MVA – Object Recognition and Artificial Vision

Assignment 3

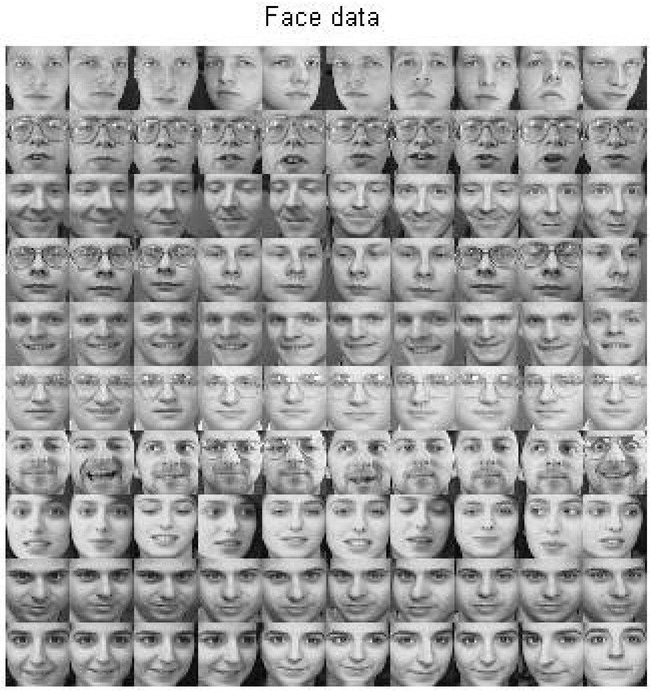
Eigenfaces for recognition

Jean-Baptiste Fiot

Nov 2008

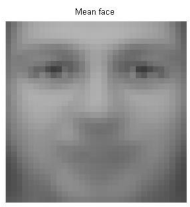
1 Algorithm outline

1.1 Subtraction of the mean picture.

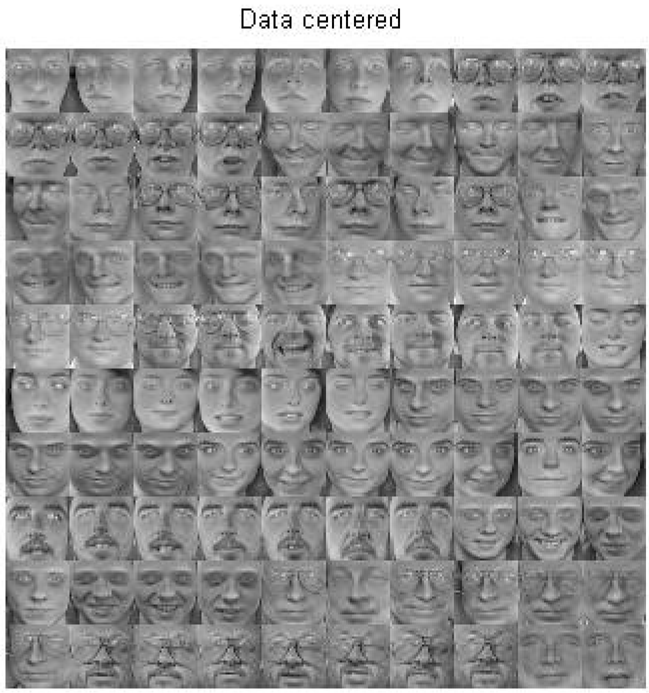


$-$

$$\boxed{I_{mean} = \frac{1}{N_{mean}} \sum_{I \in Training\,Set} I}$$



$=$



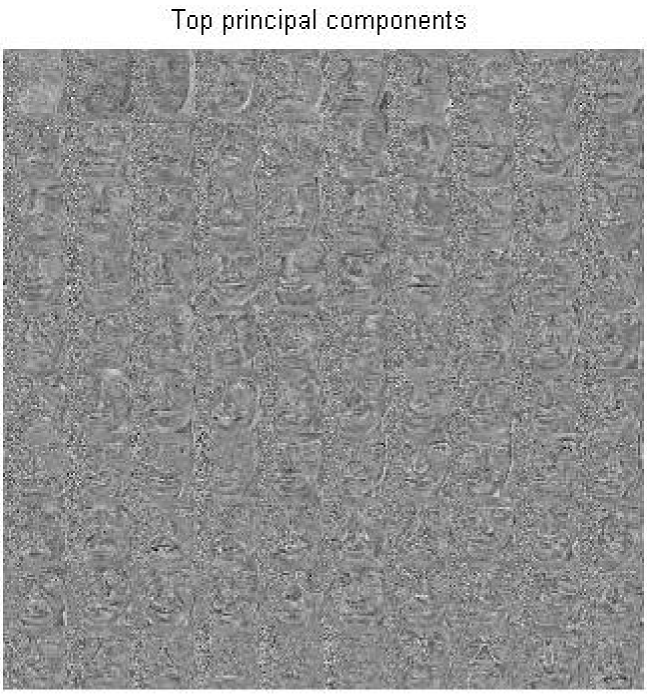
1.2 Extraction of the principal components.

Notations:

- Training pictures: $I_{1,} \ldots, I_M$ (M=280 here).
- Each picture I_i (N*N=32*32 here) is represented by a vector Φ_i with N² elements.
- $\boxed{A=[\Phi_1 \ldots \Phi_M]}$ (N² * M matrix)

Using the princomp.m function, we extract the **principal components of the training pictures**, namely N² vectors with N² elements each. The principal components are the **eigenvectors of** AA^T.

Here are some of them:



To have a representation with a lower dimension, we keep only **K (normalized) eigenvectors** $\{u_{1,} \ldots, u_K\}$, corresponding to the **K largest eigenvalues.**

We want to **project** the training pictures on $vect(u_{1,} \ldots, u_K)$:

$$\boxed{\Phi_i = \sum_1^K \alpha_{i,j} u_j \quad \forall i \in [\![1, M]\!]}$$

We get the coefficients with:

$$\boxed{\alpha_{i,j} = \langle \Phi_i | u_j \rangle = \Phi_i^T u_j \quad \forall (i, j) \in [\![1, M]\!] \times [\![1, K]\!]}$$

1.3 Recognition of an unknown face.

- We **subtract the mean image of the training set:**

$$\boxed{I_{unknown,centered} = I_{unknown} - I_{mean}}$$

- We **project** the corresponding Γ vector on $vect(u_{1,} \ldots, u_K)$:

$$\boxed{\beta_j = \langle \Gamma | u_j \rangle = \Gamma^T u_j \quad \forall\, j \in [\![1, K]\!] \Leftrightarrow \Gamma = \sum_1^K \beta_j u_j}$$

- We find the image whose low-dimension representation has the **minimum Euclidian distance** with the unknown picture's low-dimension representation:

$$\boxed{I_i\, recognized\;\; face \Leftrightarrow d(\beta, u_i) = min\{d(\beta, u_i), i \in [\![1, M]\!]\}}$$

  I implemented EuclDistClassifier.m for this part.

2 Interest of this method

A direct way to solve this problem would be to find the image which has the maximum NCC with the unknown picture.

Despite its relative loss of precision, the interest of the low-dimensional projection method relies on its **computing efficiency**. I implemented NNclassifier.m to compare both methods' performances.

With computations for all K in $[\![1{,}50]\!]$, the “direct” method takes 380 seconds, and the low-dimensional projection method takes only 6 seconds. This means a **63 folds decrease**.

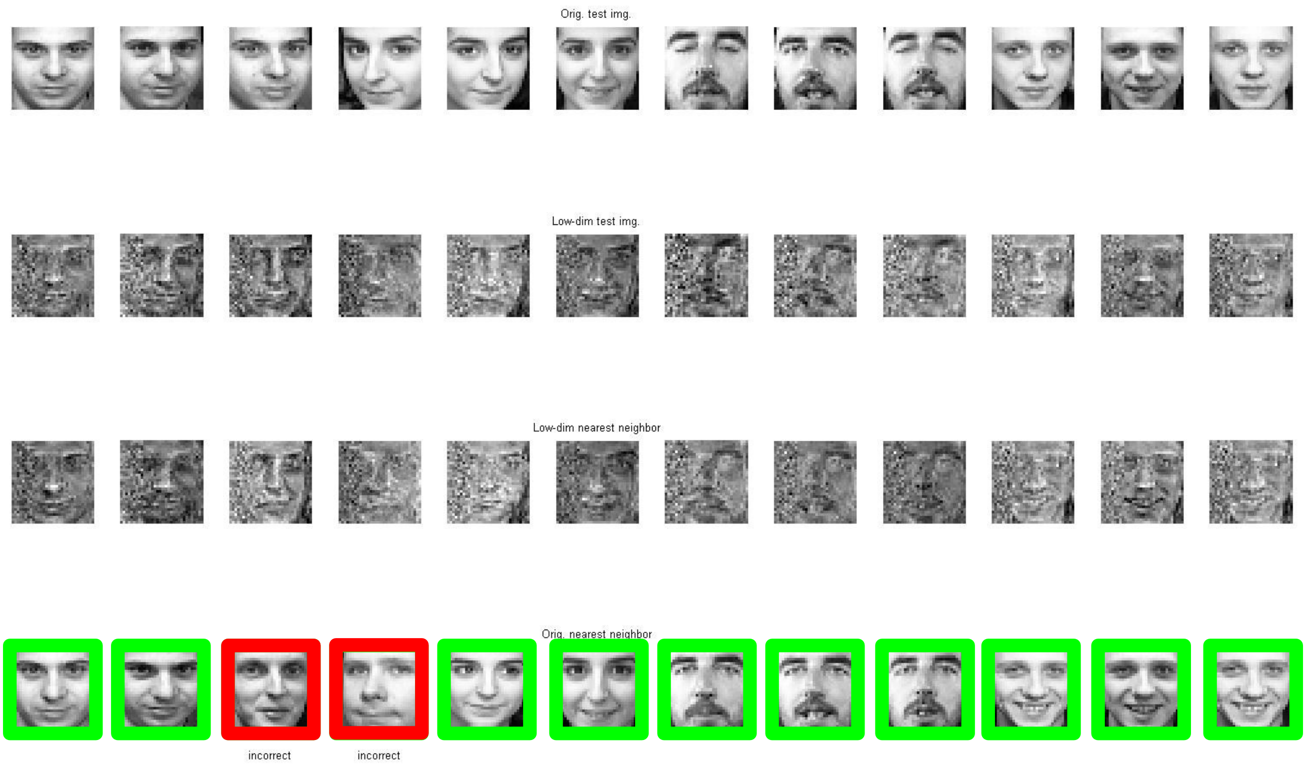
3 Results for K=100

For each group of 4 lines, we have:

– original test images
– low-dim test images
– low-dim nearest neighbors
– original nearest neighbor

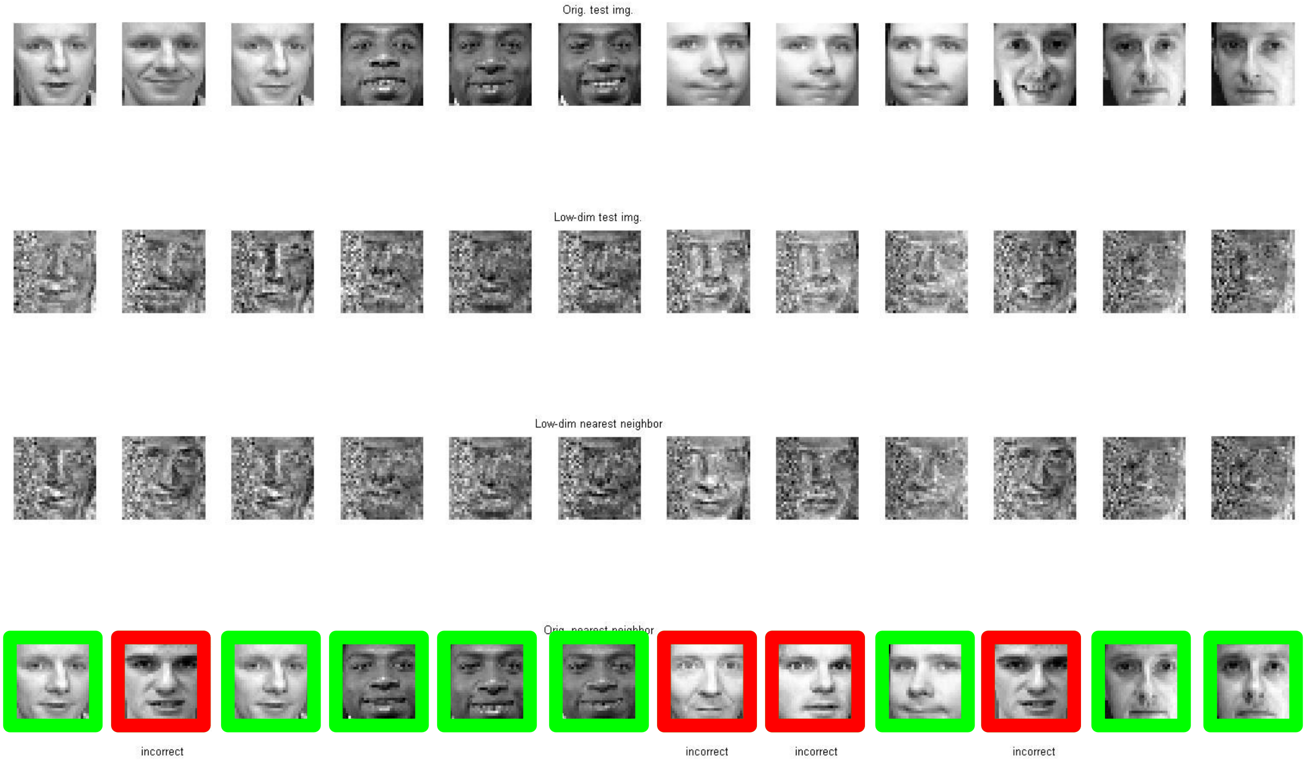
The green faces are the correct recognitions, the red the incorrect ones.


Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor
incorrect

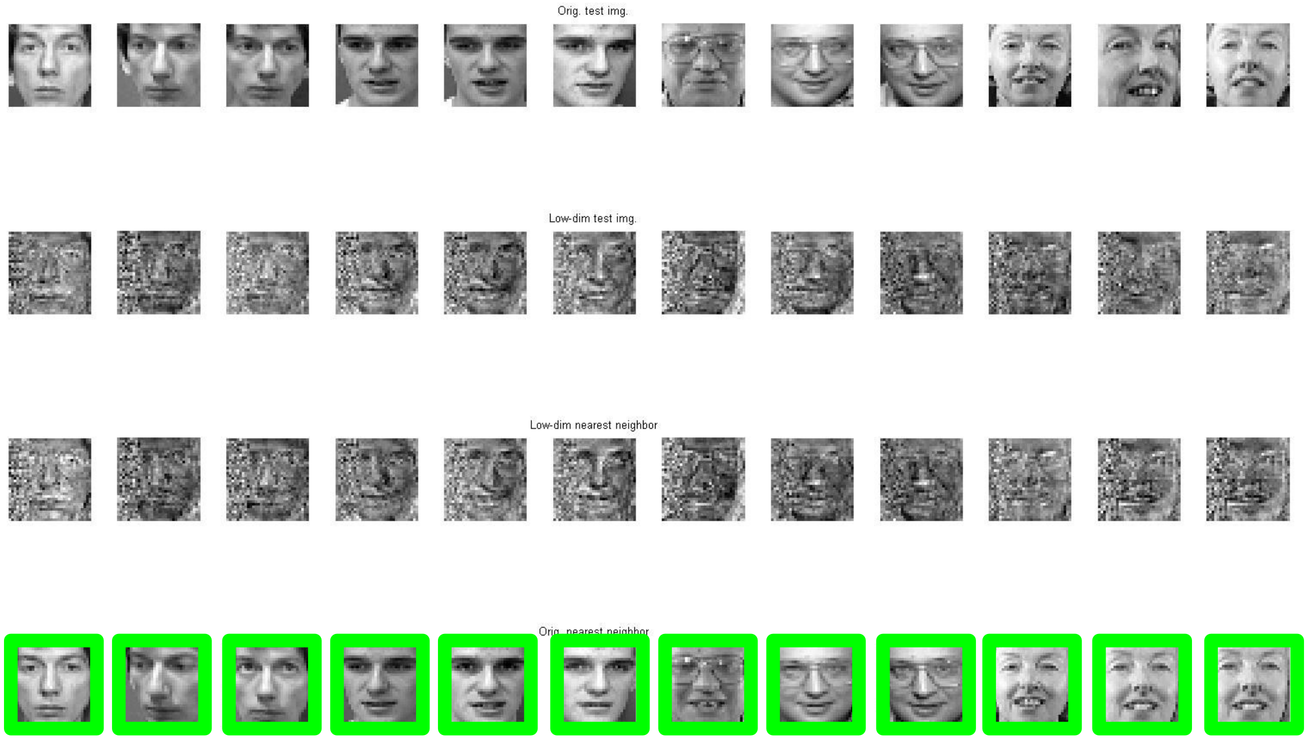

Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor
incorrect
incorrect

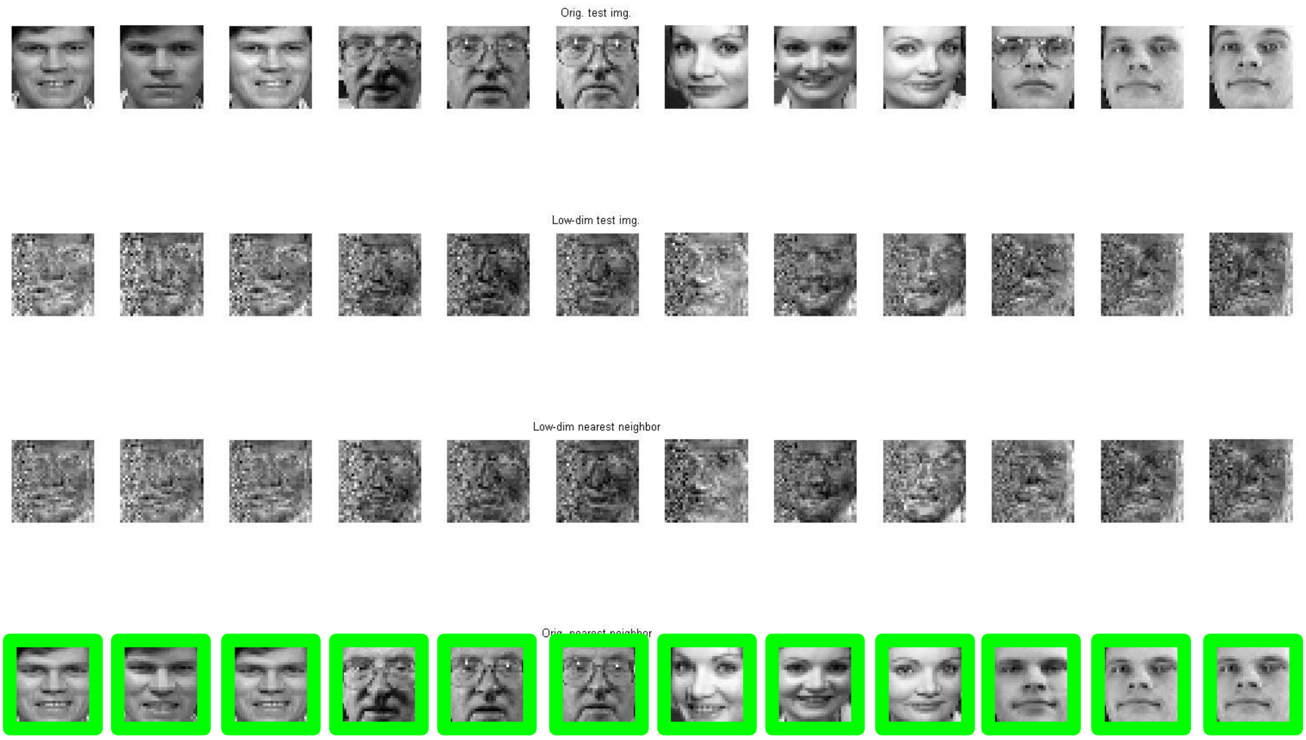

Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor
incorrect


Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor
incorrect
incorrect
incorrect


Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor
incorrect
incorrect
incorrect
incorrect


Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor
incorrect


Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor


Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor

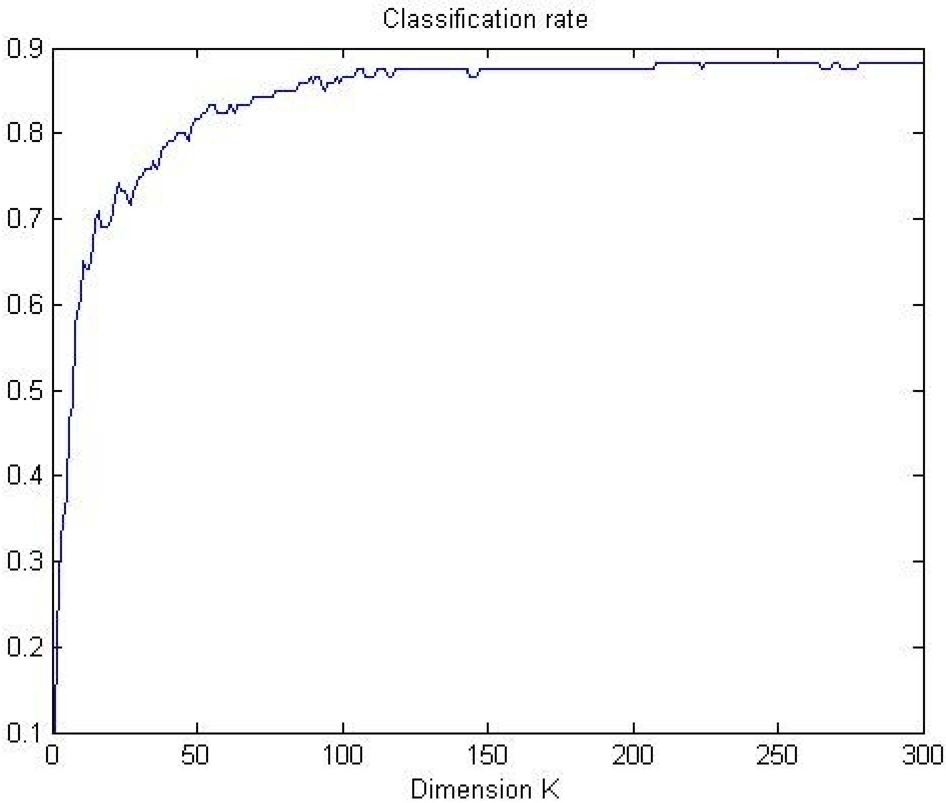

Orig. test img.
Low-dim test img.
Low-dim nearest neighbor
Orig. nearest neighbor
incorrect
incorrect
incorrect

4 Classification rate

For a given K, the classification is given by:

$$R_{classification} = \frac{Card\,[\,I \in Test\ Set\,,\ Estimated\ Label\,(I) = Test\ Label\,(I)\,]}{N_{test}}$$

Here is classification rate's evolution depending on the dimension K:



(K could go up to N² (=1024 here), but the classification rate is not displayed for values higher than 300.)

Conclusion:

For low K, the performance is extremely bad, but rises very quickly, and tends to stagnate just below 90%.