Face Recognition - Eigenfaces

Juan Camilo Ceballos Arias, Miguel Angel Mejia Muñoz, Santiago Aristizabal Toro, Juan Esteban Torres Marulanda, Danny Styvens Cardona Pineda

ABSTRACT

The aim of the project is to recognize a person's face by making a comparison between its characteristics and those of other people's faces which are already known. To perform this task, Eigenfaces method is implemented together with a previous preparation of the images.

Initially, outlier images are identified within a face and landscape dataset, using metrics such as Manhattan, Euclidean, Chebyshev and Minkowsky distances ($p = \frac{5}{2}$ and $p = \frac{\sqrt{2}}{2}$). Then the results are compared to determine which one of the metrics has better performance in the identification of rare images. In total 2.470 images (2.260 faces and 210 natural landscapes) were used. Secondly, the methodology of Eigenfaces was implemented with a total of 3059 images of faces, then by distance measurements, distances of new images are identified to the subspace generated by the Eigenfaces.

The code, datasets, Readme and a document type article are in an open Github repository at `https://github.com/mmejiam-eafit/ms_data_science_project_i`. It also includes a notebook with all the stuff necessary to replicate the work, found in the repository mentioned above.

Keywords: Distance, Metric, Outlier, Eigenfaces

1. INTRODUCTION

For face recognition, we need to identify that a given image corresponds to a face by means of comparison of its characteristics with a given knowledge base of faces, in order to compare it with. Different approaches have been proposed (e.g. Geometrical Characteristics of Faces, Eigen-Faces, Holistic methods, etc.) In the following exercise we will focus on the group of holistic methods that have shown promising results, overcoming some of the difficulties related in working with geometrical features (e.g., face symmetry) (*Zhao, Chellappa, Phillips, and Rosenfeld, 2003* [6]).

For the problem of face recognition, it is necessary to identify if a certain face belongs to a set of faces that are already known (recognition), by means of characteristics that can be extracted from the set of images. Different approaches have been proposed such as geometrical characteristics of the faces, and holistic methods. This exercise will focus on the group of holistic methods since they have shown good results, overcoming some of the difficulties related in working with geometrical features such as face symmetry (*Zhao, Chellappa, Phillips, and Rosenfeld, 2003* [6]).

Thanks to the technological advances especially in computer science, development of mathematical models and new algorithms, there have been important advances in automatic face recognition starting in the 70s and onwards (Zhao et al., 2003). There are two important topics in automatic face recognition:

1. Detection of a face, for which segmentation methods are used.
2. Features extraction algorithms using linear combinations of characteristics on a set of data, of which this project will be focusing on.

An approach as Eigenfaces and Fisherfaces, have observed good results. Eigenfaces approach is based on the Principal Component Analysis (PCA) (*Turk and Pentland, 1991* [5]) and Fisherfaces is based on the Discriminant analysis (*Etemad and Chellappa, 1997* [2]); both methods have disadvantages and advantages depending on the use classification or image representation (*Belhumeur, Hespanha, and Kriegman, 1997* [1]).

Although the holistic approaches have solved some difficulties, it is still important to address other approaches such as the variation of illumination in the images and facial expressions. These two issues are still a challenge for these methods (Zhao et al., 2003), but methodologies like deep neural network could be a good alternative to deal with them.

Eigenfaces method is developed to make comparisons for face recognition (*Turk and Pentland, 1991* [5]). Basically, what is made is to extract the most important information of a face in order to compare it with a database's. Mathematically, it consists in calculating the images set's covariance matrix eigenvectors so that a space being made of them is created. While working on image recognition the eigenvectors are called Eigenfaces and on its part, the eigenvector space, face space. Every face in the images set can be constructed as a linear combination of the vectors belonging to this, which is also the basis of the whole methodology since the recognition process refers to measuring the distance between an image and this space and depending on how long it is, to classify the face as a known or unknown one.

It has to be taken into account that the total number of possible Eigenfaces is equivalent to the number of images of the whole dataset. Likewise, the faces can be approximated using only the best M Eigenfaces, which are those with the highest eigenvalues associated with the greatest variance in the images. The number M is usually determined holistically, although other proposed methods exist in the literature (*Peres, Jackson and somers, 2005* [4]).

The process of facial recognition can be summarized in these 4 steps:

1. Calculate the eigenfaces of some images, thus defining the face space.
2. When a new image is introduced, weights of incoming image and the M eigenfaces are calculated by projecting the incoming image in each Eigenface.
3. Determine if the image is a face. This is defining whether the image is relatively close to the face space or not.
4. If the new image is a face, its weight pattern is classified to a previously known or unknown person.

This document presents the initial phase of the project, which identifies outlier images within a face and landscape dataset, by using metrics such as Manhattan, Euclidean, Mahalanobis, Chebyshev and Minkowsky distance ($p = \frac{5}{2}$) (*Macho, 2010* [3]). Additionally Minkowsky similarity ($p = \frac{\sqrt{2}}{2}$) is also included. Then, a comparison is made among the results to determine which one of the metrics has a better performance in the identification of rare images. Then the Eigenfaces method is developed to make comparisons for face recognition.

2. DATASETS

For the development of this project, a group of images hosted in a free online database will be used. This database has four different directories holding the images in different levels of difficulty as follows: faces94, faces95, faces96 and grimace. The last two are more complex due to the images variation on background and scale and the type of facial expressions in them.

The whole set has 7900 images belonging to 395 individuals. Different genders and races are shown, people wearing glasses and beards are also taken into account and, regarding the age range, most of data corresponds to first-year undergraduate students between 18 and 20 years old, even though some older people are present in the data as well. For this first phase, Dataset Faces94 will be used.

Faces94 is a collection of images consisting of a wide range of people's pictures taken speaking in front of camera. Because of the speech, this set is an introduction to the variation in facial expression. Faces94 has 153 individuals images using portrait format. It contains pictures of male, female and male staff in separate directories. The pictures background is plain green. It does not have any individual's variation on head scale and image lighting, but it does have a few on head turn, tilt and slant, and considerable on facial expression. Additionally, there is no individual hairstyle variation as the images were taken in a single session.

The methodology of Eigenfaces was implemented with a total of 3059 images of Faces94 (faces94 male,faces94 female,faces94 malestaff).

Figure 1. Faces94 face images

Additionally, natural landscape images were included. These images were obtained from ImageNet database and each one of their links are online http://image-net.org/api/text/imagenet.synset.geturls?wnid=n13104059. Cv2 package was used to read and resize the images and then a Numpy array was created with a gray scale of the images.

Figure 2. Natural landscape images

3. ARCHITECTURE, MODELS AND DATA PREPARATION

Only images in JPG format were used in the project. They were originally colored (3-D) and converted into grayscale (2-D) due to practical reasons of computation, but it is important to mention that the model could be generalized to the case of 3-D.

Python programming language is used for the development of the project which considers some stages as follows:

a) Data preparation, in which the colored images are converted into grayscale and resized in such a way that they can be compared. This process allows to have a data matrix where each element corresponds to a pixel of the image; cv2 package in python was used for that purpose.
b) Functions are defined to read and manipulate images inside the dataset which contains subfolders (nested structure).
c) Functions are written to do distances calculation, outliers detection, accuracy of the metrics determination and finally graphing the results.
d) A Jupyter Notebook was created in which functions are run and results are depicted. These other packages were used: Pandas, Numpy, Matplotlib and Collections.

The datasets, the functions and the Notebook are located in a Github repository, so that each member of the team makes their contributions after developing and testing locally.

4. RESULTS

4.1 Metrics comparison and outliers detection

In this project, comparison measures of efficiency between metrics were used. Table 1 depicts the global accuracy efficiency between metrics. In all cases, the measure was greater than 87% for both criteria detection ($Q3 + 1.5*$

IQR and $90th$ percentile).

Table 1. Accuracy of metrics based on the mean and median image.

	Outlier selection criteria*			
	Percentile 90		**Q3 +1.5*IQR**	
Metrics	**Distance to the mean**	**Distance to the median**	**Distance to the mean**	**Distance to the median**
Manhattan ($p = 1$)	88.06	89.51	91.58	92.20
Euclidean ($p = 2$)	89.68	90.49	92.63	93.40
Minkowsky ($p = 3$)	89.27	90.00	93.04	93.00
Chebyshev ($p = inf$)	89.19	88.54	90.57	91.00
Minkowsky ($p = \frac{5}{2}$)	89.51	90.16	93.52	93.20
Minkowsky ($p = \frac{\sqrt{2}}{2}$)	87.49	89.35	91.13	92.10

* accuracy (%) = (true positive + true negative)/N

Table 2 depicts false negatives and true negatives. For true negatives, the percentage with respect to the total of outliers (210) was calculated. Overall outlier detection was less than 53%.

Table 2. Number of true and false negatives obtained, based on the mean and median of images.

	True negatives (% true outlier accuracy)		**False negatives**	
	Percentile 90		**Percentile 90**	
Metrics	**Distance to the mean**	**Distance to the median**	**Distance to the mean**	**Distance to the median**
Manhattan ($p = 1$)	81 (38.57)	99 (47.14)	166	148
Euclidean ($p = 2$)	101 (48.09)	111 (52.86)	146	136
Minkowsky ($p = 3$)	96 (45.71)	105 (50.00)	151	142
Chebyshev ($p = inf$)	95 (45.24)	90 (42.86)	152	163
Minkowsky ($p = \frac{5}{2}$)	99 (47.14)	107(50.95)	148	140
Minkowsky ($p = \frac{\sqrt{2}}{2}$)	74 (35.24)	97 (46.19)	173	150

No large differences were observed when comparing the distances to the mean and the median of the images (tables 1 and 2).

For the calculation of the real median it was necessary to compute and sum the distances of all images to each other, in order to select the minimum result.

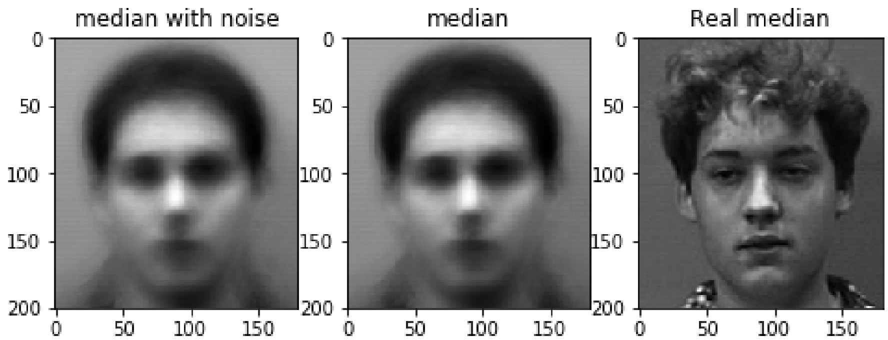

Figure 3. Mean face

Table 3. Confusion matrix of metrics/similarity, according to the mean image

Metric/similarity	True positive	False negative	False positive	True negative
Manhattan $(p = 1)$	2094	166	129	81
Euclidean $(p = 2)$	2114	146	109	101
Minkowsky $(p = 3)$	2109	151	114	96
Chebyshev $(p = inf)$	2108	152	115	95
Minkowsky $(p = \frac{5}{2})$	2112	148	111	99
Minkowsky $(p = \frac{\sqrt{2}}{2})$	2087	173	136	74

Table 4. Confusion matrix of metrics/similarity, according to the median image

Metric/similarity	True positive	False negative	False positive	True negative
Manhattan $(p = 1)$	2112	148	111	99
Euclidean $(p = 2)$	2124	136	99	111
Minkowsky $(p = 3)$	2118	142	105	105
Chebyshev $(p = inf)$	2097	163	120	90
Minkowsky $(p = \frac{5}{2})$	2120	140	103	107
Minkowsky $(p = \frac{\sqrt{2}}{2})$	2110	150	113	97

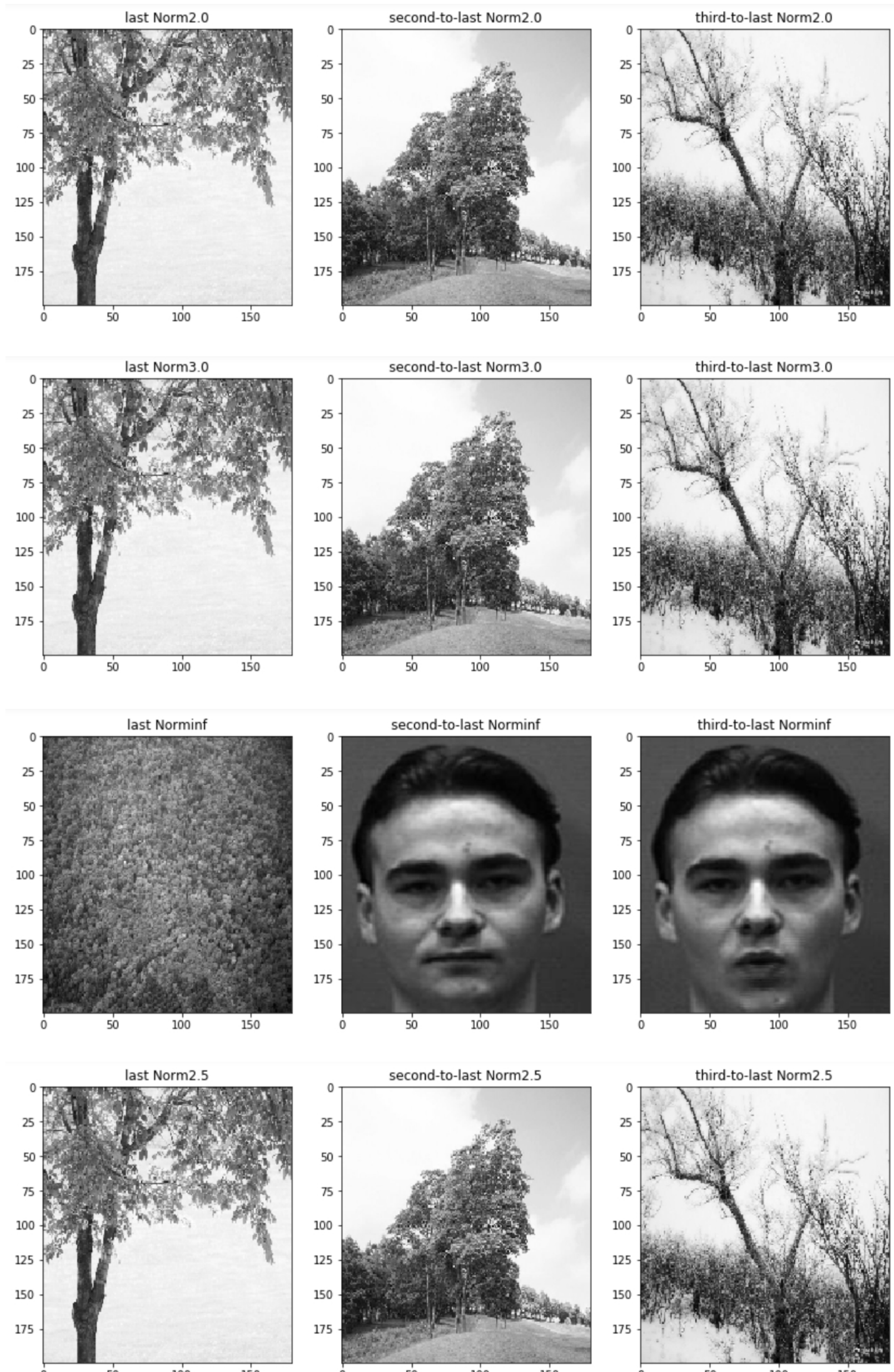

Figure 4. Outliers - against mean

4.2 Eigenfaces

The first step was to centralize the data in order to calculate the covariance matrix. This was made by means of calculating the mean image for later subtracting it to the whole dataset and performing a dot product of this last matrix and its transposed to get the covariance one. The product was done to have the covariance matrix dimensions not in terms of the dataset's images height and width but the number of images wise, as a strategy to reduce the size of the result.

4.2.1 Subspaces

After having the covariance matrix, the next step was to compute its singular value decomposition to get the eigenvectors and eigenvalues. Then, using the results, the face space started to be created by selecting the number of components of the subspace. That selection was run using two options:

- Investigator's criteria of variability captured, where the number of principal components were obtained from a prior set representation percentage. This option raises from how disperse the values are and the percentage of variability captured, so cumulative sums of variance were performed until getting the desired representation, in this study, 98%

- Investigator's criteria of threshold contribution value, where the approach to choose the number of components was to analyze the whole variation as a percentage.

Finally, option 2 was used to set the total of components to create the face space. They were the first 305 components, with 97.59% of variability captured and from which the contribution was less than 0.01%.

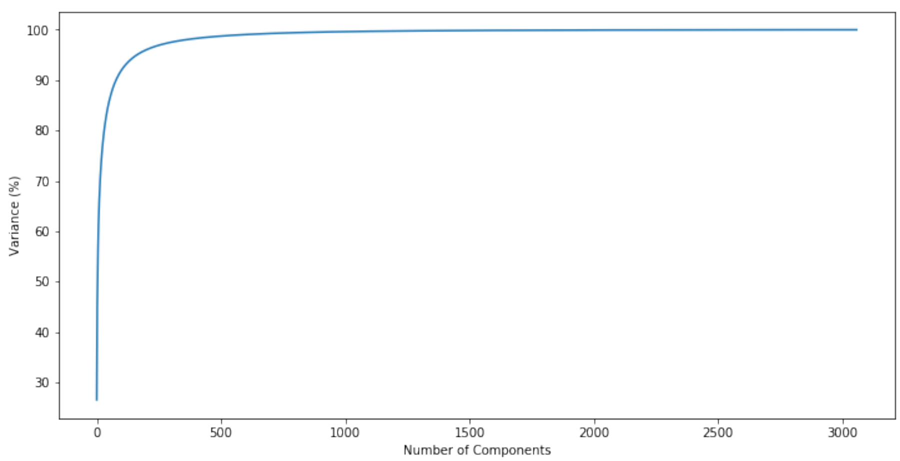

Figure 5. Cumulative summation of the explained variance

4.2.2 Face space

After setting the number of principal components that were going to be used to create the face space, the calculations were done: the dot product of the data and the eigenvectors selected was performed and the result was normalized.

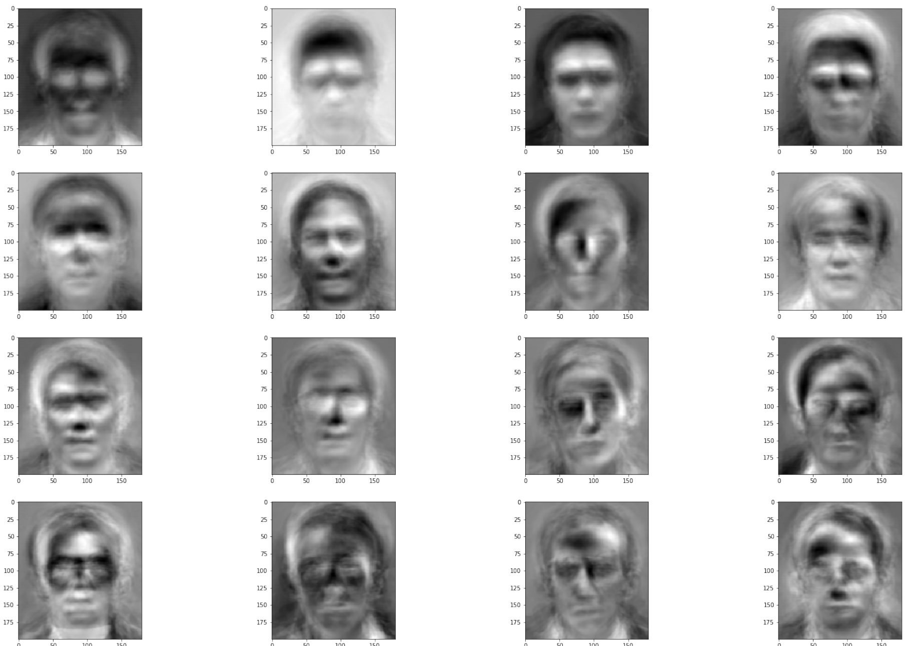

Figure 6. First sixteen components of face space

4.2.3 Projection of an image on the face space

As every image of the dataset can be constructed as a linear combination of the face space's vectors, some image projections were made to observe how they were reconstructed by the model. To begin with, the weight (w) of each Eigenface in the generated subspace due to the original image, was computed. Then, that image was reconstructed by a dot product of w and the face space, and later adding the mean image.

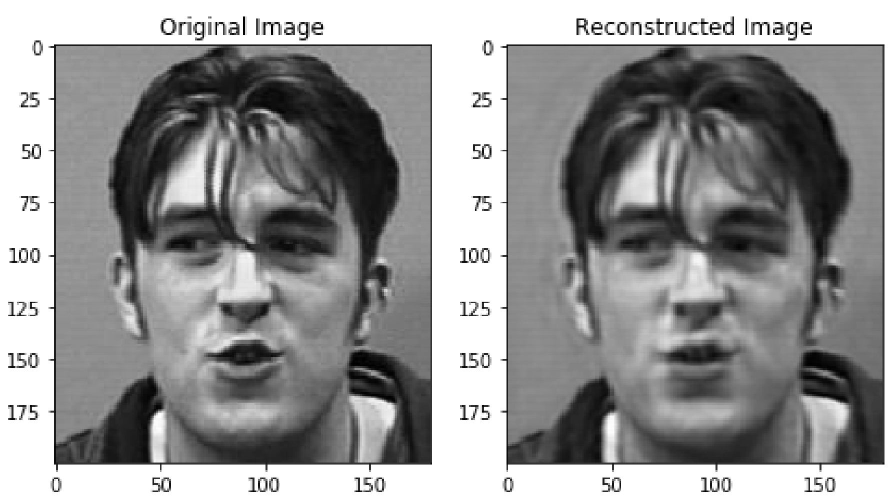

Figure 7. Projection of an image on face space

4.2.4 Distances and outliers

The distances between the original images and those reconstructed from the subspace were calculated, from these distances atypical data were identified (greater distances between the original image and those reconstructed), the ones with the smallest distances (the best reconstructed ones) were also identified.

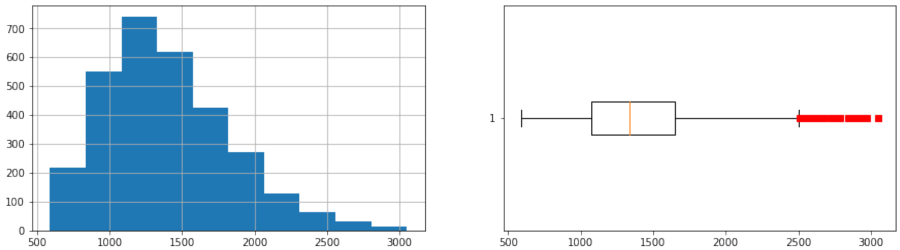

Figure 8. Distances (norm 2) distribution

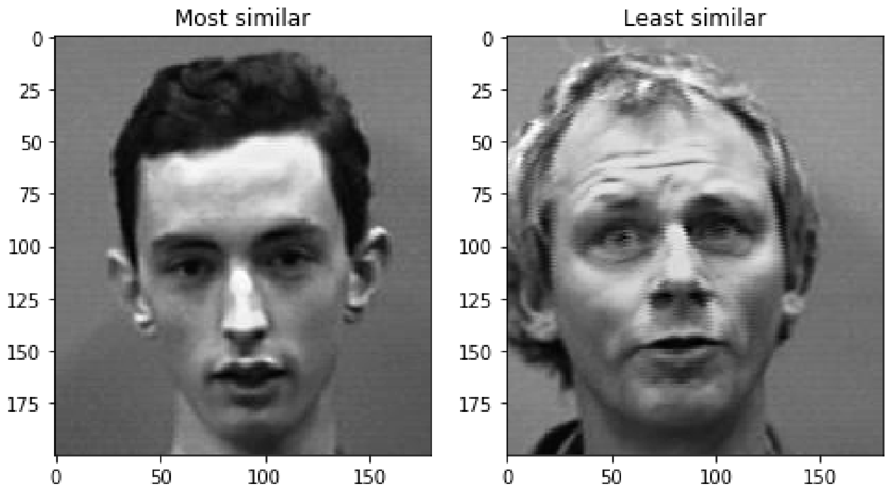

Figure 9. Most and least similar reconstructed faces to original image

Figure 10. Least nine similar reconstructed faces to original image

4.2.5 Face recognition

To test whether with the subspace built it is possible to detect if an image is a face or not, it is tested by projecting images of landscapes and its distance is calculated. Large distances are observed (outside the range of distances), it is concluded that the image is not a face.

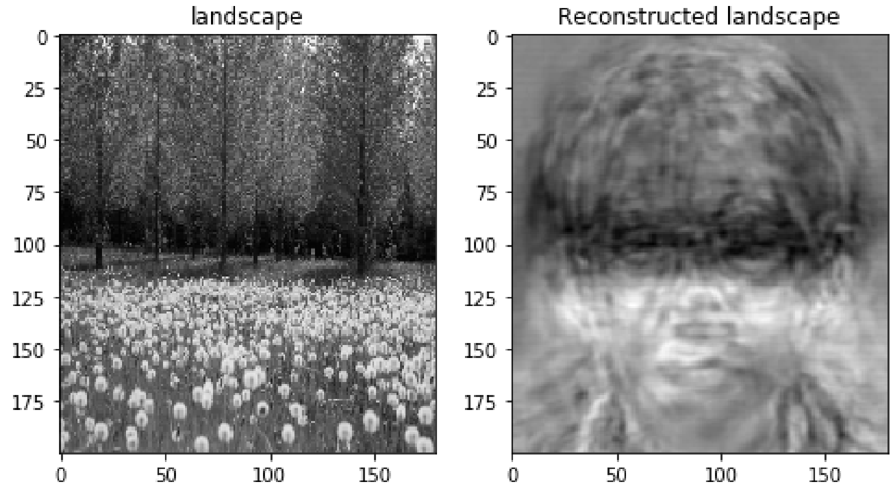

Figure 11. Image does not belong to the set of faces (distance>max(faces to face-space)

5. CONCLUSIONS

The outlier detection presented low accuracy rate, none of the metrics detect more than 53% of natural landscapes. Due to face recognition was not done in this first phase of the project, it is possible that some image pixels around faces could have been interpreted as landscapes. Also, image lighting could be a noise factor that disturbs the outlier identification.

Distances to mean and median were similar, although about 10% of images were outliers, this suggests that it is necessary to do some type of preprocessing in images to improve estimations.

To improve estimations, in next phase, other distances will be calculated, such as Mahalanobis, for which it is necessary to reduce dimensionality of data through principal component analysis (PCA) to estimate the inverse of the covariance matrix or precision matrix.

In general, all metrics behaved in a similar way on detecting true positives and negatives. When detecting false positives and negatives, outlier detection was less consistent compared to true positive and negatives detection. The worst norm was the Chebyshev one (p = inf) with nule true outlier detection.

The Eigenfaces method allowed to reduce the original dimensionality of the problem without much loss of information. With only 305 (out of 36.000) components, 97.59% of variability was captured.

Eigenfaces allowed to identify in a successfully way whether an image is a face or not, by projecting that image to the face space.

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. *Eigenfaces vs. Fisherfaces: recognition using class specific linear projection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), pp.711-720, 1997. https://doi.org/10.1109/34.598228.

[2] K. Etemad and R. Chellappa. *Discriminant analysis for recognition of human face images.* J. Opt. Soc. Am. A, 14(8), pp.1724–1733, 1997. https://doi.org/10.1364/JOSAA.14.001724.

[3] M. Macho. *TOPOLOGÍA DE ESPACIOS METRICOS*. 2010. Retrieved from http://www.ehu.eus/ mtwmastm/TEM0910.pdf.

[4] Somers K Peres P., Jackson D. *How many principal components? stopping rules for determining the number of non-trivial axes revisited.* Computational Statistics Data Analysis, 49(4), 974-997, 2005. https://doi.org/10.1016/j.csda.2004.06.015.

[5] M. A. Turk and A. P. Pentland. *Face recognition using eigenfaces.* 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Proceedings, pp.586-591, 1991. https://doi.org/10.1109/CVPR.1991.139758.

[6] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. *Face Recognition: A Literature Survey.* ACM Comput. Surv., 35(4), pp.399–458., 2003. https://doi.org/10.1145/954339.954342.