# THE EIGENFACE METHOD

## Introduction

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

$$\boldsymbol{A}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

Number $\lambda$ is the corresponding eigenvalue of the eigenvector $\boldsymbol{x}$.

Most important eigenvectors properties:

- There are $n$ eigenvectors (and corresponding eigenvalues) in a $n \times n$ matrix.
- Eigenvectors corresponding to different eigenvalues are orthogonal.

Face recognition systems are based on the assumption that each person has a specific face structure, meaning any faces possess characteristic features. These characteristic features are called *eigenfaces* because they are the eigenvectors (principal components) of the set of faces. We can extract them from the original face image using mathematical tool 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 $I(x, y)$ 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 $N^2$.

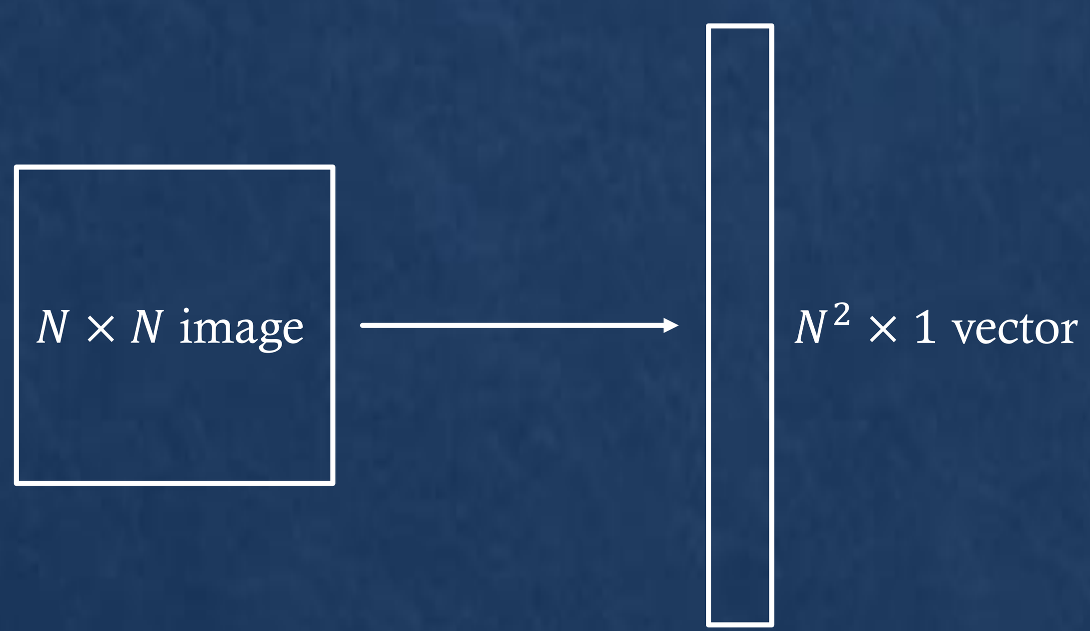


Fig. 3. Seven of the eigenfaces calculated from the trening set of Figure 1 by Turk and Pentland in [2].

## Calculate of eigenfaces using PCA

1. **Prepare the data**
   Prepare the training set of face images $\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2 \dots, \boldsymbol{\Gamma}_M$ 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 averege face**
   The average face of the training set is defined by:
   $$\boldsymbol{\Psi} = \frac{1}{M}\sum_{n=1}^{M} \boldsymbol{\Gamma}_n.$$
   Each face differs from the average by the vector $\boldsymbol{\Phi}_i = \boldsymbol{\Gamma}_i - \boldsymbol{\Psi}$, $i = 1, \dots, M$.

3. **Calculate the covariance matrix**
   The covariance matrix $\boldsymbol{C}$ is calculated according to:
   $$\boldsymbol{C} = \frac{1}{M}\sum_{n=1}^{M} \boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^T = \boldsymbol{A}\boldsymbol{A}^T,$$
   where $\boldsymbol{A} = [\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \dots, \boldsymbol{\Phi}_M]$ is $N^2 \times M$ matrix.

   Such a covariance matrix has dimension $N^2 \times N^2$, so we would have $N^2$ eigenfaces and corresponding eigenvalues.
   For a typical image of size $256 \times 256$ we would have to calculate a $65{,}536 \times 65{,}536$ matrix and 65,536 eigenvectors. This is a very large amount of data, which is not efficient in the computational. So we have to use some trick to compute the eigenfaces.

4. **Calculate the eigenvectors and eigenvalues of the covariance matrix**
   Consider the matrix
   $\boldsymbol{L} = \boldsymbol{A}^T\boldsymbol{A}$ ($M \times M$ matrix),
   where $\boldsymbol{L}_{mn} = \boldsymbol{\Phi}_m^T \boldsymbol{\Phi}_n$.
   Compute the eigenvectors $\boldsymbol{v}_i$ of matrix $\mathbf{L}$.
   $$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{\mu}_i \boldsymbol{v}_i.$$

   What is the relationship between the eigenvectors $\boldsymbol{u}_i$ of the matrix $\boldsymbol{C}$ and the eigenvectors $\boldsymbol{v}_i$?
   $$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{\mu}_i \boldsymbol{v}_i \Rightarrow \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{\mu}_i \boldsymbol{A}\boldsymbol{v}_i \Rightarrow$$
   $$\boldsymbol{C}\boldsymbol{A}\boldsymbol{v}_i = \boldsymbol{\mu}_i \boldsymbol{A}\boldsymbol{v}_i \text{ or } \boldsymbol{C}\boldsymbol{u}_i = \boldsymbol{\mu}_i \boldsymbol{u}_i,$$
   where $\boldsymbol{u}_i = \boldsymbol{A}\boldsymbol{v}_i$.
   Thus, matrices $\boldsymbol{C}$ and $\boldsymbol{L}$ have the same eigenvalues and their eigenvectors are related as follows: $\boldsymbol{u}_i = \boldsymbol{A}\boldsymbol{v}_i$.

   Vectors $\boldsymbol{v}_i$ determine linear combinations of the $M$ training set face image to form the eigenfaces $\boldsymbol{u}_l$
   $$\boldsymbol{u}_l = \sum_{k=1}^{M} \boldsymbol{v}_{ik} \boldsymbol{\Phi}_k, \ l = 1, \dots, M.$$
   We normalize the eigenfaces $\boldsymbol{u}_l$, such that $\|\boldsymbol{u}_l\| = 1$.

5. **Select the principal components**
   From $M$ eigenfaces $\boldsymbol{u}_l$ we choose only $M'$, which have the largest eigenvalues. These eigenefaces span an $M'$-dimensional subspace (face space) of the original $N^2$ image space.

   Each original face image (minus the average) $\boldsymbol{\Gamma}_i'$ of the training set can be represented as a linear combination of $M$ eigenfaces.
   $$\boldsymbol{\Gamma}_i' - \boldsymbol{\Psi} = \sum_{j=1}^{M} \omega_j^i \boldsymbol{u}_j,$$
   where $\omega_j^i = \boldsymbol{u}_j^T \boldsymbol{\Phi}_i$.
   $\boldsymbol{\Omega}_i^T = [\omega_1^i, \omega_2^i, \dots, \omega_M^i]$ is a weight vector of the image $\boldsymbol{\Gamma}_\mathrm{i}$ from the training set.

   We can also use only a part of the eigenfaces. Then we get an approximation of the original face image.
   $$\boldsymbol{\Gamma}_i' - \boldsymbol{\Psi} \approx \sum_{j=1}^{M'} \omega_j^i \boldsymbol{u}_j.$$

Fig. 1. Face images used as the training set by Turk and Pentland in [2].

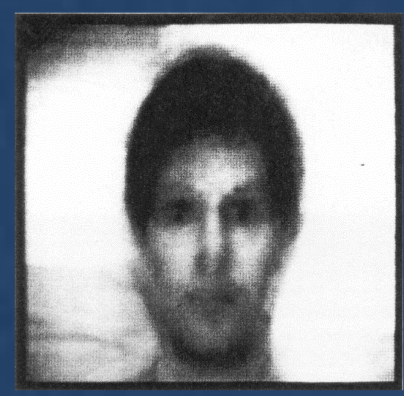

Fig. 2. The average face $\boldsymbol{\Psi}$ of the training set from Figure 1 calculated by Turk and Pentland in [2].

## Classification of face images

1. **Transform a new (unknown) face image into its eigenface components**
   A new face image $\boldsymbol{\Gamma}_\mathrm{new}$ is transformed into its eigenface components by a simple operation
   $$\omega_k = \boldsymbol{u}_k^T \boldsymbol{\Phi}, \ k = 1, \dots, M',$$
   where $\boldsymbol{\Phi} = \boldsymbol{\Gamma}_\mathrm{new} - \boldsymbol{\Psi}$.

   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
   $$\boldsymbol{\Omega}_\mathrm{new}^T = [\omega_1, \omega_2, \dots, \omega_{M'}],$$
   that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images.

2. **Comapare the new face image with other faces**
   Then the vector $\boldsymbol{\Omega}_\mathrm{new}$ is used to establish which of the predefined face classes best describes the new face. The simplest way to determine which face class provides the best description of the new face image is to find the face class $k$ that minimizes the Euclidian distance:
   $$\epsilon_k = \sqrt{\|\boldsymbol{\Omega}_\mathrm{new} - \boldsymbol{\Omega}_k\|^2},$$
   where $\boldsymbol{\Omega}_k$ is a vector describing the $k$th face class. The face classes $\boldsymbol{\Omega}_i$ are computed by averaging the result of the eigenface representation over a small number of face images of each person. A new face is classified as belonging to a some class $k$ when the minimum $\epsilon_k$ (i.e. the maximum matching score) is below a certain threshold value $\theta_\epsilon$. Otherwise, we can assume that the unknown face image $\boldsymbol{\Gamma}_\mathrm{new}$ is not a face.

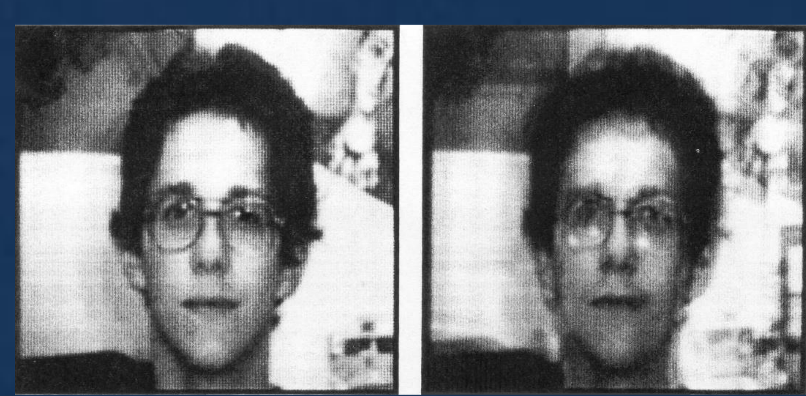

Fig. 4. The face image from the training set and its projection into the face space spanned by the eigenfaces of Figure 3.

## 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 also be implemented using modules of connectionist or natural networks, as described by Turk and Pentland (see [2] and [3]).
The eigenface method have many practical applications e.g. in security systems, access control, image and film processing, criminal identification and human-computer interaction.


### References

[1] Sirovich L., Kirby M., Low-dimensional procedure for the characterization of human faces, Journal of the Optical Society of America A., 1987, 519–524.

[2] Turk M., Pentland A., Eigenfaces for recognition, Journal of Cognitive Neuroscience, 1991, 71–86.

[3] Turk M., Pentland A., Face recognition using eigenfaces, Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1991, 586–591.

[4] Pissarenko D., Eigenface-based facial recognition, 2002.



Ministerstwo Nauki i Szkolnictwa Wyższego

Organizacja IX Konferencji Modelowanie Matematyczne w Fizyce i Technice (MMFT 2017)
- zadanie finansowane w ramach umowy 829/P-DUN/2017 ze środków Ministra Nauki i Szkolnictwa Wyższego przeznaczonych na działalność upowszechniającą naukę.