Face Recognition via Principle Component Analysis

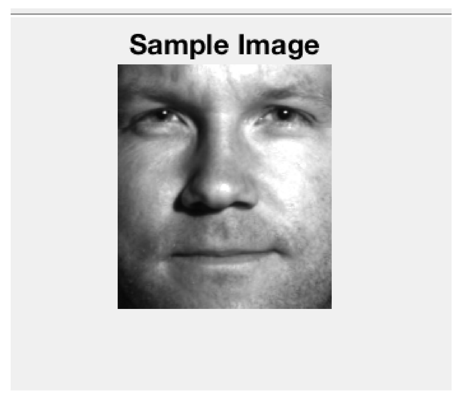
Author: Jingya Li, Yujia Chen Instructor: Henry Kvinge

Objective

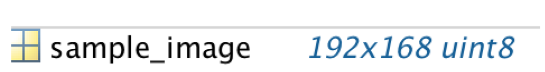
Use the singular value decomposition for a matrix to efficiently recognize human faces.

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:



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 just a few images, the computation 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. p_j's represent the pixel values.

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

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 $\{e_i\}$ which best capture the variance of $w_1, \ldots, w_M$. Specifically set

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

While in general

$$e_k =:= \arg\max_{||e||=1} \Big(\sum_i^M \langle \hat{w}_i, e \rangle^2 \Big)$$

where

$$\hat{w}_i = w_i - \sum_{i=1}^{k-1} w_i e_i e_i^T.$$

Main Result

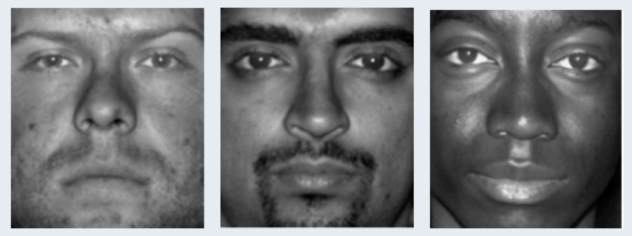


Figure 1:Faces from training set.

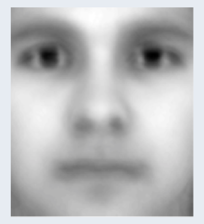


Figure 2:The average face after mean centering.

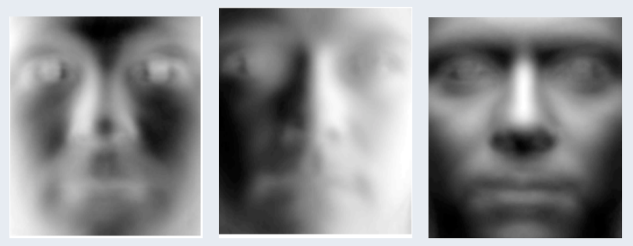


Figure 3:The corresponded eigenfaces.

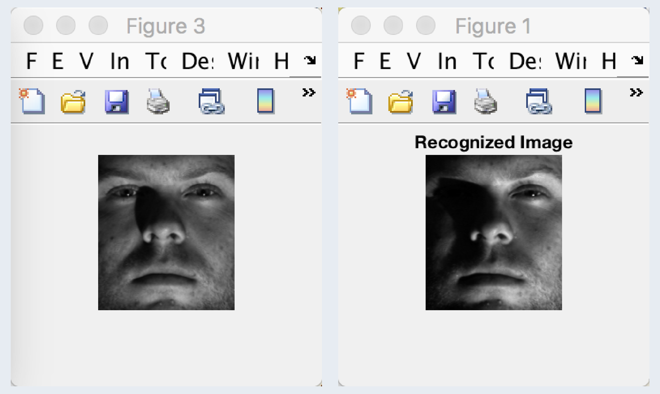


Figure 4:Test face vs. recognized face.

Figure 5:The reduction in computation.

As we explained in the Introduction, by using PCA we were able to summarize the discriminatory features of a large collection of images of an individual in just a few images.

In other words, we repeatedly extract the direction which captures the highest variance in the data. This both acts as a tool for dimensionality reduction and also feature extraction since it efficiently captures variation in the data in a very compact form.

Given a new image v_i (which we treat as a vector) we can calculate the angle between v_i and the subspace spanned by $e_1, e_2, \ldots, e_\ell$ for small ℓ. The size of this angle should efficiently indicate the similarity of the new image to the set $x_1, \ldots, x_M$.

The Algorithm

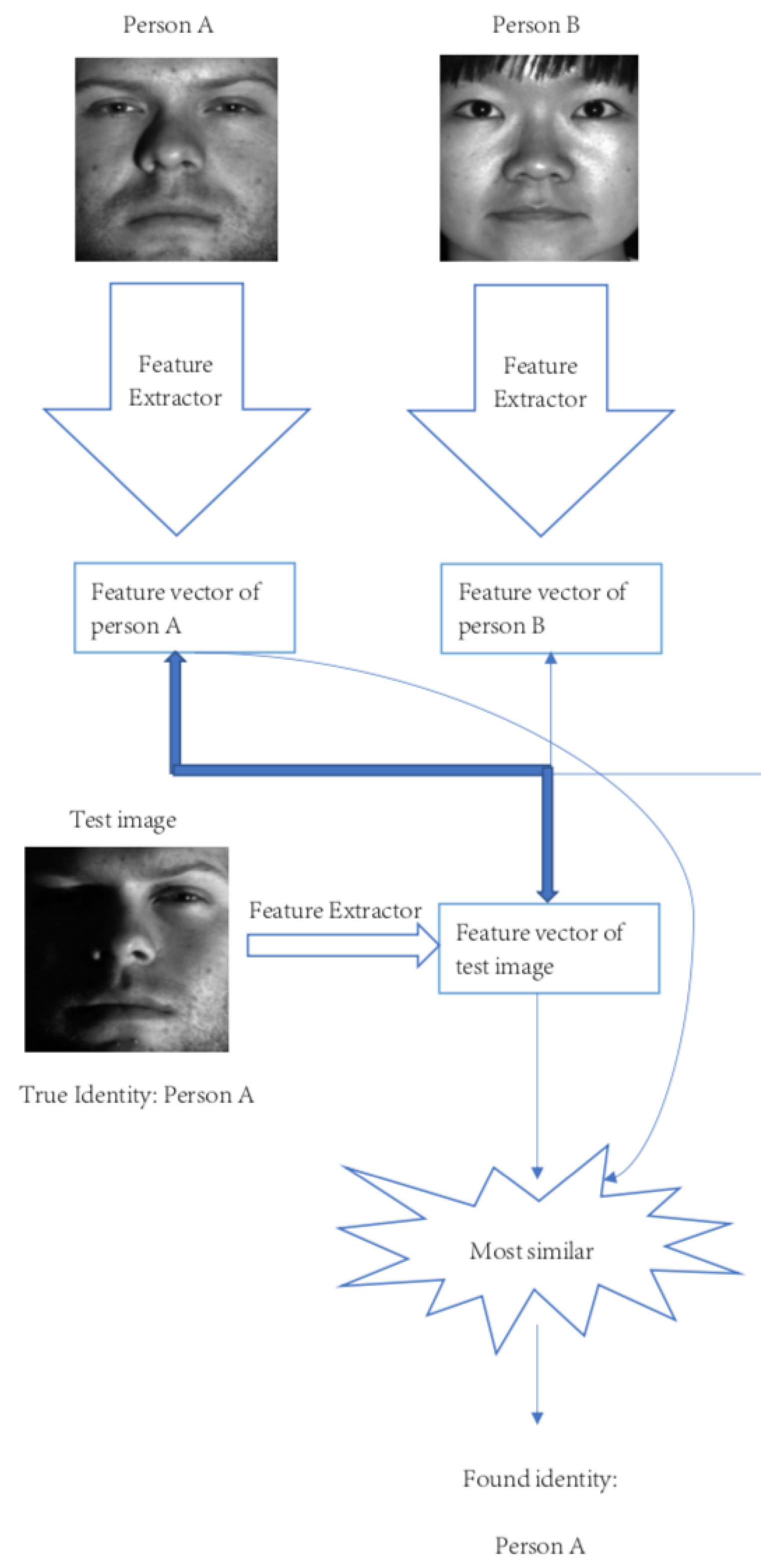


Figure 6:The flow chart.

i. The feature extractor calculates the eigenfaces thus converting discriminatory information for each persons face to a small set of column vectors.

ii. The test image is also converted to a vector. By comparing with a subset of the eigenface vectors for each person, we can predict which individual the test vector image corresponds to.

References

- The Yale Face Database B, `http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html`