ML Laboratory 05: Object Detection with SSD

1. Objective

Students should understand the principles of object detection with single-stage object detection networks, and be able use a pretrained object detection model available in Matlab.

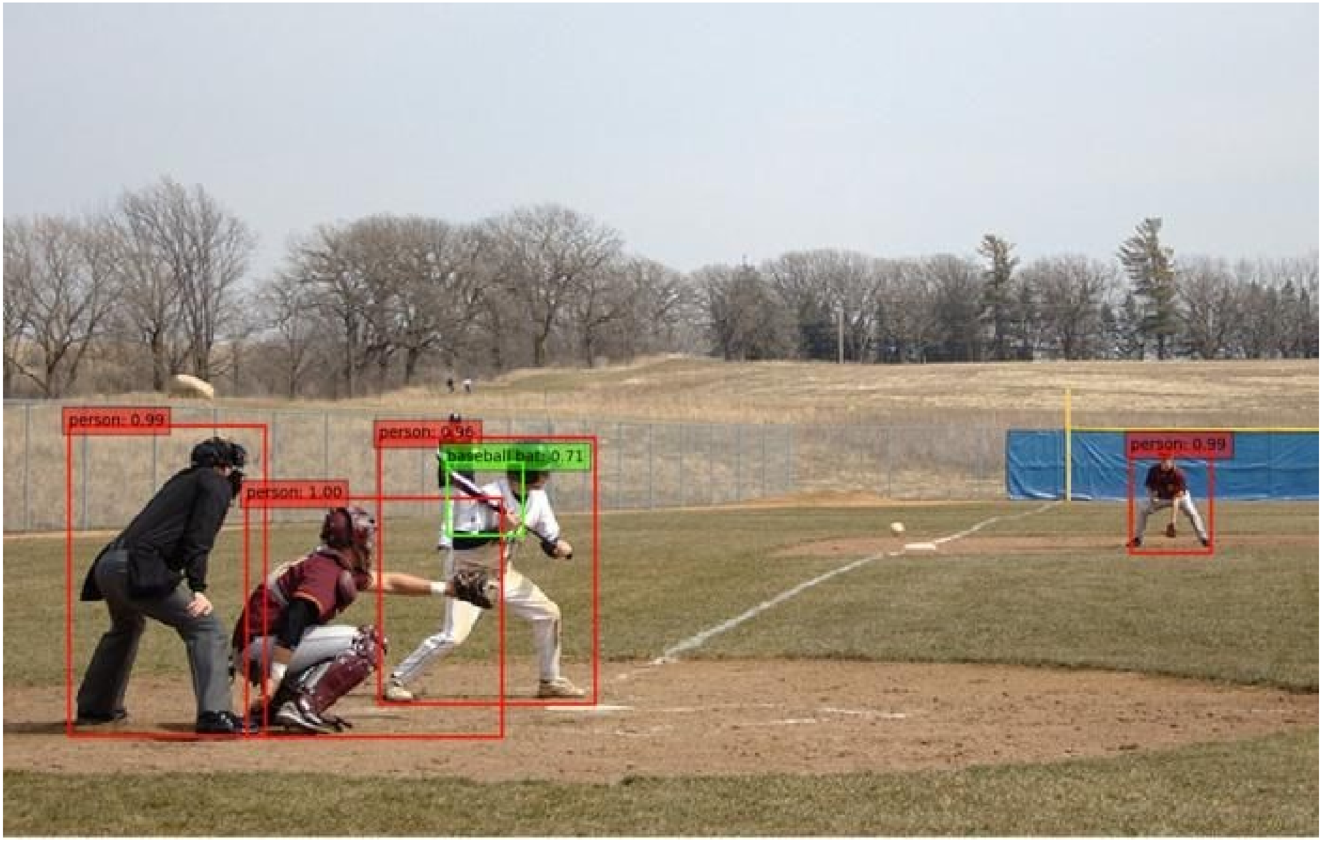
2. Theoretical aspects

Single Shot Detector (SSD) is a CNN network architecture designed for fast detection (e.g. localization) of objects in an image.

Object detection

Object detection = locating certain objects in an image, and indicate their class.

Example (image from [https://lambdalabs.com/blog/how-to-implement-ssd-object-detection-in-tensorflow/ (https://lambdalabs.com/blog/how-to-implement-ssd-object-detection-in-tensorflow/)](https://lambdalabs.com/blog/how-to-implement-ssd-object-detection-in-tensorflow/)):

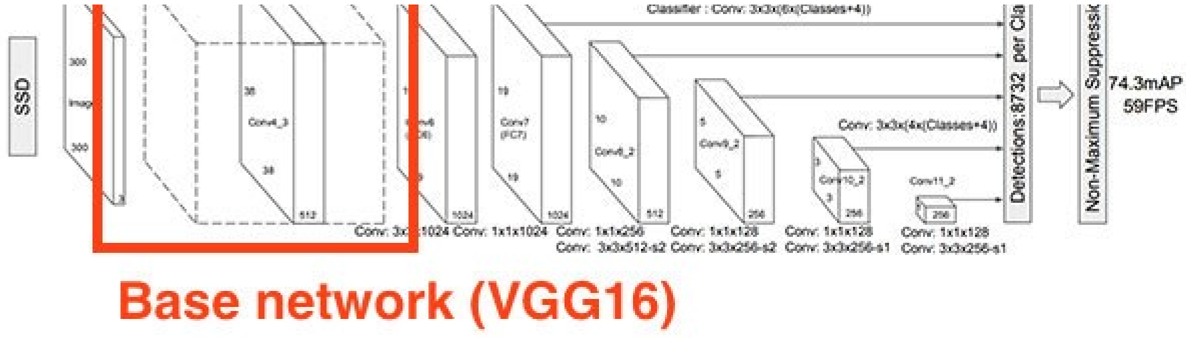


We have: a certain set of classes of objects which the model has been trained to detect (e.g. person, baseball bat, car, bicycle, , ball etc.).

We want: a list of objects detected in the image, with the following data:

1. For each detected object, we want the **bounding box** surrounding the object. This is defined by 4 pixel coordinates (e.g. bottom=140, top=300, left=74, right=128)
2. For each detected object, we want the **class** of the object (i.e. it is a person or a bicycle). The class will come with a certain confidence score (probability), just like we saw in Lab4 for classification networks.

The SSD network architecture



(image from [https://www.pyimagesearch.com/2018/05/14/a-gentle-guide-to-deep-learning-object-detection/ (https://www.pyimagesearch.com/2018/05/14/a-gentle-guide-to-deep-learning-object-detection/)](https://www.pyimagesearch.com/2018/05/14/a-gentle-guide-to-deep-learning-object-detection/)

The **input** of the network is a fixed-size $W \times H \times 3$ image tensor.

The **outputs** of the network are the parameters for all N detection boxes:

- A matrix of $N \times 4$ values with the predicted box displacements with respect to all the N anchor boxes
- A matrix of $N \times C$ with the predicted class scores for all the N boxes. C is the number of distinct classes in the dataset.

N is the total number of predicted boxes over all the image. It can be very large, e.g. 8732 boxes for a 300×300 image.

For a larger 1024×640 image, the number can be much larger, over 90000 (speaking from experience). Of course, it depends on the parameters chosen by the designer.

The model

The network model is defined by the sequence of convolutional layers in the architecture. Open the newtork in Matlab in order to inspect the architure in detail.

The model parameters

The SSD model parameters are the parameters of all the convolutional the layers. There are no fully-connected layers in the SSD network.

The cost function

The model produces two kinds of data:

- box locations
- box classification scores

The boxes are compared to the **real ground-truth** boxes available. For each ground-truth box, the best predicted box is selected (the predicted box with the correct class which overlaps the most with it), and the error is computed based on two terms:

1. The **localization error**: typically, is the absolute difference between the coordinates of the ground-truth box and the predicted box.
2. The **classification error**, which compares the class scores of the predicted box with the true ground-truth class. The cross-entropy loss function is typically used, just like it is used for all other classification tasks.

   Besides the predicted boxes which are compared to the ground-truth, there are many predicted boxes which corespond to **no** ground truth box. There are a few different methods how to treat these, some methods (e.g. Focal Loss) being better than others.

Training

Training is done with **backpropagation** and gradient descent (or some variant of it).

Backpropagation = the technique to compute the derivatives of J with respect to all parameters in the network.

Same story as for all CNN networks.

3. Practical work

Practical work is based on 3 Matlab documents:

1. Getting Started with SSD Multibox Detection (https://www.mathworks.com/help/vision/ug/getting-started-with-ssd.html)
2. Anchor Boxes for Object Detection (https://www.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html)
3. Object Detection Using SSD Deep Learning (https://www.mathworks.com/help/vision/ug/object-detection-using-single-shot-detector.html)

1. Open the tutorial Object Detection Using SSD Deep Learning (https://www.mathworks.com/help/vision/ug/object-detection-using-single-shot-detector.html)

In [ ]:
```
openExample('deeplearning_shared/ObjectDetectionUsingSSDDeepLearningExample')
```

1. Run the first steps of the tutorial (until the creation of the SSD network)
2. Investigate the network architecture by running `analyzeNetwork()` on the network variable `lgraph`

   - How many layers are there?
   - What is the required size of the input image?
   - Which are the two output layers?
   - How many boxes are predicted in all?

1. Run the next steps of the tutorial, and observer the detected boxes on the image.
2. Download a similar image from the Internet and run the model on it. Are the objects detected well?
3. Locate the call to the `detect()` function and change the detection threshold to 0.01. What happens?

   - What happens if the threshold is set too low?
   - What would happen if the threshold is set too high?
   - What is the trade-off involved in choosing a value for the threshold?

1. Set the detection parameter 'SelectStrongest' to `false` : `detect(... 'SelectStrongest', false)` . Set the threshold to 0.001. What changes? How many detected boxes are now?
2. Set the detection parameter 'SelectStrongest' to `false` : `detect(... 'SelectStrongest', false)` . Reset the threshold to a reasonable value (e.g. 0.4). What changes? How many detected boxes are now?

   - Why do we have multiple boxes around a single object?
   - Imagine a procedure to keep only a single detection box around an object (this is known as **Non-Max Suppresion (NMS)**)

1. Set the variable `doTraining` to `true` (or 1) and train the model. How fast does it work?

4. Final questions

TBD