Assignment 2

Aflever 15/3 23:59

March 2, 2020

1 Assignment 2: Face recognition using eigenfaces

The goal of this assignment is to construct a simple face recognition system based on the approach used in the notebook for week 5. You can use any material and code from the exercise notebook. Remember to make sure your code is executable and reproducable.

- Write a function, classifyImage, that uses the eigenface projection to classify images from the test data set $\mathbf{Y}$. The function must recieve information about a single test image (like what column in $\mathbf{Y}$ the image is represented by). It must then project this test image into the subspace spanned by the K eigenfaces. The function must then return information that can be used to identify the training image that is most similarity to the test image. The similarity score can be based on the cosine distance between the projected features of the training image (represented by a vector A) and the projected features of the test image (represented by a vector B), as:

$$\text{similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{K} A_i B_i}{\sqrt{\sum_{i=1}^{K} A_i^2}\sqrt{\sum_{i=1}^{K} B_i^2}}$$

In python the cosine distance is given by the spatial.distance.cosine function, such that the most similar training image can be found by:

```
similarity_score = np.zeros(200)
for n in range(200):
    A = subspace_features[:,n]
    similarity_score[n] = spatial.distance.cosine(A, B)
most_similar_tranining_index = np.argmin(similarity_score)
```

- To figure out which subject is depicted in the most similar image, write code that computes the subject number (between 0 and 39) for a given index of a training image (between 0 and 199). Also write code that computes the subject number for a given index of a test image (between 0 and 199). You can incorporate this functionality into the classifyImage function if you wish.
- Try out your function for different test images. Plot the test image together with the training image that was found to be most similar. Also print whether the two images depict the same subject (whether the image is correctly classified).

Now, we want to investigate how the number of eigenfaces influence the classification performance. - Write code that uses the classifyImage function to classify all the 200 test images and simultaneously counts the number of miss-classified instances.

- Compute and plot the number of miss-classifications when using different number of eigenfaces K, for instance in the range $K \in [1, 100]$.
- Comment on the relation between explained variance (computed and plottet earlier for the training data) and the classification performance when varying K.
- Plot the images that are miss-classified together with their most similar images. Comment on what features you expect the mis-classification to be caused by. Is it the same features that causes miss-classification of multiple images?
- Instead of basing the classification on the single most similar image, you can base it on a majority vote of the five most similar images. Compare and comment on the classification error using this procedure.