

# Exercices (4<sup>th</sup> Day)

(Eigenfaces)

**Exercise 1:** Given a training sample set of face images (Day5/images1/Training/) and a test set (Day5/images1/Test/Faces/ and Day5/images1/Test/noFaces/), implement a simple Eigenface-based face detector:

1. From the training set, compute the “mean image” and the first  $L$  Eigenfaces that arise from the Eigenvalue decomposition of the covariance matrix. Take care for an efficient implementation and visualize your results.
2. Classify the test set into faces and non-faces using the Eigenspace generated above. To this purpose, compute the reconstruction error that arises from back-projection of an image from the Eigenspace. Try to find an adequate classification threshold.
3. Experiment with the images in “Day5/images1/Test/reconstruct”. Calculate their projections into your Eigenspace and reconstruct the images from their projections. Visualize the reconstructed images. Can you explain what happened?

**useful functions:**

`prod`, `reshape`, `eigs`, `norm`

**Exercise 2: (optional)** Implement an Eigenface-based face identifier. Use the images from “Day5/images2/Training” for Eigenspace training. Try to identify the faces from “Day5/images2/TestFaces/known” and “Day5/images2/TestFaces/unknown”. For classification, use a simple nearest-neighbor-measure to the images from “Day5/images2/knownFaces”. Don’t forget to implement an adequate rejection mechanism for non-faces and unknown persons.

**useful functions:**

`abs`, `numel`