Ukrainian Catholic University

Faculty of Applied Sciences

Business Analytics Program

Face recognition

Linear Algebra final project report

Authors:

Nazarii Peniaha

Vasyl Burak

Kyrylo Shamanskyi

17 May 2023

Contents

1 Introduction

Face recognition, a technology that can use the unique characteristics of an individual's face to verify their identity, has been rapidly developing in computer vision. This technology uses sophisticated algorithms to analyze a face's characteristics and compare them with the database of known faces. The process consists of capturing an image or video of a person's face and processing it in software for the purpose of obtaining special facial features, such as eye distance, jawline shape, and nose width.

In recent years, the application of face recognition technology gained considerable momentum and has now reached a variety of fields in which it is used, for example, for security, surveillance or identity authentication on mobile devices. This technology has led to widespread adoption, due to its potential for enhancing security, improving user experience and simplifying the identity verification process. However, despite its success, face recognition technology is not without limitations and challenges.

2 Problem setting and dataset

In light of the considerations, our team has identified a specific problem where face recognition technology can offer a valuable solution. We've seen in our community a frequent problem, where neighbours often leave their keys for the colleges and lead to inconveniences and security concerns. To address this issue, we recommend the development of an alternate identity verification scheme that uses face recognition technology at entrances to colleges and universities. By leveraging the capabilities of face recognition, we aim to provide a seamless and secure means of access control.

Initially, the idea was to implement a database with residents of the collegium, but after evaluating the needs of the algorithm and the corresponding number of photos, we abandoned this idea. We decided to make a test format, which will be a more visual model,

considering only 40 people, each of which will contain 10 photos. To implement the eigenfaces algorithm we have separated the data into 70%-part on which we have trained the algorithm and 30%-part to test its work and see the accuracy of it.

We are going to apply linear algebra techniques with a view to developing an effective and accurate facial recognition system in order to realize this project. We are going to use ATT's database of faces, which is widely used as a benchmark for facial recognition. We'll look at techniques for extracting and representing facial features, reducing data dimensionality in order to train classification models so that accurate identification can be achieved using linear algebra.

3 Overview of approaches

Appearance-Based (Holistic) Methods.
Appearance-Based methods attempt to identify faces using global representations that are based on the entire image rather than local facial features. Many methods for object recognition and computer graphics are based directly on images without intermediate 3 dimensional models, most of these methods depend on image representation that induces a vector space structure and requires dense correspondence in principle.

Global facial information is fundamentally represented by a small number of features that are directly derived from the pixel information of face images; these small number of features distinctly capture the variance among different individual faces and are used to identify unique individuals. In an appearance-based method, the whole face region is considered as an input for a face detection system to perform face recognition. Appearance-based methods can be classified into linear and non-linear subspaces. Below the explanation for linear Analysis:

1. **Linear Analysis.** There are three classic linear classifiers: PCA, LDA, and ICA, each classifier has its own representation of high dimensional face vector space based

on different statistical viewpoints. The projection coefficients are used as the feature representation of each face image through the projection of the face vector onto the basis vectors. The matching score between the test face image and the training prototype is calculated between the coefficient vectors of the images, smaller matching score leads to a better matching process.

2. **Model-Based Methods.** Model-based face recognition methods aim to construct a model of the human face that captures facial variations. Prior knowledge of the human face is highly utilized to design the model. For example, model-based matching derives the distance and relative position features from the placement of internal facial elements. Model-based methods can be made invariant to size, orientation, and lighting. The other benefits of these schemes are the compactness of the representation of face images and rapid matching. Three different extraction methods are distinguished (generic methods based on edges, lines, and curves; feature template based methods, and structural matching methods that consider geometrical feature constraints). The major disadvantage of these methods is the difficulty of automatic feature detection. Implementing any of these methods needs arbitrary decisions on which features are important.

4 Algorithm Analysis

In this project we are using the eigenfaces algorithm for face recognition with the Principal Component Analysis approach. "In order to recognize a face using an already trained model, it is enough to simply find the smallest Euclidean distance between the eigenfaces of the expanded face with the training data. However, this algorithm is not efficient due to the operation of huge data. That the reason why we use PCA".

PCA, or Principal Component Analysis, is a dimensionality reduction technique commonly used in face recognition. The algorithm works by identifying the principal components, or directions of maximum variation, in a dataset and projecting the data onto

a lower-dimensional space defined by these components. In the case of face recognition, PCA can be used to extract the most important features of a face image and represent it in a lower-dimensional space.

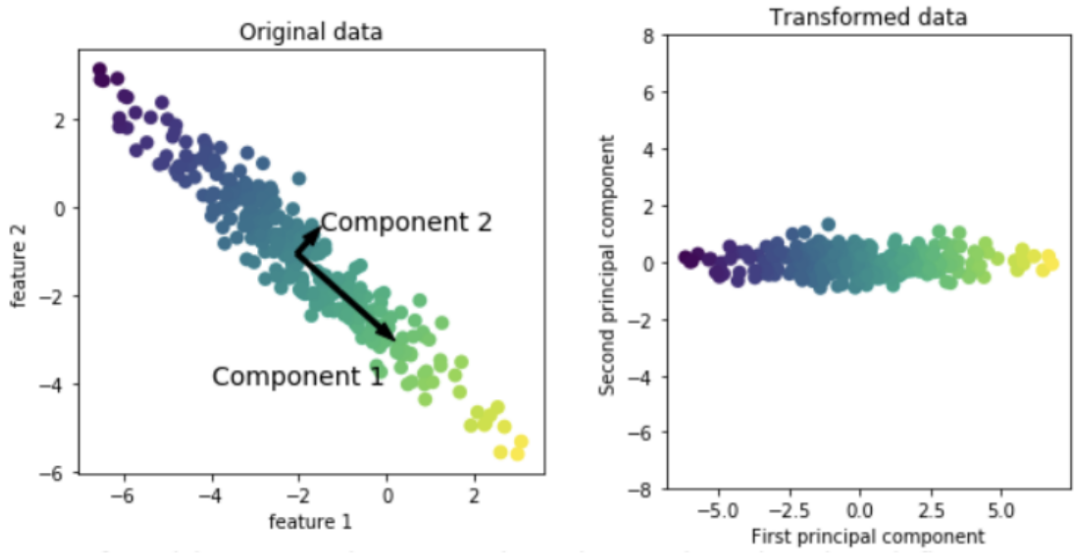


Figure 1: The above illustration shows a simple example on a synthetic two-dimensional data set. The first drawing shows the original data points colored to distinguish points. The algorithm first proceeds by finding the direction of the maximum variance labeled "Component 1". This refers to the direction in which most of the data is associated, or in other words, the properties that are most related to each other.

Pros of PCA for face recognition:

1. PCA is a well-established and widely used technique for dimensionality reduction.
2. PCA can significantly reduce the computational complexity of face recognition tasks.
3. PCA can be used to identify the most important features of a face image and eliminate noise and irrelevant information.

Cons of PCA for face recognition:

1. PCA assumes that the data is linearly correlated and may not work well for non-linear data.

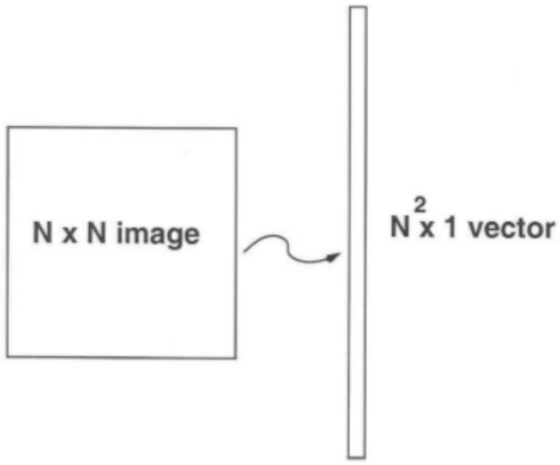
2. PCA may not be optimal for face recognition tasks with large datasets, as it can be computationally expensive to compute the eigenvectors and eigenvalues of a large covariance matrix.

3. PCA may not be robust to variations in lighting, pose, and expression, which can affect the quality of the principal components.

The algorithm of the PCA for face recognition:

1. Preparation of the dataset(shape of each image: $M \times N$)(Training data set $[X_1, X_2, X_3, \ldots . X_m]$)

2. Creating the matrix, where each row will be the flattened image into a $M \times N$-dim vector(long $M \times N$ list of grayscale pixel intensities)



   $x_1, x_2, x_3 \ldots$ which size $1 \times MN$

3. Compute the mean μ_i of each column in the matrix, giving us the average pixel intensity value for every (x, y)-coordinate in the image dataset.

$$u[j] = \frac{1}{n} \sum_{i=1}^{n} X[i, j]$$

4. Subtract the μ_i from each column c_i — this is called mean centering the data and is a required step when performing PCA.

$$a_i = c_i - \mu_i$$

$$A = [a_1, a_2, a_3, ..., a_m]$$

5. Now that our matrix M has been mean centered, compute the covariance matrix. The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

$$C = A^\top A$$

The covariance matrix is a $p \times p$ symmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 data matrix of this from:

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Figure 2: Covariance Matrix for 3-Dimensional Data

Since the covariance of a variable with itself is its variance ($Cov(a,a) = Var(a)$), in the main diagonal (Top left to bottom right) we actually have the variances of each initial variable. And since the covariance is commutative ($Cov(a,b) = Cov(b,a)$),

the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal.We can derive this also from multiplication of matrices($A^\top A$):

From definition of symmetric matrix:

$$A^\top = A$$

$$C = A^\top A$$

$$C^\top = (A^\top A)^\top = A^\top (A^\top)^\top = A^\top A = C$$

6. Perform an eigenvalue decomposition on the covariance matrix to get the eigenvalues λ_i and eigenvectors X_i.

   PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on. As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set. As we know covariance matrix is a product of $A^\top A$, which is always a symmetric matrix, here conclude that there exists such a P whose columns are eigenvectors such that they are mutually orthonormal. With matrix P we can get basis after projection onto which, we will get a matrix with diagonal values equal to the eigenvalues. In fact covariance matrix with smallest variance.

$$P^\top Q P = \lambda$$

7. Sort X_i by λ_i, largest to smallest and take the top N eigenvectors with the largest corresponding eigenvalue magnitude.

   By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance. Because the eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance (most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.

8. Transform the input data by projecting (i.e., taking the dot product) it onto the space created by the top L eigenvectors — these eigenvectors are called our eigenfaces.

$$T_{n\times L} = [X - \mu]_{n\times MN} \cdot P_{MN\times L}$$

Testing algorithm:

We've got an unknown face that needs to be detected. We subtract it from the average face, project the normalized vector into eigenspace to get the linear combination of eigenfaces, from this projection we generate the vector of the coefficient. This earlier generated vector we subtract from the training image to get the minimum distance between testing and training vectors. And these actions will allow us to predict the label for the image (recognize the face on the image).

$distance(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + .. + (x_n - y_n)^2}$

5 Implementation

1. Data.

   According to the decision taken at the beginning, we decided to make a test format, which will be a more visual model, considering only 40 people, each of which will contain 10 photos. To implement the eigenfaces algorithm we have separated ATT's database of faces into 70%-part on which we have trained the algorithm and 30%-part to test its work and see the accuracy of it.

2. Implementation of the algorithm.

   The implementation will be based on the theoretical steps, which are indicated in the part with the analysis of the algorithm above. Therefore, the implementation will consist of the following stages:

   - read_files() - a function for reading data from the dataset and forming the data structure accordingly for further use of this data in the algorithm.

- matrix_of_data() - a function for building a matrix from images obtained from a dataset. Where we display each image as a row of matrix with the corresponding representation ($K \times N$ image $\implies$ KN$\times$1 vector).
- the next stage is the calculation of the average face using the obtained matrix. Where the mean vector contains the mean value for each preformed variable.
- pca_compute() - function for performing a full-fledged PSA algorithm.
  Firstly, we subtract the mean face From all the images in our database. After that, we get eigenvalues and eigenfaces of the covariance matrix. The dimensions of each vector are the same as the dimensions of our dataset's original images, the same number of components. The eigenvectors of our covariance matrix, therefore, are what we need to know as the essential eigenface. The vectors in which our pictures are different from the average face of your dataset can be seen as direction vectors.
- The last and main stage of our algorithm is face recognition. We're creating an eigenface for the image uploaded, which we need to recognize. Then we'll calculate the Euclidean distance between these eigenfaces and the ones we've already calculated for the dataset for the matching image, which would be the smallest. Based on the obtained result, we return our prediction.

6 Conclusions

The document describes the face recognition process and the functionality of our proposed solution. Possible solutions were analyzed and appropriate conclusions were drawn based on this. Here, we have compared different algorithms for face recognition based on the provided dataset for training and images for recognition. There is an analysis of the algorithm, its advantages and disadvantages, and a fully implemented algorithm for recognition on the dataset we have chosen. As a result, we received an application for face recognition based on a dataset consisting of photos of 40 people. You can use our

application: https://github.com/smthcreative/Face-Recognition/tree/main

References

[1] International Journal of Artificial Intelligence Applications (IJAIA), Vol.3, No.2, March 2012 DOI : 10.5121/ijaia.2012.3203 23 OPTIMIZING FACE RECOGNITION USING PCA. Manal Abdullah , Majda Wazzan and Sahar Bo-saeed

[2] Jain, A.K. and Li, S.Z., 2011. *Handbook of face recognition*. New York: springer.

[3] A Step-by-Step Explanation of Principal Component Analysis (PCA) Written byZakaria Jaadi https://builtin.com/data-science/step-step-explanation-principal-component-analysis