Exercises – Deep Learning part 2

For this set of exercises, we will be using Keras with the Tensorflow backend and additional implementations of object detectors based on deep learning. As a testing sample, use the images available in `images.zip`. Note: the Tensorflow library currently does not support Python 3.7, therefore these exercises only work with Python 3.6.

1. RetinaNet (Faster R-CNN with a different loss)

The following additional python libraries are required: opencv-python, keras-resnet and matplotlib

a) Clone the git repository of the YOLOv3 implementation

```
git clone https://github.com/fizyr/keras-retinanet.git
```

b) Inside the `keras-retinanet` folder, compile the Cython code and download the previously trained weights for RetinaNet

```
python setup.py build_ext --inplace
wget https://github.com/fizyr/keras-
retinanet/releases/download/0.5.0/resnet50_coco_best_v2.1.0.h5
```

c) Open detect_retinanet.py and edit only the main() function. Start by loading the model.

```
model = models.load_model(modelpathpa, backbone_name='resnet50')
```

d) Open an image and copy it (to show the detections later).

```
image = read_image_bgr(imagepath)
imagecopy = image.copy()
imagecopy = cv2.cvtColor(imagecopy, cv2.COLOR_BGR2RGB)
```

e) Preprocess, resize image and predict the detections using the previously loaded model.

```
image = preprocess_image(image)
image, scale = resize_image(image)
boxes, scores, labels = model.predict_on_batch(np.expand_dims(image, axis=0))
```

f) Change the detected boxes to correct image scaling and visualize everything.

```
boxes /= scale
visualize_detections(imagecopy, boxes, scores, labels)
```

g) Test all images and analyse the results.

2. YOLO

The following additional python library is required: pillow

a) Clone the git repository of the YOLOv3 implementation.

```
git clone https://github.com/qqwweee/keras-yolo3.git
```

b) Inside the `keras-yolo3` folder, download the previously trained weights for YOLOv3 and convert them to the h5 format, that is used by Keras.

```
wget https://pjreddie.com/media/files/yolov3.weights
python convert.py yolov3.cfg yolov3.weights model_data/yolo.h5
```

c) Copy and open detect_yolo.py and edit only the main() function. Start by opening an image.

```
image = Image.open(imagepath)
```

d) Detect objects in the image and show the results.

```
r_image = yolo.detect_image(image)
r_image.show()
```

e) Test all images and analyse the results; finally, close the Keras session.

```
yolo.close_session()
```

Note: It is also possible to train the YOLO detector with a custom dataset and different classes using the train.py script (it may take many hours). More information in the `keras-yolo3` documentation [https://github.com/qqwweee/keras-yolo3#training]