

Submission before: 11.01.2016

Discussion on: 12.01.2016

Submission on stud.ip, submission folder for sheet.

Please submit a zip file containing the .m files for Matlab programming tasks.

Exercise 1 (*eigenvalues – 3p*)

Define an arbitrary $n \times n$ -matrix A in matlab. Calculate the eigenvectors and eigenvalues of this matrix (`eig`). What happens if you multiply A with one of the eigenvectors? How are the eigenvalues correlated with A ? Give the answer in comments in your code.

Exercise 2 (*eigenfaces – 17p*)

Implement a simple face recognition system.

- Import the images in `trainings.zip` into matlab (`dir` might be a useful command). PCA in Matlab is performed using `pca`. The `pca` function requires an array where rows correspond to observations and columns to variables. In our case, an observation is one face. Transform the images accordingly and create an input matrix which contains all faces. The result is a 20×32256 -matrix.
- The eigenvectors calculated on face images are called eigenfaces. Display the eigenfaces calculated in (a). (The return value `coeff` represents the eigenvectors).
- Now project the training face images into the eigenspace to calculate their "feature vectors", i.e. a representation with significantly lower dimension. For the projection of the face images, they have to be centered first, i.e. the mean face vector has to be subtracted.
- Test the implementation by classifying the training images. Iterate on all training images and project the training image into the eigenspace (right now this step is obsolete, but you will need it in the next task). Calculate the euclidean distance of this face feature vector to all of the training feature vectors obtained in (c). The feature vector with the smallest distance represents the winner category. Display the input image and the winner image (image corresponding to the winner feature vector) beside each other to check the correctness of your implementation.
- Now classify the images in `testing.zip`. Try to reduce the number of principal components used. How many PCs are necessary to still achieve perfect classification?

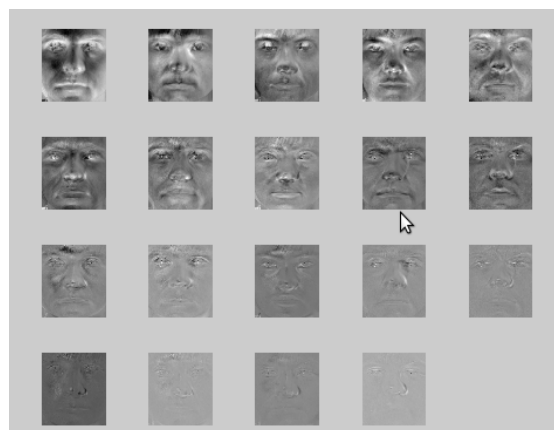


Figure 1: Eigenfaces