Department of Computer Science

TDT4265 - Computer Vision and Deep Learning

Assignment 4 Report

Authors:
Sander Aakerholt, Ludvig Brannsether Ellingsen

March, 2022

1 Introduction

This is an outline for your report to ease the amount of work required to create your report. Jupyter notebook supports markdown, and I recommend you to check out this cheat sheet. If you are not familiar with markdown.

Before delivery, **remember to convert this file to PDF**. You can do it in two ways:

- Print the webpage (ctrl+P or cmd+P)
- Export with latex. This is somewhat more difficult, but you'll get somehwat of a "prettier" PDF. Go to File → Download as → PDF via LaTeX. You might have to install nbconvert and pandoc through conda; `conda install nbconvert pandoc`.

Task 1

Task 1a)

Intersection over Union (IoU) is a measure of the difference between an objects ground truth box and the predicted box for that object. The IoU is calculated by dividing the intersection of the boxes with the union of the boxes, and the ratio is usually considered a hit if it is above 0.5 (or 50%). The formula for the ratio is:

$$a_0 = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{1}$$

Equation 1 is from section 4.2 in this article.

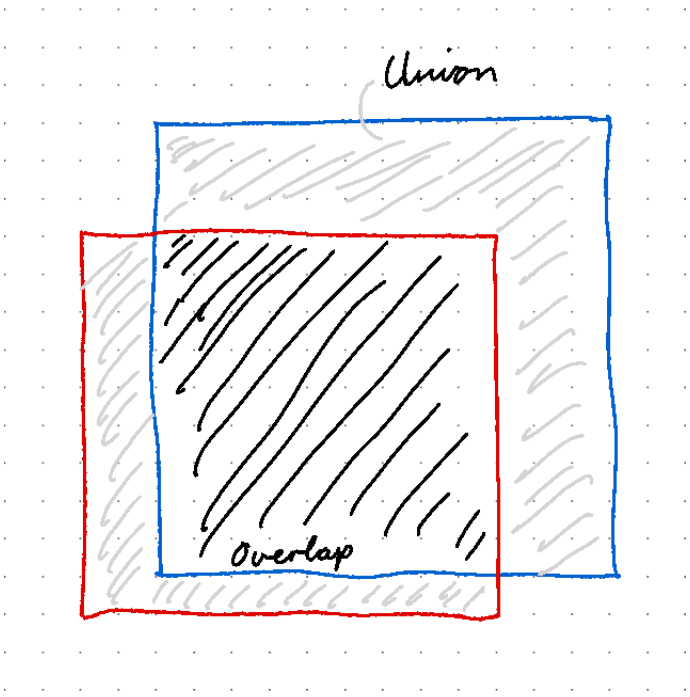

Figure 1: Drawing of equation 1. Where "Overlap" is the region $area(B_p \cap B_{gt})$ and "Union" is $area(B_p \cup B_{gt})$. And the red square indicates the predicted area, and the blue square indicates the ground truth area. Drawing inspiration from this article.

Task 1b)

The equation for precision is:

$$Precision = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{2}$$

and for recall we have:

$$Recall = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{3}$$

A true positive is when a classification is classified correctly, while a false positive is when something is classified as something it is not. Hence if you want to classify cats in a picture and the classifier classifies a dog as a cat, we have a false positive classification.

Equations 2 and 3 was found in this article.

Task 1c)

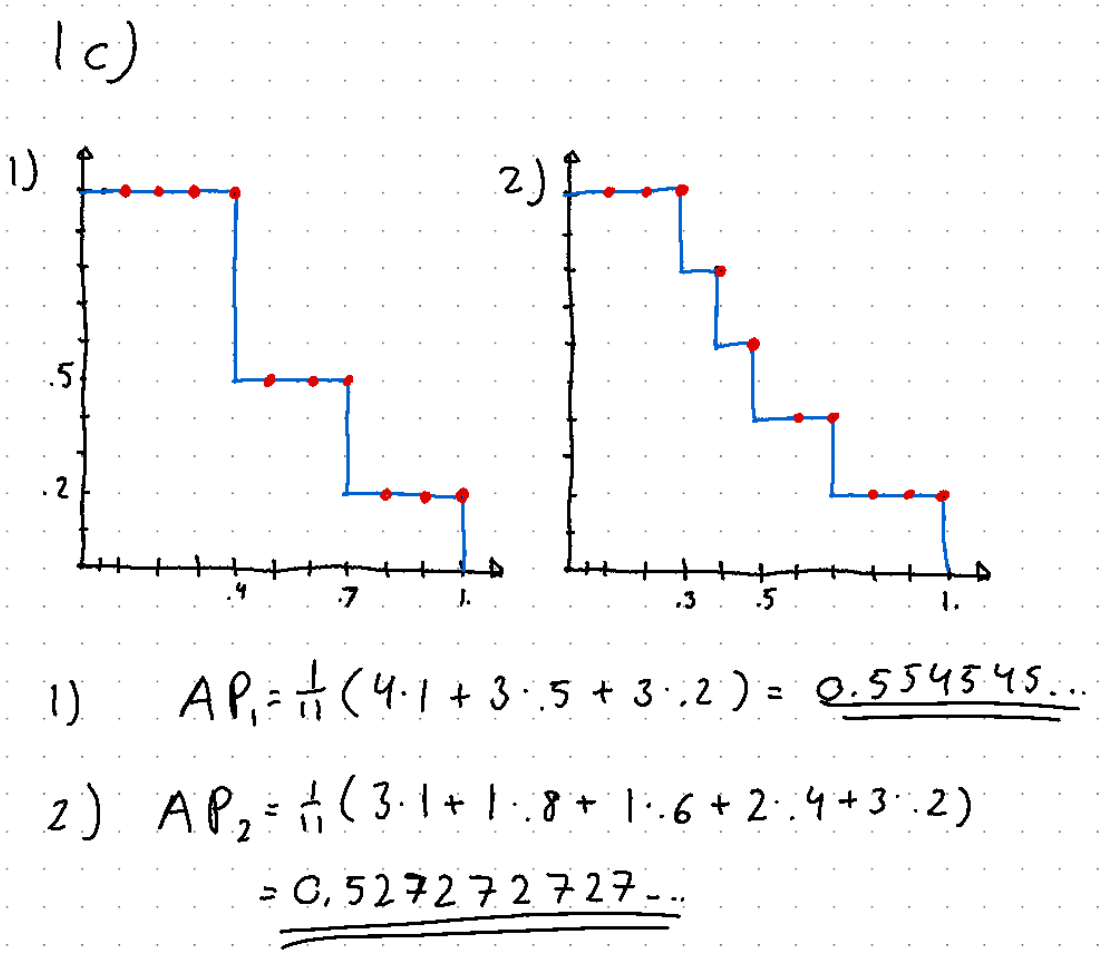

Figure 2: Solution for 1c. Formula found in this article.

Task 2

See fig. 3.

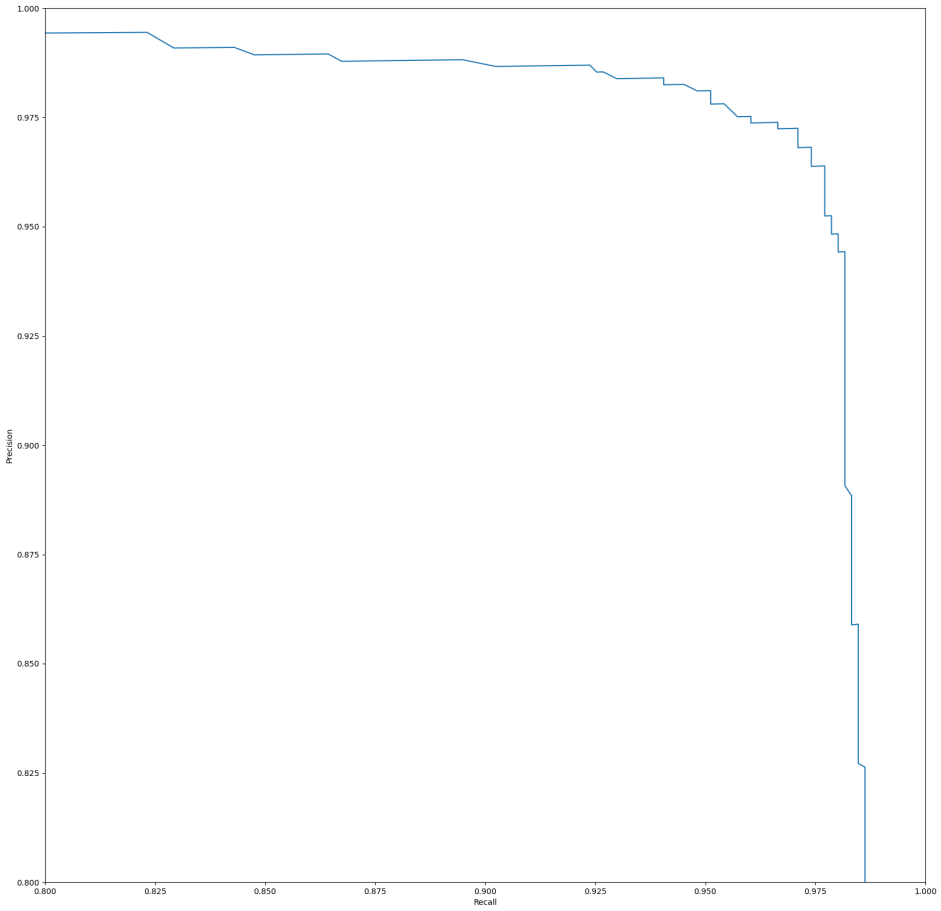

Figure 3: Precision-recall curve in task 2f

Task 3

Task 3a)

SSD uses non-maximum suppression to filter out the overlapping boxes. In this filter, the predictions are sorted by confidence score. The most confident prediction is placed at it's given location. The second most confident prediction is compared with the most confident one. If the IoU between the predictions is higher than 0.45, the second prediction is discarded as there is already a more confident prediction for that location.

Continuing this pattern, the position of the most confident remaining prediction is compared with previous predictions in search of an IoU higher than 0.45. If such a previous prediction exists, the current prediction is discarded.

Task 3b)

False, the deeper layers detect larger objects, as the resolution of the grid is courser than the "higher" layers.

Task 3c)

For each convolutional layer, SDD use a scale value to define the size of the default boxes. This is to account for different sized objects in the image. However, different objects have different shapes in addition to different sizes. A car might take up the same amount of space as a person in an image, but this space will be distributed differently. Each convolutional layer therefore uses k multiple aspect ratios to define different default box shapes. The k different default boxes is attached to k different anchors, or center locations.

Task 3d)

Yolo uses 2 fully-connected layers, whereas SDD uses convolutional layers of varying sizes. This allows SDD to look at grids of different sizes in the image, allowing for higher accuracy in classifying objects.

Other differences between SSD and YOLO are that YOLO uses k-means estimation to decide which default boxes that will be used in the layers, and that YOLO uses a larger input image resolution compared to SSD300.

Task 3e)

For a feature map with resolution 38x38, there is $38 \times 38 = 1444$ anchor locations. Since there is 6 different anchor boxes for each anchor, there is a total of 8664 anchor boxes in this feature map.

Task 3f)

Using the same method as in section 1, we can calculate the total number of anchor boxes in the network as $(38 \times 38 \times 6) + (19 \times 19 \times 6) + (10 \times 10 \times 6) + (5 \times 5 \times 6) + (3 \times 3 \times 6) + (1 \times 1 \times 6) = 11640$.

Task 4

Task 4a)

Done. See results in 4b.

Task 4b)

See fig. 4 and fig. 5. The mAP-value was 0.737 after 20 epochs. However, after 19 epochs it was above 0.75.

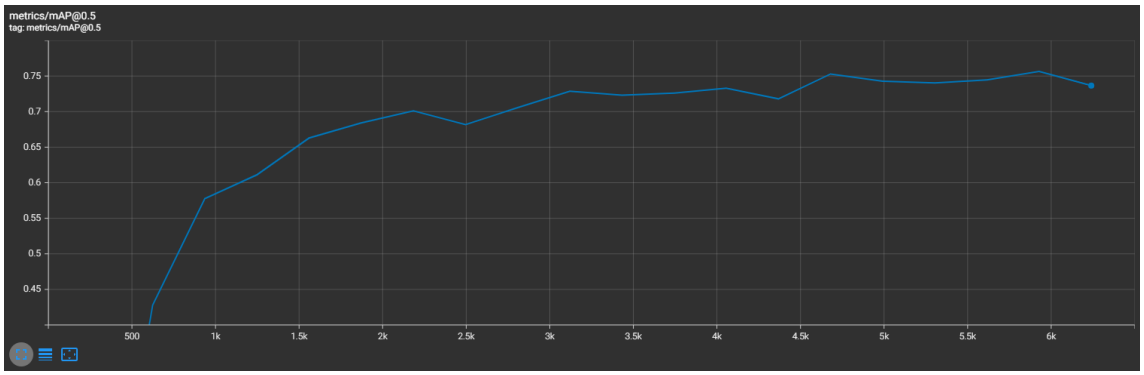

Figure 4: MAP for the network given in the assignment and IoU threshold of 0.5 after approximately 6000 iterations

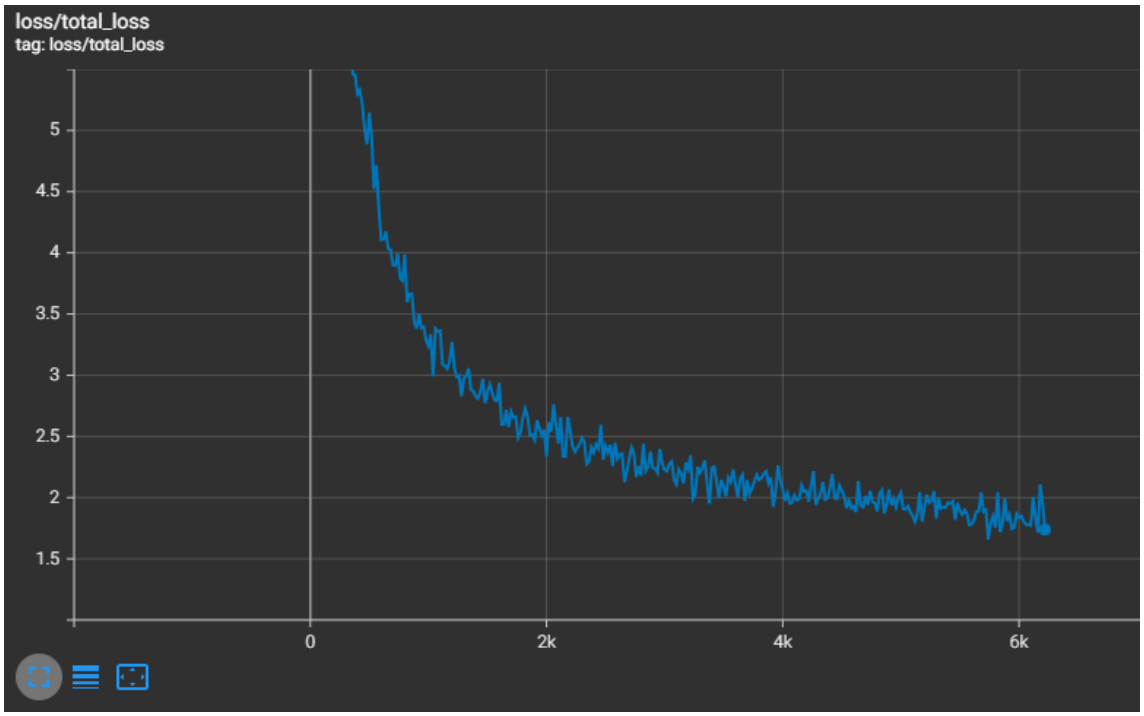

Figure 5: Total loss for the network given in the assignment after approximately 6000 iterations