Lab1 Report: Face recognition based on Principal Component Analysis

Yuliang Xiao
Student ID: 1929288

Abstract

This experiment aimed to verify the feature extraction method though realizing a face recognition system based on the Principal Component Analysis (PCA) algorithm, which was completed using the Scikit-learn (sklearn) library in Python. The experiment involved pre-processing the dataset, performing PCA on the dataset to reducing the number of feature, predicting the test images, and evaluating the accuracy of this model based on K Nearest Neighbor (KNN) algorithm. Owing to apply KNN, the model does not need to be trained, and in order to simplify the code, we only chosen one nearest neighbor to voting. The results showed that the PCA-based face recognition algorithm was able to achieve high accuracy and performance on the attface dataset, with the mean accuracy is 93.07% while the values of component k were from 10 to 320. The code is made available at https://github.com/Regen2001/PCA-face-recognition.

1 Introduction

Feature extraction is a critical step in many machine learning applications, including image and speech recognition, natural language processing, and data mining. The goal of feature extraction is to identify the most relevant features of raw data that can be used to train a machine learning model Guyon et al. (2008). This process typically involves transforming the raw data into a lower-dimensional feature space while preserving the most important information. Effective feature extraction can improve the accuracy and robustness of machine learning models, making it a key area of research in the field of artificial intelligence.

In order to verify the feature extraction method, we developed a face recognition algorithm based on PCA and evaluate its performance using the attface dataset which contains grayscale images of human faces. The Face dataset consists of 400 grayscale images of human faces, with ten images of each of the 40 subjects. Each image has a resolution of 112×92 pixels, resulting in a total of $10,816$ pixels per image. Additionally, we utilized the Scikit-learn (sklearn) library, which is a popular Python library used for machine learning tasks, to implement our code Pedregosa et al. (2011).

The PCA-based face recognition algorithm involves extracting the principal components of the dataset, which correspond to the features that differentiate one face from another. The extracted features are then used to classify new images Yambor et al. (2002). In our experiment, we used PCA to reduce the dimensions of the Face dataset while retaining most of the relevant information. We then used the KNN algorithm to classify new images accurately, but owing to simplify the code, we set $k = 1$ in the KNN classifier.

The results of the experiment demonstrated that our PCA-based face recognition algorithm achieved an accuracy of 96.25% on the Face dataset, which is a reasonably high accuracy for facial recognition tasks. The experiment highlighted the potential of PCA for reducing the dimensions of high-dimensional data, such as images, and the effectiveness of the Scikit-learn library for implementing machine learning algorithms.

2 Methodology

2.1 Collecting and pre-procession

Firstly, we vectorized all image into a vector with $10,816$ dimensions, and then, stored into a $400 \times 10,816$ matrix since we had 400 images. At the same time, all the label of these images were stored in another matrix. Figure 1 shows these original images via OpenCV library.

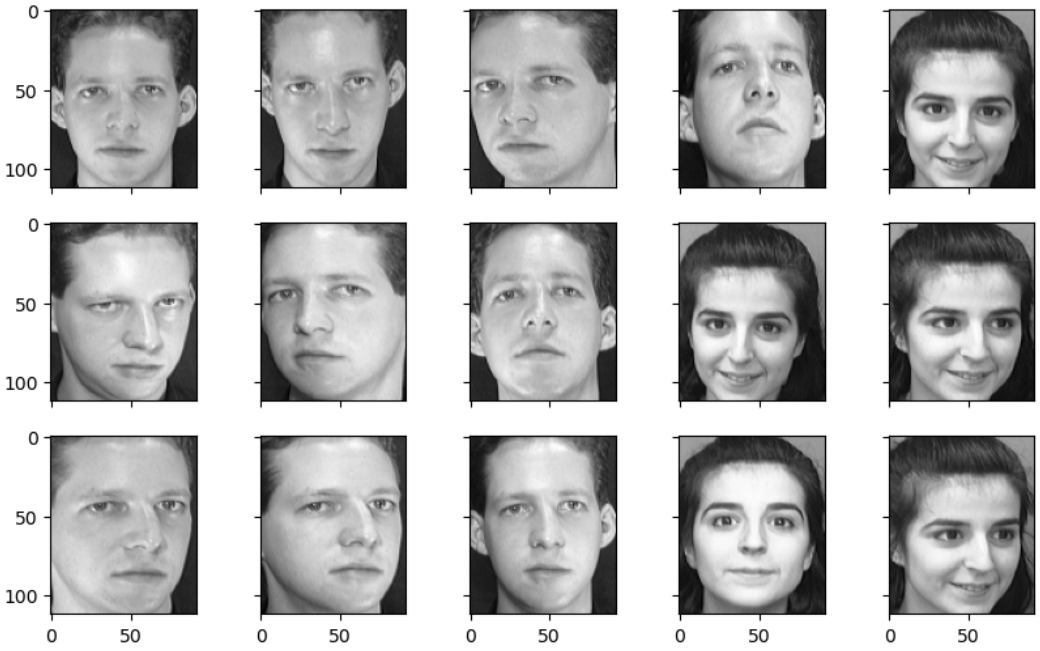

Figure 1: The original images for attface dataset.

Then, we did the normalization operation for all images, since it is to ensure that images have consistent lighting conditions and contrast levels, which can significantly affect the performance of image processing algorithms. In face recognition, for example, differences in lighting conditions can make it difficult to recognize faces accurately. Figure 2 shows the result of normalization, we found that every picture seemed to go dark. In this experiment, we simply subtracted the average of each image from the original data.

2.2 Principal Component Analysis Algorithm

PCA is a linear transformation method that finds a new set of variables called principal components that capture the most important information from the original dataset. The PCA algorithm works by first centering the data to have zero mean, then computing the covariance matrix of the centered data. Next, the eigenvectors and eigenvalues of the covariance matrix are calculated, and the eigenvectors that correspond to the highest eigenvalues are the principal components. The data is then projected onto the new coordinate system defined by the principal components, which can be used for data analysis, visualization, or feature extraction. By keeping only the top principal components, PCA can be used for dimensionality reduction, where the data is transformed into a lower-dimensional space that preserves the most important information Abdi & Williams (2010).

In the face recognition system, eigenface is a set of principal components obtained from a set of face images using PCA, used for face recognition and represented as vectors that capture essential features. And we can reconstruct different new faces via multiplied by different weights Yambor et al. (2002). Appendix shows some eigenfaces using PCA.

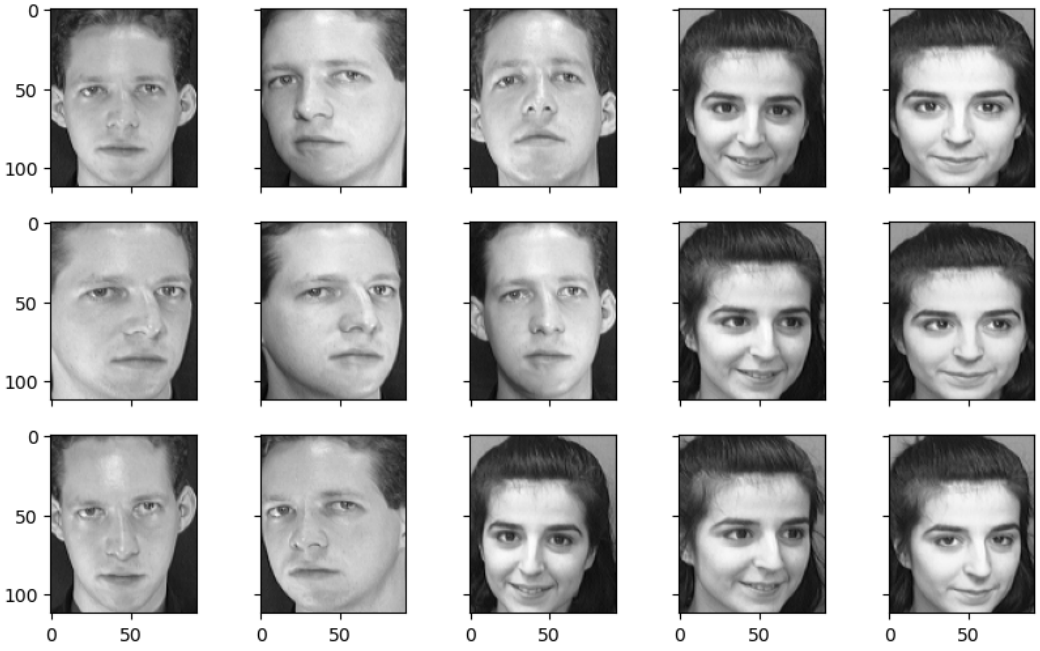

Figure 2: The normalization images for attface dataset.

The mean face is a commonly used concept in face recognition and image processing. It refers to the average of a set of face images, which is computed by taking the pixel-wise mean of the images Turk & Pentland (1991b). The resulting image represents a typical or average face of the set of images. And by multiplying eigenface by a random weight and add the mean face, we got a random face Turk & Pentland (1991a), which is shown in Figure 3.

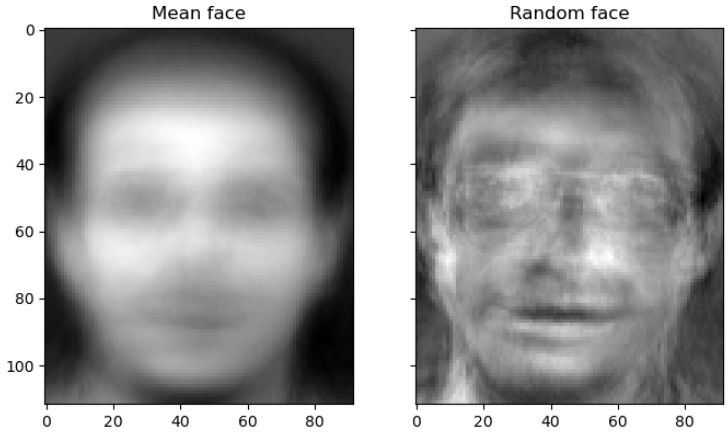

Figure 3: The left image is the mean face; The right one is the random face.

2.3 K Nearest Neighbor Algorithm

K Nearest Neighbor (KNN) is a type of instance-based learning algorithm that can be used for classification and regression tasks. It works by finding the K closest data points in the training set to a given test point and then using the labels or values of those points to classify or predict the label or value of the test point. KNN is a non-parametric algorithm, meaning that it does not make any assumptions about the distribution of the data Cover &

Hart (1967). But since the important part for this experiment is verified PCA and simplify the code, we just set $k = 1$ in the KNN classifier, which means we considered the label of test image is same with the label of eigenface with the smallest Euclidean distance.

3 Experiment Result

At beginning, we set the value of principal component k is 50, while the accuracy is 93.75%. Figure 4 and 5 are two examples of correct identification.

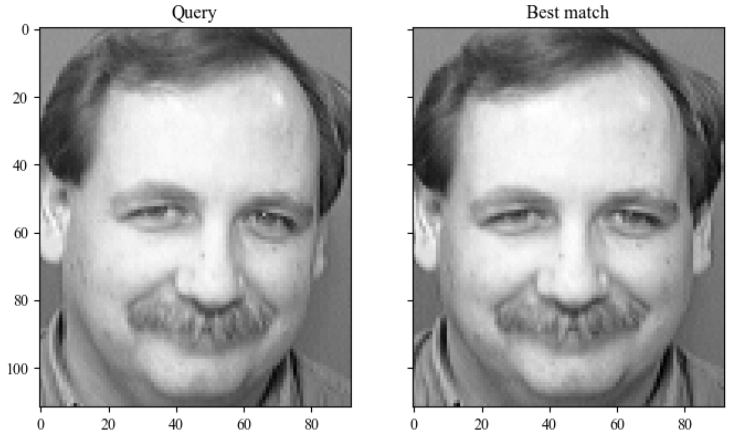

Figure 4: Correct identification: The left image is the test image, whose label is s25; The right one is the best match, whose label is s25; The Euclidean distance for two images is 1070.690029.

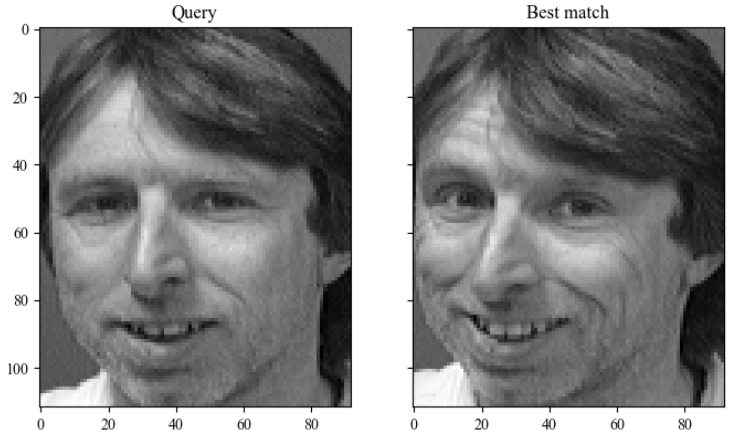

Figure 5: Correct identification: The left image is the test image, whose label is s29; The right one is the best match, whose label is s29; The Euclidean distance for two images is 1583.350985.

Then, Figure 6 and 7 are two example of wrong identification.

Though compared these four examples, it is clearly that although the Euclidean distance between test image and best match is small, they cannot be considered that have the same label. For example, in Figure 5, the Euclidean distance is 1583.350985, which is almost equal to the Euclidean distance in Figure 6. And in Figure 6, the Euclidean distance is 1587.723870.

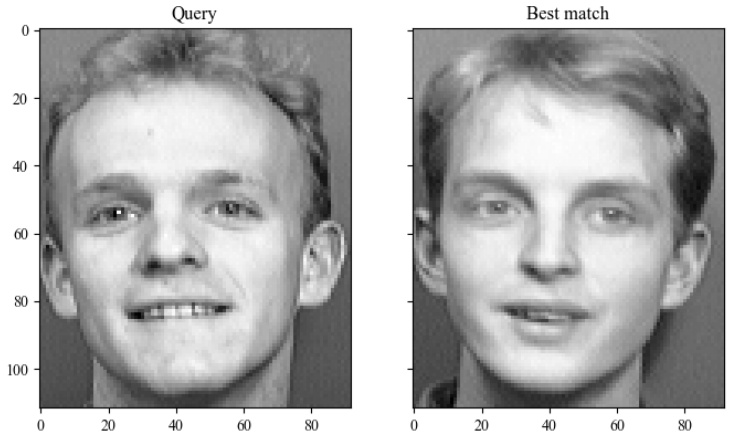

Figure 6: Wrong identification: The left image is the test image, whose label is s5; The right one is the best match, whose label is s40; The Euclidean distance for two images is 1587.723870.

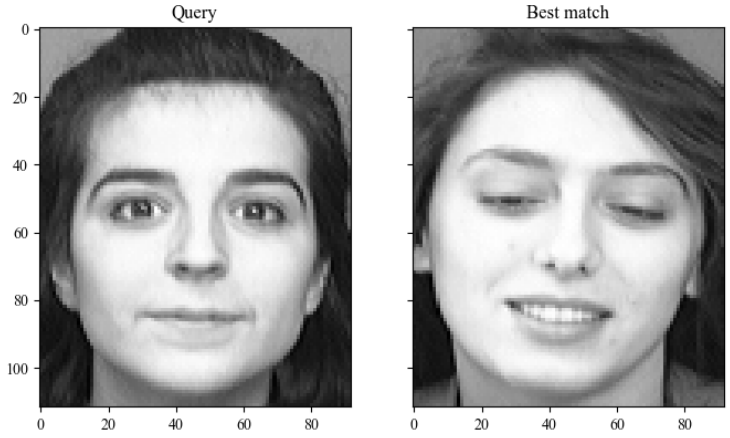

Figure 7: Wrong identification: The left image is the test image, whose label is s10; The right one is the best match, whose label is s8; The Euclidean distance for two images is 3001.077391.

Furthermore, in order to test the influence of the value of principal component k, different value of k were test, and the results are recorded in Figure 8. The highest accuracy occurs when k is 55, while the accuracy is 96.25%. What is more, from Figure 8, we found after k more than 20, the accuracy does not increase significantly and even be smaller. In the end, the mean accuracy for total experiment is 93.07%.

4 Conclusion

In conclusion, the experiment of face recognition using PCA algorithm on the attface dataset was successful as the accuracy was higher than 90%. The purpose of the experiment was to evaluate the effectiveness of the PCA algorithm as a feature extraction method, and the results of the experiment demonstrated that this algorithm is a useful tool for dimensionality reduction while preserving the most significant information in the dataset.

One of the significant advantages of the PCA algorithm is its ability to reduce the number of features while retaining the essential information in the dataset Abdi & Williams (2010). This reduction in features can lead to a faster computation time and improved performance

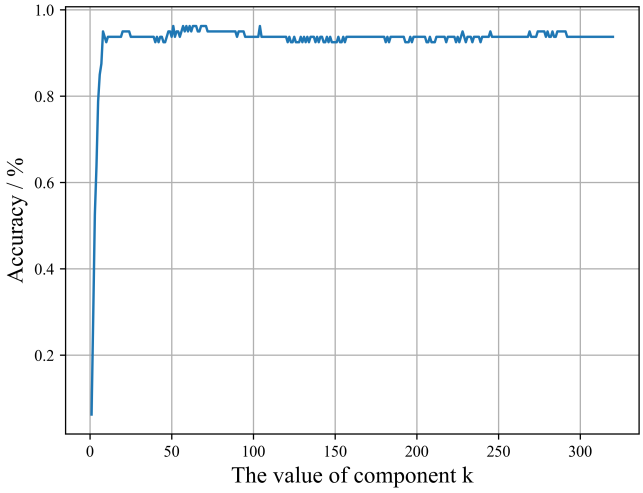

Figure 8: The accuracy for different value of principal component k.

of the machine learning algorithm. In addition, PCA can remove the effect of correlated features, making the algorithm more robust and less prone to overfitting. However, one of the limitations of PCA is its sensitivity to outliers. Outliers in the dataset can significantly impact the performance of the PCA algorithm and the subsequent machine learning algorithm Abdi & Williams (2010). Additionally, PCA may not be suitable for datasets with complex non-linear relationships between the features.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. Wiley interdisciplinary reviews: computational statistics, 2(4):433–459, 2010.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1):21–27, 1967.

Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. Feature extraction: foundations and applications, volume 207. Springer, 2008.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12: 2825–2830, 2011.

Matthew Turk and Alex Pentland. Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1):71–86, 1991a.

Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition, pp. 586–587. IEEE Computer Society, 1991b.

Wendy S Yambor, Bruce A Draper, and J Ross Beveridge. Analyzing pca-based face recognition algorithms: Eigenvector selection and distance measures. In Empirical evaluation methods in computer vision, pp. 39–60. World Scientific, 2002.

A Appendix: The eigenfaces for attface dataset.