Development and Evaluation of a Facial Recognition System with Classifiers

Marc Blánquez Barroso (1550960), Guillem Cadevall Ferreres (1632005), Marc Serrano Sanz (1632081)

Abstract—This study develops and evaluates a facial recognition system using the Olivetti Faces dataset. We employ Principal Component Analysis (PCA) to reduce data dimensionality and identify key facial features. Various models, including KNN and EigenFaces, are utilized to build our classifier. Our experiments show that the EigenFaces algorithm, combined with data augmentation techniques, achieves the highest accuracy, significantly improving facial identification performance. This system will be integrated into a robot developed for our Robotics course.

Keywords—Image classification, noisy web data, ubiquitous reweighting

---

1 INTRODUCTION

In this paper, we will explore the development of a facial recognition system using the Olivetti Faces database, Principal Component Analysis (PCA) techniques, and classifiers. The results of this project will be utilized for a robot we are developing.

The Olivetti Faces database provides a diverse set of facial images that will serve as the foundational material for our facial recognition system. By employing PCA techniques, we can reduce the dimensionality of the data and identify the most relevant facial features essential for recognition tasks.

Through this work, we aim to develop an efficient and accurate facial recognition system by leveraging the rich data from the Olivetti Faces database, the analytical power of PCA, and the capabilities of various classifiers. This will enable us to explore and better understand the functionality of facial recognition systems and their potential applications in various contexts.

2 STATE OF THE ART

OpenFace

is an open-source facial recognition library that offers tools for facial feature extraction, facial recognition, and facial verification. It employs several techniques to achieve these functionalities:

Facial Detection: Utilizes algorithms to detect and locate faces in images. These algorithms can include cascade-based detectors, feature-based detectors, or convolutional neural networks (CNNs).

Facial Alignment: Aligns the detected faces to ensure they are in a standard position and that the facial features are consistently positioned for subsequent processing.

Feature Extraction: Uses deep learning techniques to extract relevant features from faces, such as key point descriptors or facial representation embeddings.

Facial Recognition: Employs classification methods to identify faces based on the extracted features. This can include support vector machines (SVM), neural networks, or other classification techniques.

Facial Verification: Determines if two faces belong to the same person. This process involves calculating the similarity between the embeddings of the two faces and applying a decision threshold to determine if they are the same person.

3 PROPOSAL/METHODOLOGY

Firstly, a modification is made to the structure of the database to be used. Initially, this database comprises a set of 400 images, with each person having 10 images. To work more efficiently and systematically, a folder is created for each individual. The folder name corresponds to the person's identifier (target). The content of the folder consists of the images of that individual.

Additionally, all images in the database are converted to grayscale, normalized by dividing the pixel values by 255, and converted into a one-dimensional array.

Subsequently, the effectiveness of various feature extraction techniques is evaluated. The methods to be assessed include HOG, PCA, and Eigenfaces.

The HOG (Histogram of Oriented Gradients) algorithm for facial detection counts the oriented gradients in different parts of the image to describe its local structure, facilitating the identification of facial features such as contours and essential traits.

The PCA (Principal Component Analysis) algorithm reduces the dimensionality of the images by transforming them into a set of principal components. As fewer components are used, the image becomes less detailed and more simplified.

In Eigenfaces, images are converted into vectors, and their principal components are obtained through PCA.

Next, classifiers to be evaluated are selected, in this case: k-NN and Random Forest.

K-NN (k-Nearest Neighbors) is a classification and regression method in supervised learning. It uses the classes of the samples closest to a new observation to predict its label. The "proximity" between samples is measured in a multidimensional space and is defined by the parameter k, which specifies the number of neighbors to consider.

Random Forest is a supervised learning algorithm primarily used for classification and regression tasks. This algorithm creates a set of decision trees during the training process. Each tree is trained with a random subsample of the original dataset and uses a random selection of features for each split. For a new observation, each tree in the forest makes a prediction. The final prediction is determined by majority voting (in the case of classification) or averaging (in the case of regression) of all individual tree predictions.

Following the determination of the best combination of feature extraction and classifier, the impact of the database size on this efficient model is analyzed by modifying the data magnitude through various data augmentation techniques and selecting the optimal combination.

The data augmentation techniques used include: image rotation (such as mirroring), changing illumination levels, adding noise, and blurring.

The primary metric used to determine the best model evaluation method and data augmentation techniques is accuracy.

4 Experiments, Results, and Analysis

The accuracy of the following models has been analyzed: Random Forest with raw data, Random Forest with a Histogram of Oriented Gradients (HOG) applied to the data, K-Nearest Neighbors (KNN) with raw data, KNN with HOG applied to the data, KNN with both HOG and Principal Component Analysis (PCA) applied to the data, Eigenfaces, Eigenfaces with KNN, and Eigenfaces with KNN and HOG applied to the data.

Random Forest: The results with and without HOG reach at most 0.89 in accuracy. This performance, combined with a longer execution time compared to K-Nearest Neighbors (KNN), leads to the exclusion of this option as a classifier.

K-Nearest Neighbors (KNN) with raw data: 0,8916

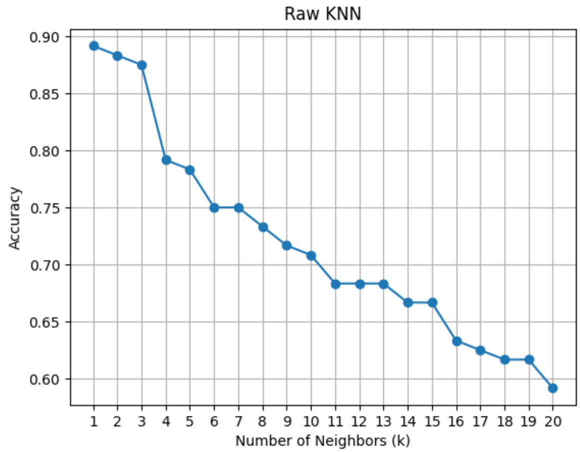

Fig. 1. Accuracy vs. Number of Neighbors Graph for K-Nearest Neighbors (KNN) with Raw Data

HOG KNN: 0,8916

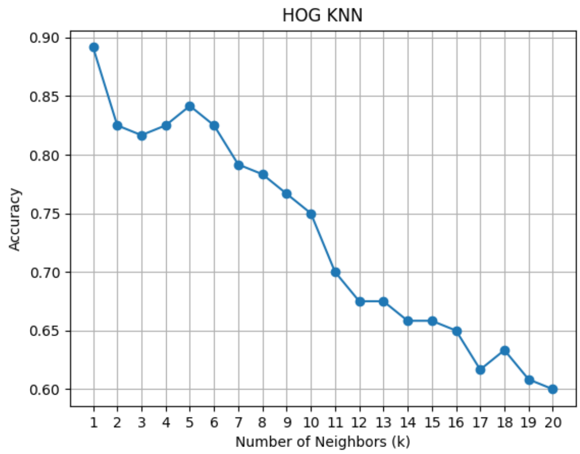

Fig. 2. Accuracy vs. Number of Neighbors Graph for K-Nearest Neighbors (KNN) with HOG Data

PCA KNN: 0,8916

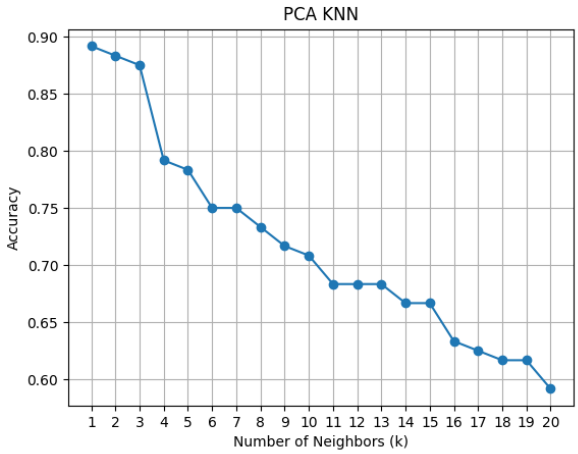

Fig. 3. Accuracy vs. Number of Neighbors Graph for K-Nearest Neighbors (KNN) with PCA Data

PCA HOG KNN: 0,9

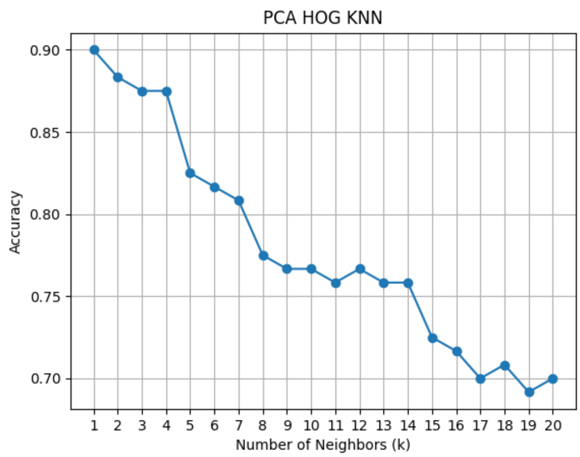

Fig. 4. Accuracy vs. Number of Neighbors Graph for K-Nearest Neighbors (KNN) with PCA and HOG Data

EigenFaces: 0,9

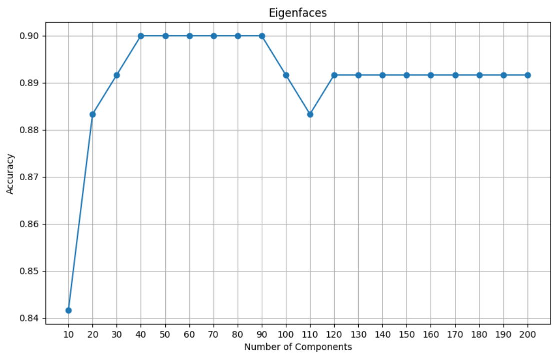

Fig. 5. Accuracy vs. Number of Components Graph for Eigenfaces with Raw Data

EigeFaces KNN: 0,9083

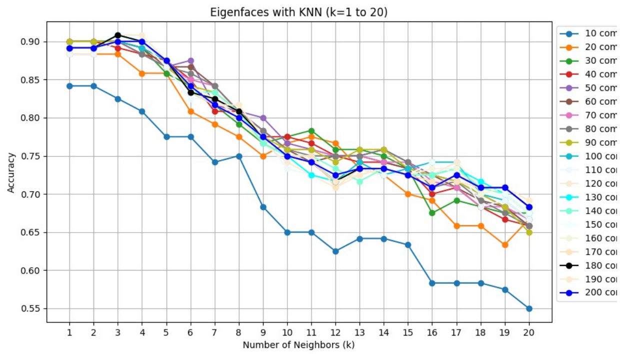

Fig. 6. Accuracy vs. Number of Neighbors and Number of Components Graph for Eigenfaces with Raw Data

EigenFaces KNN HOG: 0,9083

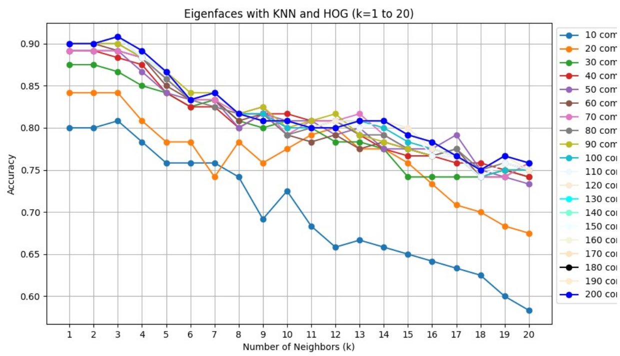

Fig. 7. Accuracy vs. Number of Neighbors and Number of Components Graph for Eigenfaces with HOG Data

After obtaining these results, it is observed that the best method is Eigenfaces when using the first 3 neighbors and 100 components. It is also noted that applying HOG to the data does not make a difference. The chosen method is the one that applies HOG, as using HOG yields better results in the case of K-Nearest Neighbors (KNN) with PCA and HOG.

Upon determining the best method and identifying the optimal value of k for this model, the data size is adjusted by applying the aforementioned techniques.

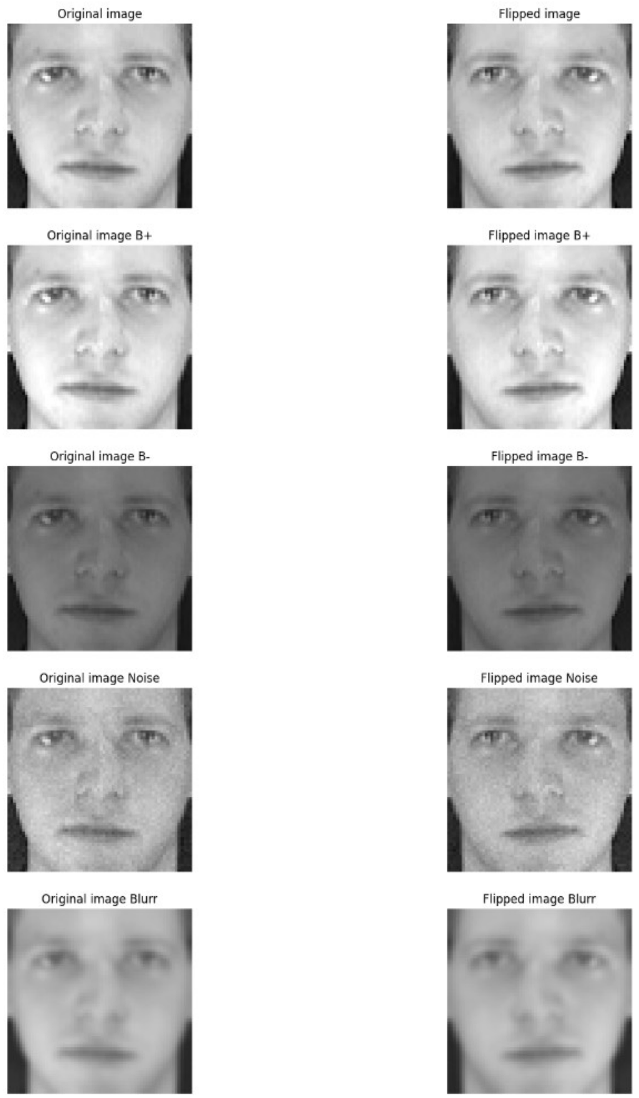

Fig. 8. Examples of All Types of Data Augmentation Performed

The following graphs have been generated to evaluate the data augmentation:

<u>Eigenfaces with Flipped images</u>: 0.9667

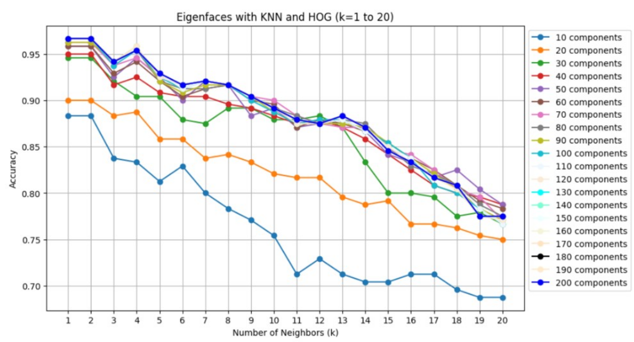

Fig. 9. Accuracy vs. Number of Neighbors and Number of Components Graph for Eigenfaces with Flipped Images

<u>Eigenfaces with Flipped Images + Intensity Augmentation/Reduction</u>: 0,5617

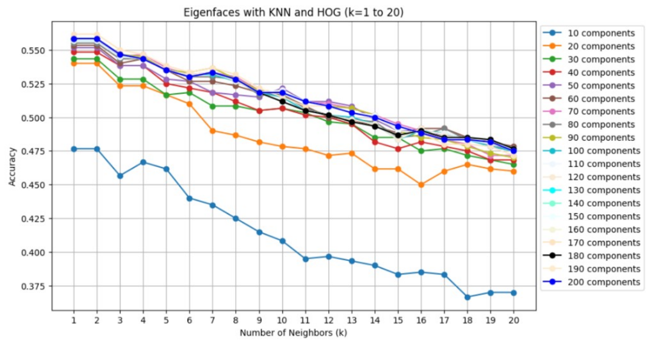

Fig. 10. Accuracy vs. Number of Neighbors and Number of Components Graph for Eigenfaces with Flipped Images and Illumination

<u>Eigenfaces with Flipped Images + Intensity Augmentation/Reduction + Noise + Blur</u>: 0,99417

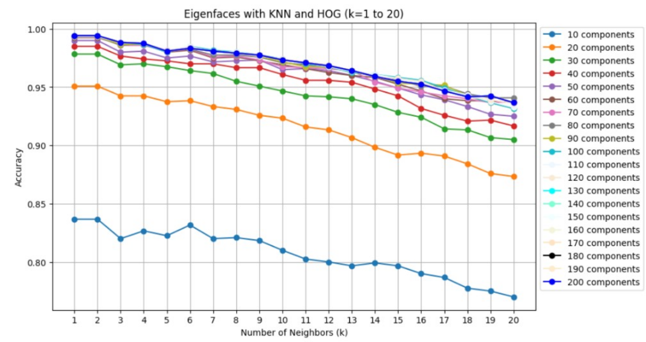

Fig. 11. Accuracy vs. Number of Neighbors and Number of Components Graph for Eigenfaces with Flipped Images, Illumination, Noise, and Blur

<u>Eigenfaces with Flipped Images + Noise + Blur</u>: 0,9805

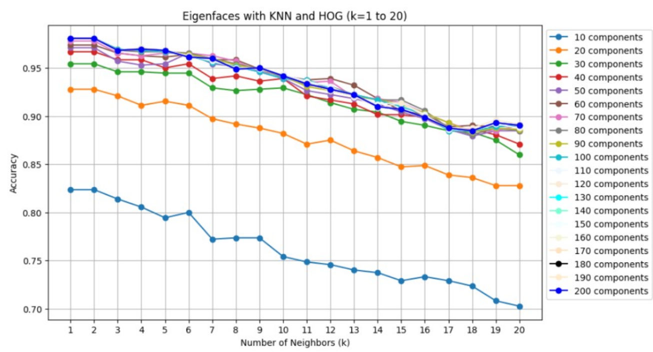

Fig. 12. Accuracy vs. Number of Neighbors and Number of Components Graph for Eigenfaces with Flipped Images + Noise + Blur

As can be observed, the best results are achieved by applying all types of data augmentation, reaching an accuracy level of 0.99417. It is also noted that using either 1 or 2 neighbors yields similar performance. The worst result was obtained when only flipping the images and making illumination adjustments; therefore, tests were conducted without these illumination modifications. Nevertheless, the best results continue to be achieved with all augmentations applied.

To conclude this project, a use case has been developed to demonstrate the results obtained during its execution. In this use case, the dataset with all implemented data augmentation mechanisms is used. Subsequently, the computer captures a photograph, extracts the face (if present), and processes it to calculate the distance using the Eigenfaces algorithm. If any image in the dataset has a proximity below a given threshold, that image is displayed along with the number identifying the person in the database.

5 Conclusions

After evaluating the performance of all models with our dataset and adjusting the dataset size to achieve the highest possible accuracy, it is concluded that the Eigenfaces algorithm with images processed using HOG yields the best results. By applying data augmentation, an accuracy of 99.4% was achieved, surpassing other models in all aspects.

In the future, exploring additional data augmentation techniques and combining different recognition methods could further enhance accuracy. Additionally, delving into neural networks could potentially improve the precision of the process.

Overall, a facial recognition classifier capable of identifying individuals with a high probability of accuracy has been developed. This has been validated by incorporating personal images into the dataset through the use case, confirming that recognition was accurate in nearly all tests performed.

Bibliografia

[1] https://github.com/TadasBaltrusaitis/OpenFace