

AssignmentReport-Group157

March 23, 2021

1 Assignment 1 Report

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: 1. Print the webpage (ctrl+P or cmd+P) 2. Export with latex. This is somewhat more difficult, but you'll get somewhat 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`.

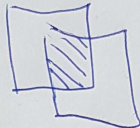
2 Task 1

2.1 task 1ab)

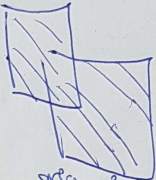
Assignment 4

Task 1

a) Intersection over Union is an evaluation metric used to measure the accuracy of an object detector and is the area of overlap over the area of union



Area of overlap



area of Union

b)

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

TP = true positives
FP = false positives
FN = false negatives

true positive is when a prediction of "positive" is True
and false positive is when a prediction of "positive" is not correct.

c) See PDF

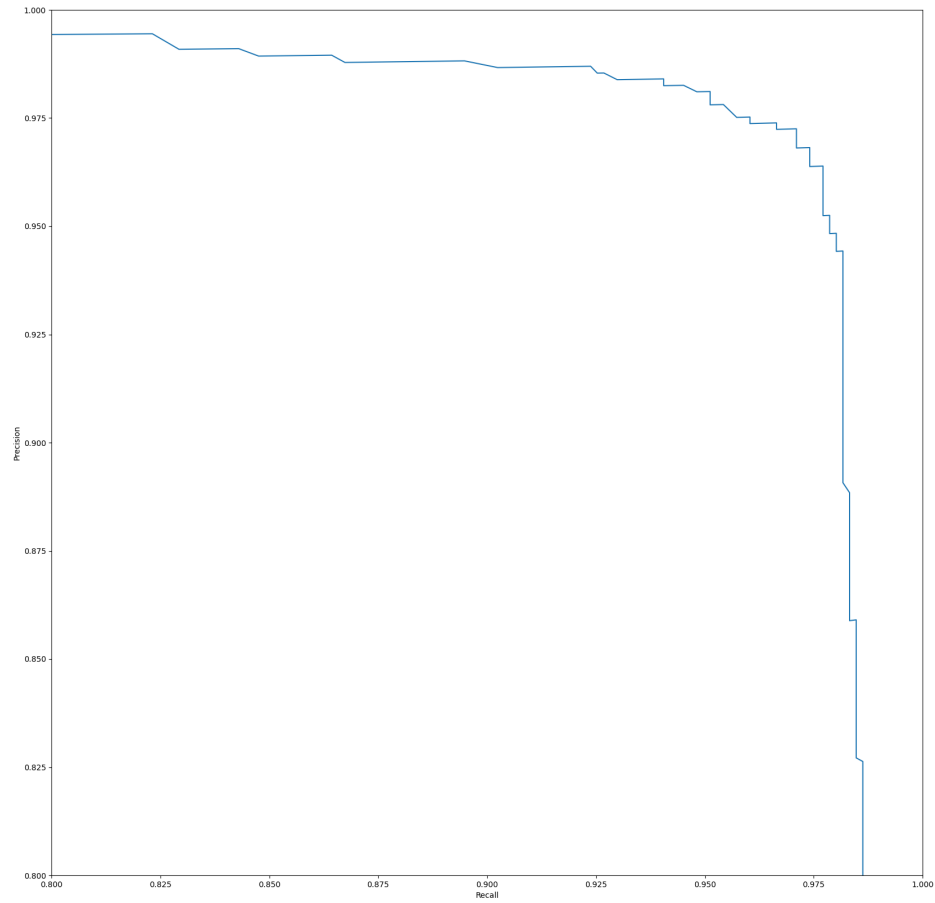
1c)

task

$$\text{mAP} = (0.73 + 0.675) / 2 = 0.7025$$

3 Task 2

3.0.1 Task 2f)



4 Task 3

4.0.1 Task 3a)

The SSD architecture produces a fixed-size number of bounding boxes and a score for each bounding box. When performing inference with SSD, we need to filter out a set of overlapping boxes. What is this filtering operation called? non-maximum suppression (NMS)

4.0.2 Task 3b)

The SSD architecture predicts bounding boxes at multiple scales to enable the network to detect objects of different sizes. Is the following true or false: Predictions from the deeper layers in SSD are responsible to detect small objects

Higher-resolution feature maps are responsible for detecting small objects. since the resolution decreases for each layer, predictions from the deeper layers are responsible for detecting bigger objects.

4.0.3 Task 3c)

SSD use k number of "anchors" with different aspect ratios at each spatial location in a feature map to predict class scores and 4 offsets relative to the original box shape. Why do they use different bounding box aspect ratios at the same spatial location?

Objects can have wildly varying shapes, as cars will often be more rectangular, and e.g. a football will have a squared bounding box. To get a bigger IoU between prediction and `gt_box`, the SSD will predict a number of objects with varying shapes at the same spatial location. The team behind SSD discovered that the model will fight between the different aspect ratios thus leaving the predictions unstable. To counter this, they start guesses based on "default" boxes that are preselected manually and carefully.

4.0.4 Task 3d)

What is the main difference between SSD and YOLOv1/v2 (The YOLO version they refer to in the SSD paper)? The main difference in the network model is that SSD adds several feature layers at the end of the base network, which predict the offsets to default boxes of different scales and aspect ratios and their associated confidences. The default bounding boxes discussed above are also not entered manually, but found through k-means clustering in the YOLO algorithm

4.0.5 Task 3e)

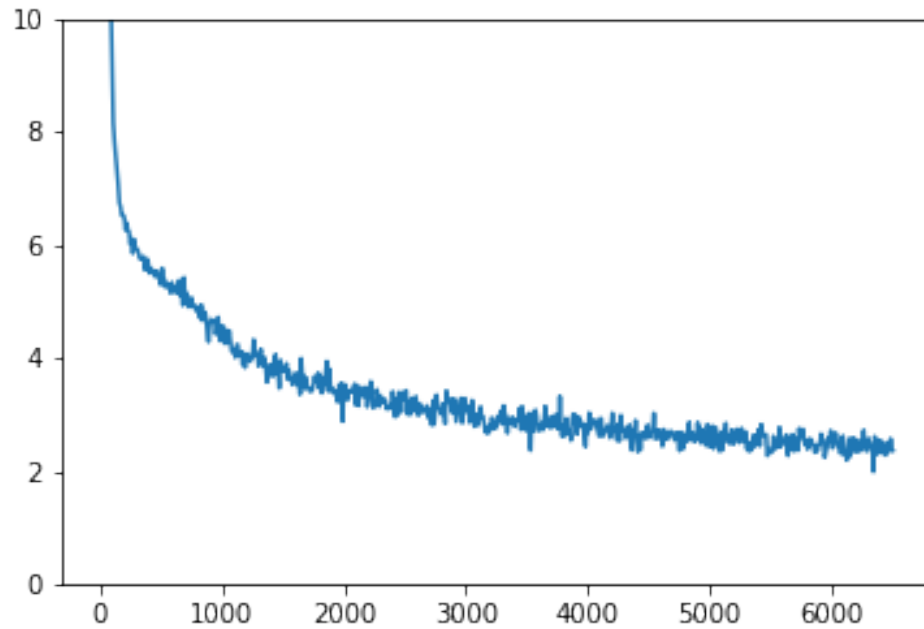
Given a SSD framework, where the first scale the network predicts at is at the last featuremap with a resolution of 38×38 ($H \times W$). For each anchor location, we place 6 different anchors with different aspect ratios. How many anchors boxes do we have in total for this feature map? We get $38 \times 38 \times 6 = 8664$ anchor boxes

4.0.6 Task 3f)

The network outlined in the previous subtask predicts at multiple resolutions, specifically 38×38 , 19×19 , 10×10 , 5×5 , 3×3 and 1×1 . It uses 6 different aspect ratios at each location in every feature map as anchors. How many anchors boxes do we have in total for the entire network? total anchors = $6 \times (38 \times 38 + 19 \times 19 + 10 \times 10 + 5 \times 5 + 3 \times 3 + 1 \times 1)$ so we get a grand total of 11640 anchor boxes

5 Task 4

5.1 Task 4b)



After 6000 iterations the SSD reached a mAP of 75.76%

```
2021-03-18 13:02:59,766 SSD.trainer INFO: iter: 005990, lr: 0.00200, total_loss: 2.496 (3.524)
2021-03-18 13:03:01,074 SSD.trainer INFO: iter: 006000, lr: 0.00200, total_loss: 2.475 (3.522)
2021-03-18 13:03:01,110 SSD.trainer INFO: Saving checkpoint to outputs/basic/model_006000.pth
2021-03-18 13:03:01,788 SSD.inference INFO: Evaluating mnist_detection_val dataset(1000 images)
2021-03-18 13:03:06,635 SSD.inference INFO: mAP: 0.7576
0          : 0.8055
1          : 0.6246
2          : 0.7436
3          : 0.7788
4          : 0.7984
5          : 0.7696
6          : 0.7826
7          : 0.7615
8          : 0.7819
9          : 0.7291
```

5.2 Task 4c)

From the last assignment we learned that batch normalization had the greatest effect on the accuracy of the data. This was thus implemented after the last convolution of each layer. Furthermore, the Adam optimizer was implemented as we saw this improved training time and accuracy as well from assignment 3. Learning rate was set to 1e-3. ReLU was replaced with LeakyReLU.

```
2021-03-18 17:09:00,364 SSD.trainer INFO: iter: 008980, lr: 0.00100, total_loss: 1.790 (2.260)
2021-03-18 17:09:01,063 SSD.trainer INFO: iter: 008990, lr: 0.00100, total_loss: 1.770 (2.259)
2021-03-18 17:09:01,730 SSD.trainer INFO: iter: 009000, lr: 0.00100, total_loss: 1.773 (2.259)
2021-03-18 17:09:01,756 SSD.trainer INFO: Saving checkpoint to outputs/improved_basic/model_009
2021-03-18 17:09:02,701 SSD.inference INFO: Evaluating mnist_detection_val dataset(1000 images)
2021-03-18 17:09:06,132 SSD.inference INFO: mAP: 0.8524
0           : 0.8794
1           : 0.7977
2           : 0.8367
3           : 0.8713
4           : 0.8644
5           : 0.8541
6           : 0.8627
7           : 0.8397
8           : 0.8778
9           : 0.8402
```

5.3 Task 4d)

Doing some analysis we think that one of the issue that needs to be addressed before you can achieve 90% mAP is that the manually coded sizes for the anchors are not specifically scaled or proportioned for this dataset. After looking at some of the images it becomes clear that a lot of the letters are really small and some are really big. the key here is that the size vary wildly from image to image. We therefore increased the size both smaller and bigger in the mnist yaml file. Batch size was increased to 32, max_iter to 15000 and threshold reduced to 0.45

MODEL:

NUM_CLASSES: 11

BACKBONE:

NAME: 'basic'

PRETRAINED: False

OUT_CHANNELS: [128, 256, 128, 128, 64, 64]

INPUT_CHANNELS: 3

PRIORS:

MAX_SIZES: [[38, 38], [90, 90], [153, 153], [207, 207], [264, 264], [312, 312]]

MIN_SIZES: [[16, 16], [38, 38], [90, 90], [153, 153], [207, 207], [264, 264]]

THRESHOLD: 0.45

INPUT:

IMAGE_SIZE: [300, 300]

DATASETS:

TRAIN: ("mnist_detection_train", "mnist_detection_val")

```

TEST: ("mnist_detection_val", )
SOLVER:
  MAX_ITER: 15000
  GAMMA: 0.1
  BATCH_SIZE: 32

```

```

OUTPUT_DIR: 'outputs/improved_basic'
DATASET_DIR: "/work/datasets"

```

Finally a mAP of 90.36% was reached after 4500 iterations. Final mAP after 15k iterations was 90.52%

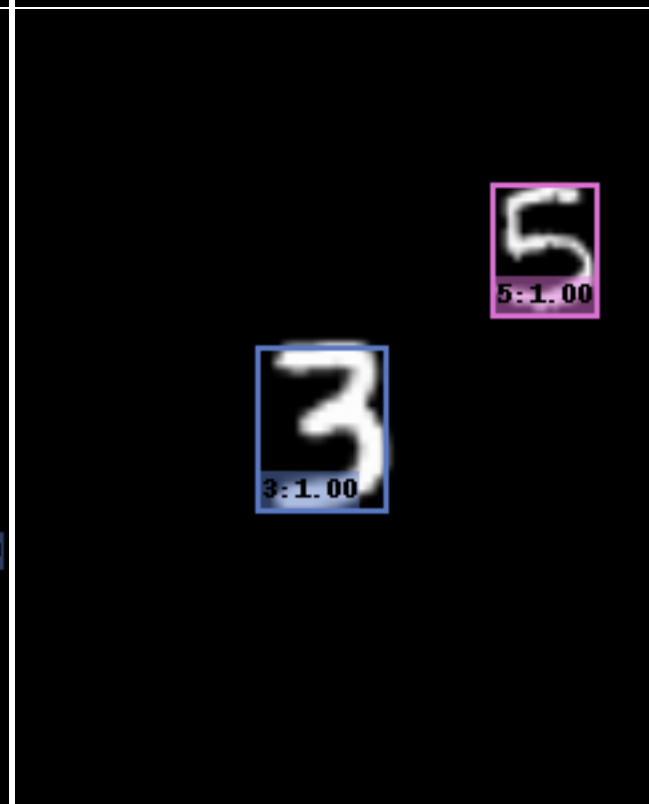
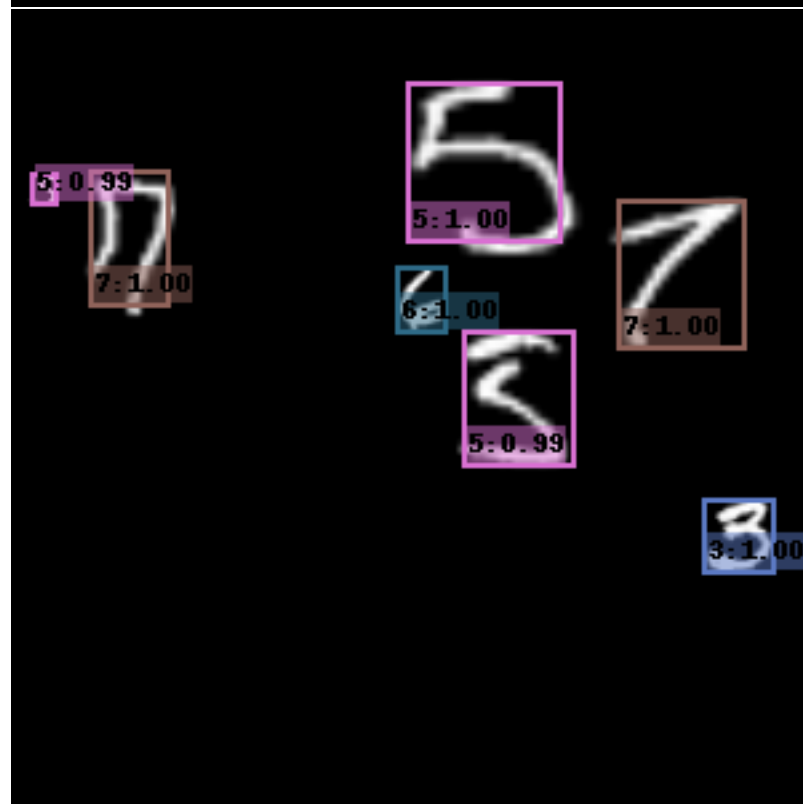
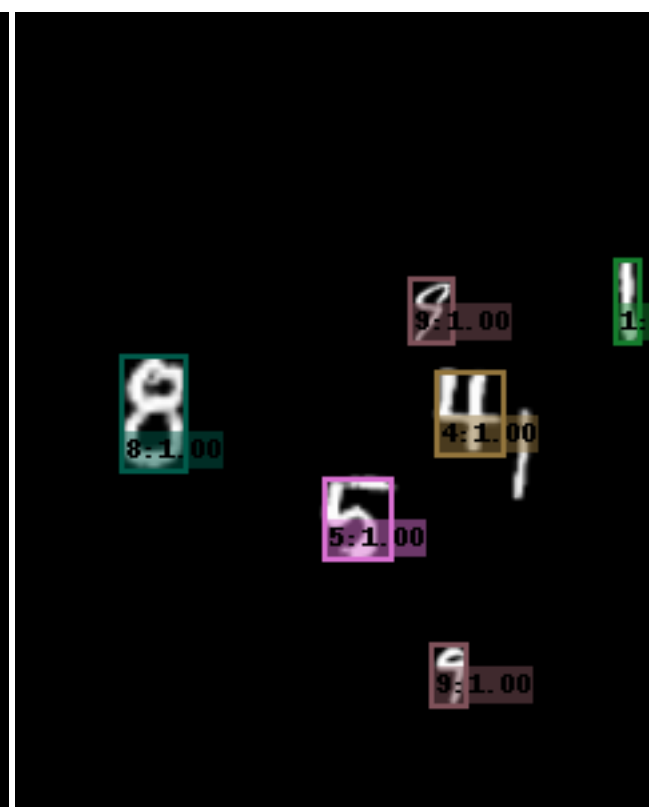
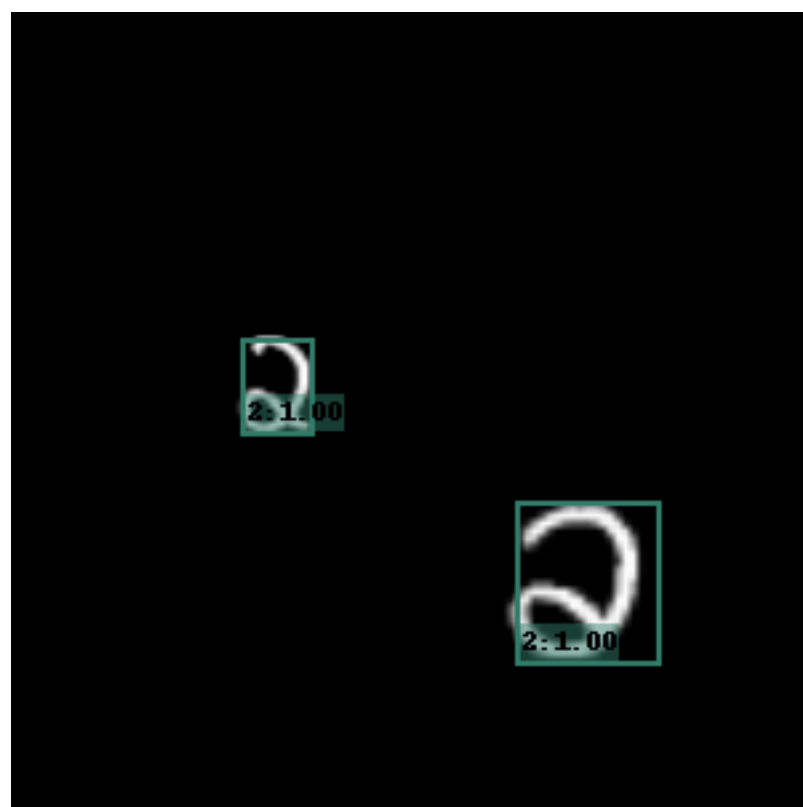
```

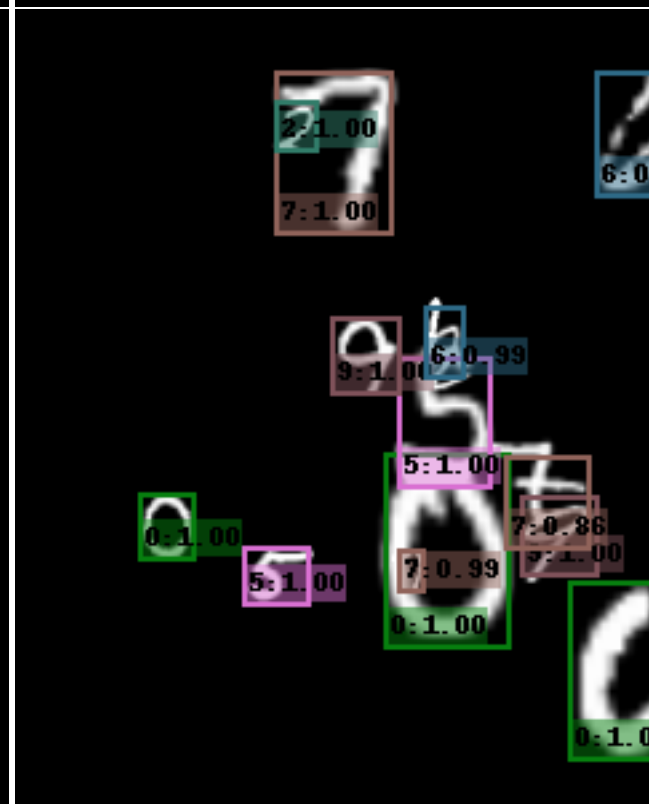
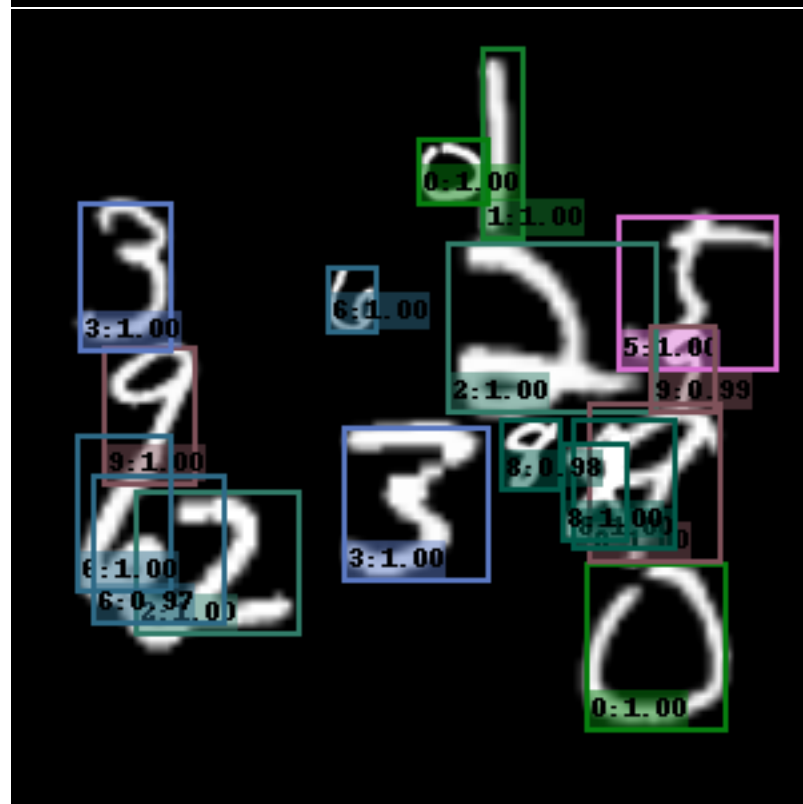
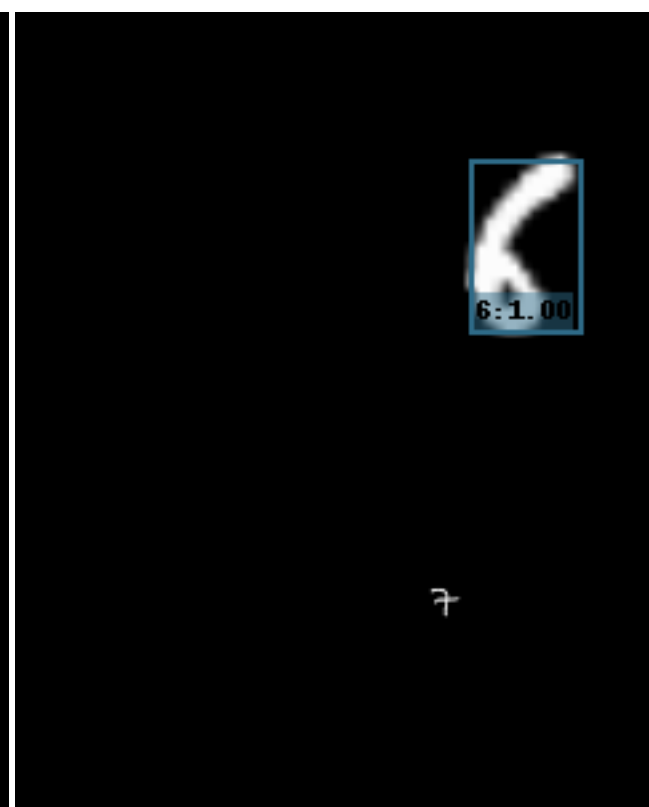
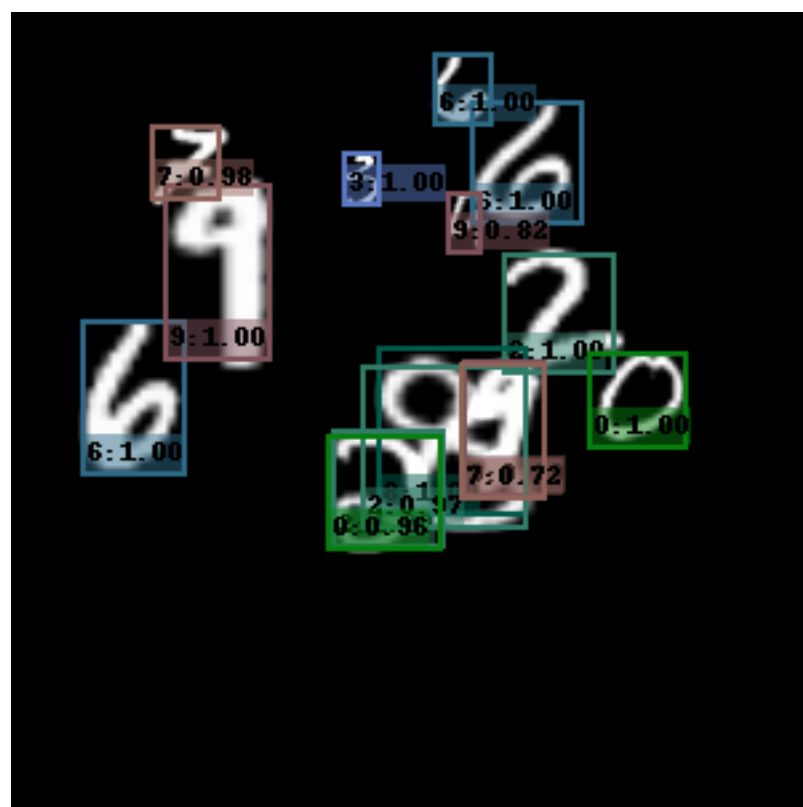
2021-03-18 20:56:36,594 SSD.trainer INFO: iter: 004470, lr: 0.00100, total_loss: 1.057 (1.633)
2021-03-18 20:56:37,902 SSD.trainer INFO: iter: 004480, lr: 0.00100, total_loss: 1.022 (1.631)
2021-03-18 20:56:39,256 SSD.trainer INFO: iter: 004490, lr: 0.00100, total_loss: 1.056 (1.630)
2021-03-18 20:56:40,599 SSD.trainer INFO: iter: 004500, lr: 0.00100, total_loss: 1.020 (1.629)
2021-03-18 20:56:40,665 SSD.trainer INFO: Saving checkpoint to outputs/improved_basic/model_00
2021-03-18 20:56:40,910 SSD.inference INFO: Evaluating mnist_detection_val dataset(1000 images,
100%|
2021-03-18 20:56:44,424 SSD.inference INFO: mAP: 0.9036
0          : 0.9081
1          : 0.8777
2          : 0.9074
3          : 0.9082
4          : 0.9065
5          : 0.9075
6          : 0.9069
7          : 0.9007
8          : 0.9081
9          : 0.9048

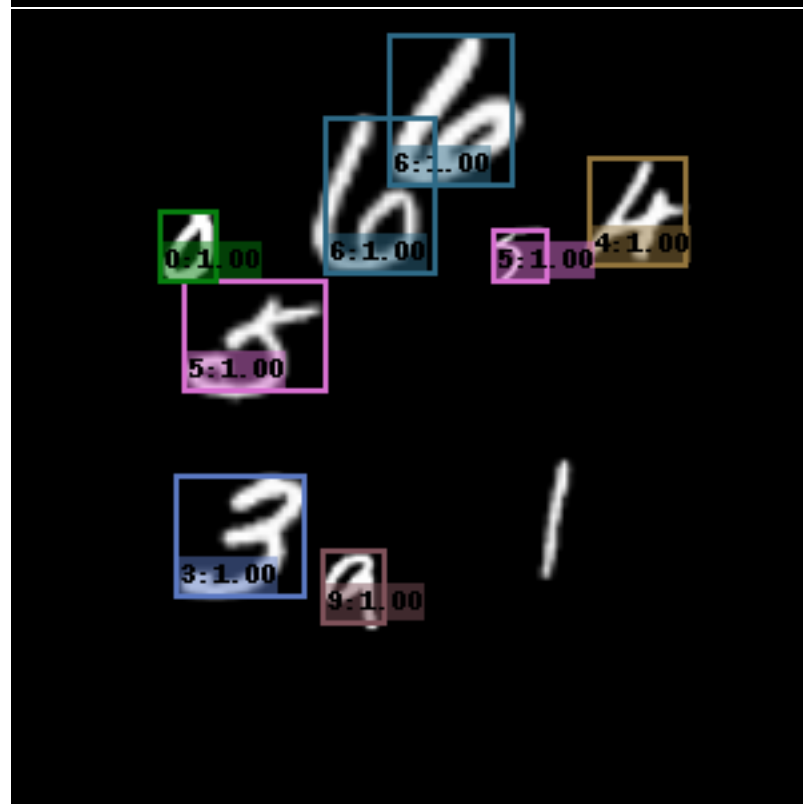
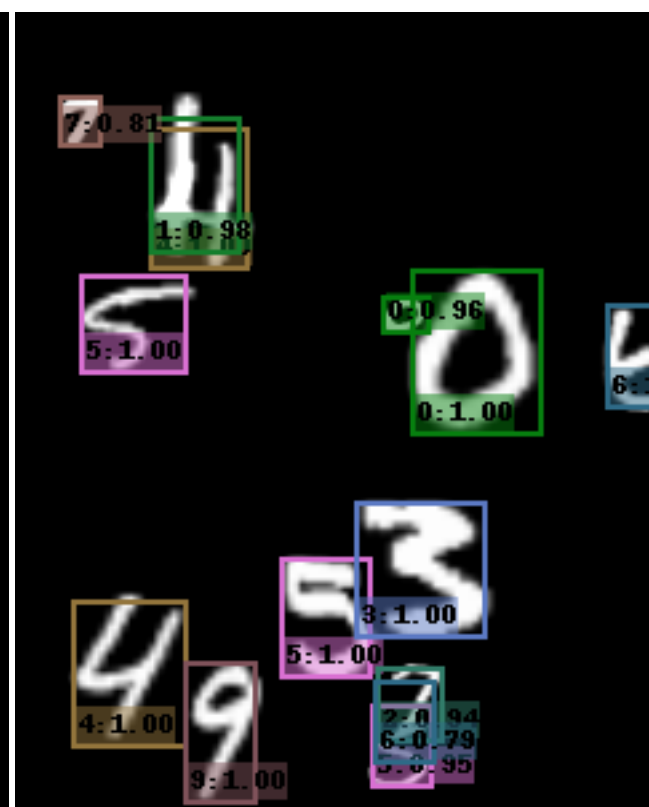
```

5.4 Task 4e)

Most digits are actually recognized, but there is some internal battle on some of the digits i.e. the SSD is not sure what the number is. Really small numbers are also not always recognized.







5.5 Task 4f)

2021-03-19 18:19:49,626 SSD.inference INFO: mAP: 0.1372

aeroplane	: 0.2276
bicycle	: 0.1524
bird	: 0.0579
boat	: 0.0913
bottle	: 0.0000
bus	: 0.1490
car	: 0.3093
cat	: 0.2334
chair	: 0.0219
cow	: 0.0830
diningtable	: 0.1284
dog	: 0.1782
horse	: 0.2548
motorbike	: 0.2046
person	: 0.2868
pottedplant	: 0.0003
sheep	: 0.1011
sofa	: 0.0393
train	: 0.1271
tvmonitor	: 0.0974

