Face Recognition*

*Olivetti Dataset

1st Tiago Pereira
MRSI
DETI
MRSI

2nd Bruno Duarte
MRSI
DETI
118326

***Abstract*—This article presents an overview on the several existing machine learning models, used in the realm of face recognition. We are going to focus on the evaluation model under different lightning application,, pose and expressions for each individual provided by the dataset. We are also going to evaluate the performance metrics of this models and find the best parameters to apply to the models.**

***Index Terms*—Face Recognition, Data, Feature Extraction, Supervised ML**

I. Introduction

This article was made in the scope of AA (*Aprendizagem Automática*) course. In this project we are going to apply different models, some of them learned in class, and other that required some extra research, in a way that we can find the best model to fit to the face recognition problem.

The facial image data comes from the Olivetti Dataset, that dataset has the been collected between April 1992 and April 1994 and contains differente images from different subjects. We are going to compare the functinalities of the machine learning algorithms, like, Support Vector Machines, Logistic Regression and Artificial Neural Networks.

Our main goal is to evaluate each of the models, and see it's response to certain variation in light, position and facial expressions, by the analysis of the outcome.

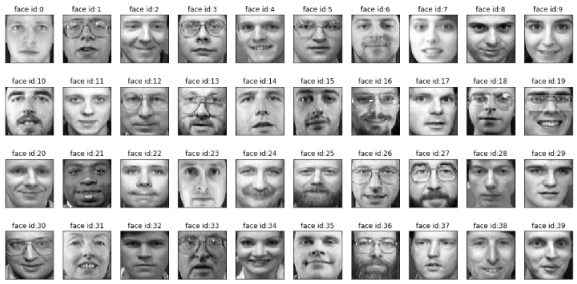

Fig. 1. The 40 participants

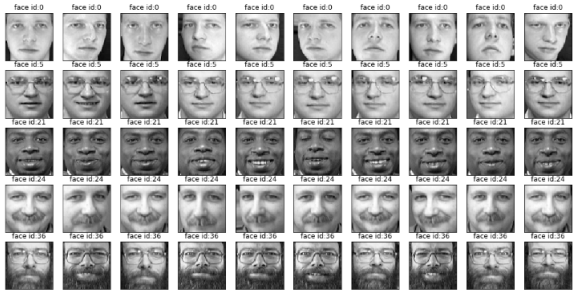

Fig. 2. All the images of a some subjects

II. Dataset Explanation

The Olivetti Dataset is a dataset that was collected between the years of 1992 and 1994, where wore taken images of 40 different people with different expositions to light, poses, facial detail, facial expression and wore taken in diferent times. The fact that the images of each subject wore taken in different time frames also can give an insight of how the faces change and evolve over time.

Given that the face photos were taken at various periods, the temporal aspect of the dataset adds a realistic touch. This temporal variance allows researchers to assess the models in dynamic environments that replicate real-world situations where people's appearances may change over time. Every face image has a black background, which reduces unnecessary visual components and draws attention to the essential aspects of each face. With each image having the same size of 64 by 64 pixels, the grayscale format ensures clarity and focus. Additionally, the pixel values have been scaled to the [0, 1] interval to enable uniform data representation and improve the dataset's suitability for use with various machine learning models.

To optimize the dataset for supervised learning tasks, the 40 individuals' names have been converted into integers between 0 and 39. This encoding simplifies the labeling process, enabling more effective training and evaluation of face recognition models. In conclusion, the Olivetti Dataset is not just a repository of facial images but a dynamic collection that captures the essence of facial diversity over time. Its careful preprocessing and thoughtful approach to image properties provide a solid foundation for researchers and practitioners aiming to advance the field of face recognition through empirical analysis and model development.

III. State of the Art

The state of the art of machine learning in general is in continuous evolve, and in the field of Machine Learning is

no different. In the recent years, Face Recognition algorithms have become a lot mode sophisticated, by the appearance of new datasets, more interest from programmers and the continuous increase in of computational power.

We started by evaluating the existing methods that the community of Kaggle applyed to this dataset, arriving to the conclusion that the fastest and more responsive method to apply was the Support Vector Method, demonstrating more proficiency in managing intricate and diverse facial features.

SVM is specially useful in face recognition as it allows models to identify and categorize people according to their facial features. In face recognition, where we have facial images with the same label very close to each other, this greatly helps to refine the algorithm.

PCA is also very used to extract features and help reduce the dimensional of the face images while also preserving the most important information, creating a more compact feature space. Because of this, machine learning algorithms greatly benefit from using PCA, as it makes the training lighter. Using PCA in conjunction with Linear Discriminant Analysis it is possible to achieve 100% accuracy, as it was shown by Sudha Sharma, Mayank Bhatt, and Pratyush Sharma in "Face Recognition System Using Machine Learning Algorithm".

IV. Methods - Theoretical Introduction

In this section we are going to give a teorical introduction to the algorithms that we are going to use to create the models.

A. Confusion Matrix

In the end of the training, one of the visual validations that we can do, is the Confusion Matrix, that essentially, this matrix give us the information in the form of a graph with the x-axis containing the classes and in y-axis with the predicted classes. In this Confusion Matrix we can obtain the information about the true positives, true false, false positives and false negatives.

To achieve this, the model analysis the image, and assigns to it a value between 0 and 1, meaning if 1 that that image belong to a category. The model then looks to all the scores, and takes the one with a bigger value for prediction. Here we can see a example of a confusion matrix

B. Subsets

To verify the performance of our model we need to verify the behaviour of our model when tested with data from the dataset. To do that, in the beginning of each program we should start by dividing the dataset in two subsets, one for training the model and update the parameter and another for testing. The training set is used to train the model, while the testing set is used to verify the model's performance, normally the biggest subset should be the train subset.During our project we made this using the train_test_split that is a function from sklearn library, that divides the dataset in two subset in a random way, with a percentage of the dataset for test and the rest to training.

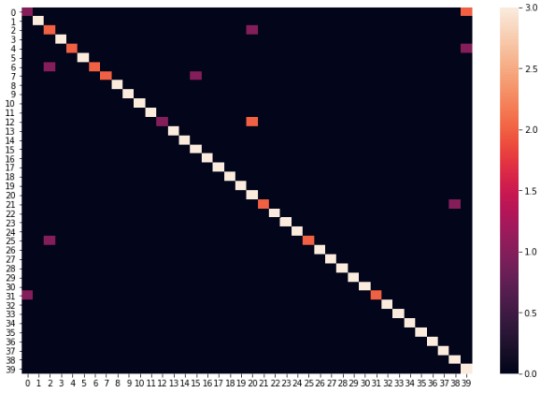

Fig. 3. Example of a Confusion Matrix (not relevant meaning)

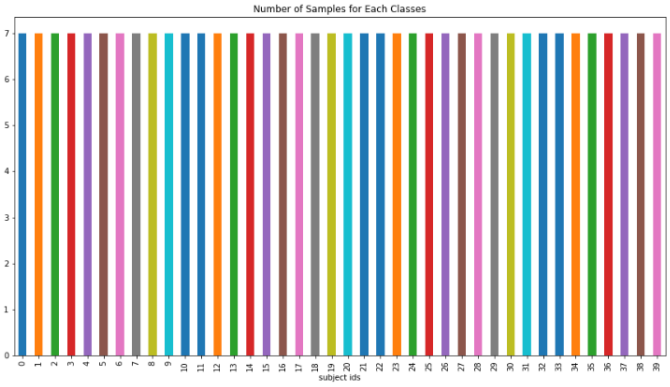

Fig. 4. Number of samples for each class

C. K-Fold Cross Validation

K-Fold Cross Validation guarantees the data balance after splitting data to avoid biased results. K-Fold Cross Validation is a method that divides the dataset in k folds, and test it in the rest of the folds. And does this test to every fold. Each of the test aren't independent, because the θ of previus iteration is then used as the initial theta of the following interaction. The results are then averaged. So this method reduces the risk of overfitting, increases reability of the algorithm and makes a efficient use of the available data.

D. Principle Components Analysis (PCA)

We use PCA to reduce the dimensionalitty of a dataset, improve the data visualization, while trying to capture the most important data. Using the original data, we are going to transform this data, into a set of orthogonal components (uncorrelated variables), while capturing the maximum variance. To do this, PCA use the covariance matrix of the data, by computing the eigenvalues and eigenvectors, that represent the magnitude and direction, respectfully. Then we are going to take all that values, and choose the greatest eigenvalues, to select the eigenvectors that we are going to apply to our new dataset.

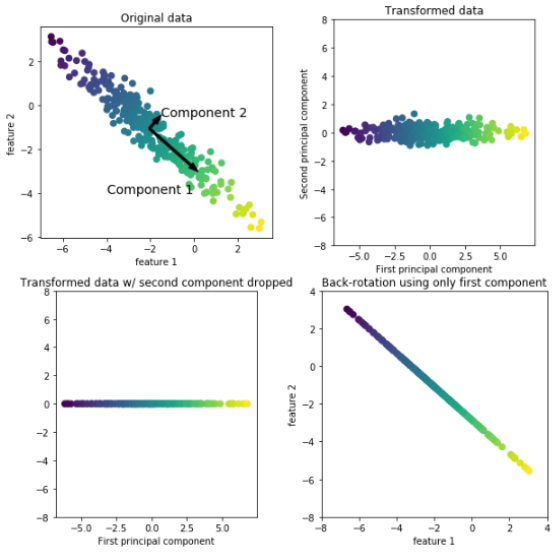

Fig. 5. PCA

In the figure, we can see, on the top left, our original dataset that has two dimensions (but PCA can be used for high-dimensional spaces). The first step is to find the direction (component 1) of the most significant variability axis. Later on the algorithm finds a orthogonal direction (component 2) to the vector of component 1.

1) Logistic Regression: Logistic Regression can be used in the field of computer vision, by the association of the features (extracted charateristics) of these images and their categories. By the analysis of the data, the model learns connections between the features and categories, refining the internal calculations, and weightts, to create a model capable of predict when he applied to the test subset.

One of the biggest advantages of Logistic Regression is that Logistic Regression offers probabilistic results, assigning to each of the categories, one probability score, allowing to more power in decision making and handling uncertainty.

While this is a strong algorithm, it is better suited for simple tasks, with a small number of features or categories.

The internal calculations of the Logistic Regression is defined by the following formula. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=10250671

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left[-y^{(i)} \log(h_\theta(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right]$$

(1)

E. Support Vector Machine

https://arxiv.org/pdf/1912.05864 Support Vector Machines (SVMs) have established themselves as a powerful technique in computer vision, particularly for image and video classification tasks. They excel at recognizing categories within datasets of moderate size.

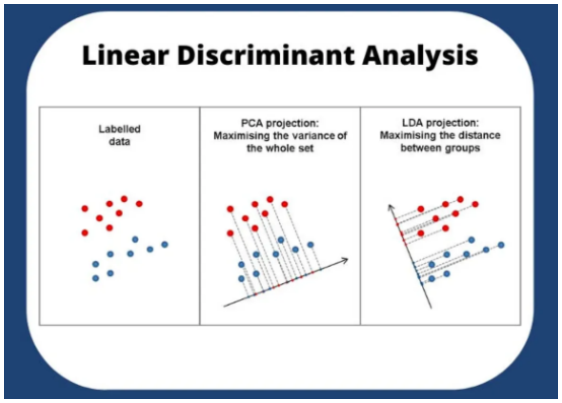

Fig. 6. LDA Example

SVMs operate by identifying dividing lines (hyperplanes) that effectively separate different classes in the data. These lines are strategically placed to maximize the margin between the closest points from each class. This approach allows for clear separation and better classification.

However, a key limitation of SVMs lies in their reliance on specific data points from the training set to define these hyperplanes. While effective, this dependence can restrict the SVM's ability to generalize to entirely new situations. In simpler terms, the SVM might struggle with data it hasn't encountered before during training. This limitation can potentially hinder accuracy on unseen data.

Despite this drawback, SVMs remain a valuable tool for image recognition, especially when dealing with manageable training data sizes. Researchers continue to explore methods for improving the generalization of SVMs, making them even more applicable across various computer vision tasks.

F. Linear Discriminant Analysis

Just like Principal Component Analysis (PCA), Linear Discriminant Analysis is also a technique used with the purpose of reducing the data dimensions. LDA aims to define, through the *W* vector, the orientation of the data, creating a new feature space, with a reduced dimensionality. To achieve this, we need to take in consideration that the classes should be well separated.

IN this figure, we can see that, initially we have two classes, (red and blue) that we are going to call R and B, now we need to find the vector that better defines the original feature space. In the middle example we see that when we transport the points to the vector, we are going to obtain a feature space were R and B are very close. But in the right example we can obtain R and B in a well space feature space.

The value of the vector is obtained by the maximization of Fisher's Criteria function, that depend on the orientation of the vector, the within-class scatter matrix, and the between-class scatter matrix.

V. Convolutional Neural Network

Convolutional Neural Network belogs to the class of Deep Neural Networks commonly used for image analysis. It can be used for image classification, object detection, image segmentation and more, related with computer vision.

It is composed of different types of layers and nodes, depending on the dataset, but there is a defined architecture.

- Input Layer
- Convolutionary Layer
- Activation Layer
- Pooling Layers
- Fully Connected Layer
- Output Layer

CNN are trained based on a process called backpropagation, where the network lear to minimize the error, trying to adjust the ”error term”, by changin the weights of the filters. It also uses optimization algorithms to optimize the learning.

THe reason why CNN is used to create models for computer vision is his ability to automate the feature extration and the efficiency presented when we need to capture low level and high level features.

VI. Implementation

We decided to create some different models to then can compare them. So, we are going to show the implementation and results for different algorithm.

A. Training and Results for PCA datasets

Initiatilly, to train the model, we are going to split the data in training data and test data. After testing, we found out that we achieve the best results when we divide the dataset in 20% for test dataset and 80% for the training dataset.

How was said before, we use PCA to reduce the dimensionality of the number of the features.

In the figure we can see in the first scatter plot, the original dataset. Were we can see that we have a very confusing dataset with the cloud point very concentrated, that would make us need of a much more computational power and to apply to this dataset a much more complex algorithm.

In the second scatter plot, we have a well separated dataset, where we can, on eye to define some classes. Making of this dataset more efficient.

We need to find to PCA the optimum number of Principles Components that we want to use. To do that, we can plot the Explained variance values for each component and check for ourselfes wich is the best number of Principle COmponents to use. In this case, we obtain this figure.

Here we found that the optimum number of components it’s around 90 components, but it can be any of the values greater than 90, but for efficiency purpose we are going to assume 90.

The reduction of features in the image, will obviously take some quality to the original images. We can see in the next fegure how is going to become one of the image for one of the subjects.

With this transformation we have all that we need to train some of the models of our project.

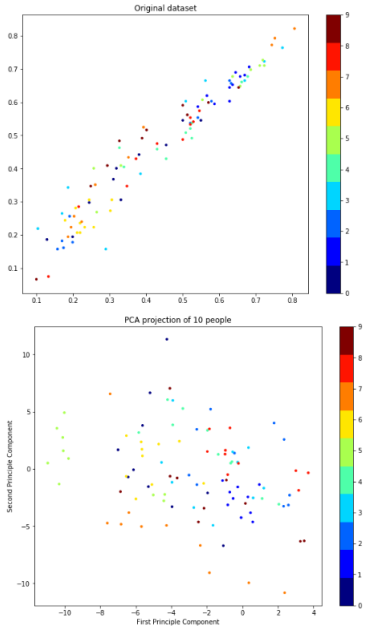

Fig. 7. Before and after PCA comparation

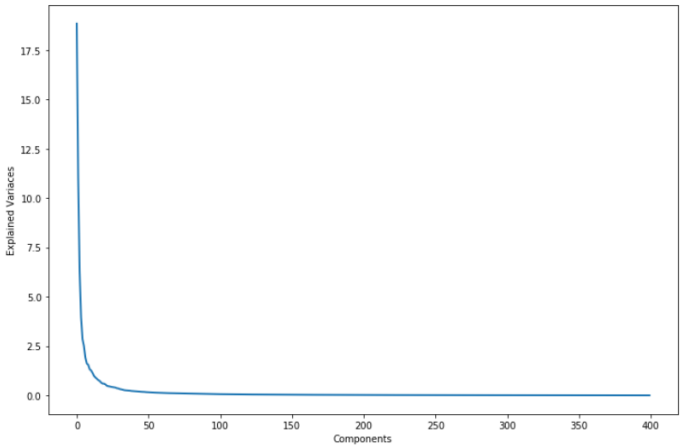

Fig. 8. Explained Variances through the number of component

Fig. 9. Average Face

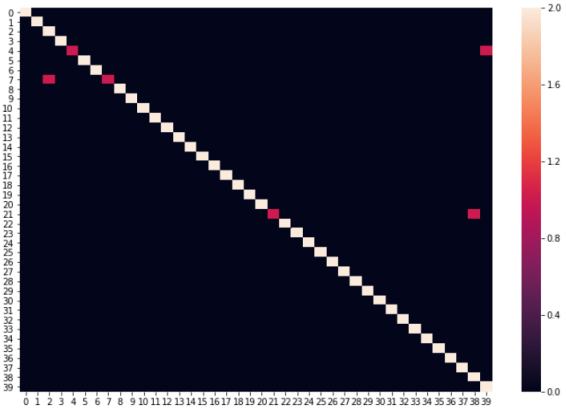

Fig. 10. Covariance Matrix for LDA

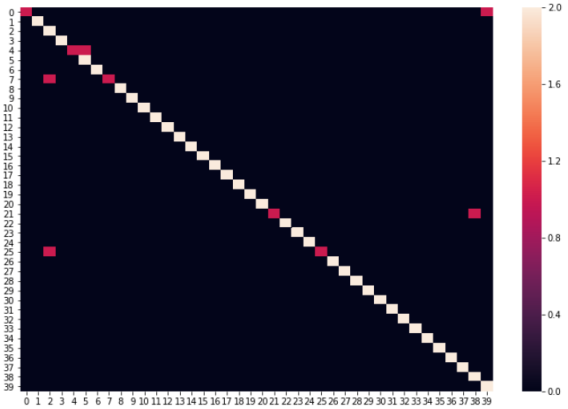

Fig. 11. Covariance Matrix for LR

We are going to train our dataset based on different classifiers the Linear Discriminant Analysis, Logistic Regression, Gaussian Naive Bayes, K-Nearest Neighbors and Support Vector Machine and check is accuracy score.

We achieve the following results

Classifier	Accuracy
LDA	0.96
LR	0.94
SVM	0.94

We can see that all the three model have a relatively god accuracy, but for a better validation, we need to apply the K-Fold Cross Validation to obtain a better perspective of our results. IN this case, we choose after some testing to use a K equal to 4 So we obtain the following Mean Cross Validations.

Classifier	K-Fold Validation
LDA	0.98
LR	0.95
SVM	0.87

Has we can see, LDA and LR model have the best results for this type of implementation, using the Olivetti Dataset.

One of the most common ways to check the accuracy, in a visual way, is the use of covariance matrix. We are going to check the covariance of the two best results.

In both of the cases we got some errors, but we can say that we achieved a good result, and this model, could be used in a real application. But the are still different ways to improve our model.

VII. Improvements

For improving this models, we could try another ways to make the features extraction and do some type of preprocessing to the images, to increase the number of data that we have available. We also could try to combine other's classifiers or the use of Neural Networks.

VIII. Conclusion

Th experience that we gain from this project has been rewarding. Along this project we have focused in studying of the Machine Learning models that are usually used in the context of Face Recognition. We learned more about some of the algorithms, just as SVM, LDA and LR and some tools to improve the efficiency of the training of our models, with the use of PCA. We think that we achieved our goal, of create and comprehend a model that is useful for Face Recognition.

IX. Contribution

In this project the work has been well divided and has been decided wich part each one of us should do, so the participation is of 50% each.

References

[1] Sudha Sharma, Mayank Bhatt, and Pratyush Sharma. "Face Recognition System Using Machine Learning Algorithm". In: 2020 5th International Conference on Communication and Electronics Systems (ICCES). 2020, pp. 1162–1168. DOI: 10.1109/ICCES48766.2020.9137850.

[2] Zoriana Rybchak Oleh Basystiuk Nataliia Melnykova. Machine Learning Methods and Tools for Facial Recognition Based on Multimodal Approach. 2023. URL: https://ceur-ws.org/Vol-3426/paper13.pdf.