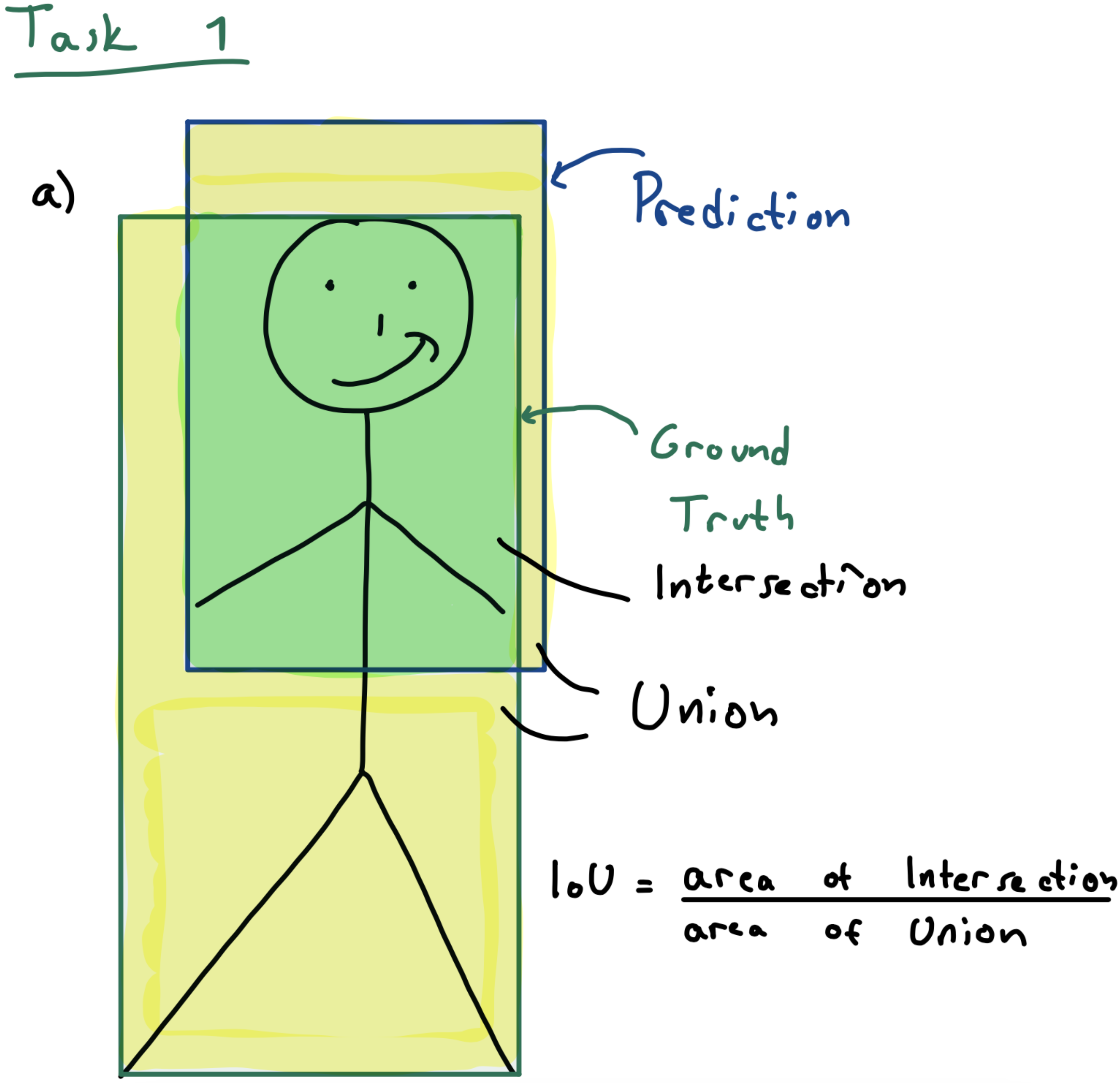
Assignment 4 Report - Group 200 - Jostein Lysberg and Ludvig Løite

Task 1

task 1a)

The Intersection over Union is simply a measure of the equality of two bounding boxes, given their respective overlap. It is often used to estimate how accurate a bounding box prediction is, compared to the ground truth. For two bounding boxes, we can calculate this value by dividing the area of overlap with the area of union.



$$IoU = \frac{\text{area of Intersection}}{\text{area of Union}}$$

task 1b)

A true positive is defined as a positive prediction on a class which is positive in reality. This prediction is correct. A false positive is also a positive prediction, but in this case the class is negative in reality. This prediction is therefore not correct.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

task 1c)

mAP for class 1:

$$mAP_1 = (5 * 1 + 3 * 0.5 + 3 * 0.2)\frac{1}{11} = 0.645$$

mAP for class 2:

$$mAP_2 = (4 * 1 + 1 * 0.8 + 1 * 0.6 + 2 * 0.5 + 3 * 0.2)\frac{1}{11} = 0.636$$

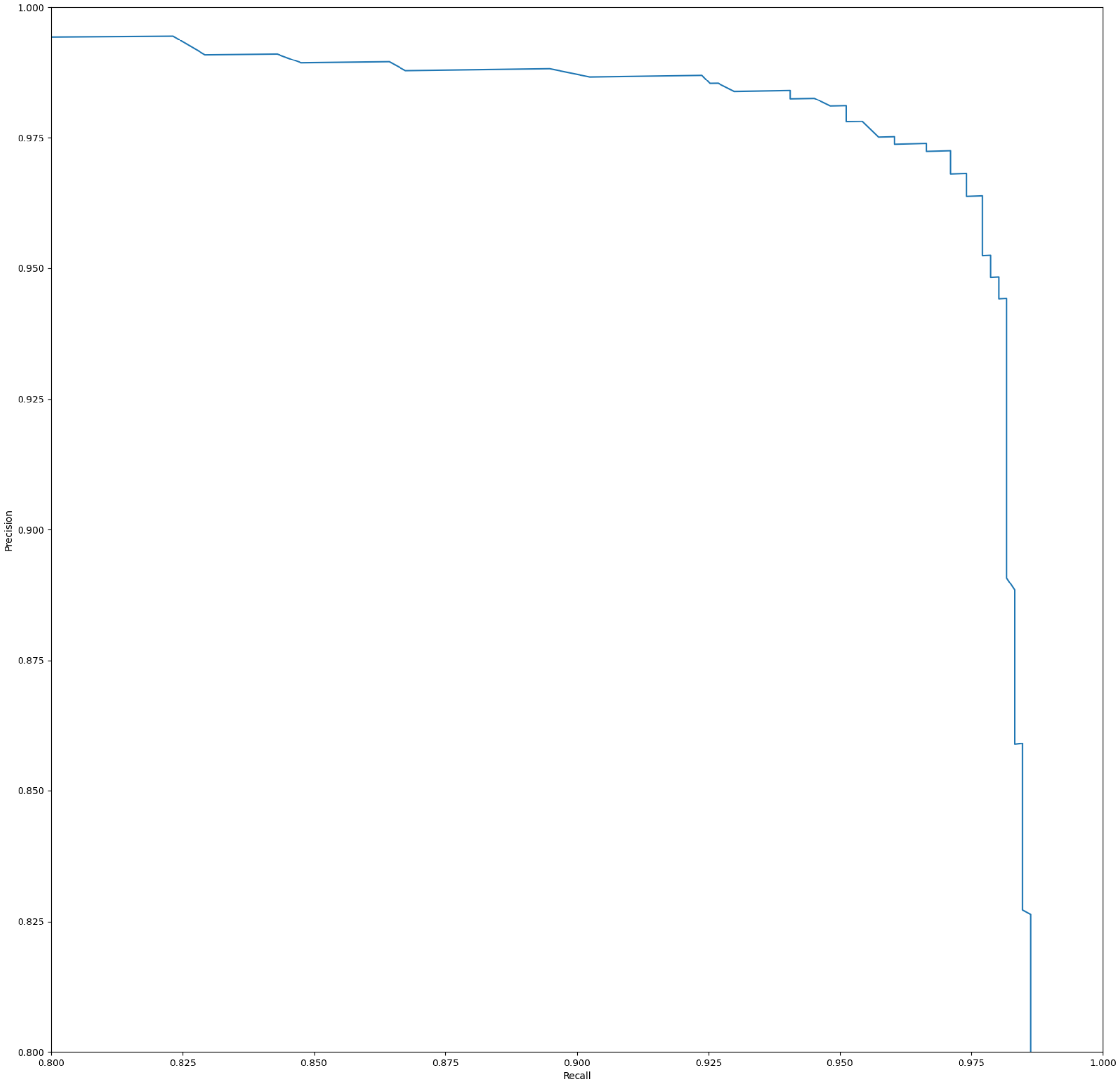
Total mAP:

$$mAP = \frac{1}{2}(mAP_1 + mAP_1) = 0.641$$

Task 2

Task 2f)

Below is the plot of our final precision-recall curve



Task 3

Task 3a)

The filtering operation of removing duplicate predictions pointing to the same object is called non-maximum suppression. This step is performed at the very end of the model, after the convolutional network has produced a collection of bounding boxes and scores for the presence of object class instances in those boxes.

Task 3b)

False. In SSD, Small objects are detected at the earlier, higher-resolution feature maps. This is because the bounding boxes come in fixed sizes, and will therefore cover larger proportions of the feature maps as the deeper layers shrink in resolution.

Task 3c)

The reason for utilizing several different bounding boxes with distinct aspect ratios is that all categories to be detected do not have arbitrary shapes and sizes. To some extent, all objects come in inherently predictable shapes, which is reflected in the wide range of bounding boxes in SSD.

Task 3d)

One major distinction between SSD and YOLO is the addition of several convolutional feature layers to the end of a base network to predict a shape offset relative to the default box coordinates, as opposed to YOLO's use of a fully connected layer for this step. This shape offset is in turn associated with the confidences of all default boxes with their different scales and aspect ratios.

Task 3e)

Assuming k=6 different default boxes and a feature map with resolution 38x38, this particular feature map will consist of 8664 anchor boxes.

$$6 * 38^2 = 8664$$

Task 3f)

Assuming k=6 different bounding boxes, the entire network would consist of 11640 anchor boxes.

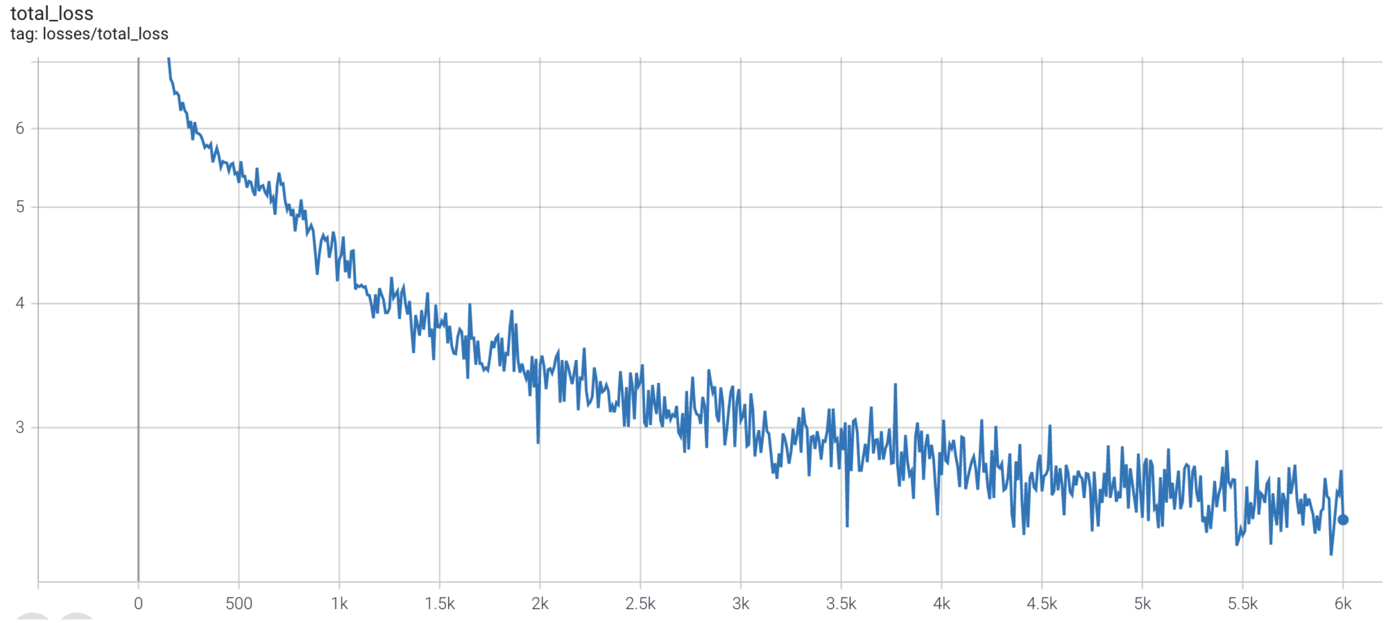
$$6 * (38^2 + 19^2 + 10^2 + 5^2 + 3^2 + 1^2) = 11640$$

Task 4

Model from task	Number of iterations	mAP
4b	6000	75.8 %
4c	10000	87.6 %
4d	14500	90.3 %
4f	5000	47.1 %

Task 4b)

Below, you can see the plot of "total_loss", taken from tensorboard. We were able to reach a mAP value of 75.8%.



Task 4c)

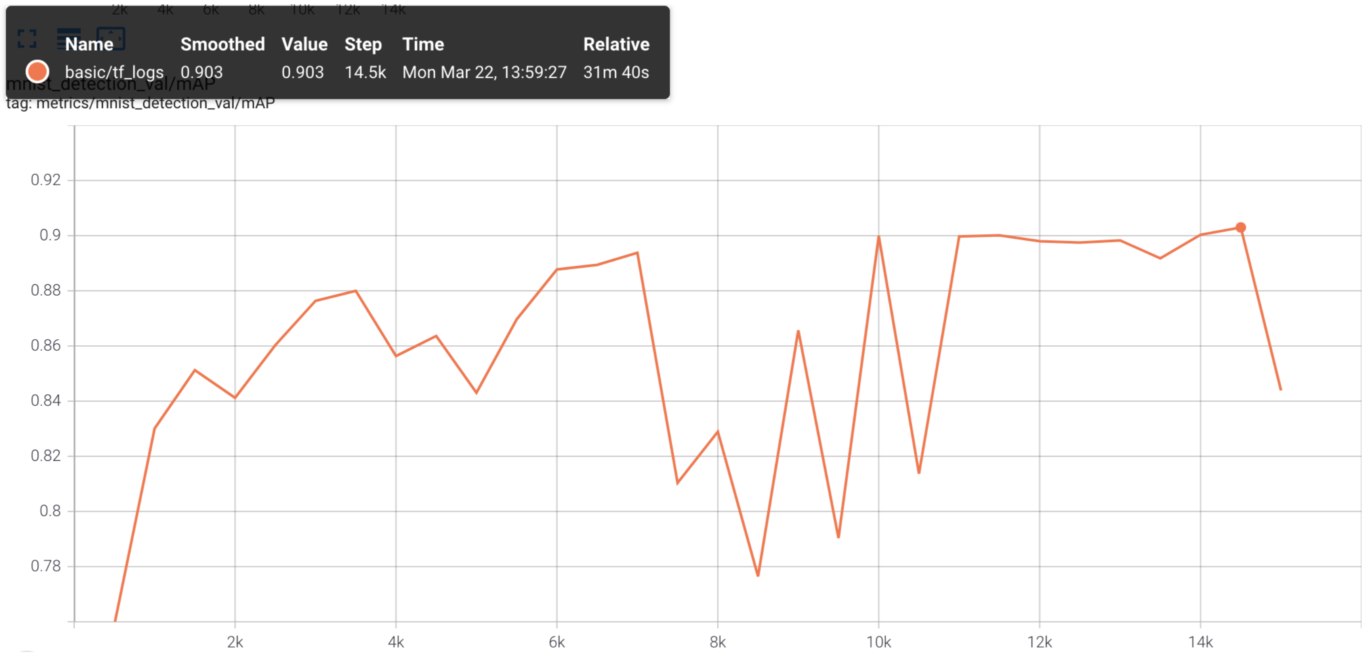
We did several changes to improve our model:

1. We added more layers to deepen our network, by adding a convolution layer to each of the 6 blocks. We ended up with around 3.9 million parameters.
2. We added Batch Normalization after each convolution layer except the first and the last in each of the 6 blocks.
3. We included the RandomSampleCrop() transform

Our final mAP value was 87.6%.

Task 4d)

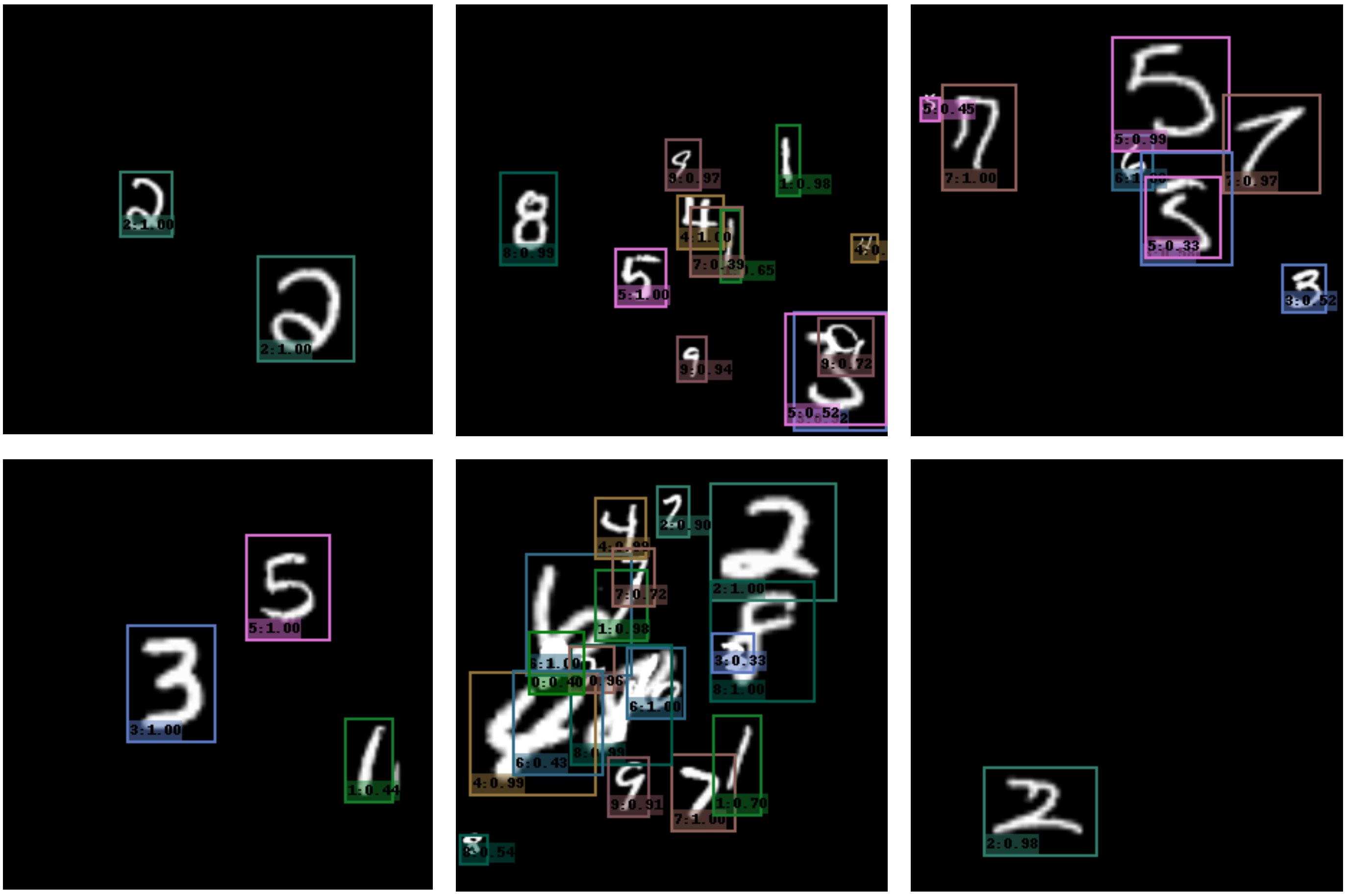
We were able to reach a mAP value of 90.3% after running 14500 iterations, as can be seen from the plot below. This was achieved by adjusting the threshold, and by lowering the minimum box sizes.

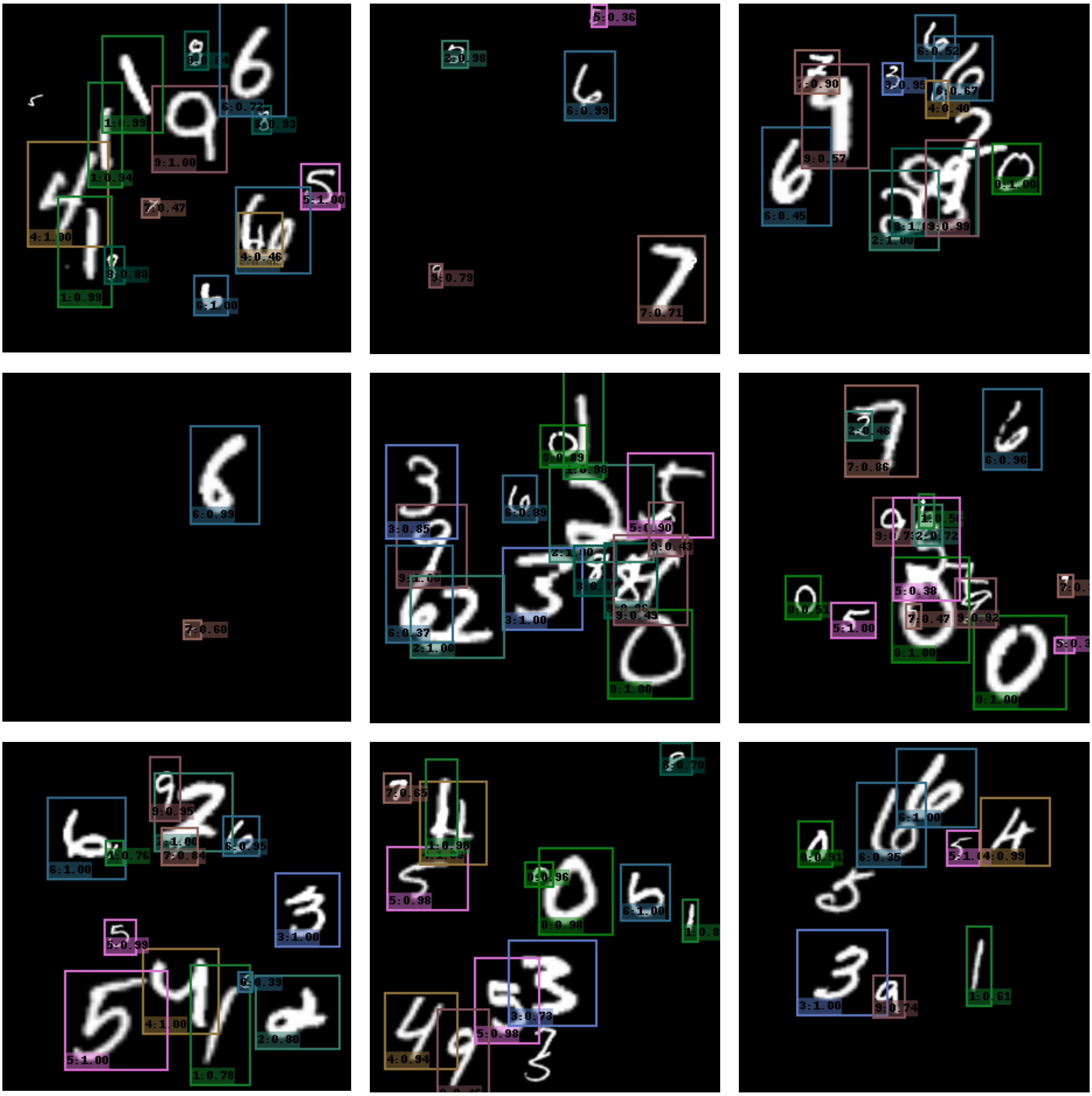


Name	Smoothed	Value	Step	Time	Relative
● basic/tf_logs	0.903	0.903	14.5k	Mon Mar 22, 13:59:27	31m 40s

Task 4e)

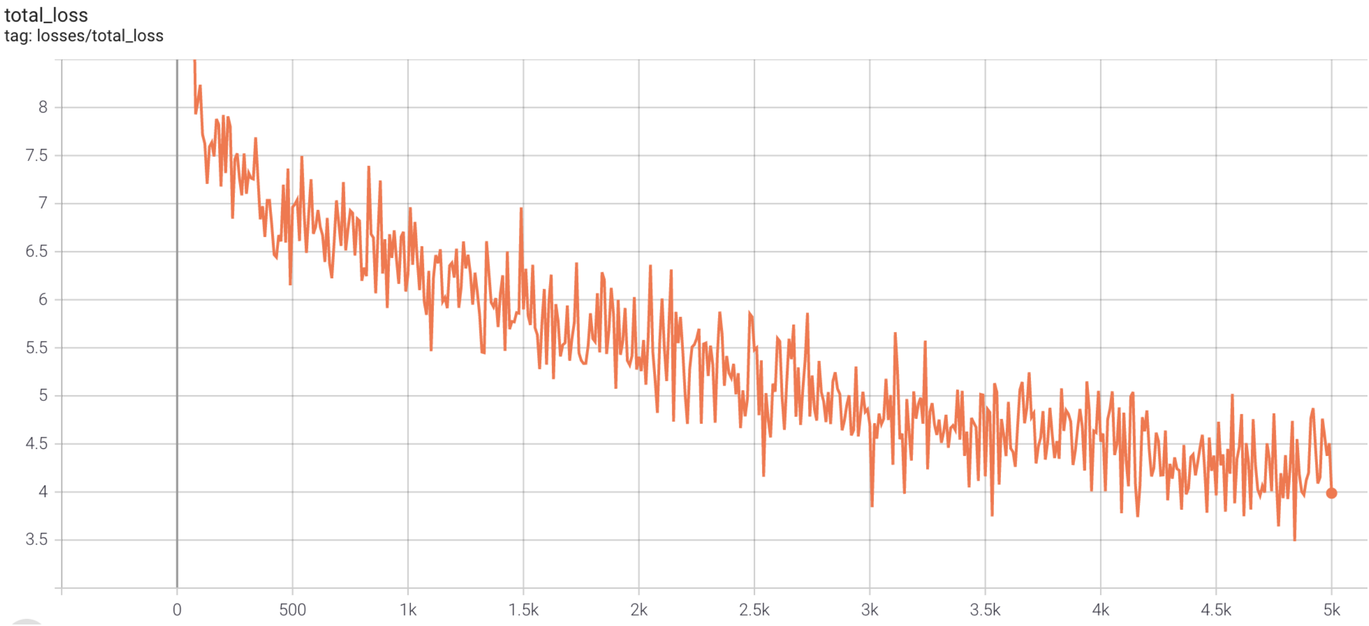
The following pictures show our classified images using a score threshold of 0.3. With this low threshold, all the detections should not be taken as the truth, at least not for critical applications. We used this low threshold to get a clear understanding of model performance. Our trained model was able to detect most digits, but was worse at detecting those contained within small bounding boxes. Most of the digits that were significantly smaller than the rest, had a score of less than 0.5. Some were even lower than our score threshold of 0.3, and were not classified at all. This is a known weakness of SSD, and is likely a result of its method for handling different object scales. The fact that prediction is performed in the feature maps of several different layers of a single network, as opposed to processing the image at different sizes and combining the result afterwards, might affect the small objects that consistently rely on being classified in the earlier, high-resolution convolutional layers. As mentioned in the paper by Liu, Anguelov, Erhan, Szegedy, Reed, Fu and Berg, this problem is reduced to some degree by introducing random sample cropping, as we have shown in task 4c and 4d. We also saw that the digit one consistently scored worse than the rest of the digits during training. In the final model, however, one is classified approximately equal to the other digits. We think the model's bad performance on classifying this particular digit might be explained by the fact that ones generally cover small bounding boxes, which has proven to be a weakness of SSD. Another explaination could be the fact that parts of other digits are very similar to one. For example, most occurences of seven, and many poorly written occurences of nines or fours could prove to be a problem.

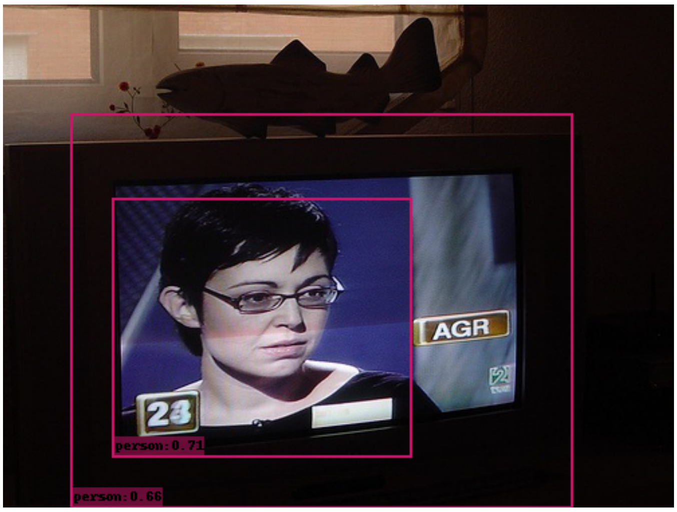




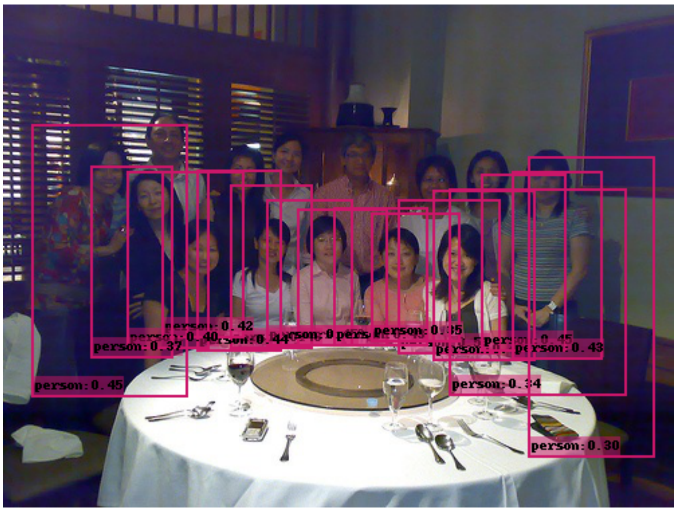
Task 4f)

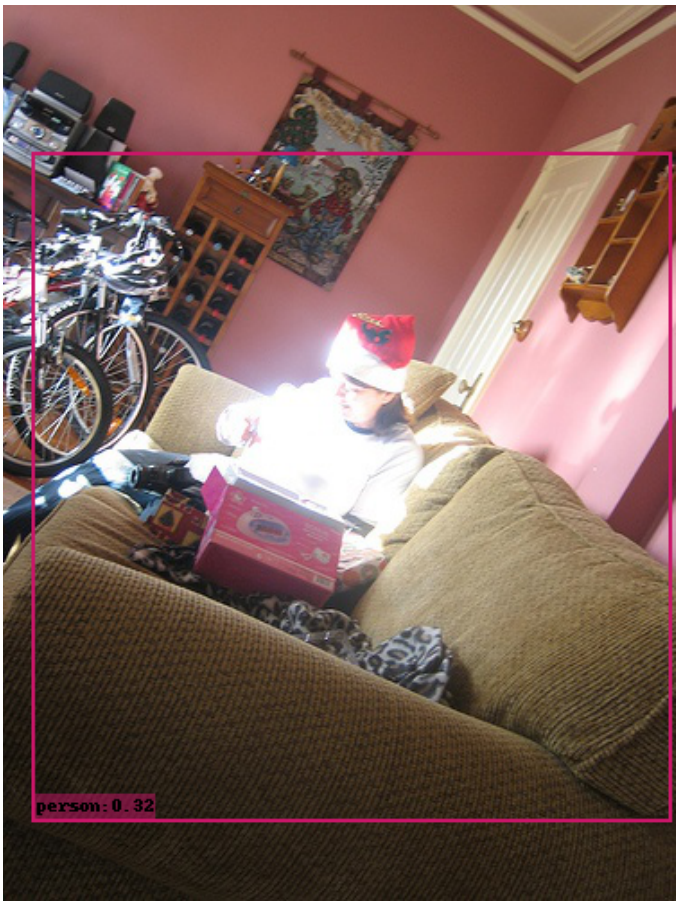
Our final mAP was 47.1%. The plot of "total loss" and the classified images can be seen below. We used a threshold of 0.3 to get further insight into classification of our model, than by using the default of 0.7.


total_loss
tag: losses/total_loss
8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
0
500
1k
1.5k
2k
2.5k
3k
3.5k
4k
4.5k
5k


AGR
28
person: 0.71
person: 0.66

at: 0.87


person: 0.42
person: 0.40
person: 0.44
person: 0.35
person: 0.45
person: 0.37
person: 0.43
person: 0.45
person: 0.34
person: 0.30


person: 0.32