DATA.ML.300 Computer Vision
Exercise Round 3

For these exercises you will need Python. Return your answers as a pdf along with your modified code to Moodle. Exercise points will be granted after a teaching assistant has checked your answers. Returns done before the solution session will result in maximum of 4 points, whereas returns after the session will result in maximum of 1 point.
Load the file containing all the necessary data/code for the following tasks.

Before continuing you have to install OpenCV library for Python. You can do that by using the command below.

```
pip install opencv-python
```

Task 1. HOG descriptors for people detection. (Programming exercise) (1 point)
We'll start by implementing a simple people detector using Histogram of Oriented Gradients as descriptor for our detector. Open hog_detector.py and follow the instructions written in the comments. **You only have to return your version of hog_detector.py**

Task 2. Basics of SSD object detector. (Programming exercise) (1 point)
We'll be looking at a CNN based object detector, Single Shot MultiBox Detector (SSD). The goal of this task is to learn the basics of SSD and deep learning implementation in Python.
Open the SSD exercise folder, follow the steps below and write down your observations.

1. We are using (a small portion of) the Udacity road traffic dataset as an example, where the target objects are vehicles and traffic lights. The dataset can be found in the *datasets* directory and the target values can be found in the .csv files. What is the form the targets are presented in? What is the difference between training and validation datasets in a general sense?

2. Next we'll take a look at the general architecture of the model. The keras_ssd7.py file implements a smaller version of the SSD detector. Open keras_ssd7.py under the *models*-directory, and locate the *build_model* function. Try to find where the first convolutional part (before the convolutional predictor layers) of the network is defined. How many convolutional "blocks" are there, and what kind of layers is each block built from?

3. SSD has it's own loss function, defined in chapter 2.2 in the original publication. What are the two attributes this loss function observes? How are these defined (short explanation without any formulas is sufficient)? The publication can be found in the exercise folder or here.

Task 3. Yolo for real-time object detection. (Programming exercise) (2 points)
This time we'll be using Yolo with pretrained weights to implement real-time object detection for a video feed. Follow the instructions in yolo.py and use the OpenCV documentation to find out how certain functions work. **Include a screenshot of the video/webcam feed with a detected object in your pdf. Also return your version of yolo.py**
If you are interested, you can also try the previous HOG detector for webcam feed.