Exercises (3rd Day)
(What the machine really sees)

Open the "data/feret_176.png" image. This image shows volunteer 176 from the color feret database which is widely used for face recognition. There are multiple copies of this image in the data directory that will help you understand what a machine really sees in different feature representations:

- The original png image
- A darker version of the original image
- A jpg compressed image (50 percent quality)
- A jpg compressed image with a high compression ratio (20 percent quality)

**Exercise 1:** Write a method that computes an LBP image which displays the LBP code at every pixel. Use a simple 8-neighborhood for comparing the pixel intensities. Display the LBP image for all four versions of the image.

How do different lightning conditions influence the LBP features?
How does jpg compression influence the LBP features?

**Optional:** Have a look at rotation invariant LBPs by shifting each code until the value is minimal.

How do the codes differ from the original ones?

**Exercise 2:** Write a method that computes an HoG visualization in order to understand what the HoG descriptor sees.

Given the number of bins for the orientation histogram and a fixed cell size in pixels, compute the values of the orientation histogram in each cell.

You can then visualize them using the opencv line drawing function. Visualize the HoG features for the same images as the LBPs above.

How do different lightning conditions influence the HoG features?
How does jpg compression influence the HoG features?

**Exercise 3:** Implement a simple object detector using the image sequence in "data/detection/".

There you will find a template image of the HBahn of TU Dortmund University that can be used for detection.
First, you need to compute the HoG representation by concatenating the orientation histograms of each cell.
Now detect the HBahn in the images in "data/detection/sequence". You will have to implement a small sliding window detector that returns you the rectangle with the lowest distance between the HoG descriptors.
The returned rectangle will be visualized.

**Note:** The interface has slightly changed to a fixed number of cells instead of their pixel size. Hence, you will have to compute the pixel size of a single cell. For simplicity we consider all cells to be in a single Block so that you don’t have to worry about concatenating the cells into multiple overlapping blocks.

**Optional:** Improve the visualization using RGB images.

**Optional:** How could you further improve the detection results? → There are at least two cues that you could exploit.

**Optional:** Try to detect the HBahn in the lower resolution images of the small sequence. → You will see that the change of the interface will come in handy.

F. Wolf, N. Azizi, M. Tavakol, G. A. Fink 
TU Dortmund University, CS XII, Group for Pattern Recognition in Intelligent Systems