Face Recognition via Principle Component Analysis

Author: Jingya Li, Yujia Chen. Instructor: Henry Kvinge

Department of Mathematics, Colorado State University

Objective

Use the singular value decomposition for a matrix and Euclidean distance to efficiently recognize human faces.

Introduction

Euclidean distance is an obvious way to try and distinguish between different human faces. Calculating the Euclidean distance between two length kn vectors (corresponding to two $k \times n$ images), requires $(kn)^2$ subtractions and multiplications respectively. Using Matlab, the image:

can be treated a 192×168 matrix:

sample_image 192x168 uint8

Thus calculating the Euclidean distance for any two images of these dimensions requires $192 \times 168 = 32256$ multiplications and subtractions. Therefore, if the important features of a large collection of images can be summarized by fewer numbers (via dimensionality reduction), the computational cost can be reduced and the recognition speed can be improved. This can be achieved by PCA (Principle Component Analysis).

Mathematics of PCA

Suppose we have M vectors of size N representing a set of sampled images and p_j represents a pixel value:

$$x_i = [p_1, p_2, ..., p_{j=N}]^T, \quad \text{for } i = 1, 2, ..., M.$$

Let m be the pixelwise mean of the images and let w_i be defined as the mean centered image corresponding to x_i:

$$w_i = x_i - m.$$

Find a set of vectors $\{e_i\}$ (via PCA) which best capture the variance of the sample images. For example

$$e_1 := \arg\max_{||e||=1}\Big(\sum_i^M \langle w_i, e\rangle^2\Big).$$

For known individual with face w_i, we calculate the weight vector Ω_i,

$$\Omega_i := [e_1^T w_i, e_2^T w_i, \ldots, e_K^T w_i].$$

Because of the way we created $e_1, e_2, \ldots$, we can capture most information about the face even with $K << N$.

Main Result

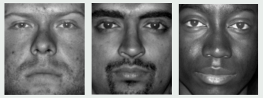

Figure: Faces from training set.

Figure: The average face after mean centering.

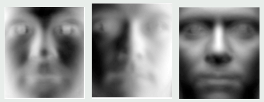

Figure: The corresponded eigenfaces.

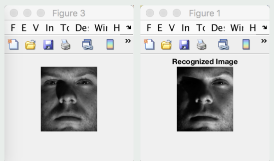

Figure: Test face vs. recognized face.

Img_Mat 32256x60 double | project_sample 59x60 double

Figure: The reduction in computation.

As we explained in the Introduction, by using PCA we were able to reduce the computation from 32256×60 to 59×60.

Given a new unknown input image w, we project w to face space to get

$$\Omega := [e_1^T w, e_2^T w, \ldots, e_K^T w].$$

Then we find the face closest to Ω by computing

$$\epsilon_i = \|\Omega - \Omega_i\| \qquad \text{for all known faces}$$

and the ϵ_i is the Euclidean distance between the unknown face and the known face.

The Algorithm

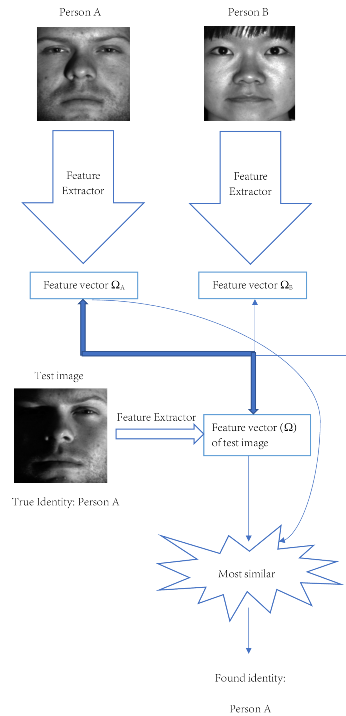

Figure: The flow chart.

i. The feature extractor reduces the dimension of the training faces using PCA.
ii. The test image is also reduced and then we compare it to known faces Ω_i in the face space and find the closest known face in terms of Euclidean distance.

References

- The Yale Face Database B, `http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html`
- Turk, Matthew A., and Alex P. Pentland. "Face recognition using eigenfaces." Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. IEEE, 1991.