Introduction

Let \( A \) be an \( n \times n \) real matrix and let \( \lambda \in \mathbb{R} \). Vector \( x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \) is called eigenvector of a matrix \( A \), if it satisfies the following condition:

\[
Ax = \lambda x
\]

Number \( \lambda \) is the corresponding eigenvalue of the eigenvector \( x \).

The eigenface method

Facial images are characterized by a narrow range of their variation. Facial recognition systems based on the assumption that each face has a specific structure, meaning that faces own characteristic features. The characteristic features are called eigenface because they are the eigenvectors (principal components) of the set of faces. We can extract them from the original face image using mathematical tools called Principal Component Analysis (PCA).

Using PCA technique we can transform any original face image from the training set into a corresponding eigenface. Recognition occurs by projecting a new unknown face image into the subspace spanned by the eigenfaces. This subspace is called “face space.” Then we can classify the face by comparing its position in face space with the faces position of the training set.

We assume that any face image \( f(x,y) \) consists of \( N \) pixels. So we can present any image as an array of \( N \times N \). We may also consider that the face image is a vector (or point) of dimension \( N^2 \).

\[
\begin{align*}
N \times N & \text{ image} \\
& \rightarrow \quad N^2 \times 1 \text{ vector}
\end{align*}
\]

Calculate of eigenfaces using PCA

1. Prepare the data

Prepare the training set of face images \( \Gamma_1, \Gamma_2, \ldots, \Gamma_M \) for processing. The face images must be centered, in grayscale and of the same size. Training set should include a number of images for each person, with some variation in expression and in the lighting. An example training set is shown in Figure 1.

2. Calculate the average face

The average face of the training set is defined by:

\[
\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_i
\]

Each face differs from the average by the vector \( \Phi_i = \Gamma_i - \Psi \), \( i = 1, \ldots, M \).

3. Calculate the covariance matrix

The covariance matrix \( \Sigma \) is calculated according to:

\[
\Sigma = \frac{1}{M-1} \sum_{i=1}^{M} (\Phi_i \Phi_i^T)
\]

Such a covariance matrix has dimension \( N^2 \times N^2 \), so we would have \( N \) eigenfaces and corresponding eigenvalues.

4. Calculate the eigenvectors and eigenvalues of the covariance matrix

Consider the matrix

\[
L = A^T A (M \times M \text{ matrix}),
\]

where \( A_{mn} = \Phi_m^T \Phi_n \). Compute the eigenvectors \( u_i \) of matrix \( L \):

\[
A^T A u_i = \lambda_i u_i
\]

What is the relationship between the eigenvalues \( \lambda_i \) and the eigenvectors \( u_i \)?

\[
A^T A v_i = \lambda_i v_i \Rightarrow A^T v_i = \lambda_i u_i \Rightarrow A v_i = \lambda_i A u_i \text{ or } C v_i = \lambda_i u_i,
\]

where \( u_i = A v_i \).

Thus, matrices \( C \) and \( L \) have the same eigenvalues and their eigenvectors are related as follows: \( u_i = A v_i \).

Vectors \( v_i \) determine linear combinations of the \( M \) training set face images to form the eigenfaces \( u_i \):

\[
\Omega_i = \sum_{j=1}^{M} v_{ij} \Gamma_j, \quad i = 1, \ldots, M.
\]

We normalize the eigenfaces \( u_i \), such that \( ||u_i|| = 1 \).

5. Select the principal components

From \( M \) eigenfaces \( u_i \) we choose only \( M' \), which have the largest eigenvalues. These eigenfaces span an \( M' \)-dimensional subspace (face space) of the original \( N \) image space.

Each original face image (minus the average) \( \Gamma_i' \) of the training set can be represented as a linear combination of \( M \) eigenfaces.

\[
\Gamma_i' = \Omega_i = \sum_{j=1}^{M} \Omega_{ij} u_j, \quad \Omega_i = [\Omega_{i1}, \Omega_{i2}, \ldots, \Omega_{iM}] \quad \text{is a weight vector of the image} \ \Gamma_i \ \text{from the training set}.
\]

We can also use only a part of the eigenfaces. Then we get an approximation of the original face image.

\[
\Gamma_i' = \Psi + \sum_{j=1}^{k} \Omega_{ij} u_j.
\]

Classification of face images

1. Transform a new (unknown) face image into its eigenface components

A new face image \( \Gamma_{\text{new}} \) is transformed into its eigenface components by a simple operation

\[
\Phi = \Gamma_{\text{new}} - \Psi
\]

It is exactly the projection the face image into the face space spanned by eigenfaces.

Figure 4 shows example of such projection.

The resulting weights form the weight vector \( \Omega_{\text{new}} = [\omega_{1}, \omega_{2}, \ldots, \omega_{M}] \) that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images.

2. Compare the new face image with other faces

Then the vector \( \Omega_{\text{new}} \) is used to establish which of the predefined face classes best describes the new face. The simplest way to determine which face classifier provides the best description of the new face image is to find the face class \( k \) that minimizes the Euclidean distance:

\[
e_k = ||\Omega_{\text{new}} - \Omega_{k}\|^2
\]

where \( \Omega_k \) is a vector describing the \( k \)th face class.

The face classes \( \Omega_k \) are computed by averaging the result of the eigenface representation over a small number of training images for each person. A new face is classified as belonging to a class \( k \) when the minimum \( e_k \) (i.e. the maximum matching score) is below a certain threshold value \( \theta_k \). Otherwise, we can assume that the unknown face image \( \Gamma_{\text{new}} \) is not a face.

Conclusion

The eigenface approach does provide a practical solution to the problem of face recognition. It is fast, reasonable simple and accurate. It works in constrained environments such as an office or household. It can be also implemented using modules of connectionist or natural networks, as described by Turk and Pentland (see [2] and [3]).

The eigenface method has many practical applications e.g. in security systems, access control, image and film processing, criminal identification and human-computer interaction.