Face Recognition with Olivetti dataset

Lucius Filho 96123
DETI
University of Aveiro
Aveiro, Portugal
luciusviniciusf@ua.pt

Tomé Carvalho 97939
DETI
University of Aveiro
Aveiro, Portugal
tomecarvalho@ua.pt

***Abstract*—The Olivetti dataset was chosen from the proposal list in the project instructions, available on Kaggle. [1]. Facial recognition is a technology capable of matching a human face from a digital image or a video frame against a database of faces in order to identify the person. Our main goal in this project is to explore different machine learning models that can be used in such a system and compare their performance.**

***Index Terms*—facial recognition, machine learning, classification, data analysis, logistic, c-support vector classification, linear discriminant analysis, gaussian naive bayes, k neighbors, decision tree, convolutional neural network, deep learning, principal component analysis**

I. Introduction

Nowadays, facial recognition is ubiquitous in our day-to-day life. It has various different applications [3].

The most well-known application is likely security and surveillance. Governments, businesses and other entities can use it to identify criminal suspects.

Another common use is for biometric authentication. The most intuitive example is unlocking smartphones, using technologies such as Android's Face Unlock or Apple's Face ID.

Additionally, it is also used for purposes such as assisting in the search for missing people, facial reconstruction (e.g., of witnesses), etc.

As it is quite a relevant technology with a multitude of different applications, it is useful to investigate which models lead to robust implementations, our objective in this project.

II. State-of-the-art Review

Paper [5] states that CNN have been found to be the most useful and accurate type of deep learning method for face recognition, the benefit of using neural networks is that it can reduce the dimensionality and can be trained as a classifier and that the most common architectures used for CNN are VGGnet and GoogleNet, both achieving a very comparable level of face recognition accuracy.

Paper [7] highlights six types of algorithms: SVM, CNN, Eigenface based algorithms, Gabor Wavelet, PCA, and HMM. It referenced other papers, comparing the performance results of each type of algorithm across them, although the results seemed to vary a significant amount.

Fig. 1, from paper [4] compares the accuracy of multiple traditional (non-deep learning) machine learning methods. Support Vector Machines (SVM) showed the best results, 98.80%, followed by Artificial Neural Networks with 97.25% (ANN) and Logistic Regression (LR) with 96.44%. K-nearest Neighbors (KNN), Naive Bayes (NB) and Random Forest (RF) paled in comparison to these, with 95.14%, 94.82% and 95.56% accuracy, respectively.

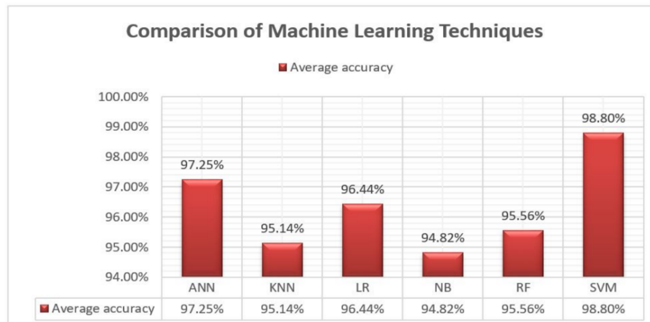

Fig. 1. Comparison of Machine Learning Techniques [4]

Name	Method	Images (Millions)	Accuracy
Baidu[23](Announced)	CNN	-	0.9985 ± -
Google FaceNet[5]	**CNN**	**200.0**	**0.9963 ± 0.0009**
DeepID3[24]	CNN	0.29	0.9953 ± 0.0010
MFRS[25]	CNN	5.0	0.9950 ± 0.0036
DeepID2+[26]	CNN	0.29	0.9947 ± 0.0012
DeepID2[3]	CNN	0.16	0.9915 ± 0.0013
DeepID[27]	CNN	0.2	0.9745 ± 0.0026
DeepFace[6]	CNN	4.4	0.9735 ± 0.0025
FR+FCN[28]	CNN	0.087	0.9645 ± 0.0025
TL Joint Bayesian[29]	Joint Bayesian	0.099	0.9633 ± 0.0108
High-dim LBP[30]	LBP	0.099	0.9517 ± 0.0113

Fig. 2. Comparison of Facial Recognition Systems [4]

Fig. 2, from paper [6] compares the accuracy of eleven state-of-the-art facial recognition systems, highlighting Google FaceNet, with an outstanding accuracy of 99.63% as the winner. Out of the systems listed, nine use Convolutional Neural Network (CNN) as the machine learning model. This was a great incentive for us to apply the CNN deep learning model to our dataset.

Fig. 3, from paper [8] compares specific methods' performance across several databases. The vast majority of them are out of the scope of the subject. Nevertheless, CNN is mentioned again in the second approach, which had an accuracy of 99.3% on the ORL database, further incentivizing us to implement CNN.

Table 1: Summary of global approach reviews

Ref. no.	Method	Database	Performance
[10]	multi-scale Gabor and center-symmetric local binary pattern (CSLBP)	ORL, Yale-B, and Yale databases	ORL (100% rank-1), Yale-B (97.5% rank-1), Yale databases (93.3% rank-1)
[26]	AHF based on Eigenface and CNN	ORL database	99.3%
[27]	group representation-based classification method	ORL, Georgia-Tech, FERET, CM-PIE and Libor face databases	92.5% (ORL), 60.55% (Georgia-Tech), 77.37% (CM-PIE), 83.35% (Libor)
[28]	TLSRWF	ORL, FERET, LFW	95% (ORL), 95% (FERET), 83.33% (LFW)
[29]	2D Discrete Wavelet Transform with Single-Level and Gaussian Low-Pass Filter, The Local Binary Pattern, Gray Level Co-Occurrence Matrix, and the Gabor filter were used for feature extraction	ORL and Yale	100% for both databases
[30]	Local Cross Pattern (LCP)	AT&T, CIE, Face94, and FERET	AT&T (97.3%), CIE (100%), Face94 (100%), and FERET (96.3%)
[31]	H-CRC	ORL, Georgia Tech, FERET, FRGC, PIE and LFW	ORL (99.17%), Georgia Tech (79.88%), FERET (88.77%), FRGC (95.47%), PIE (93.82%) and LFW (52.53%)
[32]	LOG-DTCWT 1	Extended Yale Face Database B and the CMU-PIE face database	Extended Yale Face Database B (87.32%) and the CMU-PIE 100%)
[33]	KL-HC-HOG	CMU database and Ploy-U database	CMU (99.4%), Ploy-U (97.45%)
[34]	self-residual attention-based convolutional neural network (SRANet)	LFW, AgeDB, CFP	LFW (99.83%), AgeDB (98.47%), CFP (95.6%)
[35]	The Gaussian Filter, Local Binary Patterns	FACES94, FACES95, FACES96, Grimace	93.6%, 90.6%, 91.6%, and 96.6%
[36]	contourlet transform (CNT) and curvelet transform (CLT)	JAFFE, ORL and FERET	JAFFE (97.19%), ORL (98.79%), FERET (98.1%)
[37]	Lambertian reflectance model	A database combined from Yale B and extended Yale B	95.06%
[38]	the rule-based method derived from Canny Edge features	CALTECH face databases	Approximately 89.1%

Fig. 3. Summary of global approach reviews [8]

III. Dataset

A. Dataset description

The dataset used is available on Kaggle [1]. Credits to AT&T Laboratories Cambridge [2]. It comprises a total of 400 face images, with the following properties:

- Images taken between April 1992 and April 1994 at AT&T Laboratories Cambridge.
- There are ten different images of each of the 40 distinct people.
- The images were taken at different times with variations of lighting, facial expressions and facial details.
- The background of the images is black.
- The images are grayscale.

Fig. 4. An image from each distinct individual in the dataset.

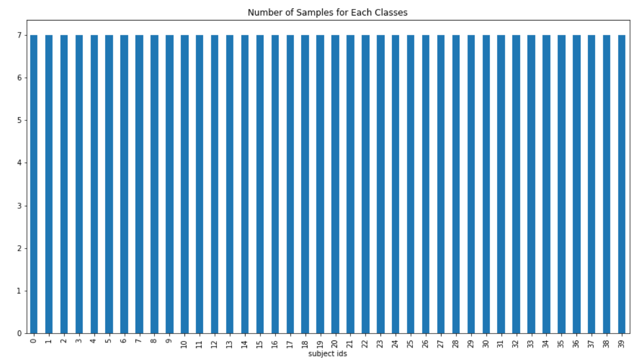

Fig. 5. Distribution of images for each person.

- The dimension of each image is 64x64 pixels.
- Image pixel values were scaled down to the [0, 1] interval.
- The names of the 40 people were encoded to integer IDs, from 0 to 39.

B. Preprocessing

As previously stated, the images are grayscale, and, in addition, their pixels' values have already been downscaled to values in [0, 1]. That being the case, it was not necessary to apply normalization.

It is, however, worth nothing that we had to reshape the training and test X data to (-1, 64, 64, 1) so that it would be compatible with the 2D convolution layers' input shape.

C. Feature distribution

Fig. 5 confirms that the dataset's 400 images are distributed among 40 different people, with 10 images for each one.

Fig. 6 shows the 10 images of 5 randomly selected subjects from the dataset.

According with Fig. 17, its possible to see that, using LDA, most faces are correctly categorized frequently. However, there are a few exceptions which were more often confused.

D. Principal Component Analysis

Principal Component Analysis (PCA) was used for dimensionality reduction. In order to make an appropriate choice of the number of components, we made use of a convenient property of scikit-learn's PCA object's *n_components* parameter:

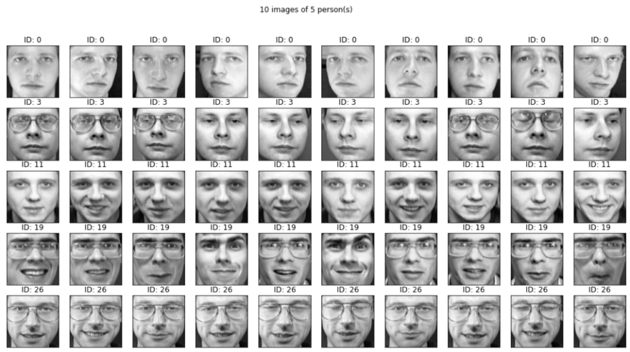

Fig. 6. Group of face images for each person.

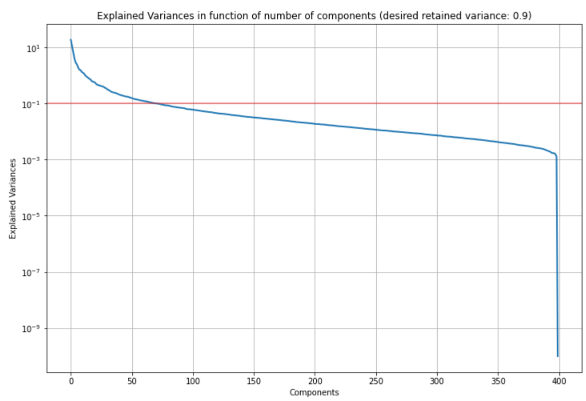

Fig. 7. Explained Variances in function of number of components.

passing a number in]0, 1[will select the minimum number of components such that the explained variance is greater than the percentage specified by *n_components*. We experimented setting the desired retained variance to values between 90% and 99%, with steps of 1%. We had also experimented with 0.89 earlier on, but our observations were not relevant. That is the reason why LOO cross-validation does not have data for this value of the desired retained variance. We found 90% to be the value that yielded the best results. As evidenced in Figs. 7 and 8, the corresponding number of components k was 66.

The average face resulting from PCA is visible in Fig. 9.

Fig. 10 plots the first two principal components of ten people from the dataset. The most notable example is seen in the left side of the graph, where a lot of the fifth person's images are found close together. We must keep in mind that this visualization encompasses only the first two principal components, which is only 3% of the number of components kept.

Each PCA component corresponds to an eigenface, which is the graphical representation of an eigenvector. Eigenfaces are a useful method for facial recognition and detection, by determining the variance of faces in a collection of face images and using those variances to encode and decode a face in a machine learning way without the full information, reducing computation and space complexity. All 66 eigenfaces

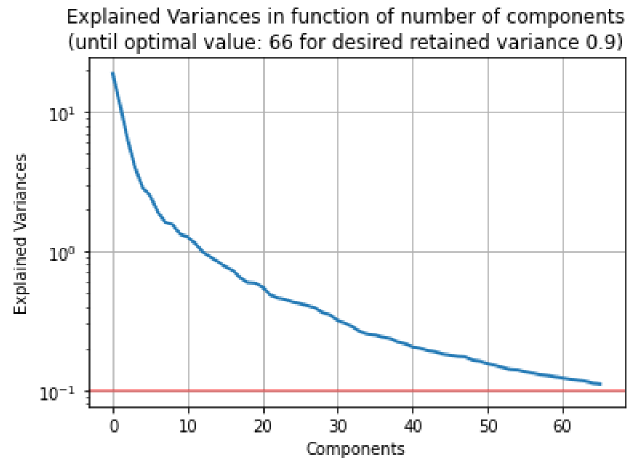

Fig. 8. Explained Variances in function of number of components with logarithm scale.

Fig. 9. Average Face determined by the PCA.

are displayed, in the corresponding PCA component order, in Fig. 11.

IV. Regression Model

A. Types of regression

Since the goal is to identify the face's person, the use of a classification model is necessary. We applied the following classification methods:

- **Linear Discriminant Analysis (LDA)** [10]: fits a Gaussian density to each class, assuming that all classes share the same covariance matrix.
- **Logistic Regression (LR)** [11]: predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
- **Gaussian NB (GNB)** [13]: performs online updates to model parameters via $partial_{f}it$.

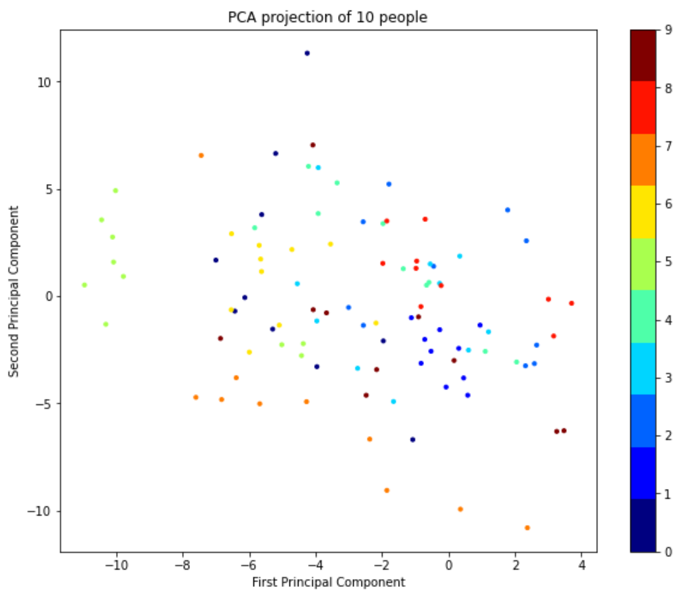

Fig. 10. PCA: Projection of the first two components of 10 people.

Fig. 11. Eigenfaces

- **Decision Tree Classifier (DTC)** [14]: has the capability of capturing descriptive decision making knowledge from the supplied data.
- **C-Support Vector Classification (SVC)** [15]: implements LIBSVM [16], which implements linear SVMs and logistic regression models trained using a coordinate descent algorithm.
- **Convolutional Neural Network (CNN)** [17]: a neural network, in this particular case, with 11 layers, including 2D convolution , 2D max pooling, dropout, flatten, and dense layers.

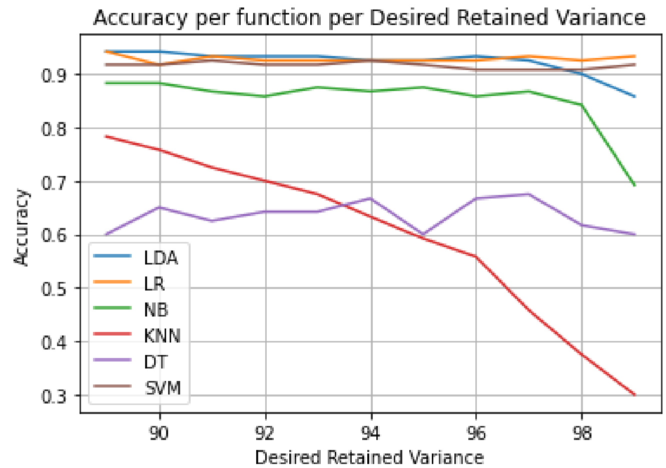

Fig. 12. Accuracy of each classifier in the Simple Validation method.

TABLE I
MAXIMUM ACCURACY BY EACH METHOD

Method	Best Classifier	Max. Accuracy
Simple Validation	LDA	0.942
K-FCV	LDA	0.980
LOOCV	LDA	0.980
Deep Learning	CNN	0.985

B. Testing accuracy

To obtain the most accurate model, we proceeded to analyze the accuracy of each aforementioned model using the following methods:

- **Simple Validation**: Distribution of the Train and Test samples (7:3 ratio). The model then fits the training samples and evaluate the test. In Fig. 12 shows the accuracies obtained with this method.
- **K-Fold Cross Validation (K-FCV)**: Distributes all samples between Train and Test groups (folds) multiple times. Fig. 13 represents its accuracy.
- **Leave-One-Out Cross Validation (LOOCV)**: Because K-FCV is fold-based and the dataset does not have a lot of data, but only 400 images, LOOCV was used as a better alternative to K-FCV. The former uses a similar strategy, but instead of splitting the data into K folds, each sample is used once as a test set (singleton) while the remaining samples form the training set. Its results are shown in Fig. 14.
- **Deep Learning**: CNN, being a deep learning method, had to be handled separately. Unlike the basis we used [9], we did not use a generator and *fit_generator*, using *fit* instead and then plotting the accuracy and loss over the epochs.

C. Graphical representation of results

Observing Table I along with Figs. 12, 13, 14 and 15, it is possible to conclude that LDA was the best performing method

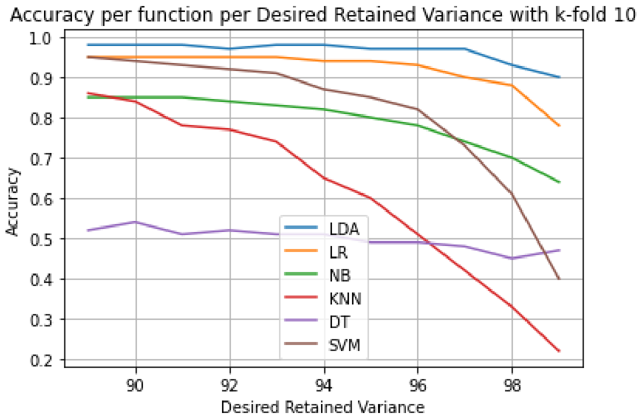

Fig. 13. Accuracy of each classifier in the K-FCV method.

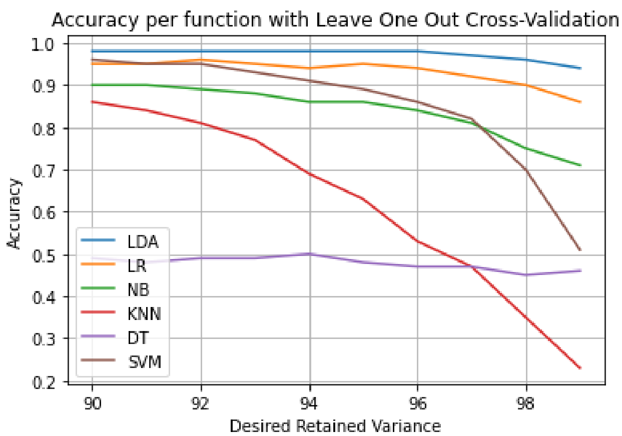

Fig. 14. Accuracy of each classifier in the LOOCV method.

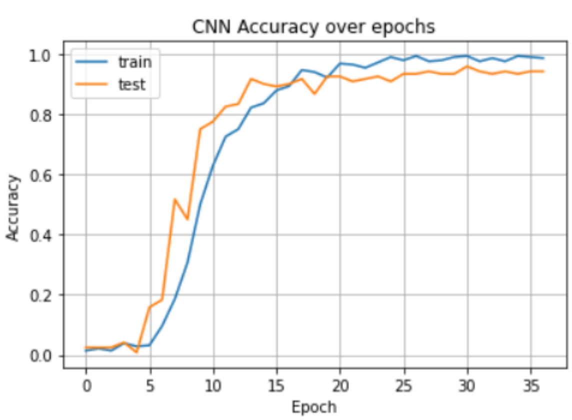

Fig. 15. CNN's accuracy in the Deep Learning method.

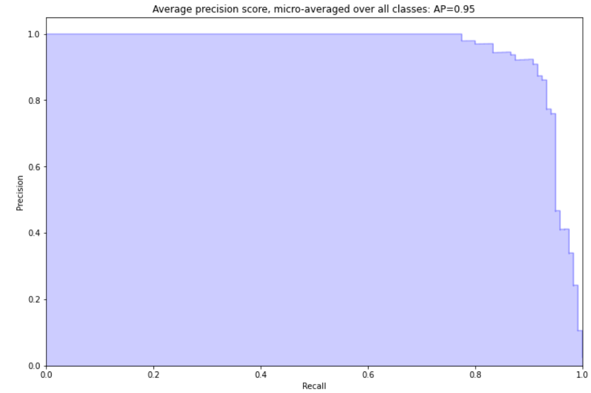

Fig. 16. LDA Precision Recall Curve

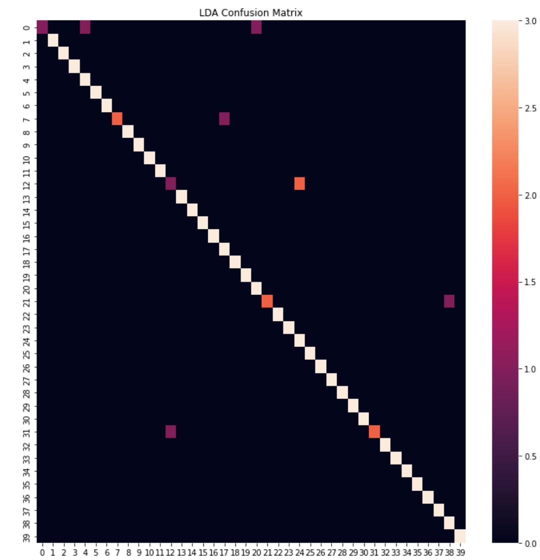

Fig. 17. LDA Confusion Matrix.

among the traditional ones, while CNN, the deep learning method, was able to perform slightly better.

1) Linear Discriminant Analysis Confusion Matrix: Since LDA proved to perform the best out of the traditional machine learning models, we found it relevant to plot its confusion matrix at Fig. 17. Additionally, Fig. 16, shows that LDA was able to achieve both good accuracy and precision values.

D. Feature Importances in the Random Forest Regression

V. Conclusion

Regarding our results, LDA had the best accuracy among the traditional machine learning methods, for all validation strategies (simple validation 94.2%, K-fold cross-validation and leave-one-out cross-validation 98%). The deep learning method, CNN, achieved a slightly higher accuracy of 98.5%.

We were able to accomplish the goals of our project successfully. Namely, investigating state-of-the-art facial recognition methods, analyzing the dataset, applying and comparing multiple methods of classification, including a deep learning one, as well as drawing conclusions from the results obtained.

VI. Division of Labor

Both students collaborated an equal amount through online meetings for synchronous development, in order to achieve collective responsibility over the entirety of the project.

Glossary

CNN Convolutional Neural Network. 4–6

DTC Decision Tree Classifier. 4, 6

GNB Gaussian NB. 3, 6

K-FCV K-Fold Cross Validation. 4–6

LDA Linear Discriminant Analysis. 3–6

LOOCV Leave-One-Out Cross Validation. 4–6

LR Logistic Regression. 3, 6

PCA Principal Component Analysis. 2, 3, 6

SVC C-Support Vector Classification. 4, 6

References

[1] Olivetti Dataset
[2] AT&T Laboratories Cambridge
[3] Trio: Facial Recognition Applications
[4] Comparative study of machine learning algorithms for face recognition
[5] Machine learning approaches to face detection and recognition
[6] Deep learning and face recognition: the state of the art
[7] A State-of-the-Art Survey on Face Recognition Methods
[8] Face Recognition State-of-the-art, Enablers, Challenges and Solutions: Review
[9] Olivetti Classification using KNN and CNN
[10] Sklearn - Linear Discriminant Analysis
[11] Sklearn - Random Forest Regressor
[12] What is Logistic Regression - Tech Target
[13] Sklearn - GaussianNB
[14] Decision Tree Classifier - an overview - ScienceDirect Topics
[15] Sklearn - SVC
[16] LIBSVM - Wikipedia
[17] CNN - Wikipedia