SGN-45006 Fundamentals of Robot Vision
Exercise Round 3

March 18, 2019

For these exercises you will need Python and a webcam which should be available on the university computers. 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 3 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. For TC303 computers, open command prompt my searching *cmd* and use the command below to install the package (note that there are two dashes before *user*).

```
pip install –user opencv-python
```

You can use any editor for these tasks, however Spyder is a good choice for scientific programming.

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. **Write your chosen parameters in you pdf as well as the output image with detections. You do not 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. **You do not have to return your version of ssd7_training.py**

1. We are using (a small part of) the Udacity road traffic dataset, 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?

2. Next we'll take a look at the architecture of the model. Open keras_ssd7.py under the *models*-directory, and locate the *build_model* function. Try to find where the convolutional part of the network is defined. How many convolutional "blocks" are there, and what kind of layers is each block build from? How do you think this differs from the larger ssd300 version?

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.

4. Now it's time to train the model on a small dataset for a few epochs. Open ssd7_training.py and run the file. The training shouldn't take too long and a plot consisting of training and validation losses should show up after the training is completed. What is the difference between training and validation datasets?

Task 3. SSD300 for real-time object detection. (Programming exercise) (1 points)
This time we'll be using a larger version of SSD with pretrained weights to implement real-time object detection for webcam feed. **Download the pretrained weights here and save them to the *weights folder*.** Follow the instructions in ssd300_webcam.py and use the OpenCV documentation to find out how certain functions work. **Include a screenshot of the webcam feed with a detected object in your pdf, e.g. a monitor or a chair. Also return your version of ssd300_webcam.py.**
If you are interested, you can also try the previous HOG detector for webcam feed.