Face Recognition Using Eigenfaces

Rafael Luís

Computational Methods for Detection, Estimation and Identification
Department of Electrical & Computer Engineering
University of Coimbra

26th July, 2023

I Introduction

The use of data reduction for description of the most relevant features of a dataset has several applications. Particularly, the **Principal Component Analysis** (PCA) has been extensively used for applications in imaging.

In this project we aim to apply a popular method using PCA for facial recognition purposes, **Eigenfaces**. This method was first published in 1991, authored by Matthew Turk and Alex Pentland [1].

A Principal Component Analysis

PCA is a popular technique for analyzing large datasets containing a high number of dimensions/features per observation, increasing the interpretability of data while preserving the maximum amount of information.

It is a statistical technique for reducing the dimensionality of a dataset. This is accomplished by linearly transforming the data into a new coordinate system where most of the variation in the data can be described with fewer dimensions than the initial data.

B Singular Values Decomposition to perform Principal Component Analysis

But how exaclty can we use the SVD of the data matrix to perform dimensionality reduction?

Let the real values data matrix $\mathbf{C}$ be of $n \times p$ size, where n is the number of samples and p is the number of variables. Let us assume that it is centered, i.e. column means have been subtracted and are now equal to zero.

Then the $p \times p$ covariance matrix $\mathbf{C}$ is given by $\mathbf{C} = \mathbf{X}^T\mathbf{X}/(n-1)$. It is a symmetric matrix and so it can be diagonalized:

$$\mathbf{C} = \mathbf{V}\mathbf{L}\mathbf{V}^T \tag{1}$$

where $\mathbf{V}$ is a matrix of eigenvectors (each column is an eigenvector) and L is a diagonal matrix with eigenvalues λ_i in the decreasing order on the diagonal

The eigenvectors are called *principal axes* or *principal directions* of the data. Projections of the data on the principal axes are called *principal components*, also known as *PC scores*, these can be seen as new, transformed, variables. The j^{th} principal component is given by j^{th} column of $\mathbf{XV}$. The coordinates of the i^{th} data point in the new PC space are given by the i^{th} row of $\mathbf{XV}$.

If we now perform SVD of $\mathbf{X}$, we obtain a decomposition

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{2}$$

where $\mathbf{U}$ is a unitary matrix (with columns called left singular vectors), $\mathbf{S}$ is the diagonal matrix of singular values $\mathbf{s}_i$ and $\mathbf{V}$ columns are called right singular vectors. From here one can easily see that

$$\mathbf{C} = \frac{\mathbf{V}\mathbf{S}\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}}{n-1} = \mathbf{V}\frac{\mathbf{S}^2}{n-1}\mathbf{V}^T \tag{3}$$

meaning that right singular vectors $\mathbf{V}$ are principal directions (eigenvectors) and that singular values are related to the eigenvalues of covariance matrix via

$$\lambda_i = \frac{s_i^2}{n-1} \tag{4}$$

Principal components are given by

$$\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V} = \mathbf{U}\mathbf{S} \tag{5}$$

C Eigenfaces

An $n \times n$ matrix $\mathbf{A}$ has an eigenvalue λ with an associated eigenvector $\mathbf{x}$, and

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \tag{6}$$

To compute the eigenvalues and eigenvales, lets multiply it by the identity matrix,

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{I}\mathbf{x}$$

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = 0$$

To find nonzero eigenvectors, the matrix $\mathbf{A} - \lambda\mathbf{I}$ must be singular,

$$det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

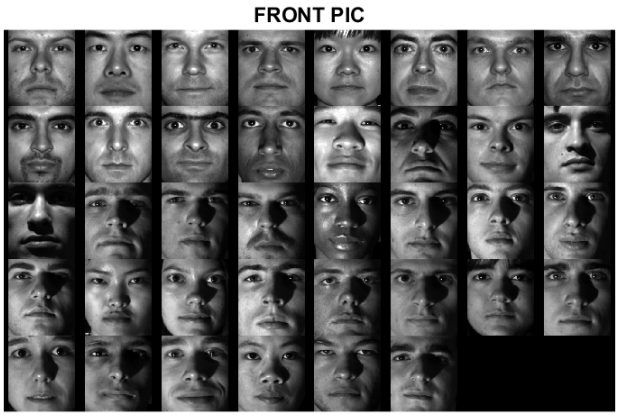
We locate the input data set's eigenfaces at a high level. These can be viewed as a basic collection of faces that can be used to represent other faces through linear combinations. The weight coefficients required to represent all of the faces in the data set by the eigenfaces can be calculated and stored. We can calculate the weight coefficients required to represent a new face with our eigenfaces when it is provided to us. By comparing the computed weights of the new face with the weights of all previously known faces in the database, we can perform basic facial recognition.

II Work

A SVD and Eigenfaces

The Extended Yale B dataset was used in the first part of this assignment. This dataset is composed of 2414 frontal-face images sized 192x168, 64 images from each of the 38 persons (subjects). This images were captured with different facial expressions and light conditions. Figure 1a illustrates some of the different light conditions on different subjects and figure 1b shows all the subjects with front light.

(a) Faces randomly choosen



(b) Faces from a single subject

Figure 1. Subjects

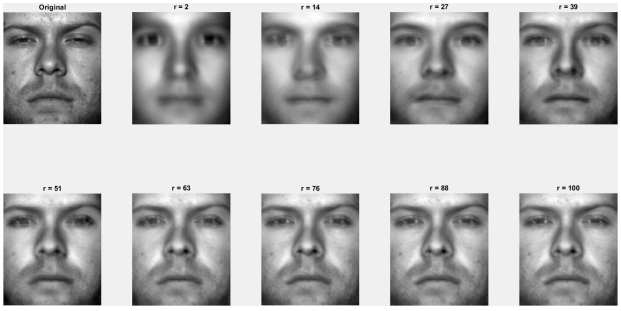
This dataset underwent a preprocessing step to prepare for further analysis. The images were converted to a grayscale and the pixels were normalizated to a range between [0,1]. The images were next stored in a matrix where each column represents one image. The result was a matrix of size 32256x2414. Following this procedure, the images were divided into training and testing sets, and the mean face was calculated using the training images. The mean face is an average representation of the faces from the dataset. It represents the collective features and characteristics among all the faces in the dataset. It can be obtained by calculating the average values of the pixels across all the training images. The obtained mean face is represented in 2

Figure 2. Mean Face

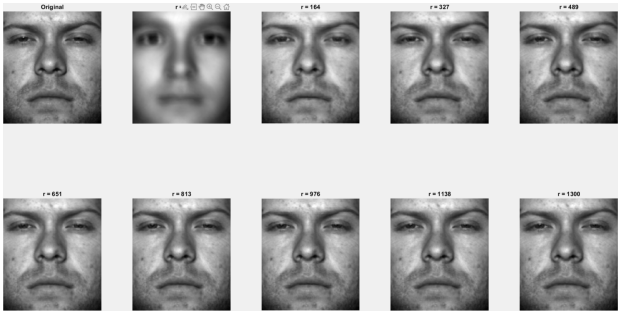
In order to reconstruct images using eigenfaces, it is performed a Singular Value Decomposition SVD. The SVD is applied to the training images, which have previously undergone preprocessing steps to normalize the pixels and convert them to grayscale. The eigenfaces are the principal variations in the face images. These eigenfaces capture the most significant facial features in the training dataset.

To reconstruct an image, a certain number r of eigenfaces were selected. These eigenfaces represent different levels of detail and features that contribute to reconstruct the image and the more eigenfaces are presented the closer to the original image the result will be. This result is archieved by calculating the dot product between eigenfaces and the difference between them and the mean face. The result is added to the mean face.

(a) Reconstruction using from 2 to 50 eigenvalues



(b) Reconstruction using from 2 to 100 eigenvalues

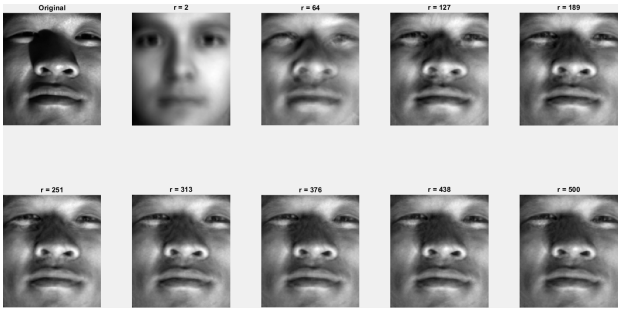


(c) Reconstruction using from 2 to 1300 eigenvalues

Figure 3. Reconstruction using different numbers of eigenvalues

These reconstructions were made to several images showing similar results.

(a) Reconstruction using from 2 to 500 eigenvalues



(b) Reconstruction using from 2 to 500 eigenvalues

(c) Reconstruction using from 2 to 500 eigenvalues

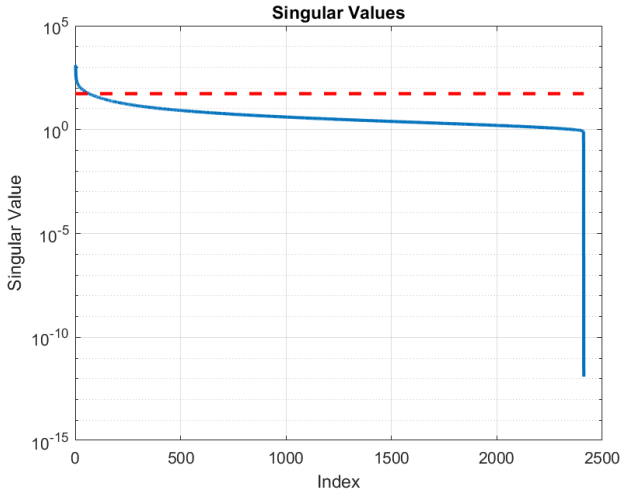
Figure 4. Reconstruction of different images

The singular values in diag(S)

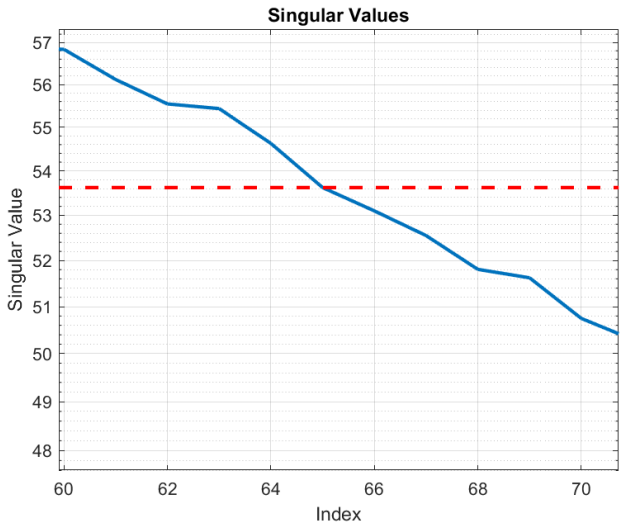
$$A = U \cdot S \cdot V'$$

can be used to determine the optimal hard threshold for retaining the most significant components of a matrix. One common approach is to compute the cumulative energy of the singular values. The energy represents the proportion of total variability captured by each singular value. By summing up the squared singular values and normalizing the result by the total sum of squared singular values, the cumulative energy can be obtained. Based on the cumulative energy plot, one can determine a threshold value that retains a desired percentage of the total energy. This threshold allows to reduce the dimensionality or compress the data by defining the number of singular values to keep, preserving the most significant information.

By running the code and analyzing the cumulative energy plot, we can make informed decisions about the level of dimensionality reduction or data compression needed. A threshold of 0.95 requires preserving 65 eigenfaces, which captures 95% of the total energy and retains a significant portion of the data's essential information (Figure 5). In scenarios where a higher level of data compression is acceptable, reducing the threshold to 0.9 allows retaining 24 eigenfaces while still preserving 90% of the information. On the other hand, for applications where maximum information retention is critical, a threshold of 0.99 necessitates keeping 310 eigenfaces, ensuring that 99% of the total energy is retained.



(a) Singular Values



(b) Singular Values (Amplied)

Figure 5. Singular Values

B Facial Recognition

For this project, two different approaches were tested to implement a facial recognition system using the YaleB dataset. Both approaches utilize Principal Component Analysis (PCA) and Eigenfaces for face representation, but they differ in the face recognition step. One crucial factor in the testing process was that the recognition system was evaluated with a one-image-at-a-time approach. This decision was made to preserve the most eigenfaces obtained from the dataset during training, allowing for a more comprehensive representation of facial features as in previous testing a significant drop in accuracy was shown sometimes due to the reduced size of the dataset. So for instance removing ten or fifteen percent of the images for testing would in some cases reduce so much the training set that the accuracy of the model would severly worsen.

The first approach *Full Image Comparison* employed the Mahalanobis distance metric to recognize faces by comparing them to all the training images. After obtaining the eigenfaces and projecting the training images into a reduced dimensional space, the Mahalanobis distance was computed between the test image and the training images. The closest match was determined, and the guessed label for the test image was assigned based on the label of the matching training image. This approach exhibited promising results, achieving higher accuracy rates for various combinations of threshold and test size percentage.

The second approach *Feature Vector Comparison*, was much faster but showed much lower accuracy. It involved building feature vectors for each subject using their training images. The feature vectors represented the averaged eigenfaces for each subject, capturing essential facial characteristics in a compact form. During face recognition, the test image's eigenface projection was computed, and the Mahalanobis distance metric was used to compare it with the feature vectors. The identity with the closest feature vector was considered the guessed label for the test image.

One third approach would use an image, or a set of images from the training set to testing. Unsurprisingly the accuracy would fall on 100% every time as the model had previously seen those exact pictures. While achieving perfect accuracy could be the perfect scenario this approach does not provide a realistic evaluation of the model's performance as the true test of a facial recognition system lies in its ability to generalize to new and unseen faces accurately.

	Energy threshold			
Test Size	0.1	0.15	0.2	0.25
0.1000	2.07%	1.38%	1.66%	1.99%
0.5000	2.07%	2.21%	2.48%	1.16%
0.7500	0.41%	0.83%	1.04%	0.99%
0.8500	48.55%	44.75%	41.82%	44.21%
0.9000	72.61%	74.86%	71.64%	74.67%
0.9500	82.99%	83.98%	81.78%	81.62%
0.9900	87.14%	80.94%	83.64%	81.95%
0.9990	82.57%	81.22%	81.37%	80.63%
0.9999	58.92%	58.01%	53.62%	59.11%
	0.3	0.5	0.75	1.0
0.1000	1.93%	1.99%	1.82%	1.66%
0.5000	2.62%	2.90%	2.87%	2.44%
0.7500	0.41%	0.66%	0.55%	0.50%
0.8500	42.96%	41.92%	42.13%	40.51%
0.9000	70.58%	69.43%	73.33%	72.16%
0.9500	79.56%	80.45%	80.73%	80.65%
0.9900	82.87%	85.09%	83.16%	83.89%
0.9990	81.63%	79.45%	81.56%	80.65%
0.9999	52.49%	54.68%	54.61%	54.06%

(a) Facial Recognition Accuracy for Full Image Comparison

	Energy threshold			
Test Size	0.1	0.15	0.2	0.25
0.1000	3.32%	3.59%	5.59%	3.15%
0.5000	0.83%	2.76%	3.93%	3.31%
0.7500	4.56%	7.18%	2.28%	3.15%
0.8500	5.81%	6.63%	6.42%	7.45%
0.9000	9.96%	9.67%	11.80%	12.58%
0.9500	18.26%	18.51%	16.15%	14.40%
0.9900	18.26%	16.85%	18.84%	19.21%
0.9990	16.60%	18.78%	20.08%	18.87%
	0.3	0.5	0.75	1.0
0.1000	2.90%	2.98%	3.59%	2.07%
0.5000	2.49%	3.31%	3.53%	3.19%
0.7500	3.15%	3.65%	3.87%	3.77%
0.8500	6.35%	6.55%	6.35%	7.17%
0.9000	11.74%	9.44%	11.87%	10.19%
0.9500	15.06%	16.57%	13.91%	14.83%
0.9900	18.23%	18.31%	18.17%	16.86%
0.9990	20.72%	18.64%	19.55%	17.40%

(b) Facial Recognition Accuracy for Feature Vector Comparison

Table 1. Facial Recognition Accuracy for Different Thresholds and Test Sizes

C Effects on different train sizes

Because of the approach type, by removing one image at the time from the dataset before calculating the eigen faces and performing the training, the train size was not affected. As it is possible to observe in table 1, the train size does not really matter because in this approach it is not really the test size. It is the number of iterations with different images for testing that the program did but the train size was for each iteration equal to the size of the whole dataset less one. Due to the reduced size of the dataset was observed that without this approach on the train size, the accuracy would fall almost immediately with even reduced test sizes. The results for smaller test sizes were very sensitive and inaccurate given that as there were just a few images, depending on the random images that would be selected, the accuracy could be either very high or very low, but not constant or reliable at all.

D Effects on the number of eigenfaces

Based on the results of table 1a, as this model was the one that performed better, the results show that generally the accuracy tends to improve as the energy threshold (scaled from [0-1]) used during Singular Value Decomposition (SVD) increases.

At lower thresholds of 0.1 and 0.5, the accuracy remains around 2-3%, a very low performance. This is likely because of the limited number of eigenfaces retained, resulting in a very simple and poor representation of the facial features.

As the energy threshold is increased to 0.75 and 0.85, the accuracy shows improvements, reaching around 40-50%. This suggests that with a higher number of eigenfaces retained the model is able to capture more variations of facial characteristics from the training data, leading to enhanced recognition performance.

The most significant increase in accuracy is observed at thresholds of 0.9 and 0.95, where the accuracy substantially improves and reaches to approximately 70-83%. At these thresholds, a significant number of eigenfaces is retained, effectively capturing the most essential facial features and achieving a good balance between model complexity and generalization. In table 2, measured for 20% of the images of the dataset is also possible to observe how the accuracy rapidly decreases with energy thresholds right under 0.9. There is a high and significant drop at this values sustaining the conclusions previously reached.

Threshold	Face Recognition Accuracy
0.7500	0.62%
0.7600	2.90%
0.7700	1.66%
0.7800	2.90%
0.7900	10.97%
0.8000	15.32%
0.8100	16.15%
0.8200	16.36%
0.8300	24.22%
0.8400	31.88%
0.8500	41.20%
0.8600	50.93%
0.8700	54.87%
0.8800	62.32%
0.8900	65.01%
0.9000	73.71%

Table 2. Facial Recognition Accuracy for Different Thresholds

However, an intriguing trend occurs after the threshold of 0.99, where accuracy starts to decrease, reaching around 52-59%. This reduction in accuracy may be attributed to overfitting. Overfitting occurs when the model becomes too specialized in recognizing the specific faces from the training set, making it less capable of generalizing to unseen faces during testing.

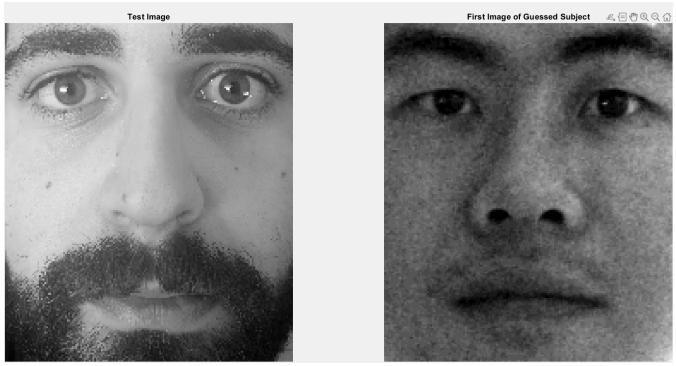
Choosing between a threshold of 0.95 and 0.99 for Singular Value Decomposition (SVD) in facial recognition must take into account consideration to the specific requirements of the application. A threshold of 0.95 offers a favorable balance by

retaining an adequate number of eigenfaces to represent crucial facial features while mitigating the risk of overfitting. This choice generally results in a relatively faster computation time compared to a threshold of 0.99, which retains more eigenfaces with the cost of increased complexity and potentially longer processing times. The decision should consider the desired accuracy, the complexity of the dataset, and the available computational resources to ensure an effective and practical facial recognition system.

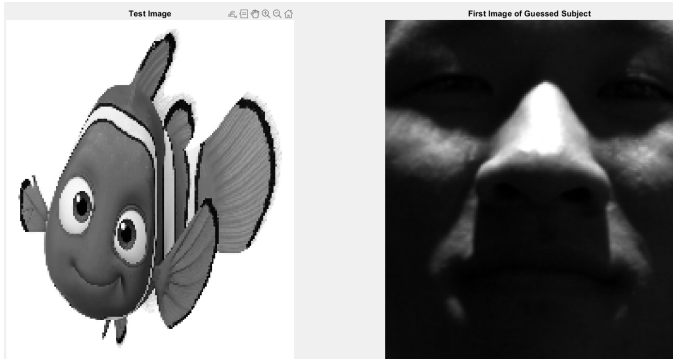
E Relevant variables

When testing images of people that are not present in the dataset or images that may not even be of people, the model will assign anyway the picture to the subject on the closest match, even if it is very dissimilar to any of the known subjects. So, to explore this behaviour some experiments were conducted in this subject. Firstly it was tested a picture from one of the subjects, then a picture of me in a similar pose as the ones from the dataset and finally one of *Nemo*. The results of these experiments are presented in figure 6. There are many possible answers on how to prevent this phenomenon from happening. For example if we take a close look to the distances to the mean face on average an image from a subject with other pictures on the dataset such as the one in figure 6a will not surpass 20000 units. For my picture in 6b this value was slightly bigger as it was not in the dataset but still is a face similar to the others and in the same position. This value was of 26120. The picture of *Nemo* had the value of 38000. Based on these observations, it is possible to set arbitrary thresholds for image rejection. For example, using a threshold of 2 times the mean distance of the images in the dataset from the mean face could be used to reject images that are not faces. Similarly, a threshold of 1.5 times the mean distance could be used to reject images of faces that do not belong to any of the known subjects.

(a) Subject test



(b) Test with myself



(c) Nemo test

Figure 6. Conducted tests

F Conclusion

The approach of Full Image Comparison likely yielded better results than Feature Vector Comparison because it is able to capture more comprehensive information about the facial features and variations. In Full Image Comparison, the entire face image is considered as a whole, what will allow the model to take into account specific facial details, expressions, and lighting conditions. On the other hand, Feature Vector Comparison relies on extracting specific facial features and encoding them into vectors, which may result in the loss of some detailed information as there are few images of each subject for this to have better results.

The limitations of the dataset might have affected the results significantly. Having images of faces in only one particular position varying the lighting conditions could lead to a lack of variation in the training data as using a particular image for testing removes all images of a subject in those lighting conditions, making the model less capable of generalizing and struggling when exposed to different lighting conditions during testing. Similarly, having only one expression per face could limit the model's ability to recognize faces with different expressions, making it less robust in real-world scenarios where people have diverse facial expressions.

This truly represents the importance of having a diverse, large and representative training data to build a robust facial recognition model that can perform well under various conditions and handle unknown faces effectively.

In conclusion, the Full Image Comparison approach might have outperformed Feature Vector Comparison due to its ability to capture more comprehensive facial information. However, the dataset's limitations, such as lack of pose variation, limited expressions, and insufficient lighting conditions, could have impacted the model's performance negatively. To achieve better results, collecting a more diverse and extensive dataset that covers a wide range of facial variations and expressions would be crucial. Additionally, training the model on more samples of each individual under different conditions would enhance its ability to recognize faces accurately and improve its performance in real-world applications.

References

[1] Turk, MatthewA. AlexP. Pentland: *Face Recognition Using Eigenfaces*, Volume 3 *Number 1*. Vision and Modeling Group, The Media Laboratory Massachusetts Institute of Technology, 1991.

[2] Brunton, Steve: *SVD: Eigenfaces Matlab Python*. Youtube, 2020.