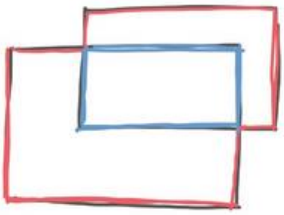
Assignment 4 - group 98

Task 1

1(a)

IoU - Intersection over union



Area of the union - area the two boxes are covering

Area of the overlap

$IoU = \frac{A_{overlap}}{A_{union}}$ ← says how much the boxes are overlapping - 1.00 the boxes covers exacly the same area

1(b)

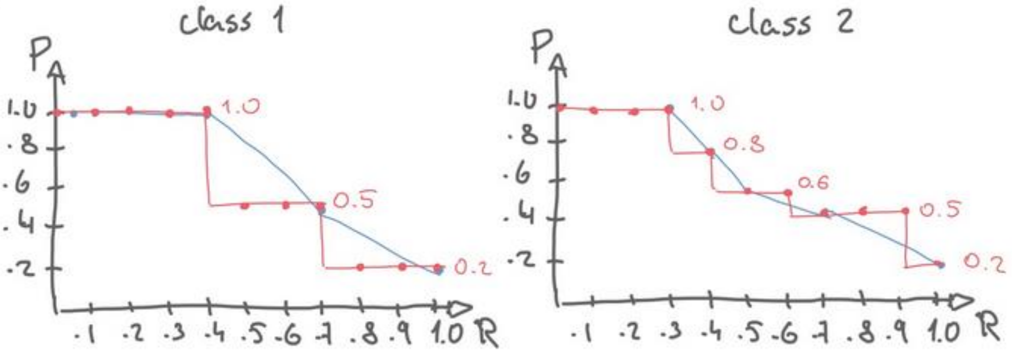
$Precision = \frac{TP}{TP+FP}$ – how many positives are true positive

$Recall = \frac{TP}{TP+FN}$ – how many positives out of all that should be positive

TP - true positive - getting "true" result when we should get "true" (corrcet)

FP - false positive - getting "true" result when we should get "false" result

1c)

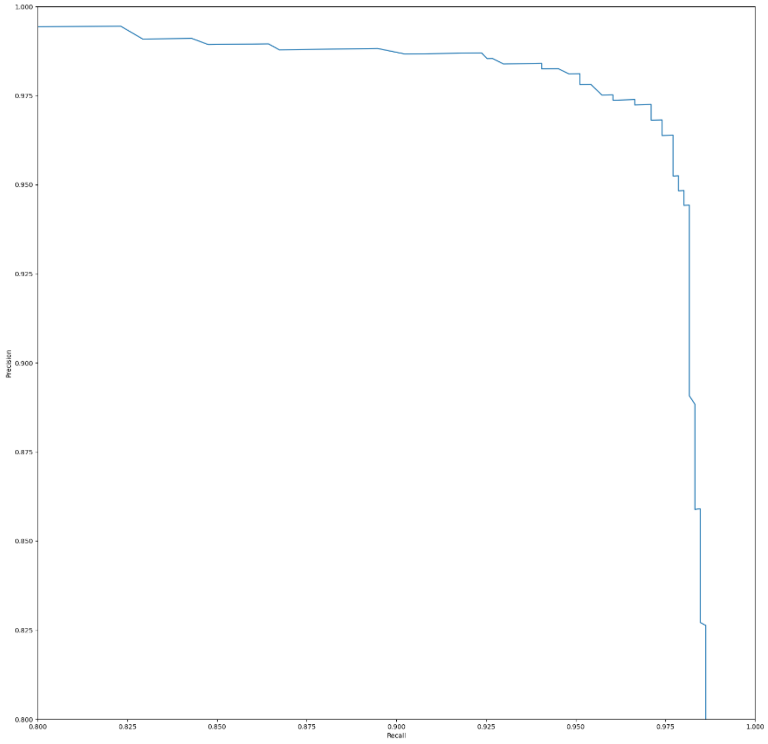


$$AP_1 = \frac{1}{11} \cdot (5 \cdot 1 + 3 \cdot 0.5 + 3 \cdot 0.2) = 0.645$$

$$AP_2 = \frac{1}{11} \cdot (4 \cdot 1 + 0.8 + 2 \cdot 0.6 + 3 \cdot 0.5 + 0.2) = 0.7$$

$$mAP = \frac{1}{2} \cdot (0.645 + 0.7) = \underline{\underline{0.673}}$$

Task 2



Task 3

a)	To filter out overlapping boxes SSD uses “non-maximum suppression” (nms)

b)	Leftmost layers with the highest resolution are only able to detect small objects. Lower resolution layers, deeper in the network, are responsible for detecting bigger objects. Statement is FALSE.

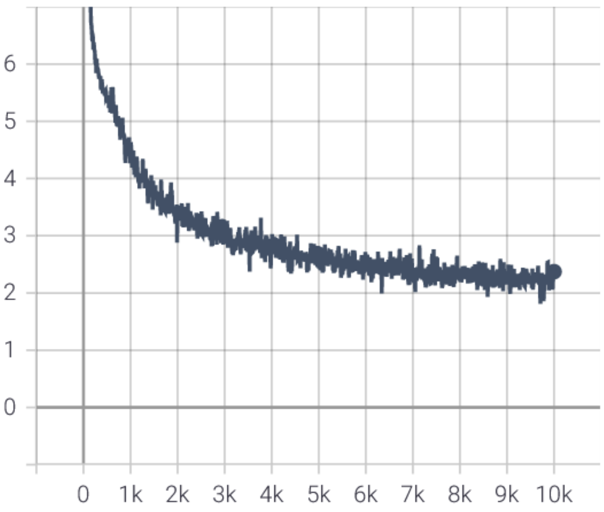
c)	Box with different aspect ratios can be matched to more different real-life object that will naturally have varying shapes - aspect ratios. This way predicted boxes will more likely match the ground truth boxes

d)	YOLO uses single scale feature map, while SSD uses multi scale feture maps for detection.

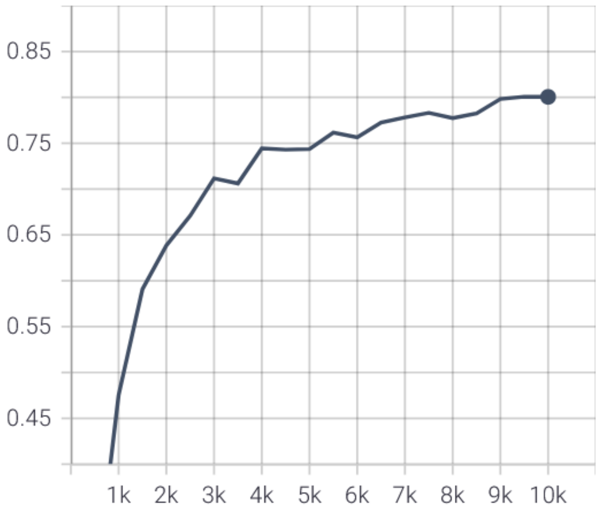
e)	w x h x 6 = 7776

Task 4

4b



Total Loss



mAP

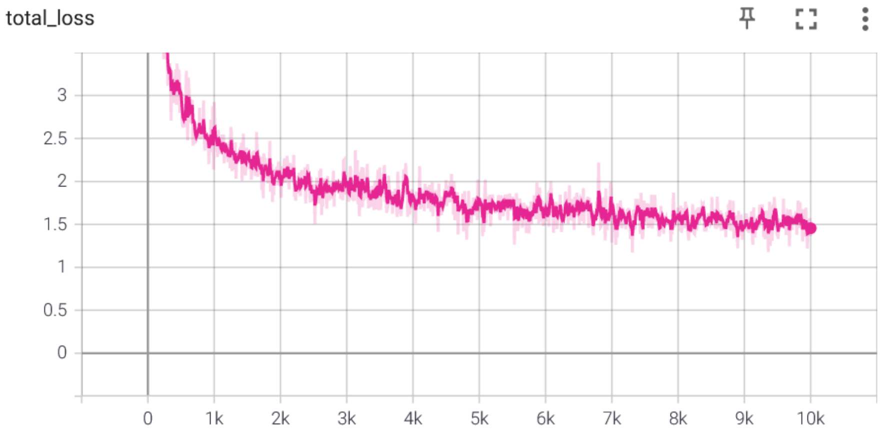
- final mAP (10k steps) = 0.80
- mAP (6k steps) = 0.75

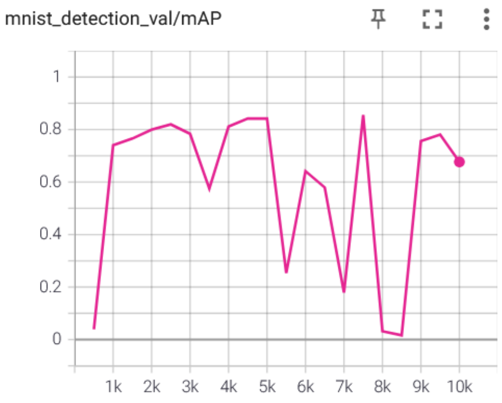
4c

The main improvements to the network was batch normalization that resulted in mAP=0.85 after 5k iterations. Other small changes were to change the optimizer to AdamW, reduce learning rate and reduce the batch size. I was however not happy with the results, since the mAP becomes very unstable after 5k iterations. I continued to improve the network and results can be seen in the next task, where the network reached mAP=0.9.

Configs:

- `Optimizer: AdamW`
    - `lr = 1e-3`
    - `amsgrad = True`
- `batch size = 10`
- `OUT_CHANNELS = [128, 128, 256, 128, 128, 512]`





Is Output	Layer Type	Number of Filter	Stride	Kernel size
	Conv2d MaxPool2d ReLU Batchnorm2d	32	1	3
	Conv2d MaxPool2d ReLU Batchnorm2d	64	1	3
	Conv2d ReLU Batchnorm2d	64	1	3
Yes – res: 38x38	Conv2D	Ouput_channels[0]	2	3
	ReLU Batchnorm2d			
	Conv2d ReLU Batchnorm2d	128	1	5
	Conv2d ReLU Batchnorm2d	256	1	5
Yes – res: 19x19	Conv2D	Ouput_channels[1]	2	3
	ReLU Batchnorm2d			
	Conv2d ReLU Batchnorm2d	256	1	3
	Conv2d ReLU Batchnorm2d	512	1	3
Yes – res: 9x9	Conv2D	Ouput_channels[2]	2	3
	ReLU Batchnorm2d			
	Conv2d ReLU Batchnorm2d	128	1	3
Yes – res: 5x5	Conv2D	Ouput_channels[3]	2	3
	ReLU Batchnorm2d			
	Conv2d ReLU Batchnorm2d	128	1	3
Yes – res: 3x3	Conv2D	Ouput_channels[4]	2	3
	ReLU Batchnorm2d			
	Conv2d ReLU	128	1	3
Yes – res: 1x1	Conv2D	Ouput_channels[5]	1	3

4d

To achieve 0.90 mAP the network from previous task has been redesigned a bit. Main inspiration here was the original VGG19 network. The main difference is the it has been added more convolutional layers at the begging, at the resolutions 150x150 and 75x75. This should help to detected small scaled numbers. The complete network is described in a table on the next page. I addition I am now using decreasing learning rate, which had improved end result and some learning stability. After suggestion from a classmate I have also changed PIXEL _MEAN and PIXEL_STD, which also improved the result slightly. No data augmentation was used. Under the training process the mAP is varying a bit. This could be reduced by using dropout and lower learning rate, which I might try later, but I didn't want to worsen the end result. The final mAP was 0.9013. This is this network that was used for the demo in the next task.

Configs:

- `Optimizer: Adam`
  - `lr = 1e-3`
  - `amsgrad = True`
- `batch size = 5`
- `Input:`
  - `IMAGE_SIZE: [300, 300]`
  - `PIXEL_MEAN: [0.485,0.456,0.406]`

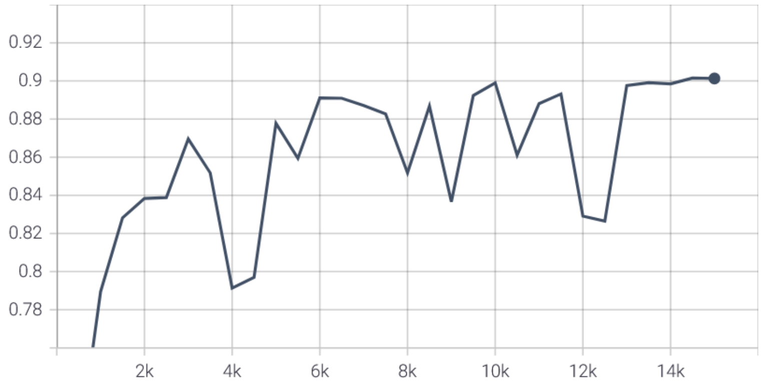


Figure 1 mAP

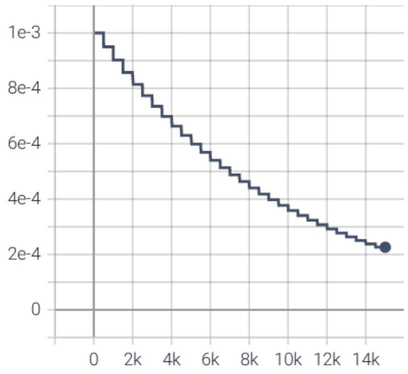
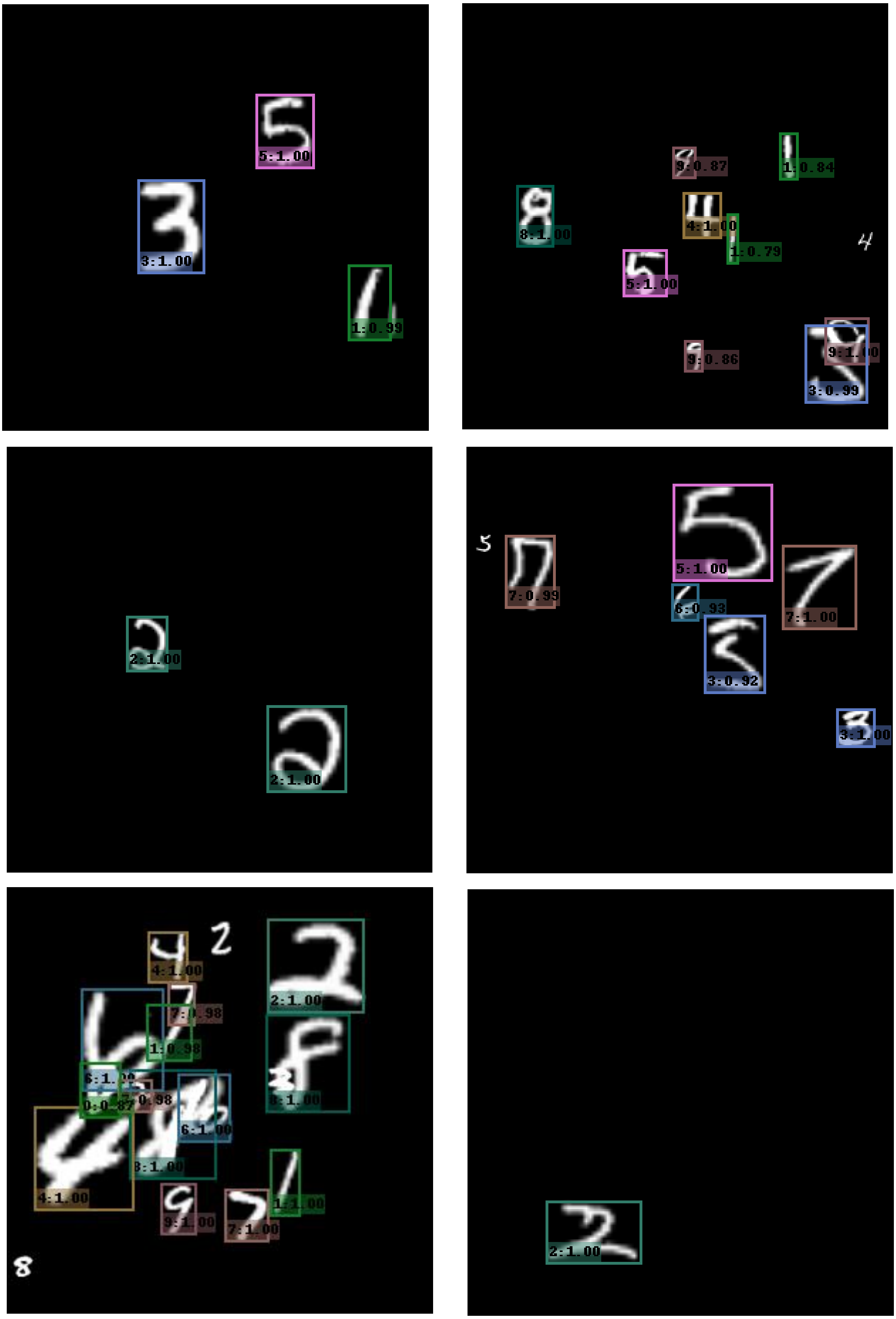
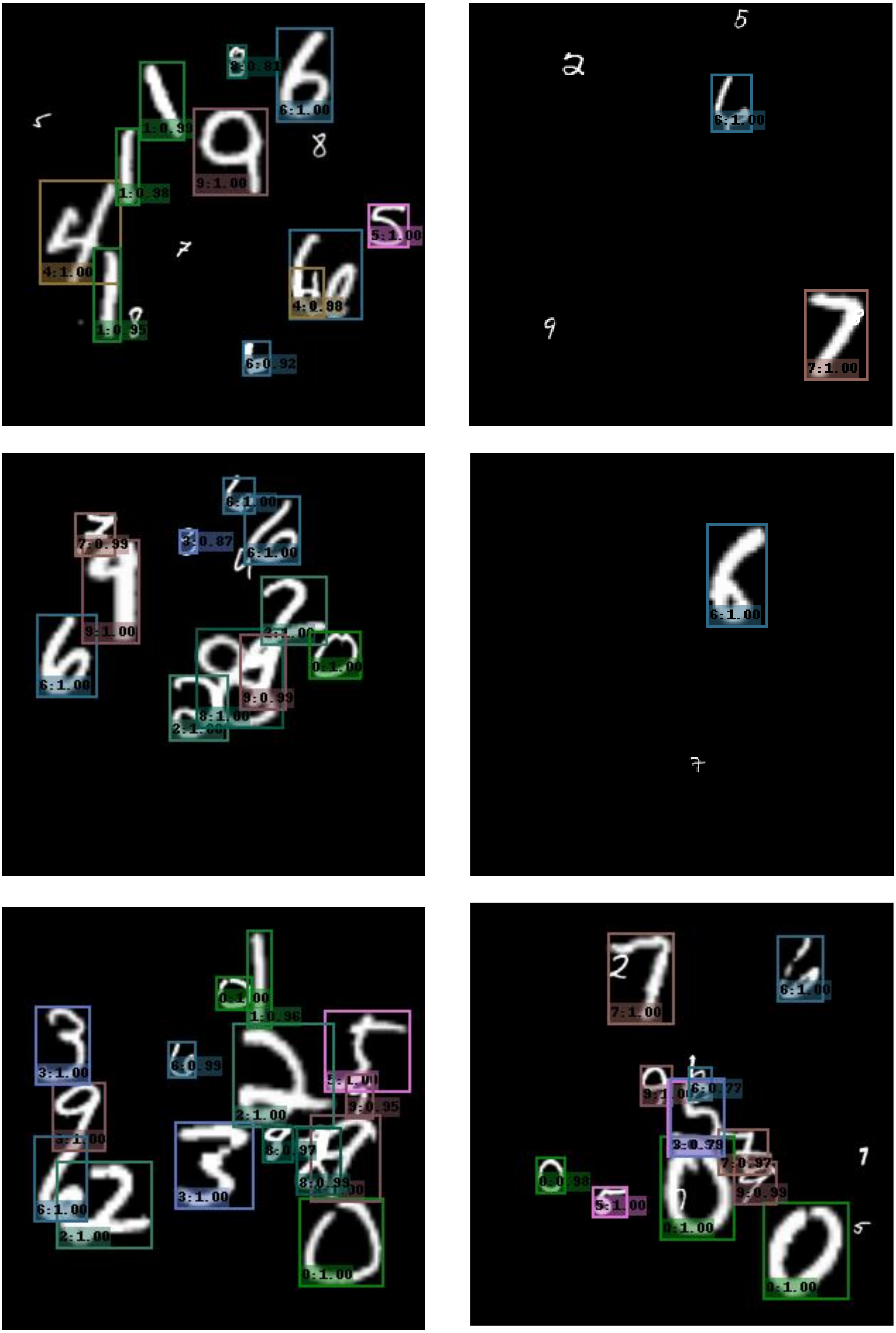


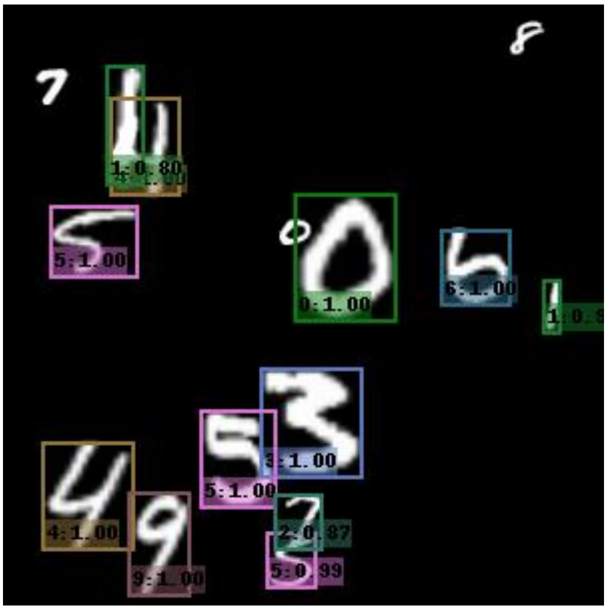
Figure 2 learning rate

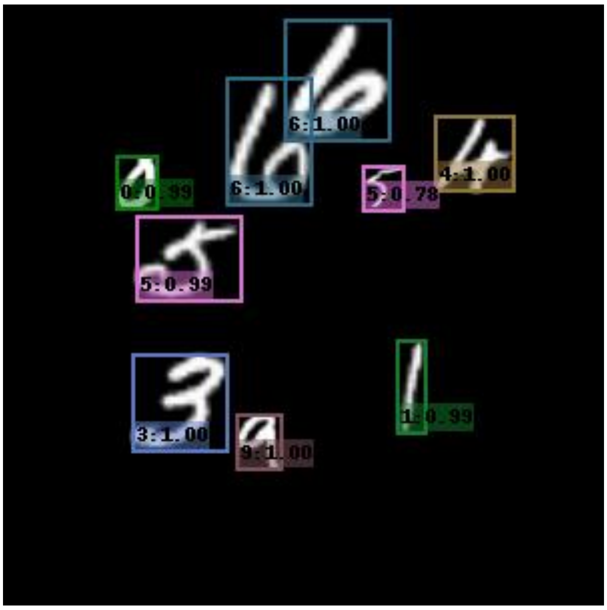
Is Output	Layer Type	Number of Filter	Stride	Kernel size
	Conv2d ReLU Batchnorm2d	32	1	3
	Conv2d ReLU Batchnorm2d	32	1	3
150x150	Conv2d MaxPool2d ReLU Batchnorm2d	32	1	3
	Conv2d ReLU Batchnorm2d	64	1	3
	Conv2d ReLU Batchnorm2d	64	1	3
75x75	Conv2d MaxPool2d ReLU Batchnorm2d	64	1	3
	Conv2d ReLU Batchnorm2d	128	1	3
	Conv2d ReLU Batchnorm2d	128	1	3
Yes - res: 38x38	Conv2d ReLU Batchnorm2d	128	2	3
	Conv2d ReLU Batchnorm2d	256	1	3
	Conv2d ReLU Batchnorm2d	256	1	3
Yes - res: 19x19	Conv2d ReLU Batchnorm2d	256	2	3
Yes - res: 10x10	Conv2d ReLU Batchnorm2d	256	2	3
	Conv2d ReLU Batchnorm2d	256	1	3
Yes - res: 5x5	Conv2d ReLU Batchnorm2d	256	2	3
	Conv2d ReLU Batchnorm2d	512	1	3
Yes - res: 3x3	Conv2d ReLU Batchnorm2d	512	2	3
	Conv2d ReLU Batchnorm2d	512	1	3
Yes - res: 1x1	Conv2D	512	1	3

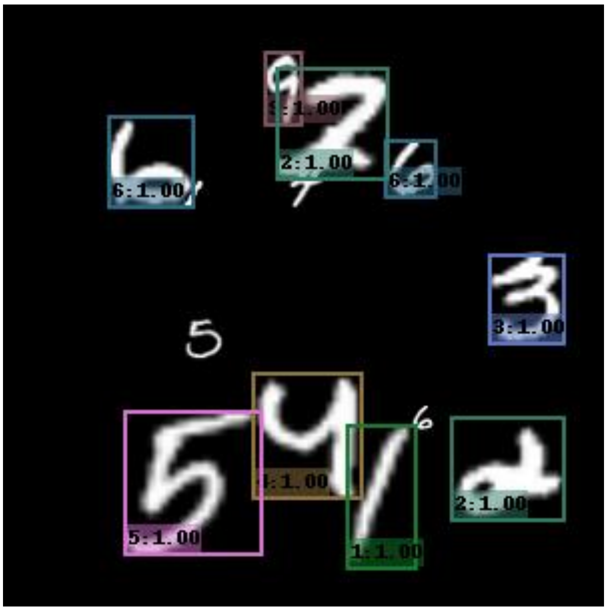
4e


5:1.00
3:1.00
1:0.99
9:0.87
1:0.84
8:1.00
4:1.00
1:0.79
4
5:1.00
9:0.86
9:1.00
3:0.99
2:1.00
2:1.00
5
5:1.00
7:0.99
6:0.93
7:1.00
3:0.92
3:1.00
4:1.00
2
2:1.00
7:0.98
1:0.98
6:1.00
0:0.87
0.98
6:1.00
8:1.00
8:1.00
4:1.00
9:1.00
7:1.00
1:1.00
8
2:1.00


8:0.81
6:1.00
1:0.99
9:1.00
1:0.98
5:1.00
4:1.00
4:0.98
1:0.95
6:0.92
6:1.00
7:1.00
6:1.00
7:0.99
3:0.87
6:1.00
9:1.00
2:1.00
0:1.00
6:1.00
9:0.99
8:1.00
2:1.00
6:1.00
0:1.00
1:0.96
3:1.00
6:0.99
5:1.00
9:0.95
2:1.00
9:1.00
8:0.97
6:1.00
3:1.00
8:0.99
2:1.00
0:1.00
7:1.00
6:1.00
9:1.00
6:0.77
3:0.79
7:0.97
0:0.98
9:0.99
5:1.00
0:1.00
0:1.00







We can see that model is strugling with small numbers, but those doesn’t seem to be detected by ssd at all. Most digits are however catagorized correctly.

4f

Final mAP of the network was 0.2172. It wasn’t able to detect many objects in the provided images either. The only image that got recognized was the one of a cat.

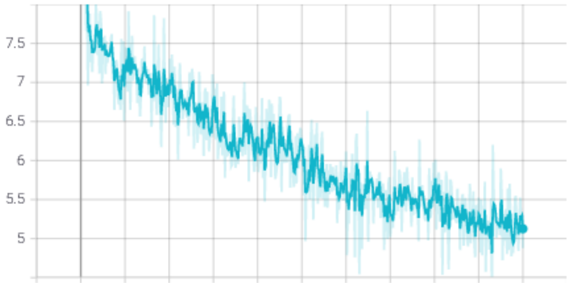


Figure 3 Total loss