (192x168 = 32256 by 1) column vector. When reshaped to and (192 by 168) matrix, an image arises, as seen on figure 5.

Figure 5: Average Face of YaleB dataset

Although the training dataset presents images in different lighting conditions, the they average out and form an image that closely resembles an human face.

The obtained average face is then subtracted from each one of the training images, resulting in a centered training dataset.

6 Getting Eigenfaces

With a now centered and ready training dataset, the next step is to finally obtain the Eigenfaces. In order to do so, the SVD is applied to the training dataset. In Matlab, *svd()* function is used, with the argument *'econ'*, in order to obtain only the first eigenfaces, corresponding to the first non zeros singular values. The first 25 eigenfaces can be seen on figures 6 and 7.

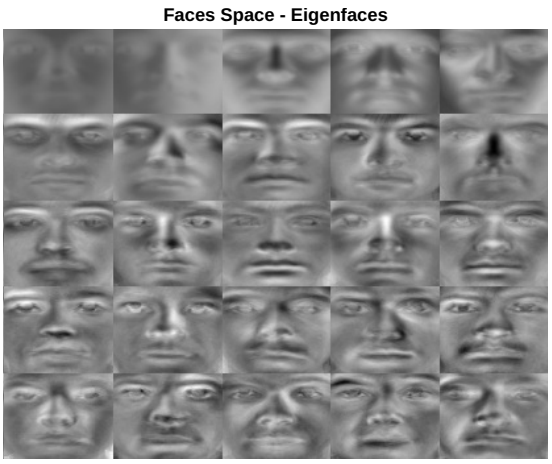

Figure 6: First 25 eigenfaces

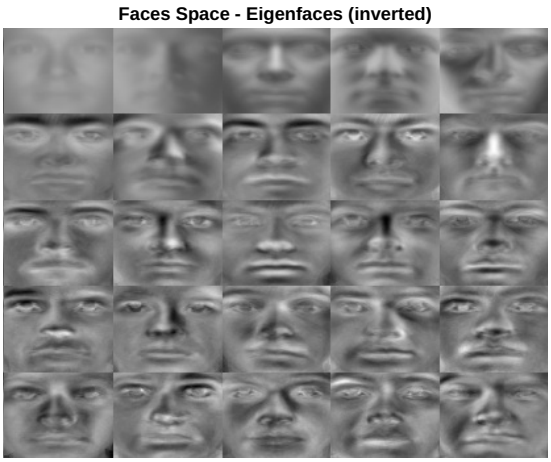

Figure 7: First 25 eigenfaces (inverted)

In order to perceive more details, the inverted variant (multiplying each value by -1) is also shown. This is only for visualization purposes, as all the calculations were and will be made using the unaltered values.

As it is possible to see, the first 3 to 5 eigenfaces present the traces common to all human faces (eyes, nose, mouth), and as we go further into the eigenfaces they start to contain more specific characteristics of each subject. This happens because the first eigenfaces (and singular values) capture most of the variance and energy of the data, and as so contain the traits common to all humans. As the energy decreases, so does the common factor.

7 Truncation

In order to represent the face images in a lower-dimensional space, which can simplify subsequent computations and reduce storage requirements and computation times, it is ideal to select only a subset of the most informative singular vectors, by truncating the singular values at a certain threshold.

The first try in applying a dimensional reduction consisted in applying the Discrete Picard Condition to evaluate a threshold.

Using the same code developed for Assignment #4, a value of p = 2290 was obtained.

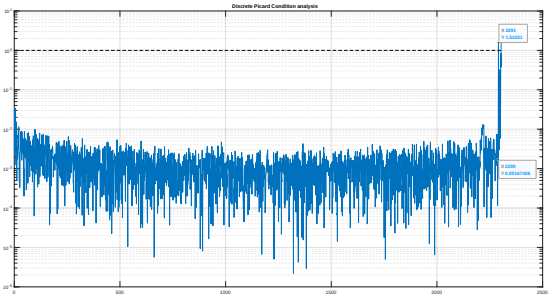

Figure 8: Discrete Picard condition Analysis

This value is not helpful, since preserving 2290 of 2304 does not consists of a valuable dimensionality reduction.

After some research, a new criterion was selected. The selected criterion was the Gavish-Donoho hard threshold [4]. It stands that for a non-square m-by-n matrix with $m \neq n$, the approximate optimal location when σ is unknown is $\hat{\tau^*}$, as shown in equation 1.

$$\hat{\tau^*} = \omega(\beta).y_{med} \tag{1}$$

In equation 1, $\omega(\beta)$ is a value that can be computed by equation 2, y_{med} is the median empirical singular value and $\beta = m \div n$.

$$\omega(\beta) \approx 0.56\beta^3 - 0.95\beta^2 + 1.82\beta + 1.43 \tag{2}$$

Applying equations 1 and 2, a new value for p arises, as seen if figure 9 . This value, from now on referred to as newP = 434. This new value for a truncation is much more reasonable and, although still a bit high, proves to provide good results.

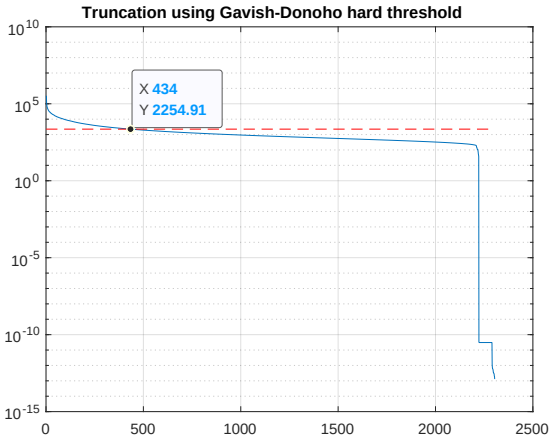

Figure 9: Gavish-Donoho threshold Analysis

Figures 10 and 11 show the reconstruction of faces from the linear combination of Eigenfaces. As can be seen, the newP value of 434 creates a reconstruction with better quality than those that were truncated earlier and a quality as good as that of models truncated later.

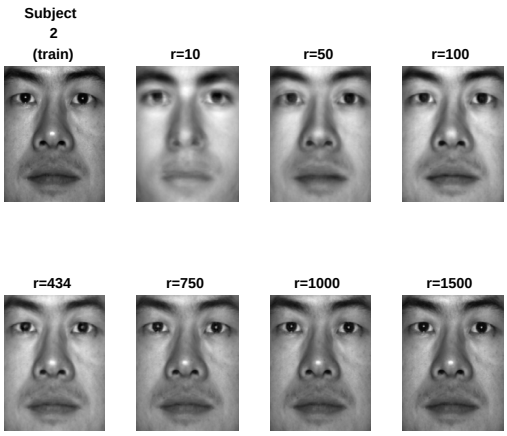

Figure 10: Train image Reconstruction

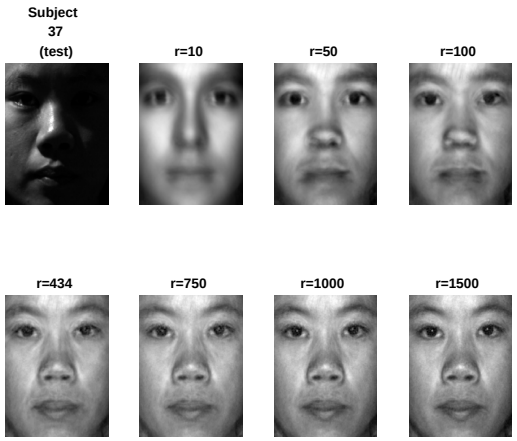

Figure 11: Test image Reconstruction

On the developed script, is possible to explore the effect of the truncation with a dynamic slider. It is possible to verify that with more eigenfaces being used, the reconstruction closely resembles the test image, has it has more details. With less eigenfaces being used, the final result appears more generalized, as it would be expected of the use of eigenfaces with more energy and information.

Also, as expected, the model proves to more easily reconstruct faces from the training dataset than those of the test dataset, as the first ones were used on the training and their specific characteristics is embedded in it, while the lattes are totally new to the model.

8 Face Recognition

In order to successfully recognize and distinguish several people using the Eigenfaces method, is necessary to build an unique feature vector of each subject, named α. It serves as a fingerprint of the subject, uniquely identifying it amongst all the subjects.

8.1 Study Model

In order to fully understand the concept, an initial test run was conducted with only 3 subjects (1, 5 and 38). For the test, the identity of each subject was built using 28 input images. To do so, 28 of their images were centered, stretched and concatenated into a single matrix. Each matrix was then projected into the face space composed of all the the images of the dataset. The projection of a subject identity yields a set of coordinates, in N dimensions (N is the number of Eigenfaces considered).

In this test, N = 3, being the the components used 5, 6 and 7. From the projection into this 3 components arises a set of 28 3D points for each identity, that compose a cluster, as seen of figure 12.

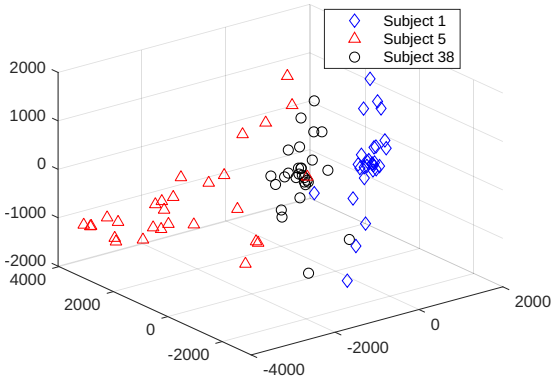

Figure 12: Clusters for each identity

Note that the components used were 5, 6 and 7 and not the first ones (like 1, 2 and 3). This arises from the fact that the first components, as mentioned before, contain the majority of the variance and energy of the dataset, and as so, contain the traits common to all. Using not the first components (such as 5, 6 and 7) works best since this components contain more detailed traits of each individual, as as so, are more useful to distinguish between subjects.

To test the model, one image from each subject, not used in the projection was used. They were centered and stretched in order to being regularized, as seen in figure 13.

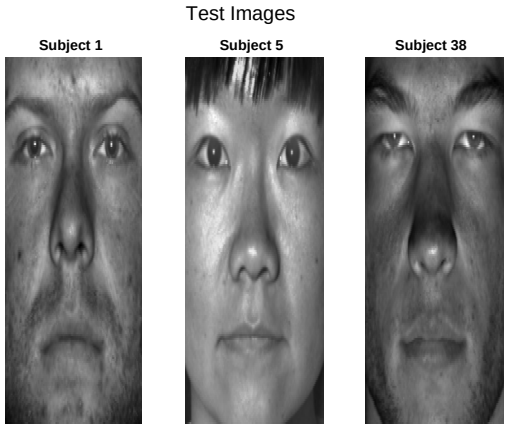

Figure 13: Treated test images

Each one of them was projected into the face space, yielding a triplet of coordinates from each image. Using the Mahalanobis distance (Matlab function *mahal()*), the distance between each test identity and the identities for each model subject was calculated. The results can be seen on table 1.

Distance to / Test Subject	Cluster 1	Cluster 5	Cluster 38
Subject 1	0.3702	10.8232	23.5216
Subject 5	456.0694	1.7502	163.7230
Subject 38	25.9872	4.5678	0.7117

Table 1: Distance of test image to each subject cluster (N = 3)

From this first run, the results proved to be positive, as each test image was successfully identified.

In order to study the effect of changing the number of input images for each identity, a slider was inserted that easily permits to change the number of images. Be conducting some tests, the conclusion is that the increase in input images leads to a more dense cluster for each identity in the feature space.

The effect of changing the number of Eigenfaces of the feature space was also experimented on. By changing the components used from N = 3 to N = 7, each identity cluster becomes more isolated and further from one another. This change leads to an increase in the Mahalanobis distance when trying to classify each test image, as seen on table 2

Distance to / Test Subject	Cluster 1	Cluster 5	Cluster 38
Subject 1	7.4274	106.7970	1038.5
Subject 5	2647.1	5.6967	6163.9
Subject 38	547.8760	128.8134	7.4627

Table 2: Distance of test image to each subject cluster (N = 7)

Although the distances have overall increased from the test image to each cluster, the classification continues to be correct.

8.2 Final Model

Gathering the techniques used to build this simple model, a more complex model, capable of identifying a subject from any of the 38 was implemented. For this model, and using the exploration on the parameters done on the study model, it was defined that each training identity would have 60 input images, randomly chosen from the 64 of each subject. The remaining 4 will be used for testing. The eigenfaces used will be from 5 to 11, giving N = 7.

Having prepared all data, the calculation of the coordinates of the feature space and of the test images was then done. The feature space values were stored in a (7 by 60 by 38) matrix (7 components by 60 images by 38 subjects), and the test values were stored in a (7 by 4 by 38) matrix (7 components by 4 images by 38 subjects).

Inside a nest of for loops, the distance of each test coordinates to each feature cluster was then calculated, and compared with all the others. The smallest distance would then dictate the prediction. If it was right, a right counter was implemented. If it was wrong, a wrong counter would be incremented. The accuracy was then calculated by dividing the right counter by the total number of tests.

The results were not satisfying, as the accuracy stayed always around **50%**. Several adjustments to the variables were tried, but the accuracy did not improve significantly.

In order to easily visualize the results, a confusion matrix was also created and populated at each test. It can be seen in figure 14

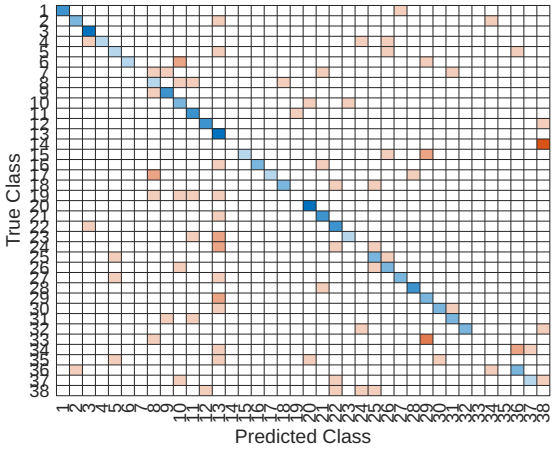

Figure 14: Confusion Matrix

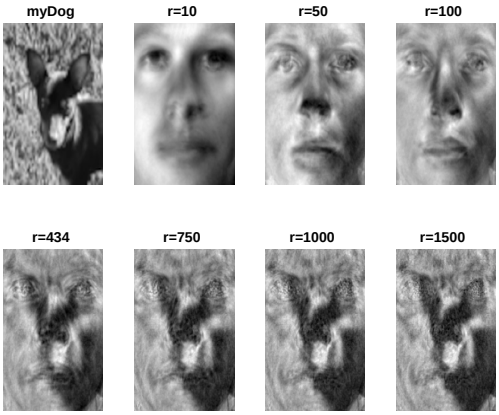

Figure 16: Nina image Reconstruction

9 Only Faces? Well...

While researching about Eigenfaces, one example that frequently appeared was the use of Eigenfaces to reconstruct images of other subjects that not faces. Basically, by projecting an image from virtually anything, is possible to reconstruct it using the face space.

In order to try this concept, an image from my cat, Pitucha, from my dog, Nina, and from the robot with I've been working on IS3L, Guida, were used as test images. The results, seen of figures 15, 16 and 17 prove to be very good.

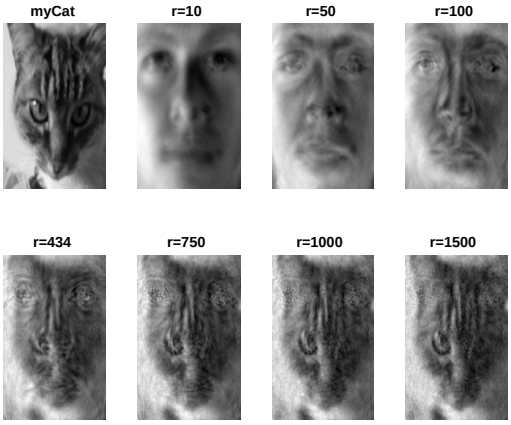

Figure 15: Pitucha image Reconstruction

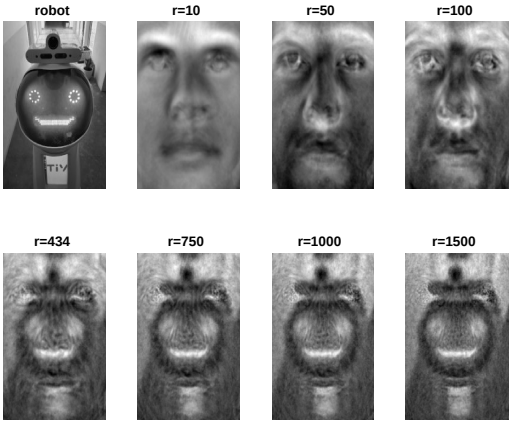

Figure 17: Guida image Reconstruction

10 Conclusion

I finish this assignment happy with the work developed, not only because I genuinely liked to experiment with Eigenfaces, but also I'm satisfied with the knowledge I've gained.

In terms of results, the reconstruction of test images using Eigenfaces proved to be very successful. On the other side, although the Face Recognition model results are not the best, they prove that it's possible to create a Facial Recognition system based on the Eigenfaces method.

Although this method is has now more than 30 years since first developed, and although nowadays new methods prove to be more efficient and accurate, the overall simplicity of it fascinates me.

References

[1] *AT&T Dataset*. URL: https://www.kaggle.com/datasets/kasikrit/att-database-of-faces?resource=down.

[2] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. DOI: 10.1017/9781108380690.

[3] *Extended Yale B Dataset*. URL: https://paperswithcode.com/dataset/extended-yale-b-1.

[4] Matan Gavish and David L. Donoho. *The Optimal Hard Threshold for Singular Values is 4/sqrt(3)*. 2014. arXiv: 1305.5870 [stat.ME].

[5] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. "From few to many: illumination cone models for face recognition under variable lighting and pose". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.6 (2001), pp. 643–660. DOI: 10.1109/34.927464.

[6] Matthew Turk and Alex Pentland. "Eigenfaces for recognition." In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86. DOI: https://doi.org/10.1162/jocn.1991.3.1.71.

[7] *Yale Dataset*. URL: https://www.kaggle.com/datasets/olgabelitskaya/yale-face-database.