TDT4265 Assignment 4 Report

Group 16

Hanna Waage Hjelmeland, Siri Holde Hegsvold

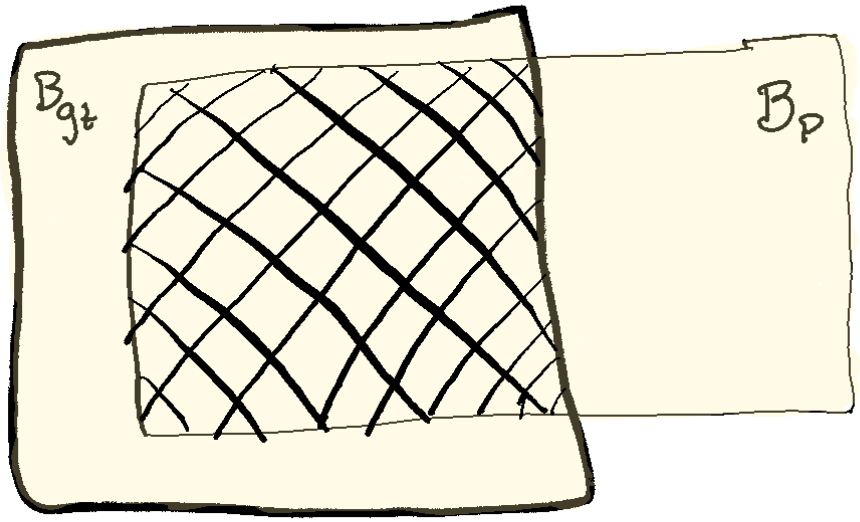
Task 1

Task 1

1. a) Let B_p be the predicted bounding box and B_{gt} be the ground truth. The intersection over union is then defined as

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})}$$

Illustrated:



$$IoU = \frac{\text{area}}{\text{area}}$$

1.b) TP = True positive
TN = True negative
FP = False positive
FN = False negative

$$\text{Precision} = \frac{TP}{TP+FP} \qquad \text{Recall} = \frac{TP}{TP+FN}$$

True positive is when we make a correct prediction, i.e. when we detect & classify an object corresponding to the ground truth.

False positive is when we make a prediction that isn't there, i.e. when we detect & classify a part of a picture where no ground truth is present, or it is present, but the IoU is below the threshold.

1.c) The mAP is given by

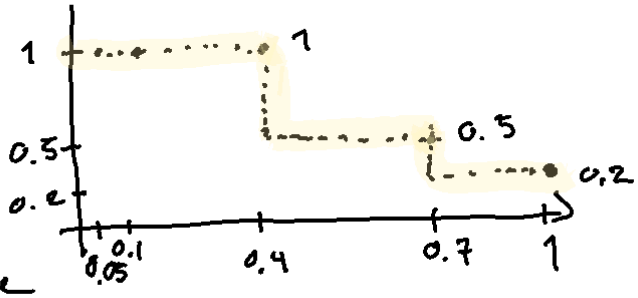
$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} p_{interp}(r)$$

where

$$p_{interp}(r) = \max_{\tilde{r} : \tilde{r} \geq r} p(\tilde{r})$$

↳ maximum precision measured for a method for which the corresponding recall exceeds r.

so, for class 1:
we smooth the curve we get from the given points by replacing the precision value at every recall level with the maximum precision value to the right of the level.

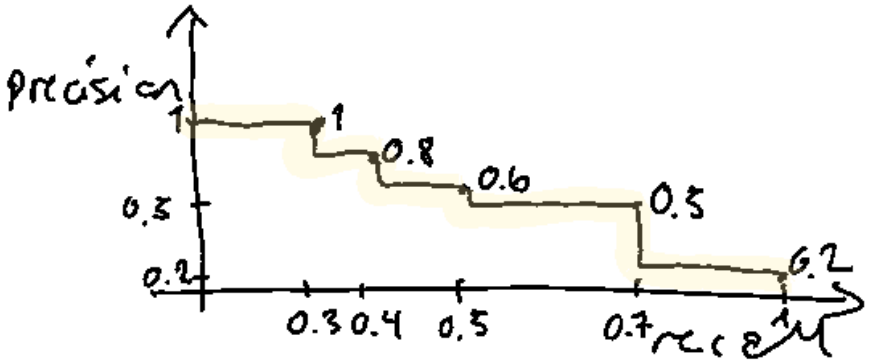


$$mAP = \frac{1}{11}\left(p_{interp}(0) + p_{interp}(0.1) + \ldots + p_{interp}(1)\right)$$

$$= \frac{1}{11}\left(5 \cdot 1 + 3 \cdot 0.5 + 3 \cdot 0.2\right) = \underline{0.65}$$

... 7. c)

class 2:



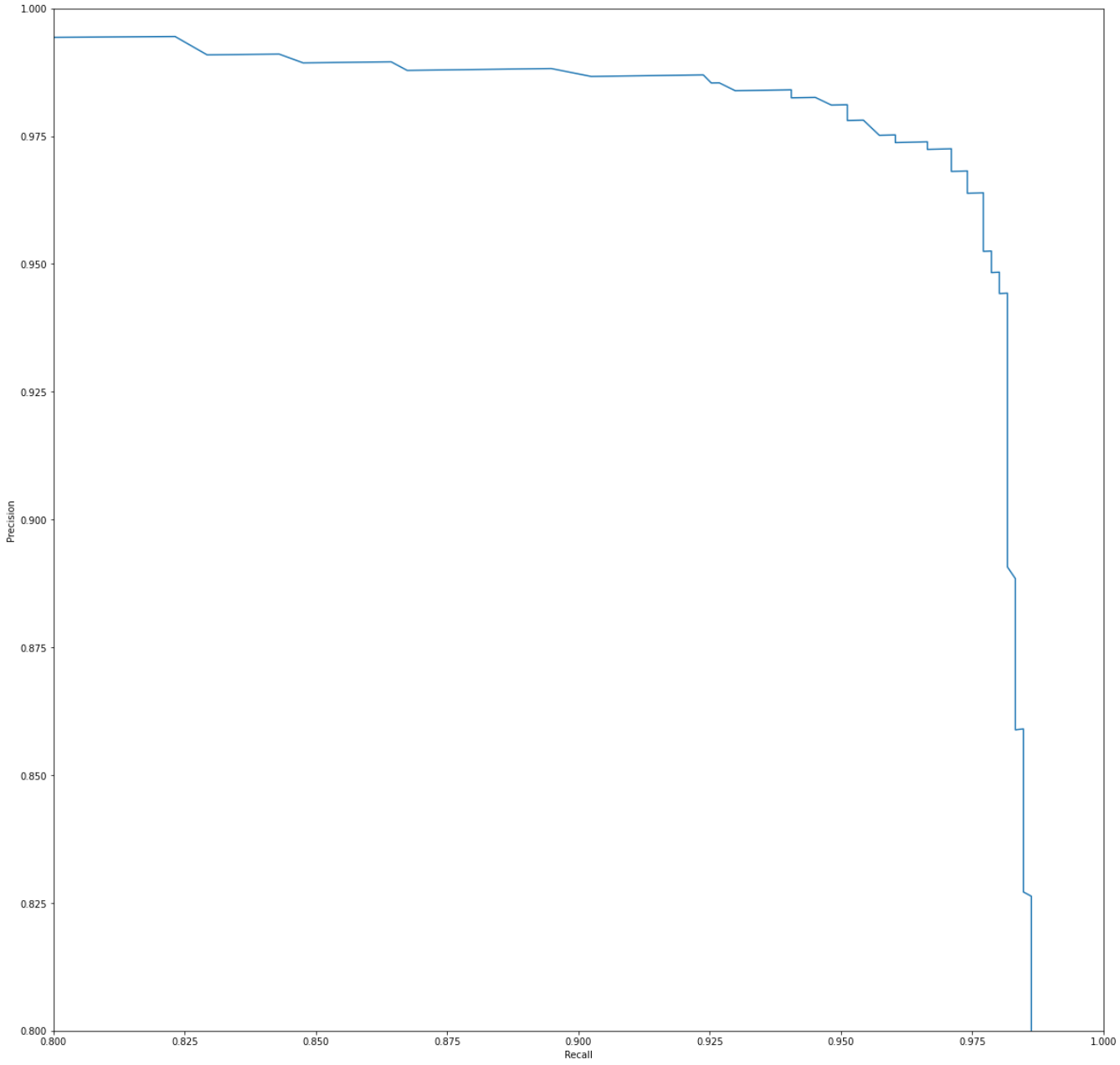
$$mAP = \frac{1}{11}(4 \cdot 1 + 0.8 + 0.6 + 2 \cdot 0.5 + 3 \cdot 0.2)$$

$$= \underline{0.64}$$

Task 2

Task 2f)

```
Mean average precision: 0.9066
```



Task 3

Task 3a)

To filter out the set of overlapping boxes the SSD use a filter operation called Non-Maximum Suppression

Task 3b)

False: deeper layers in SSD are responsible to detect large objects. Which makes sense since the deeper layers is more generalized, as discussed in assignment 3 task 4.

Task 3c)

Answer based on the SSD-paper linked in the assignment text:

We use different aspect ratios at the same spatial location because different object sizes and shapes are easier detected by some aspect ratios then another. Using different aspect ratios let us handle object of different sizes and shapes. This means that a prediction is responible for some spesific shapes of the ground truth. This will lead to predictions closer to the corresponing default box. An example from the SSD-paper is that a dog is matched with 4x4 feature map and not 8x8, while the cat is matched with 8x8. The learning problem is simplified and the network can have multiple overlapping default boxes insted of just one.

Task 3d)

Answer based on the SSD-paper linked in the assignment text:

The main difference between SSD and YOLO is that the architecture of YOLO uses fully connected layers instead of convolutional filters. The SSD architecture has several feature layers added at the end of the base network, which predict the offset to default boxes of different scales and aspect ratio and their confidence. The YOLO architectures does not have these extra layers. This makes it possible for SSD to operate on multi-scale feature maps, which means that the layers decrease in size progressively and allow predictions of detections at multiple scales. YOLO only operate on a single scale feature map.

Task 3e)

38×38 feature map with 6 anchors at each location. This gives 38×38×6 = 8664 predictions in total. Each prediction corresponding to an specific anchor box, making it 8664 anchors boxes in total for this feature map.

Task 3f)

38×38×6 = 8664, 19×19×6 = 2166, 10×10×6 = 600, 5×5×6 = 150, 3×3×6 = 54, 1×1×6 = 6,

In total = 11 640

For the entire network we have 11 640 predictions --> 11 640 anchors boxes.

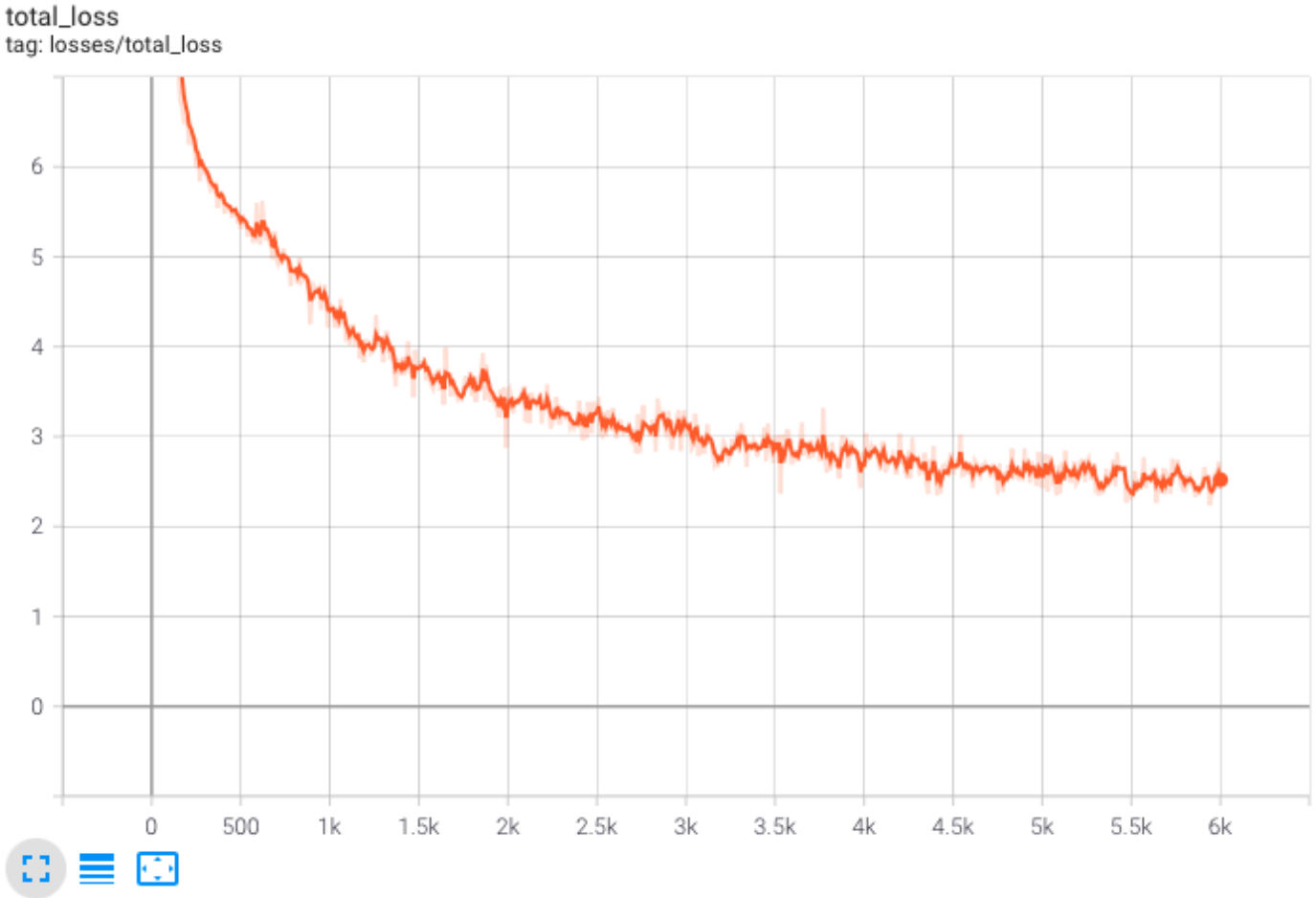
Task 4

Task 4b)

Output at final checkpoint:

```
2021-03-16 00:05:57,050 SSD.trainer INFO: Saving checkpoint to outputs/basic/model_final.pth
2021-03-16 00:05:57,094 SSD.trainer INFO: Total training time: 0:19:02 (0.1903 s / it)
2021-03-16 00:05:57,102 SSD INFO: Start evaluating...
2021-03-16 00:05:57,249 SSD.inference INFO: Evaluating mnist_detection_val dataset(1000 images):
100% 100/100 [00:05<00:00, 19.92it/s]
2021-03-16 00:06:02,961 SSD.inference INFO: mAP: 0.7566
0               : 0.8061
1               : 0.6201
2               : 0.7438
3               : 0.7787
4               : 0.7962
5               : 0.7701
6               : 0.7827
7               : 0.7598
8               : 0.7791
9               : 0.7297
```

The total loss: (screenshot from tensorboard)



mAP at 6000 iterations: 0.7566 --> 75,7%

Task 4c)

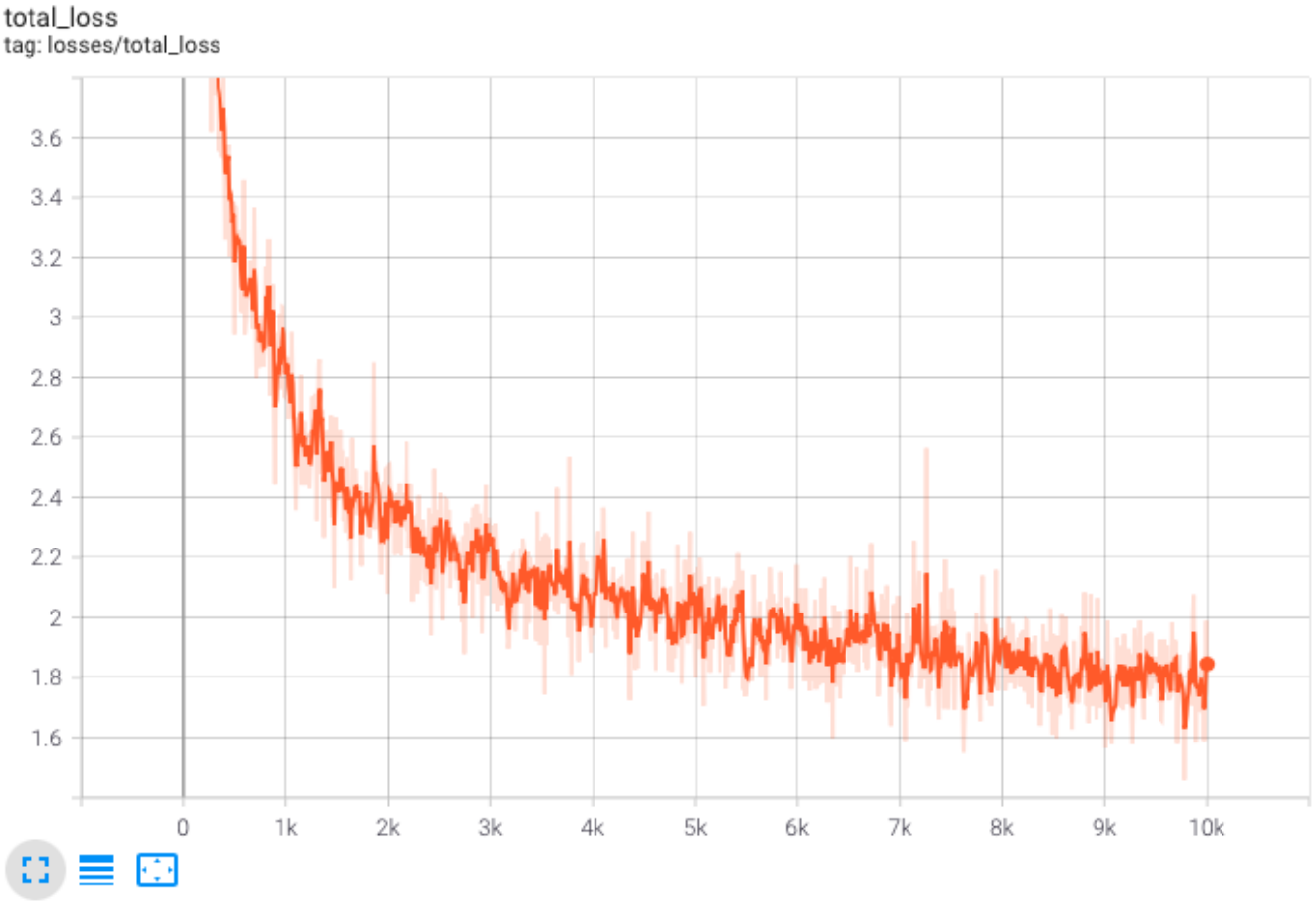
Output at final checkpoint:

```
2021-03-16 10:17:20,087 SSD.trainer INFO: Saving checkpoint to outputs/basic/model_final.pth
2021-03-16 10:17:20,159 SSD.trainer INFO: Total training time: 0:34:43 (0.2083 s / it)
2021-03-16 10:17:20,169 SSD INFO: Start evaluating...
2021-03-16 10:17:20,352 SSD.inference INFO: Evaluating mnist_detection_val dataset(1000 images):
100% 100/100 [00:05<00:00, 18.24it/s]
2021-03-16 10:17:26,540 SSD.inference INFO: mAP: 0.8511
0               : 0.8833
1               : 0.7822
2               : 0.8450
3               : 0.8580
4               : 0.8678
5               : 0.8459
6               : 0.8657
7               : 0.8379
8               : 0.8702
9               : 0.8554
```

After 9000 iteration we have a mAP over 85%

The best mAP is found at 9500 iterations with 86,3%, and the final mAP at 10 000 iterations is 85,1%

We have here introduced batch normalization, leaky relu and adam optimizer into our model. The total loss becomes:



Task 4d)

To reach the mAP of 90%, we continued with the model from c) but increased the number of convolution layers in the first layer by 3 and doubled the number of filters in every layer.

Result:

We reach 90.1% at iteration 14,5K. At 15K, the mAP drops to 62,5%, so we disregard this iteration and save the second to last as the best model for further use.

Output at 14500:

```
2021-03-18 14:21:23,012 SSD.inference INFO: Evaluating mnist_detection_val dataset(1000 images):
100% 100/100 [00:06<00:00, 16.37it/s]
2021-03-18 14:21:29,745 SSD.inference INFO: mAP: 0.9010
0               : 0.9082
1               : 0.8801
2               : 0.8957
3               : 0.9037
4               : 0.9041
5               : 0.9034
6               : 0.9045
7               : 0.8985
8               : 0.9063
9               : 0.9051
```

Task 4e)

Using the model with mAP of 90,1% (second to last, at 14500 iterations):

We recall the mAP of the model:

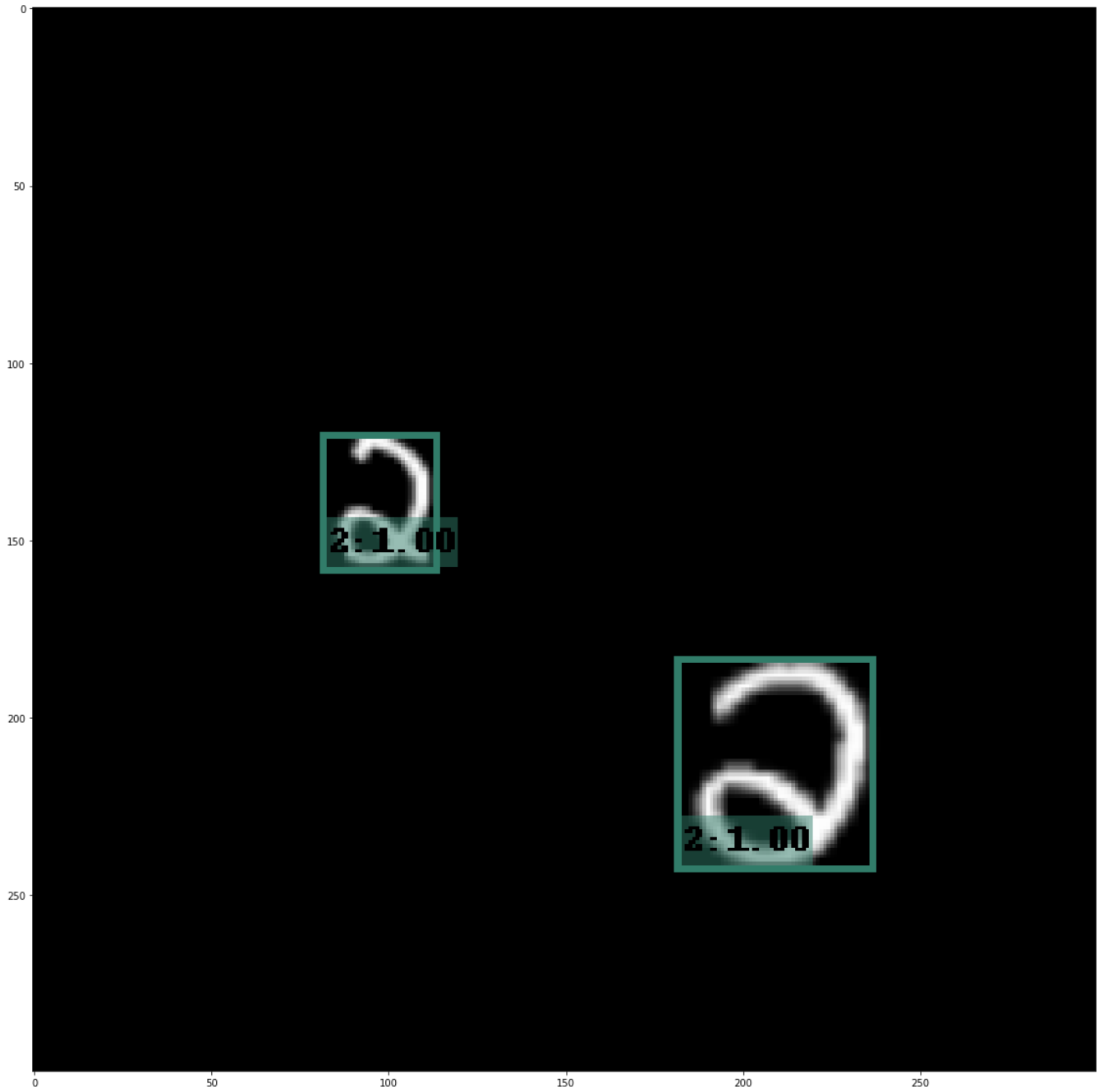
mAP: 0.9010

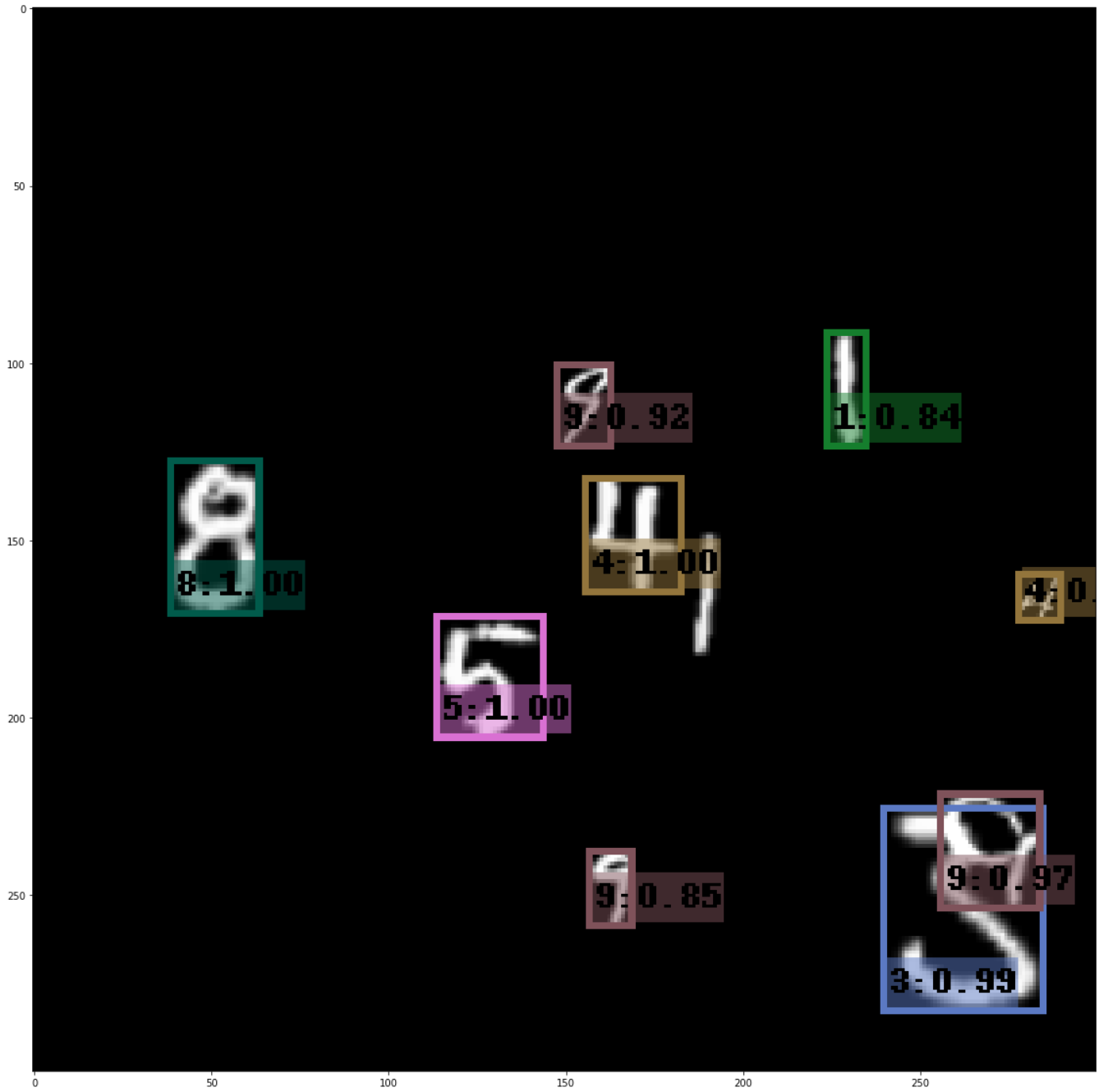
- 0 : 0.9082
- 1 : 0.8801
- 2 : 0.8957
- 3 : 0.9037
- 4 : 0.9041
- 5 : 0.9034
- 6 : 0.9045
- 7 : 0.8985
- 8 : 0.9063
- 9 : 0.9051

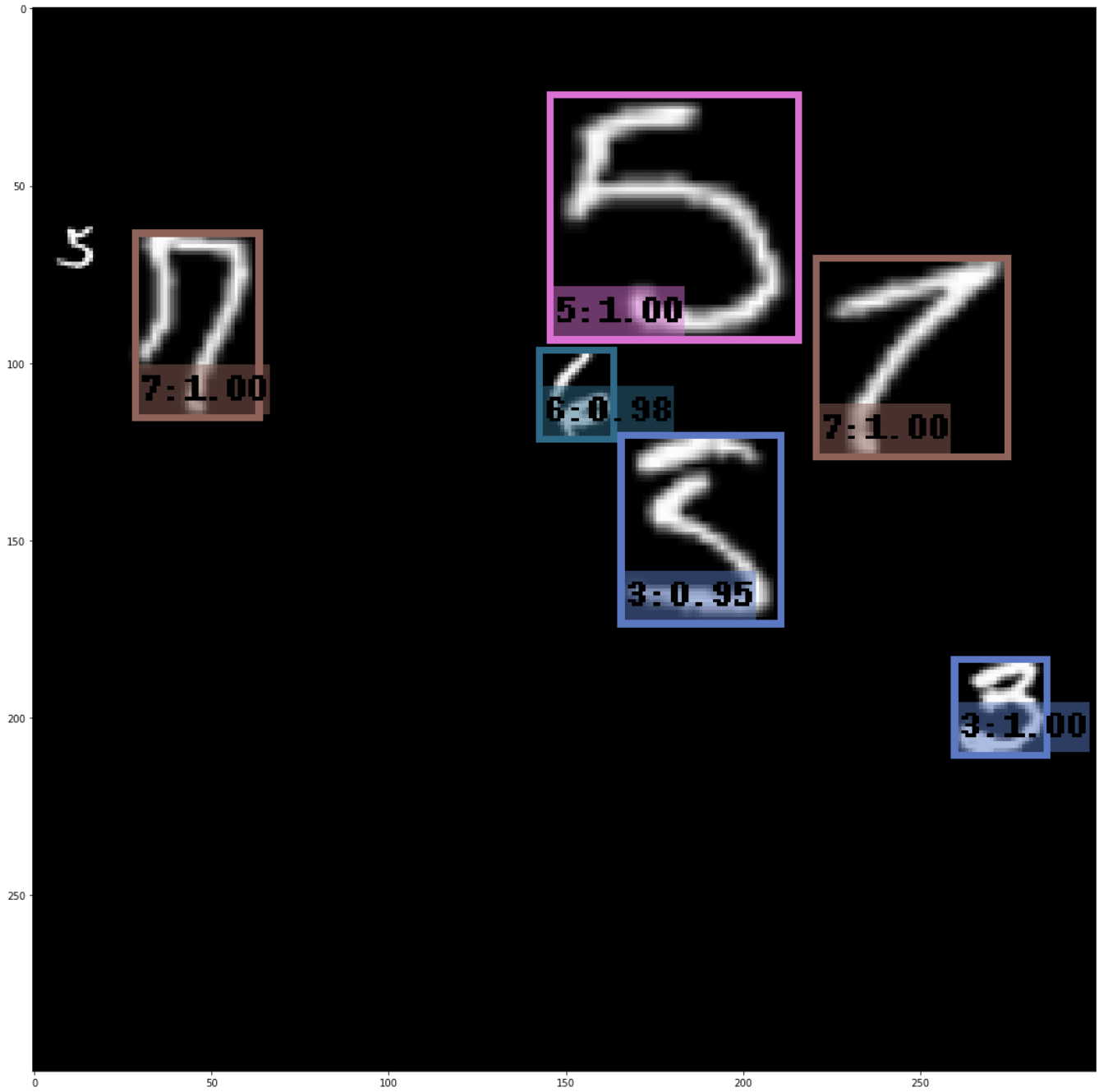
Result:

Overview:

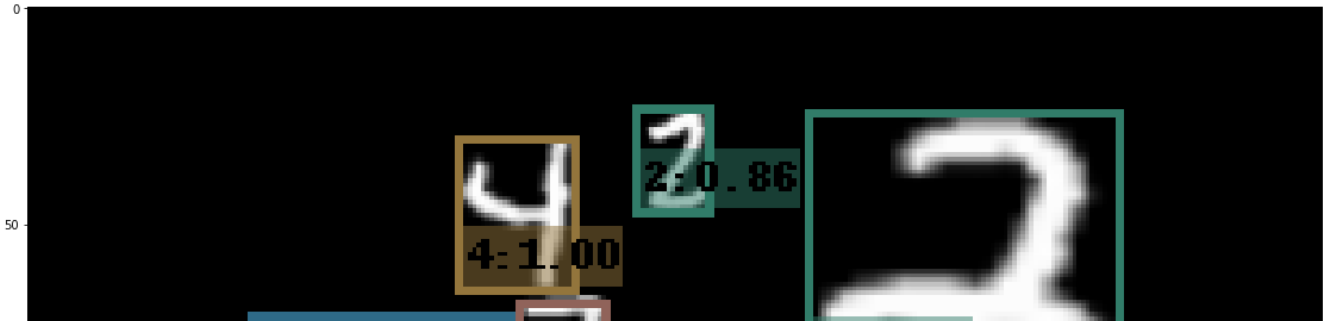
Up close:


2: 1.00
2: 1.00

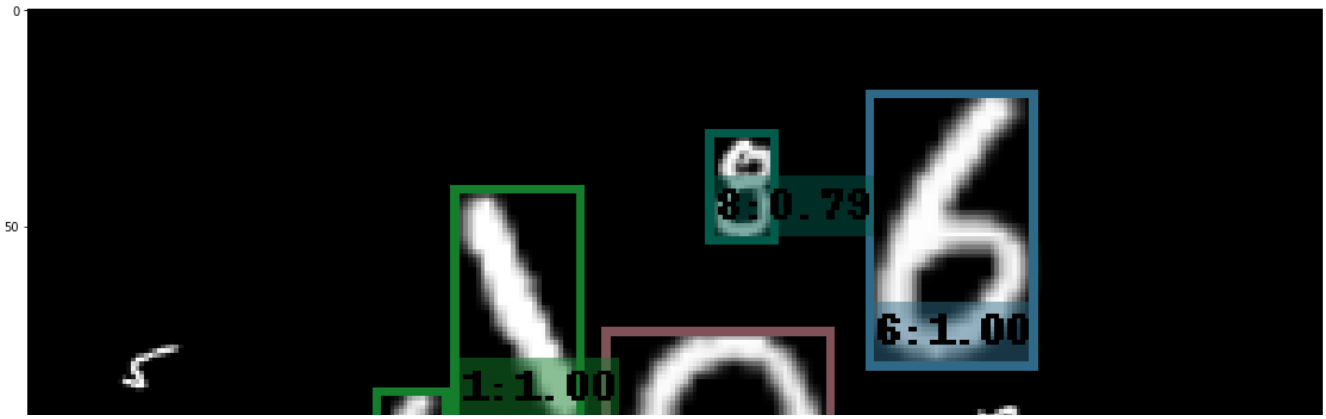

9: 0.92
1: 0.84
8: 1.00
4: 1.00
5: 1.00
9: 0.85
9: 0.97
3: 0.99

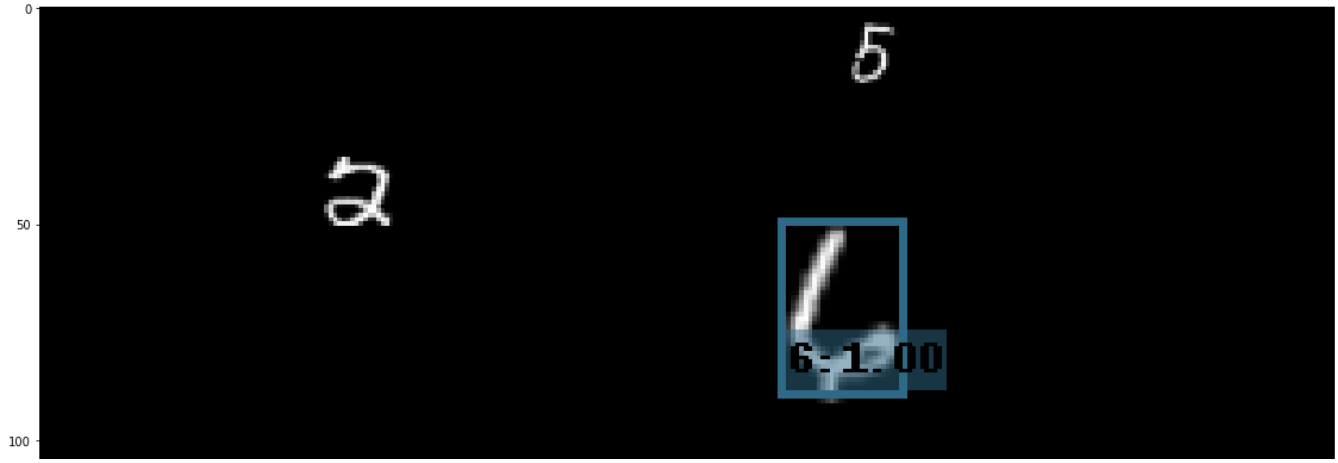

5: 1.00
7: 1.00
6: 0.98
7: 1.00
3: 0.95
3: 1.00

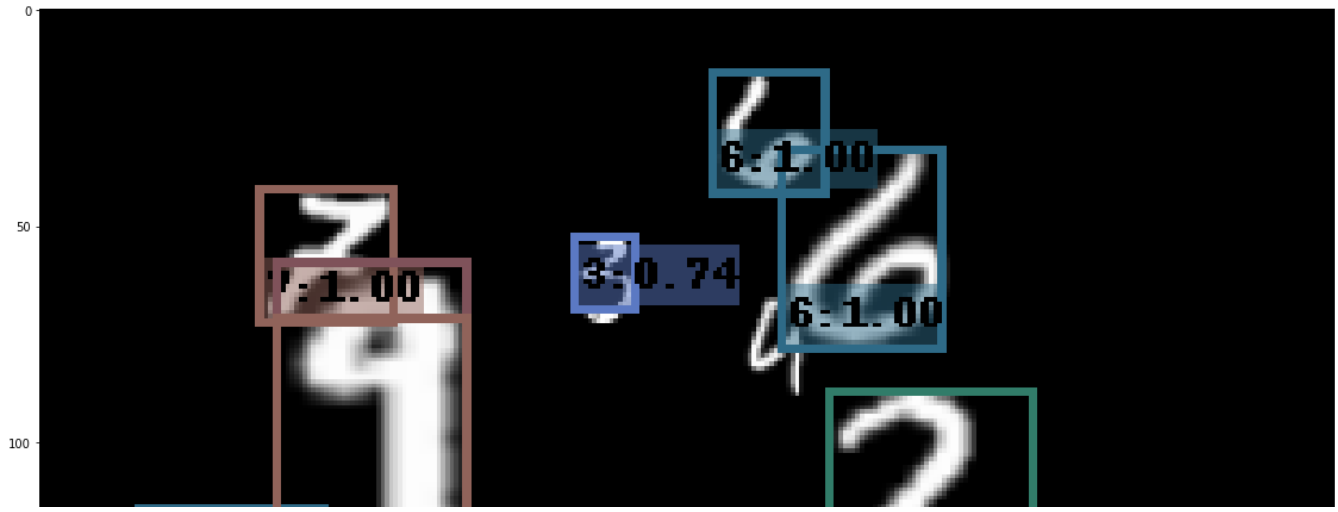

0

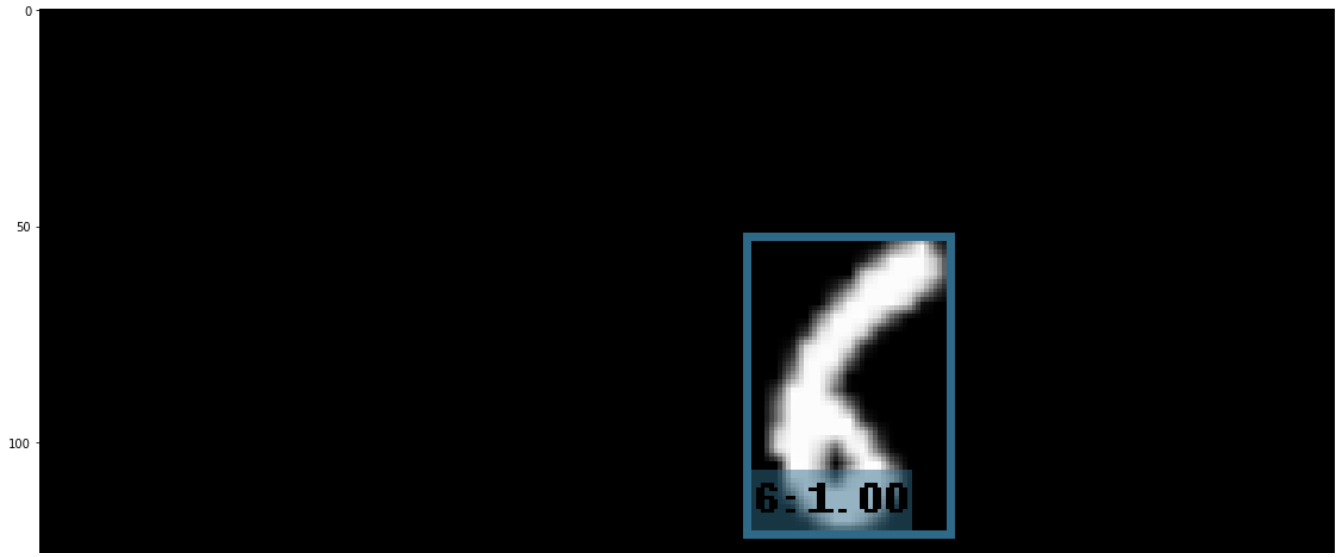

0
50
4: 1.00
2: 0.86

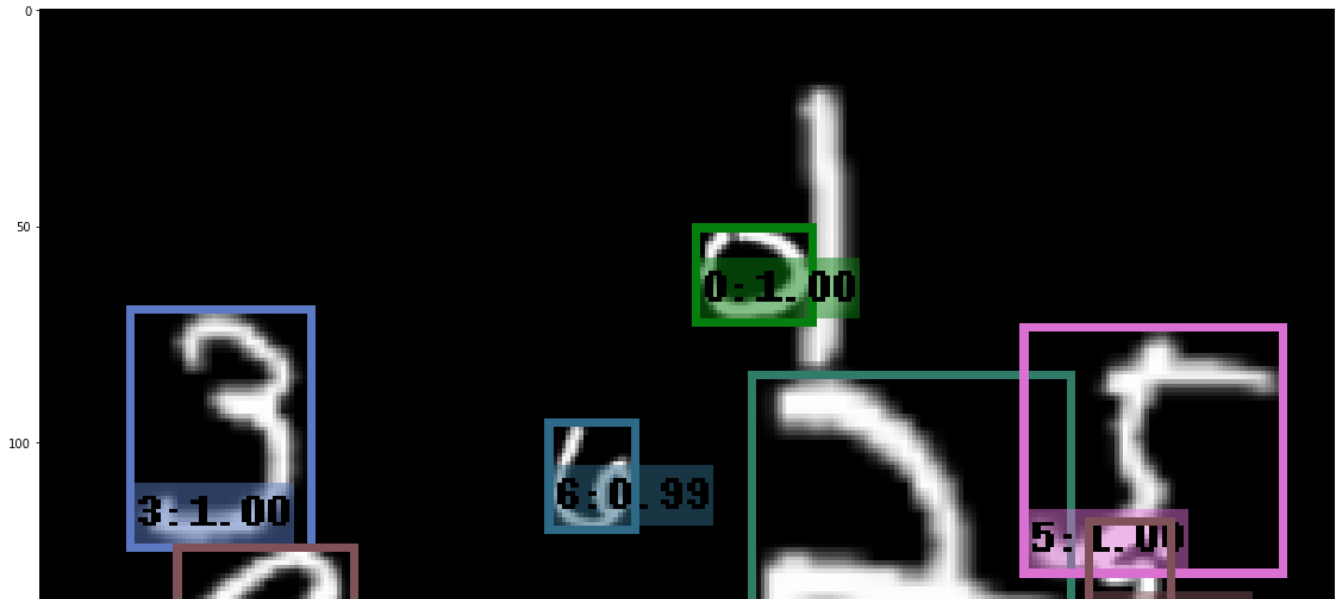

0
50

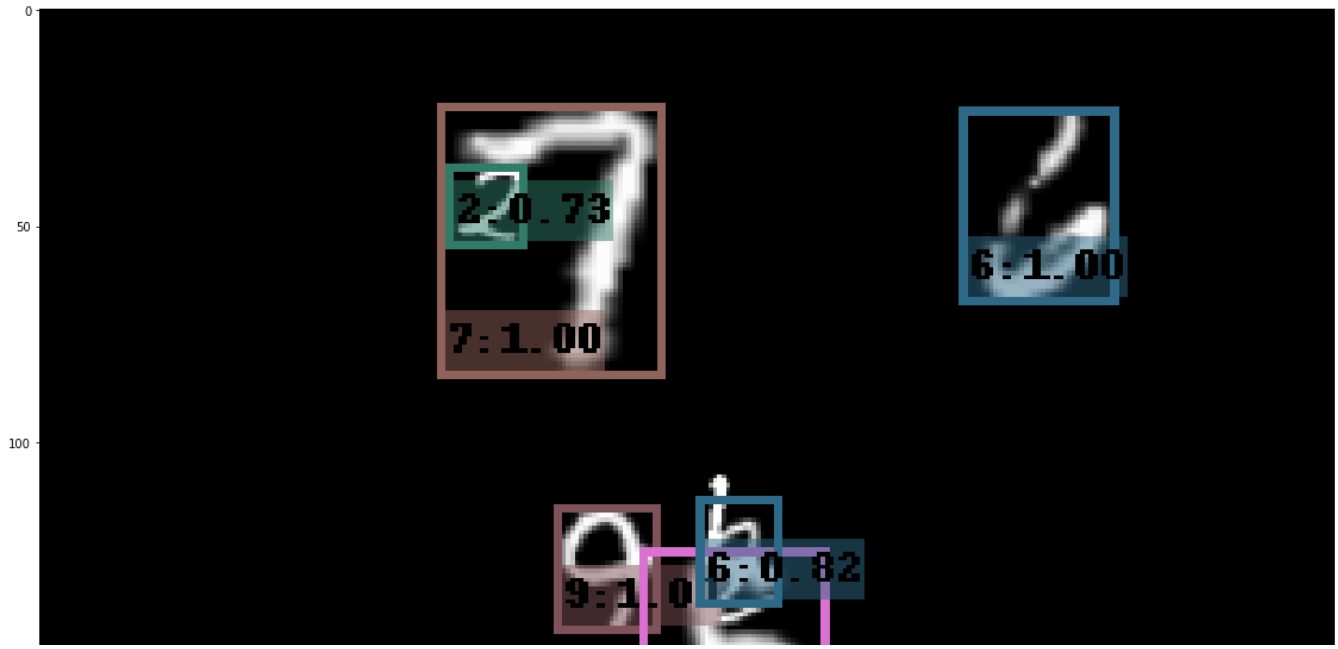

0
50
8: 0.79
6: 1.00
1: 1.00

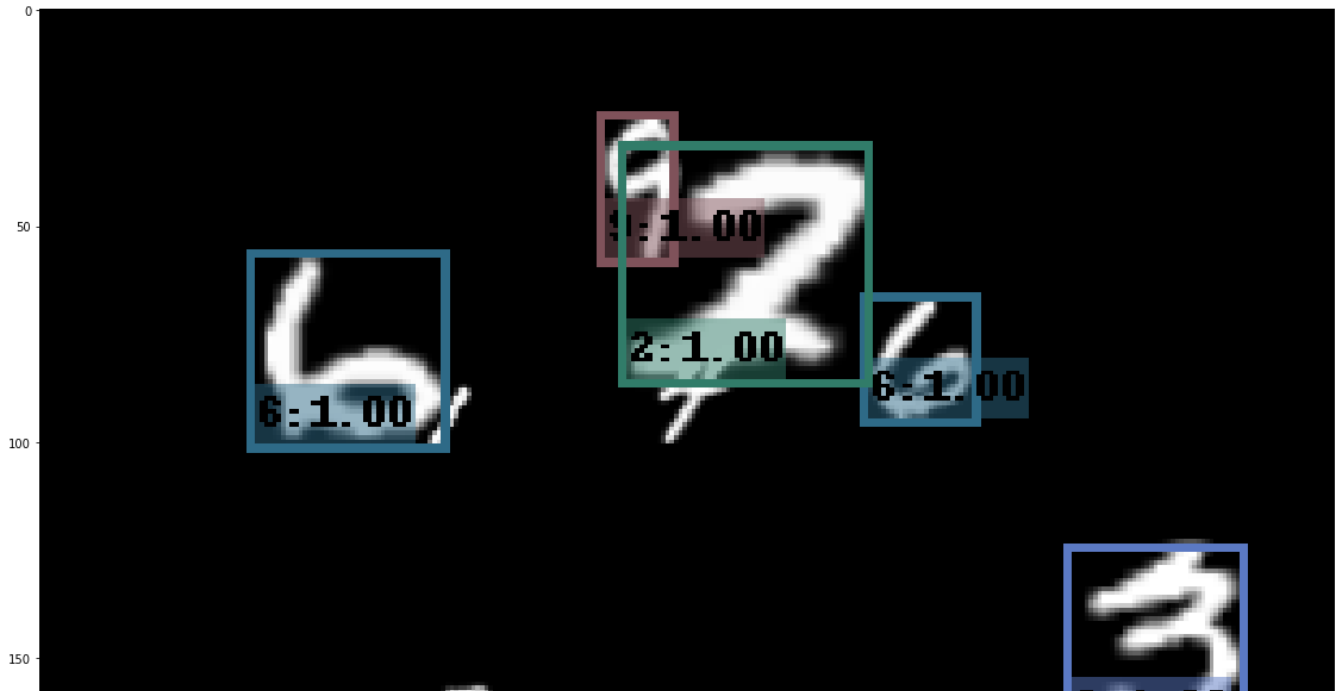

0
50
100
6_1_00

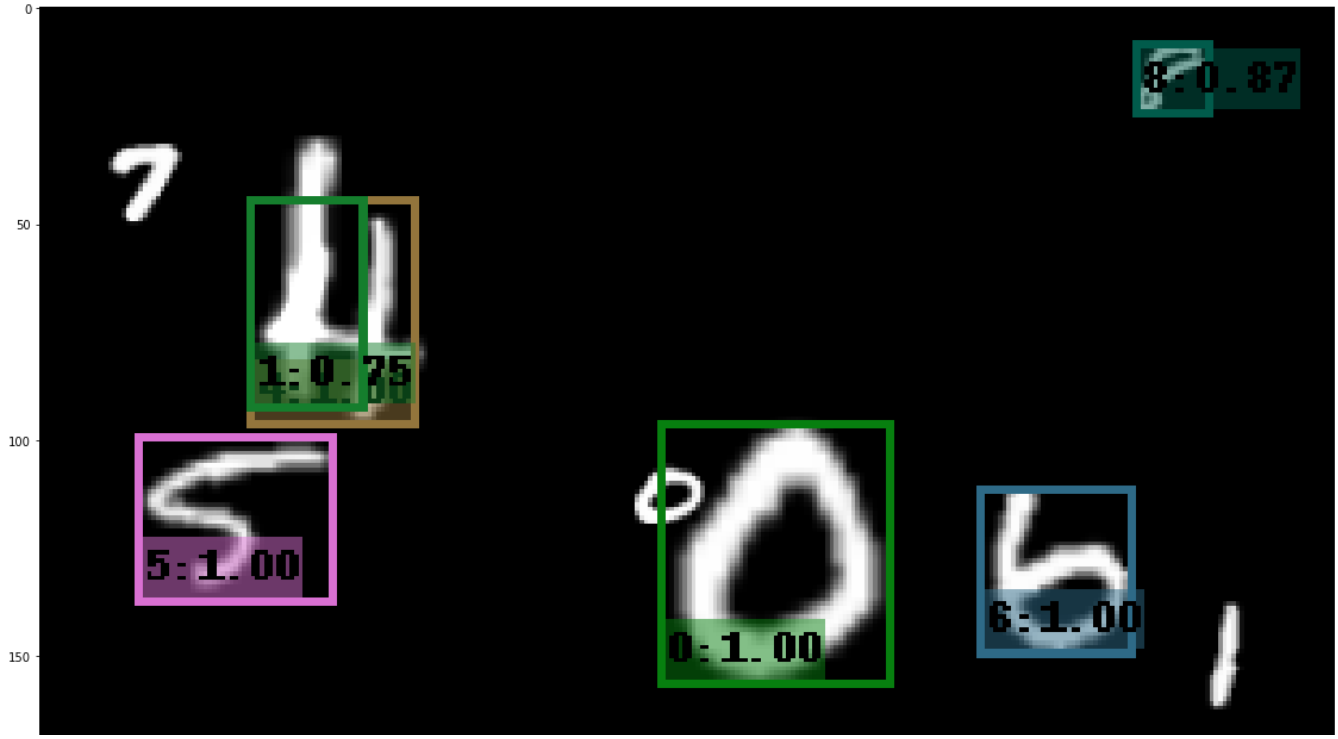

0
50
100
6:1.00
3:0.74
6:1.00


0
50
100
6:1.00


0
50
100
0: 1.00
3: 1.00
6: 0.99

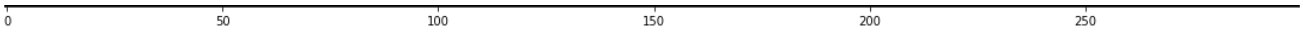

0
50
100
2:0.73
7:1.00
6:1.00
9:1.0
6:0.82


0
50
100
150
6: 1.00
2: 1.00
6: 1.00


0
50
100
150
8: 0.87
1: 0.75
5: 1.00
0: 1.00
6: 1.00

From this demo we see that the model especially struggles to classify small ones, twos, sevens and fives. We observe that the AP for 1, 2 and 7 are below average in training, so this resonnates with our prior knowledge. 5 is above average, but we see that the fives that are missed in this demo are somewhat unclear which may explain the misses.

Task 4f)



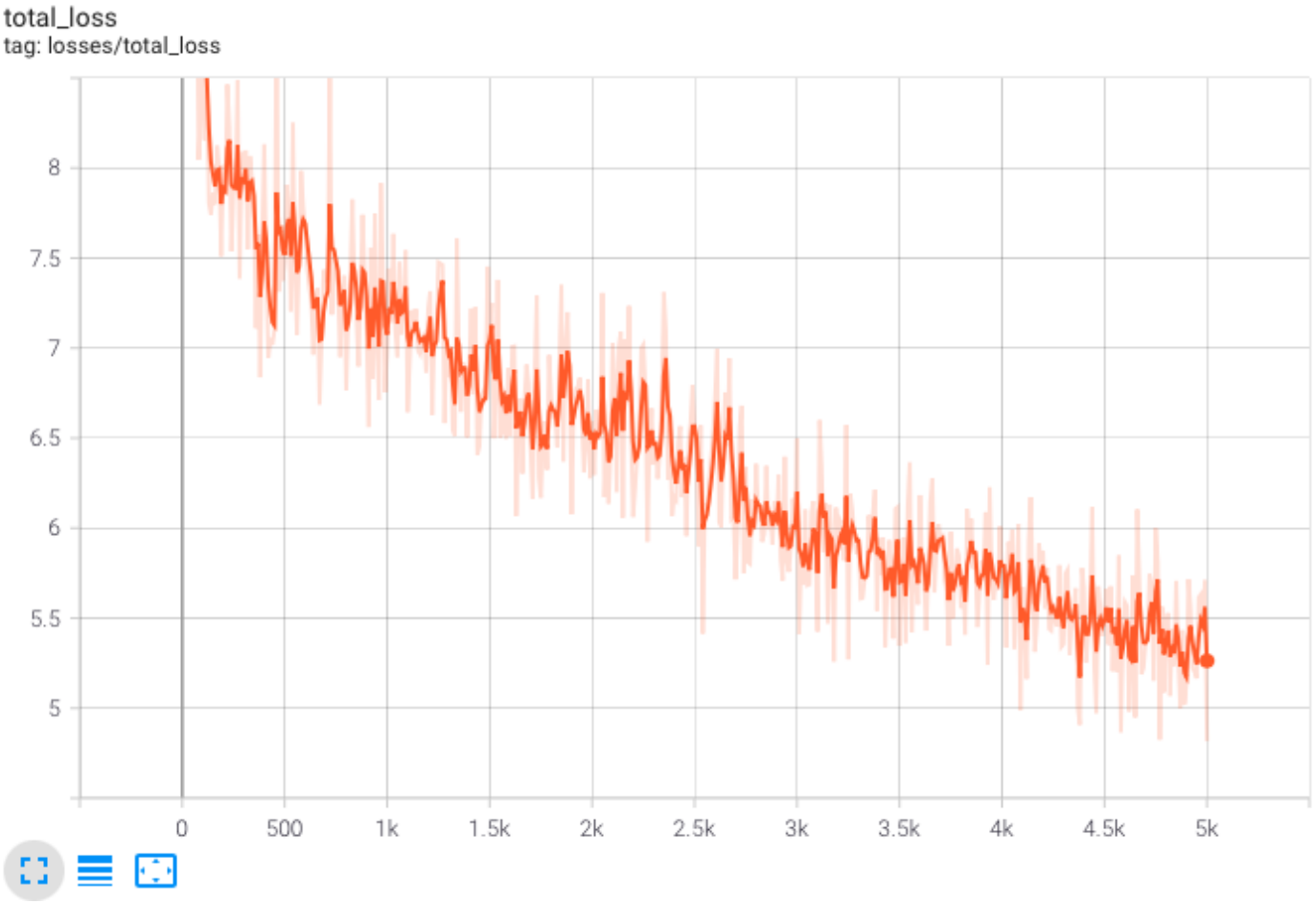
With all implementations from previous tasks removed, we trained the VGG16-model on the PASCAL VOC dataset.

Result:

```
2021-03-19 12:16:04,539 SSD.trainer INFO: Saving checkpoint to outputs/vgg_VOC/model_final.pth
2021-03-19 12:16:05,757 SSD.trainer INFO: Total training time: 3:49:05 (2.7490 s / it)
2021-03-19 12:16:05,760 SSD INFO: Start evaluating...
2021-03-19 12:16:05,766 SSD.inference INFO: Evaluating voc_2007_test dataset(4952 images):
100% 496/496 [05:10<00:00,  1.60it/s]
2021-03-19 12:21:31,542 SSD.inference INFO: mAP: 0.1962
aeroplane       : 0.2500
bicycle         : 0.1339
bird            : 0.1186
boat            : 0.0385
bottle          : 0.0125
bus             : 0.2480
car             : 0.4594
cat             : 0.2900
chair           : 0.1232
cow             : 0.2043
diningtable     : 0.1042
dog             : 0.2689
horse           : 0.3650
motorbike       : 0.2307
person          : 0.3602
pottedplant     : 0.1008
sheep           : 0.1620
sofa            : 0.1388
train           : 0.1755
tvmonitor       : 0.1395
```

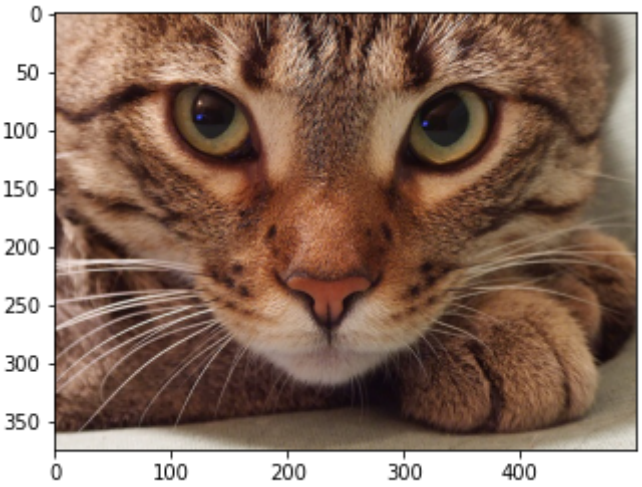
The final mAP after 5000 iterations for the validation set is 19,62%

- The total loss is plotted below:

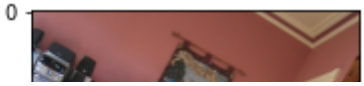


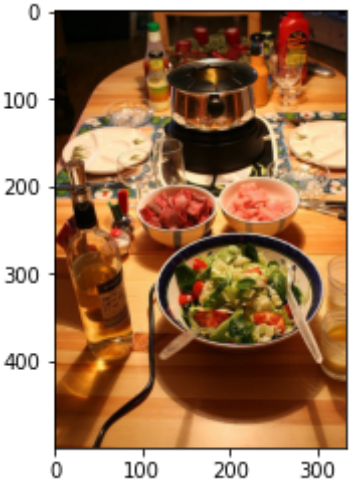
Test of model on images using demo:


0
50


0
50
100
150
200
250
300
350
0
100
200
300
400


0
50
100
150
200
250
300
350
0
100
200
300
400





The model was unable to identify any objects when running the demo. This was surprising, but the mAP obtained was quite low which can explain why we do not detect any objects.