CS-E4850 Computer Vision
Exercise Round 10

The problems should be solved before the exercise session and solutions returned via MyCourses. As your solution upload a pdf file containing your answers to the questions in Exercise 1 and 2 below.

The following instructions have been tested with Matlab R2024a on Linux computers in room T7 in Computer Science building and are likely to work also with other Aalto Linux computers.

To install the required libraries do as follows:

1. Download the latest matconvnet 1.0-beta25 from `http://www.vlfeat.org/matconvnet/`.
2. Unpack the library and add the resulting matconvnet-1.0-beta25 folder to your Matlab path.
3. Add the text `#include <limits>` to the beginning of file `matconvnet-1.0-beta25/matlab/src/bits/nnpooling.cu`
4. Start Matlab by typing the following command in the terminal window: `LD_PRELOAD=/lib/x86_64-linux-gnu/libstdc++.so.6 matlab`
5. Setup MatConvNet and install mcnSSD with the instructions in Listing 1 below.
6. Get the example images and `run_ssd.m`-file by downloading `Exercise10.zip` from MyCourses and unzip them in the folder `matconvnet-1.0-beta25/contrib/mcnSSD`

Listing 1: Instructions for installing MatConvNet and SSD in Matlab

```
% In the the folder matconvnet-1.0-beta25 run the following commands:
cd matlab
vl_compilenn

% make sure that autonn is in path
vl_contrib('install', 'autonn') ;
vl_contrib setup autonn ;

% install SSD detector
vl_contrib('install', 'mcnSSD') ;
vl_contrib('compile', 'mcnSSD') ;
vl_contrib('setup', 'mcnSSD') ;

% You can test if the installation was succesfull by running ssd_demo
run ../contrib/mcnSSD/ssd_demo
```

Exercise 1. Object detection with SSD, in PyTorch.
The goal of this task is to learn the basics of deep learning based object detection with SSD by experimenting with the provided code and by reading the original Single Shot MultiBox Detector publication by Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang, and Alexander C. Berg from 2016.

Read the research paper linked above and experiment with the provided sample code according to the instructions below. Then answer the questions *a), b), c)* below and return your answers. Note that scientific publications are written for domain experts and some details may be challenging to understand if necessary background information is missing. However, don't worry if you don't understand all details. You should be able to grasp the overall idea and answer the questions even if some details would be difficult to understand.

Look through the code in `run_ssd.m` and experiment with it by running `run_ssd()`, read the original SSD publication and answer the questions below. The function `run_ssd()` implements the following steps to demonstrate the SSD object detector:

1. Build the SSD300 architecture and load pretrained weights on the VOC07 trainval dataset.

2. Load 4 random sample images from the VOC07 dataset.

3. Preprocess the images to the correct input form.

4. Run the sample images through the SSD network, parse the detections and show the results.

a) In the beginning of the publication the authors argue about the relevance and novelty of their work. Summarise their main arguments and the evidence they present to support their arguments.

b) SSD consists of two networks: a "truncated base network" and a network of added "convolutional feature layers to the end of the truncated base network." What is the purpose of the base network? Which base network do the authors of the SSD publication use, and what dataset was used to train the base network?

c) 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) and how do they help the network to minimise the object detection error?

Exercise 2. Evaluating the SSD network on some more challenging input images.
To better detect challenging inputs the authors have implemented data augmentation described in Section 2.2 and 3.2 of the publication and in reference 14. Read the relevant sections in the publication and selectively read the main points of reference 14 and answer the following questions.

2.1 Run SSD with the first test image at threshold 0.6 using the command `run_ssd(0.6,1)` and answer the following questions:

(a) Based on the result what kind of objects are easier for the network to detect?

(b) Would switching from input size 300x300 to 512x512 make the problem better, worse, or have no impact? Elaborate on why or why not. Would designing a detector that uses HD resolution of 1920x1080 or even higher images as inputs be a good idea? Elaborate on why or why not.

2.2 Run SSD with the second set of test images at threshold 0.6 using the command `run_ssd(0.6,2)` and answer the following questions:

(a) Based on the results elaborate why the detector has trouble detecting the objects?

(b) Have the authors of the SSD publication tried to mitigate this problem? If yes, briefly explain how.

2.3 Run SSD with the third set of test images at threshold 0.6 using the command `run_ssd(0.6,3)` and answer the following questions:

(a) Based on the results elaborate why the detector has trouble detecting the objects?

(b) Have the authors of the SSD publication tried to mitigate this problem? If yes, briefly explain how.

2.4 Run SSD with the fourth set of test images at threshold 0.6 using the command `run_ssd(0.6,4)` and answer the following questions:

(a) Based on the results elaborate why the detector has trouble detecting objects in the first/upper image, but is able to detect objects in the second/lower image which contains the left part of the upper image?

(b) Have the authors of the SSD publication tried to mitigate this problem? If yes, briefly explain how.

2.5 Run SSD with the fifth set of test images at threshold 0.6 using the command `run_ssd(0.6,5)` and answer the following questions:

(a) One of the test images in this set was perhaps accidentally flipped vertically and as can be seen the detector has trouble detecting objects if there's a significant rotation. Have the authors of the SSD publication tried to mitigate this problem? If yes, briefly explain how. If no, propose a naive method to mitigate it and explain why it usually might not be necessary or a good idea (more harm than good).

(b) Is a convolutional network naturally, without specifically adressing the issue, able to detect objects if there are small rotations? Briefly explain which part of the network helps to mitigate the effects of rotation and why. (Hint: What layers are usually between two convolutional layers?)

2.6 Incrementally lower the detection confidence threshold and run SSD again on the test images, observe the results and answer the following questions:

(a) What are the upsides and downsides of lowering the confidence threshold? How could you measure the effect of changing the confidence threshold? Assume you have some object detection task that you want to apply the detector to. What kind of object detection tasks could benefit from a lower confidence threshold and what kind of tasks need a high confidence threshold?

(b) Watch the following YouTube video of an object detector that is similar to SSD called YOLO V2 (You Only Look Once) and use the knowledge from previous tasks to answer the following questions: In what kind of object detection tasks could an automatic object detector be especially useful and able to perform better than a human? In what object detection tasks is a human better than a state of the art object detector? Give examples of both.