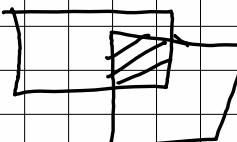


# Assignment 4

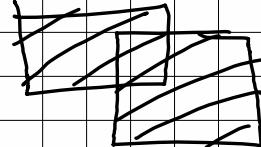
## Task 1

a) Explain Why the Intersection over Union is used  
Here we can find it for two bounding boxes

It is used to evaluate accuracy of object detection.

$$IOU = \frac{\text{Area of overlap}}{\text{Area of Union}} =$$


The result is the area between the ground truth and prediction box.



b) Write down the equation of precision and recall,  
and briefly explain what are the terms  
true pos and false pos.

TP: Correct detection, Actual is Pos and Predicted is Pos

FP: False detection, Actual is Neg and predicted is Pos

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

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

\*  
false negative

(c) Find mean average precision

Class 1: Precision =  $\{1, 1, 1, 0.5, 0.2\}$   
Recall =  $\{0.05, 0.1, 0.1, 0.3, 1\}$

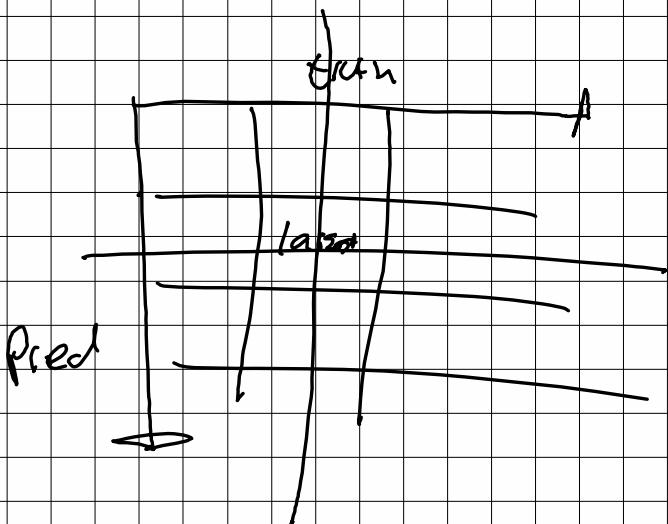
Class 2: Precision =  $\{1, 0.8, 0.6, 0.5, 0.20\}$   
Recall =  $\{0.3, 0.4, 0.3, 0.7, 1\}$

$$\begin{aligned} &= 0.1 \cdot 1 + 0.3 \cdot 1 + 0.05 \cdot 1 + 0.1(0.8 + 1) \\ &\quad + 0.5(0.6) + 0.7(0.5 + 0.5) + 1(0.2 + 0.2 + 0.5) \\ &\quad + 0.1 + 0.05 + 0.3 \\ &= 3.42 \end{aligned}$$

$$\text{Mav} = \frac{\underline{3.42}}{5} = \underline{\underline{0.684}} = 68.4 \%$$

2)

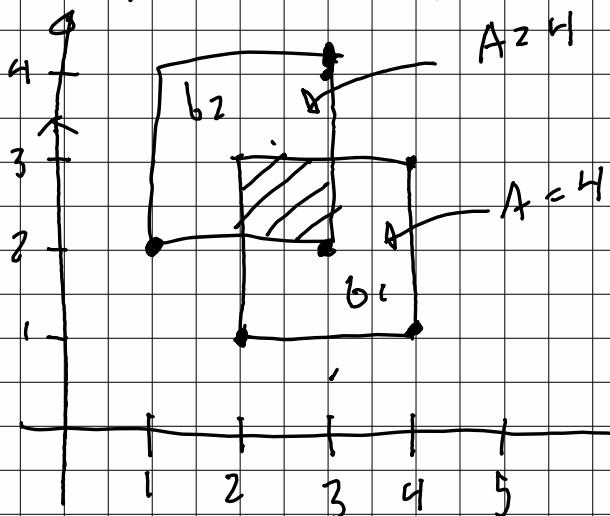
ku mas



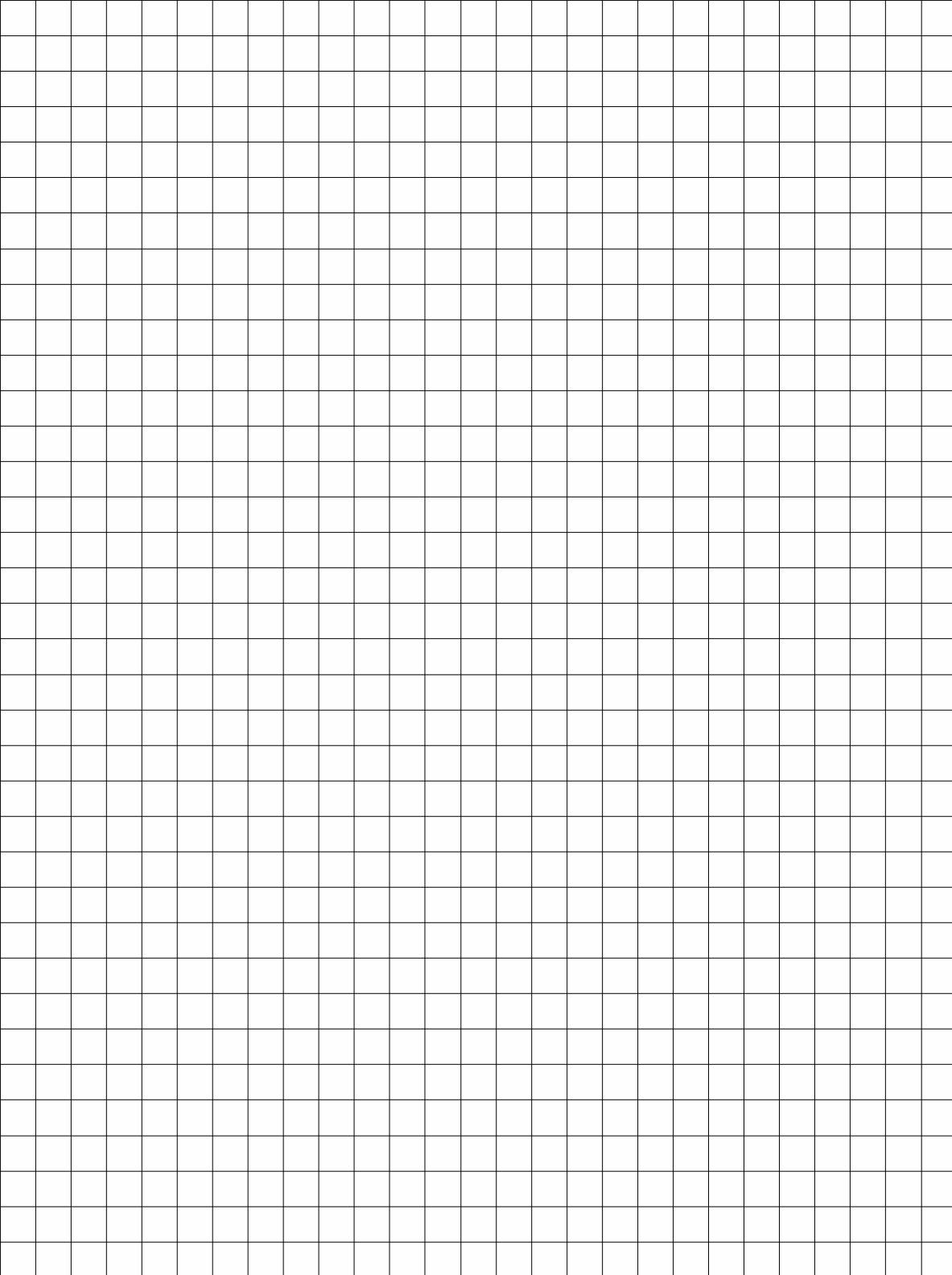
Pred(2) and true(2) mask

$$b_1 = [2, 1, 4, 3] \quad \dots [x_{\min}, y_{\min}, x_{\max}, y_{\max}]$$

$$b_2 = [1, 2, 3, 4]$$



$$ku = \frac{1}{4+4-1} = \frac{1}{7}$$



# 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](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) (<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>). 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`.

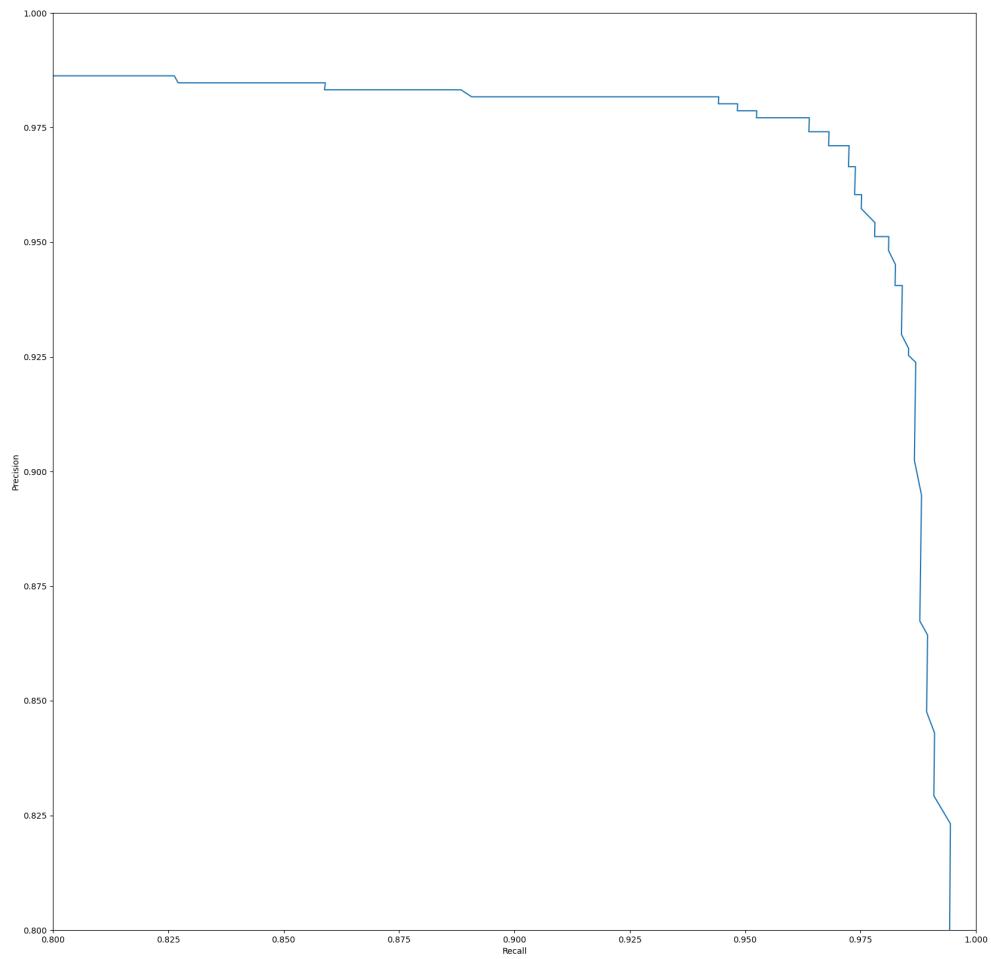
## Task 1

All of Task 1 is shown as handwriting, as well as some figures showing my work on other tasks!

## Task 2

### Task 2f)

Mean average precision: 0.9066



## Task 3

I kept these answers a little short mabey

### Task 3a)

non-maximum suppression (nms): Sorts the predictions by confidence scores. Then starts from the top and checks if the prev boxes have a higher iou then 0.45. If found, keep the current pred. Keep max 200 predictions.

### Task 3b)

False, small objects can only be detected in higher resolution layers

### Task 3c)

Using different bounding box aspect ratio at the same location makes it cover more various input object sizes and different shapes.

### Task 3d)

Were YOLO use two fully connected layers and SSD uses multiple convolutional layers. This predicts the offset to default boxes.

### Task 3e)

Total anchor boxes =  $H \times W \times 6$  different = 38386 = 8664

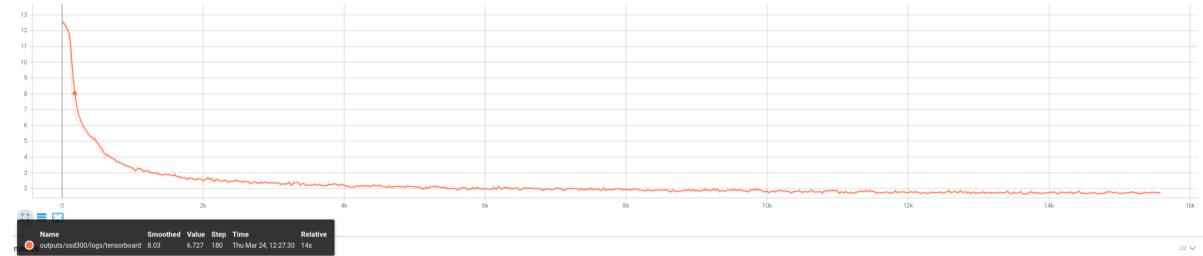
### Task 3f)

Total fo network =  $(3838+1919+1010+55+33+1)6 = 11640$

## Task 4

### Task 4b)

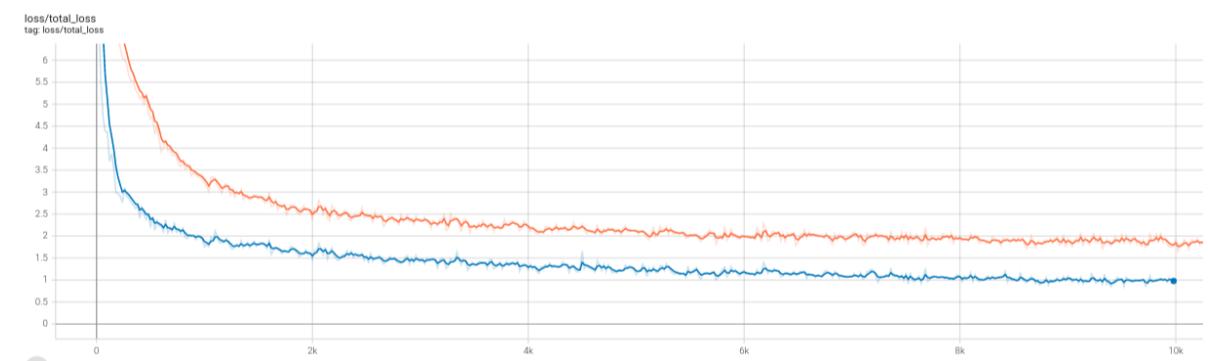
metrics/mAP@0.5: 0.775

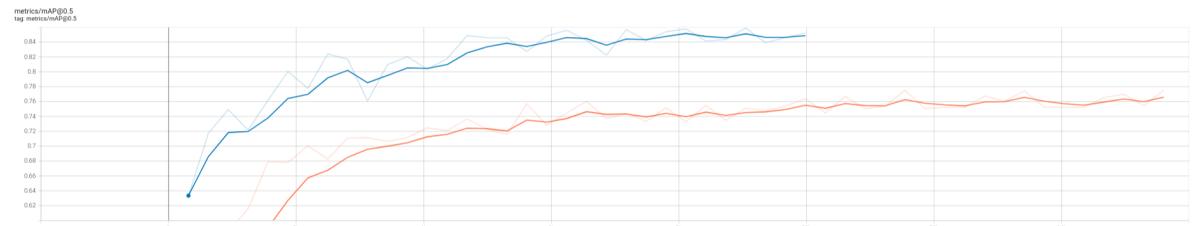


### Task 4c)

Making the size of the samllest achor box smaller. From 30,30 to 15,15. In order to detect the smallest numbers! Changing to adam optimizer. Changing epochs to 32 in order to do 10k iterations Adding some bachnormalization after the conv layers. Still not sure what the best placement of normalization layers are.

metrics/mAP@0.5: 0.852





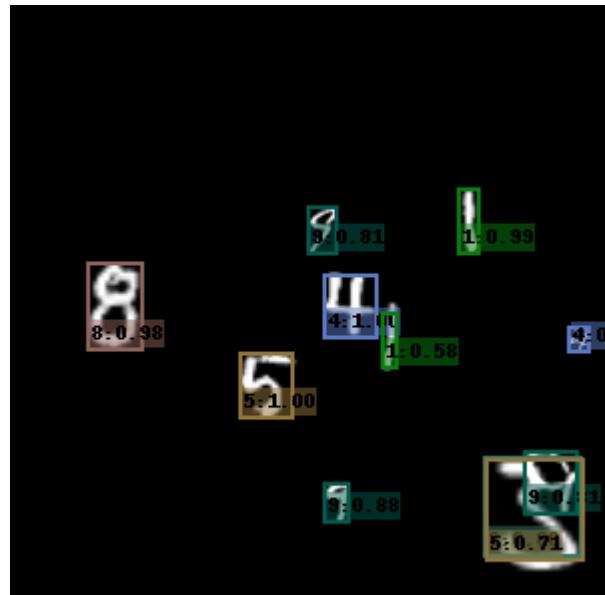
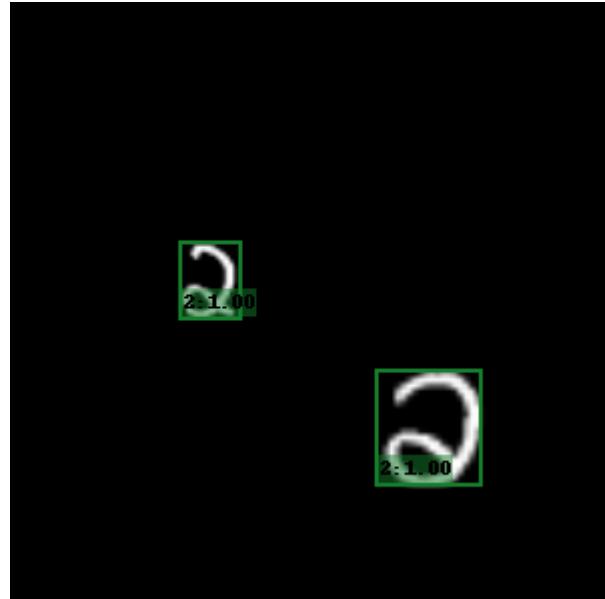
Blue is the improved model!

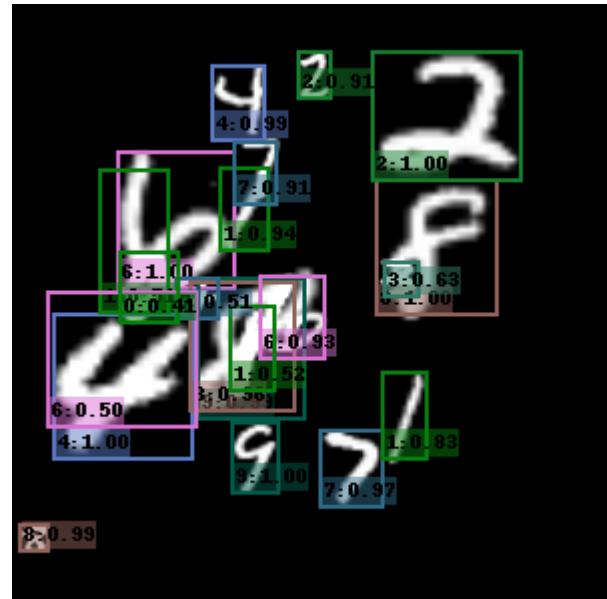
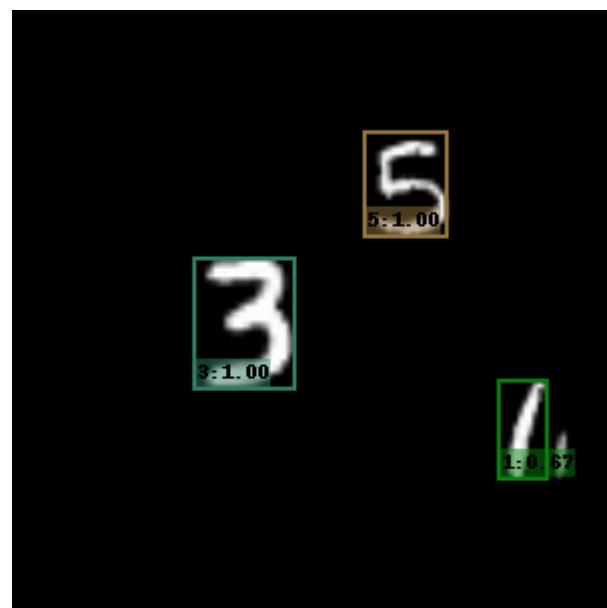
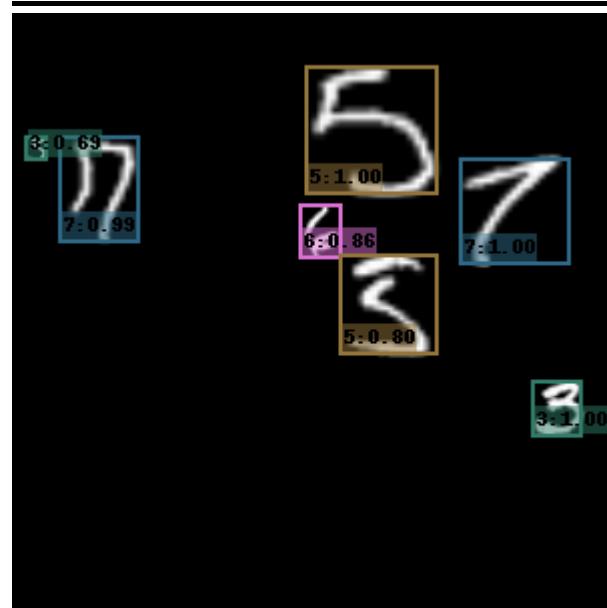
## Task 4d)

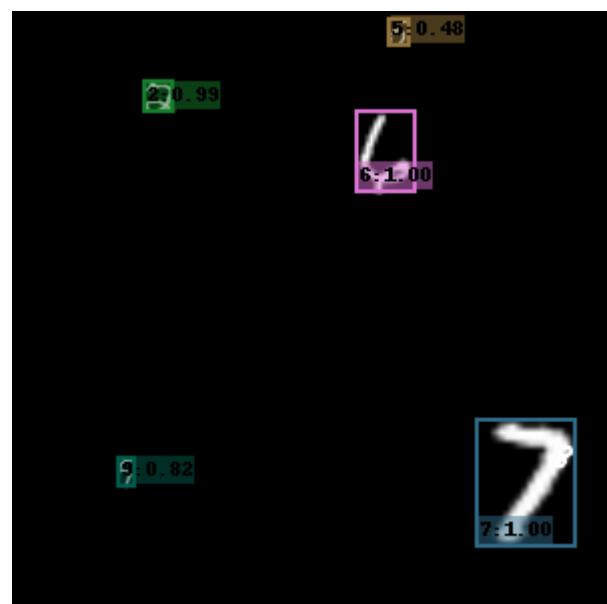
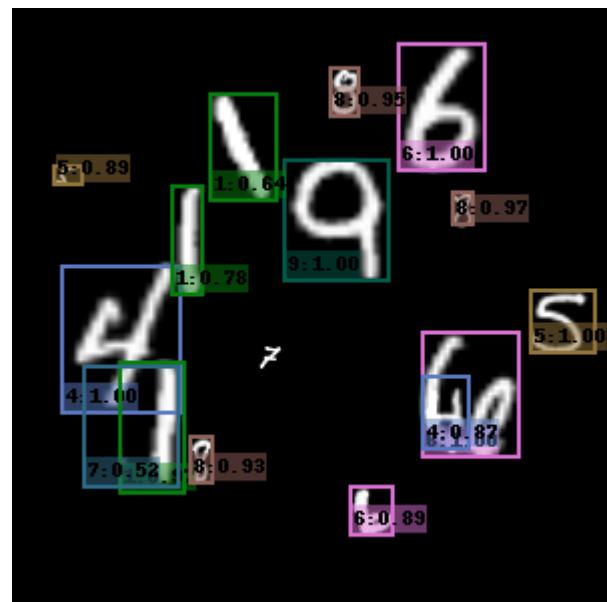
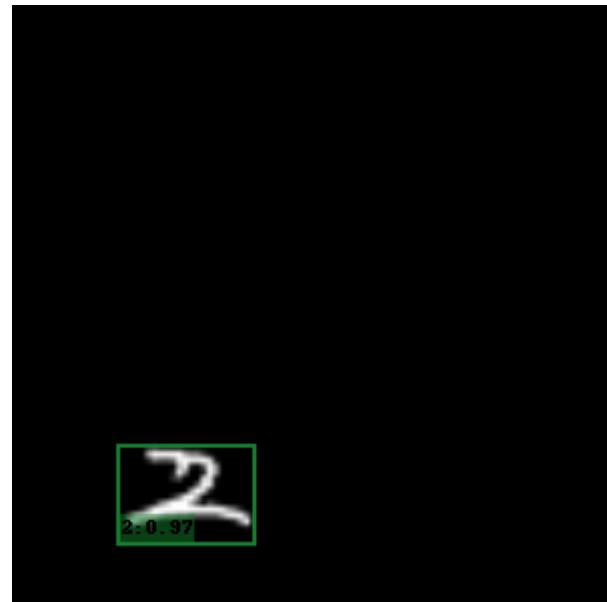
I didnt not manage to do this Didnt understand what the TA were trying to tell me :)

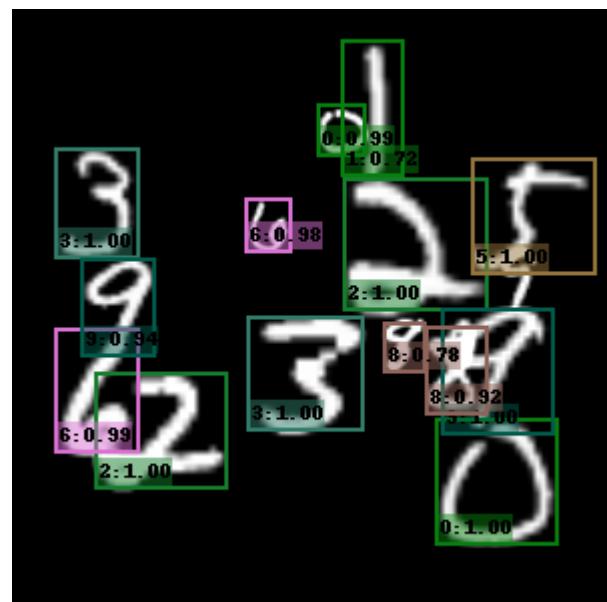
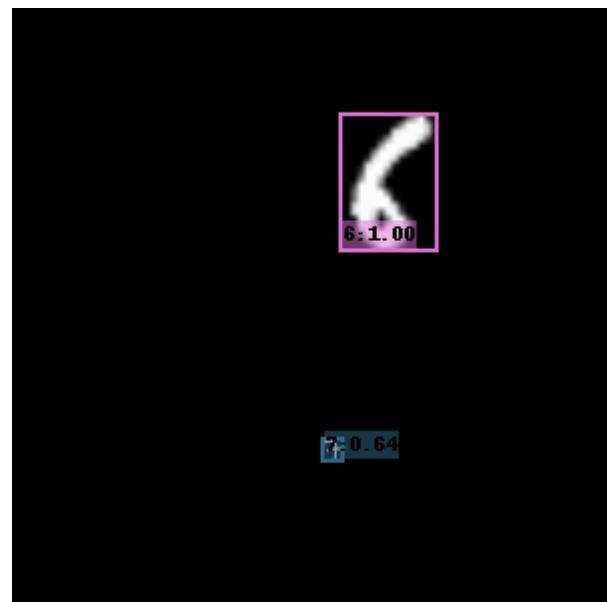
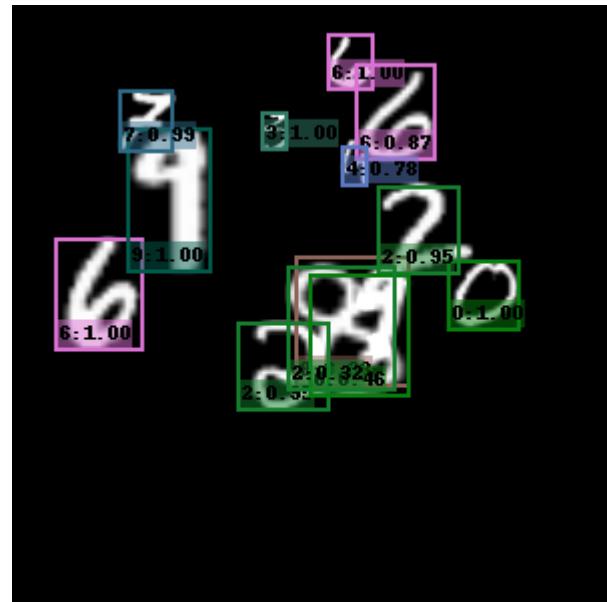
## Task 4e)

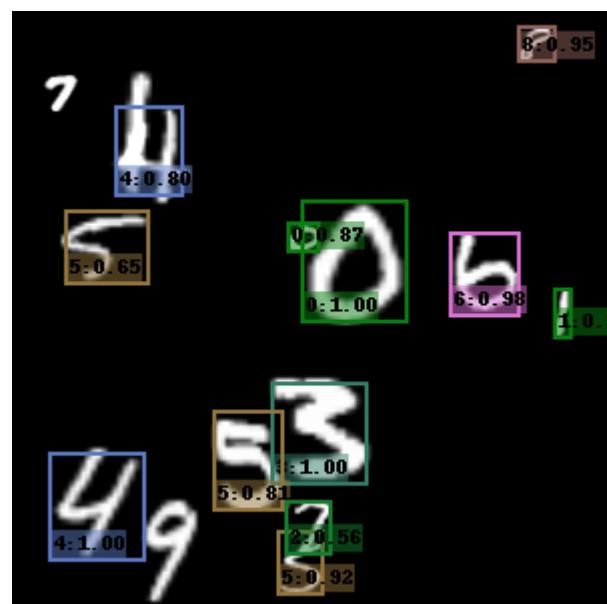
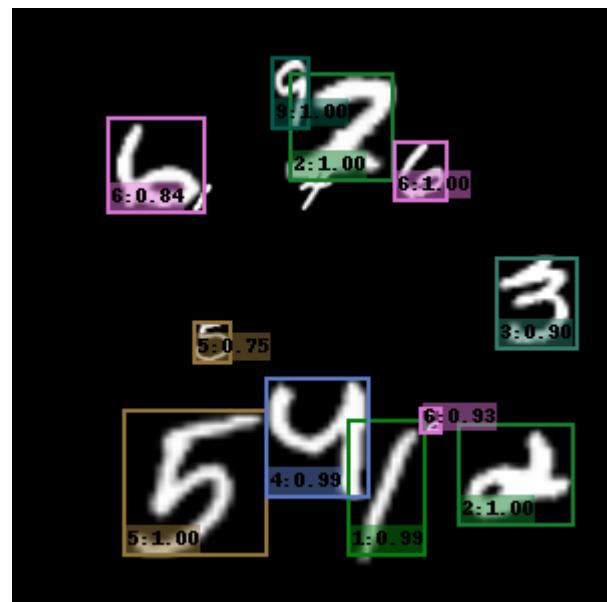
Sorry alle these photos. It still seems to struggle with the smallest photos, and seven and ones, and six! It also seems to struggle with some of the normal size numbers. Dont really know why... Could probarly include more achor box sizes. Also on one image it seams to predict to many numbers, but cant really tell myself.....

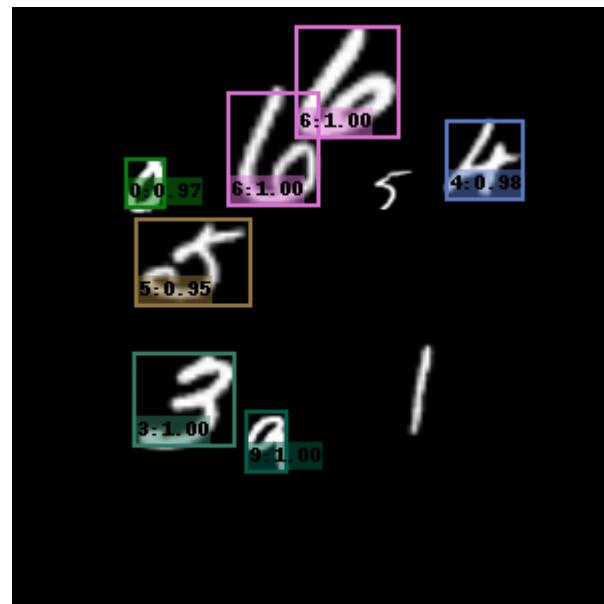






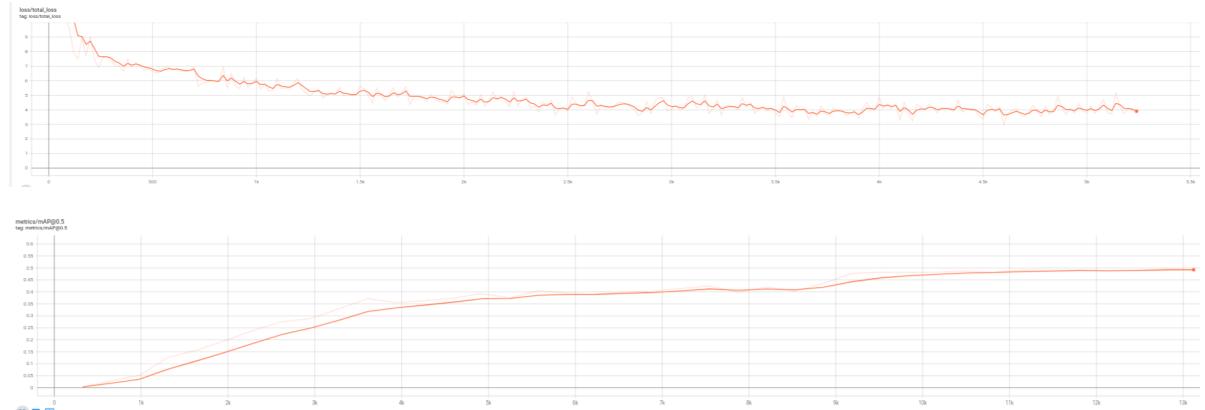






## Task 4f)

I had to do this on a clean install from github. Not sure if the code provided in the zip will do the same as this, because I only got errors with the old code.



I was wondering if there is an easy way to print text instead of just class number for easier readability? I only got numbers 1-20, not text label so I found this online and it seems to match my results:

21:10



gist.github.com

what about "Boat" ?

**sandeep22-v** commented on 29 Apr 2021

...

The above labels are of COCO dataset and not  
PASCAL VOC dataset

1:'aeroplane'  
2:'bicycle'  
3:'bird'  
4:'boat'  
5:'bottle'  
6:'bus'  
7:'car'  
8:'cat'  
9:'chair'  
10:'cow'  
11:'diningtable'  
12:'dog'  
13:'horse'  
14:'motorbike'  
15:'person'  
16:'pottedplant'  
17:'sheep'  
18:'sofa'

19:'train'

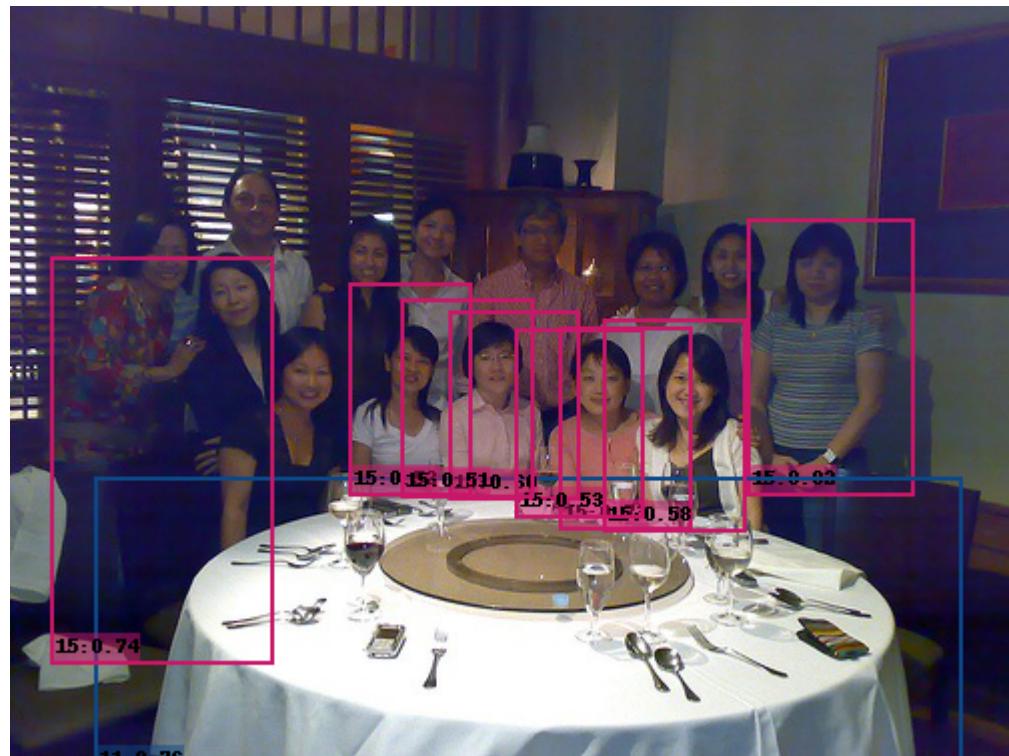
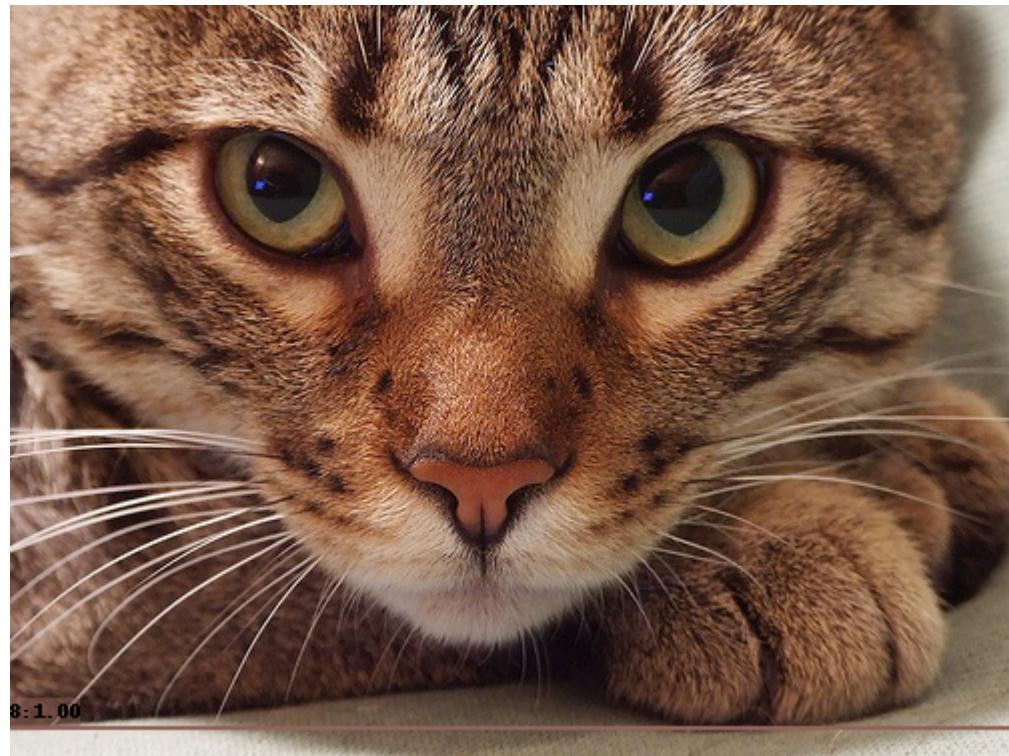
20:'tvmonitor'

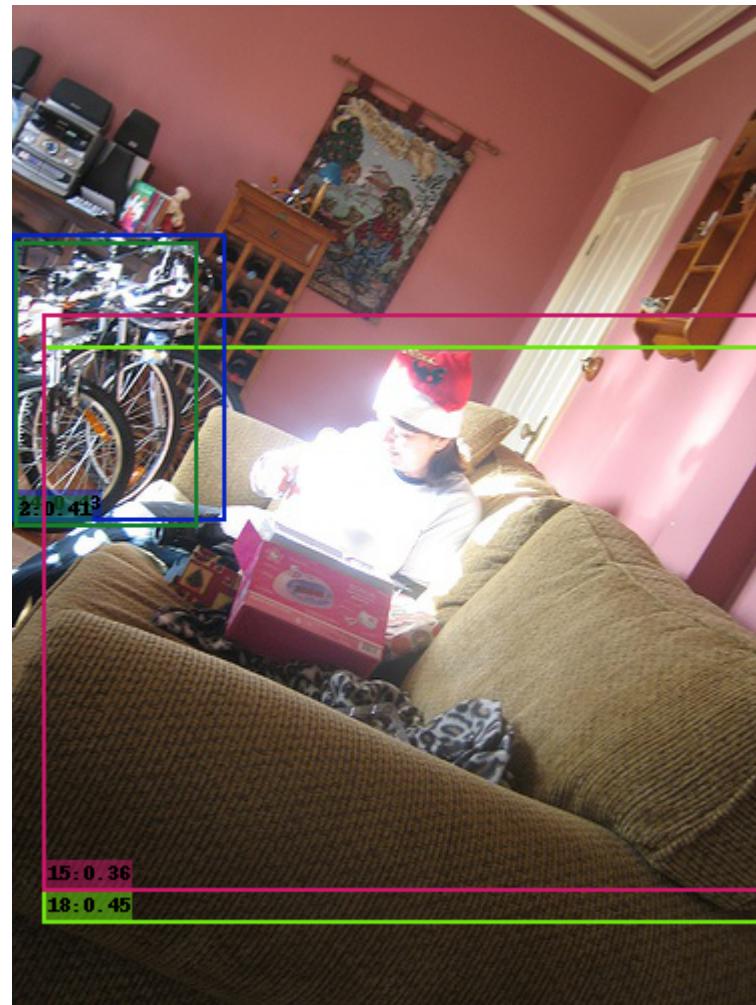
Isn't this the correct one??

**anujonthemove** commented on 22 May 2021 ...

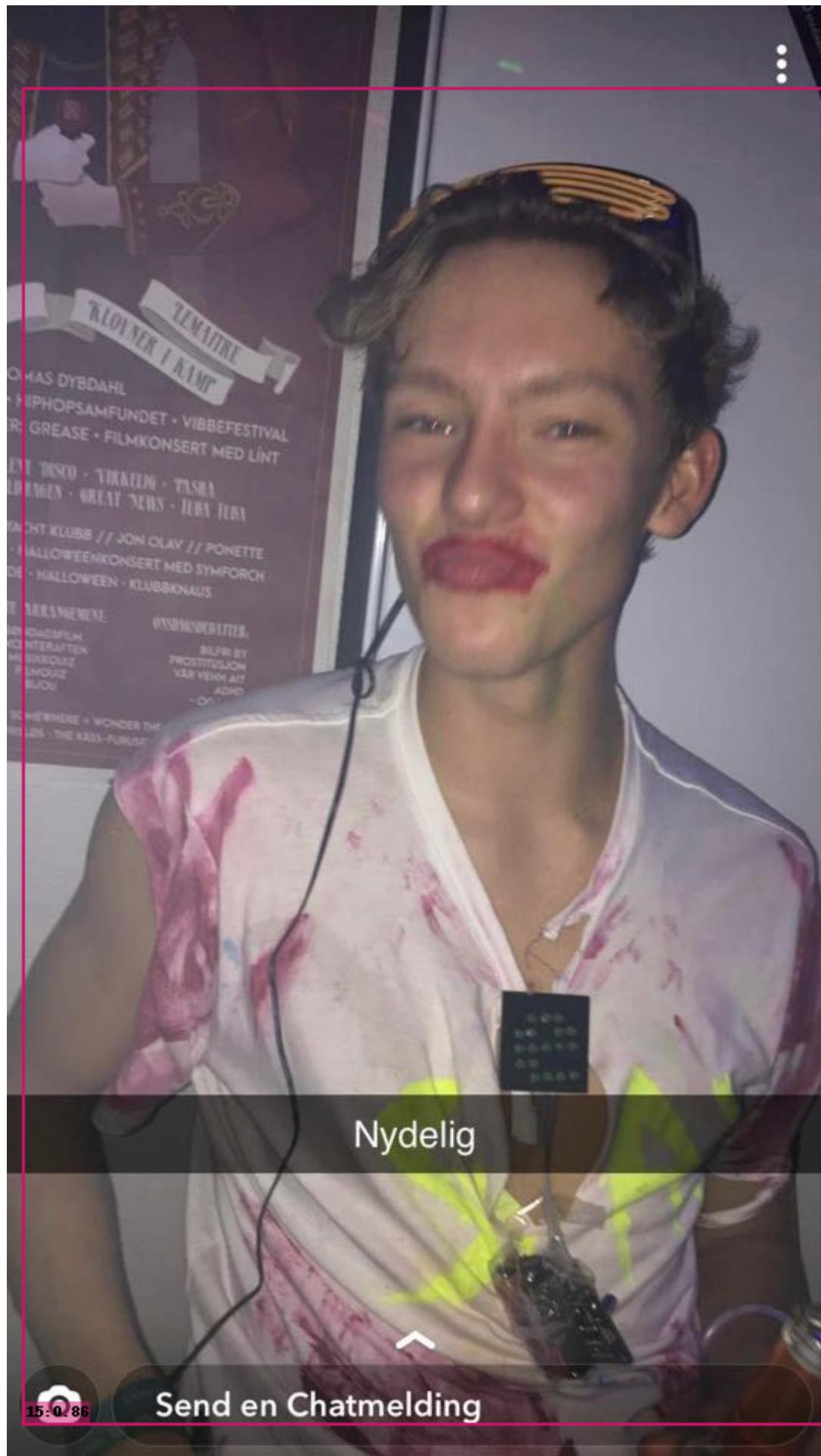
The above labels are of ~~0000~~ dataset and  
Anyways here are my results, they seem to be alright condisering 15 is a person, have fun :)





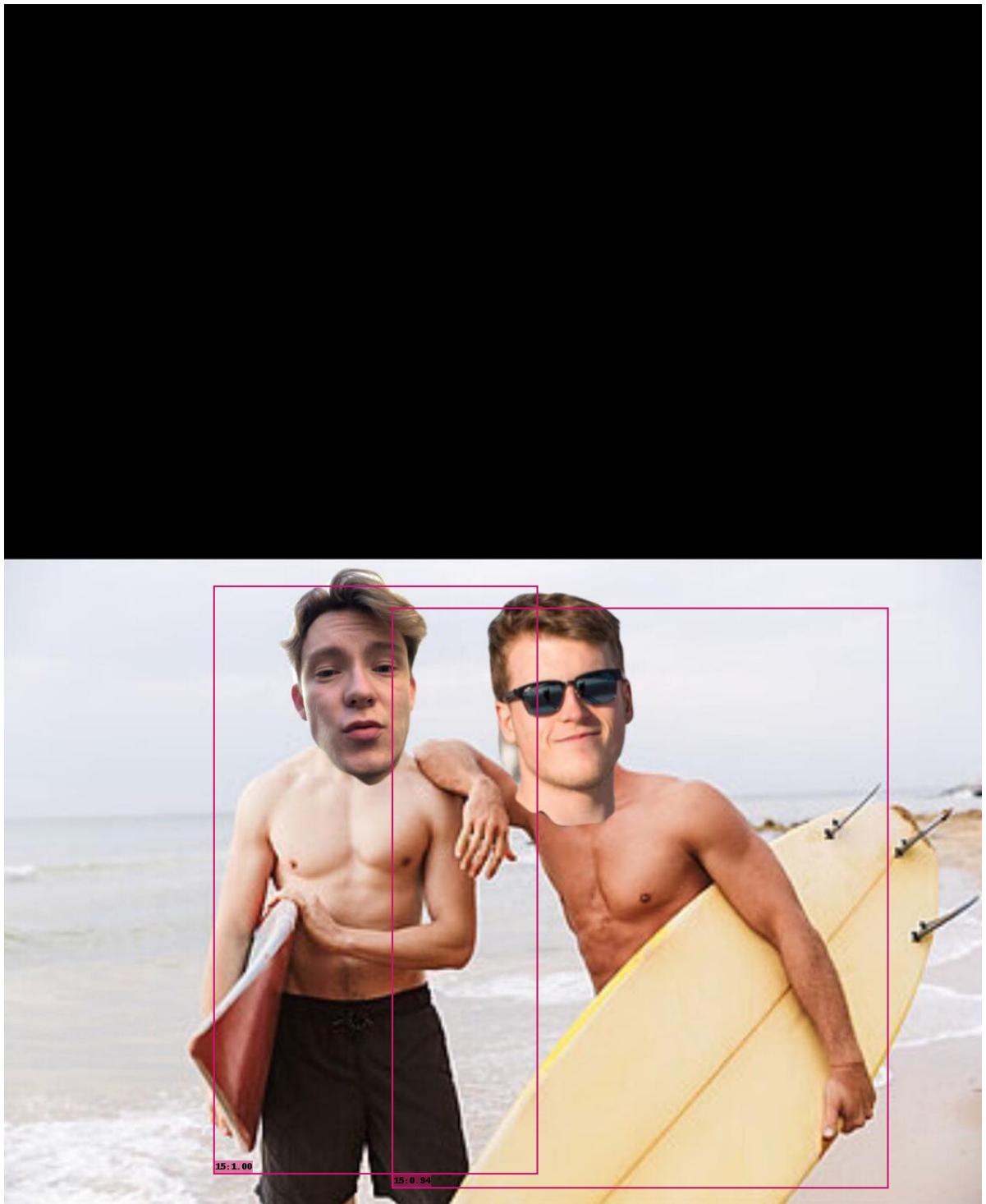












© CanStockPhoto.com





