CIS 680: Project 2

Junfan Pan

10/04/2020

7 Submission and Evaluation

7.5 plot of the mean Average Precision over training

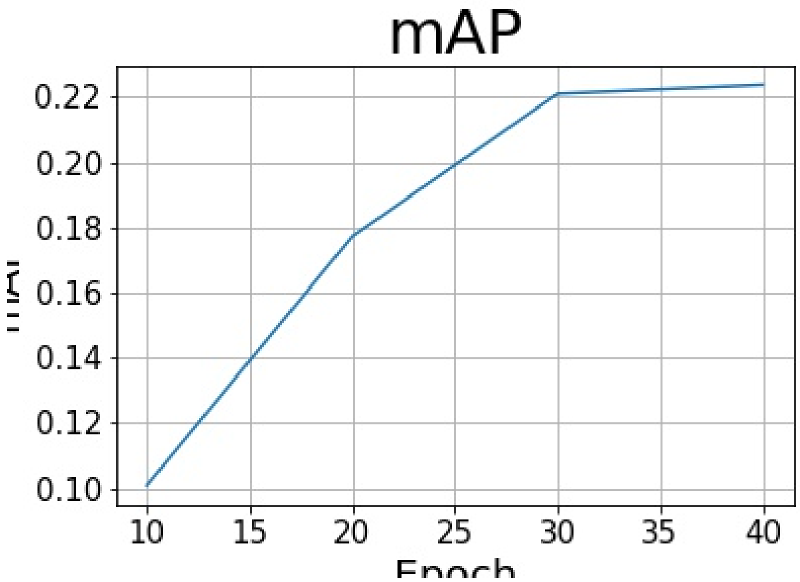
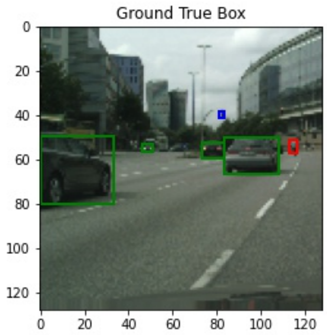


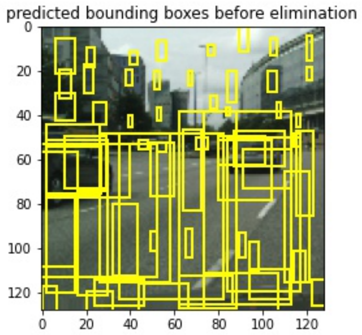
Figure 1: mAP plot

I plot the mAP for the first 40 training epochs since I observed that after 40 epochs, the mAP constantly decreases.

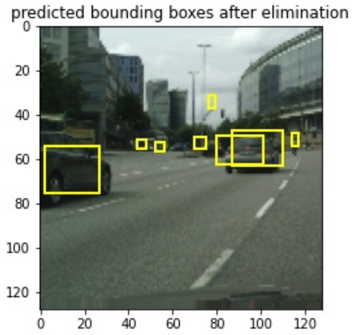
7.6 and 7.7 Data Postprocessing



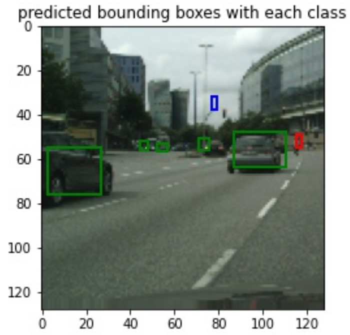
(a) Ground True Box



(b) predicted bounding boxes before elimination



(a) predicted bounding boxes after elimination



(b) predicted bounding boxes (after NMS) with each class

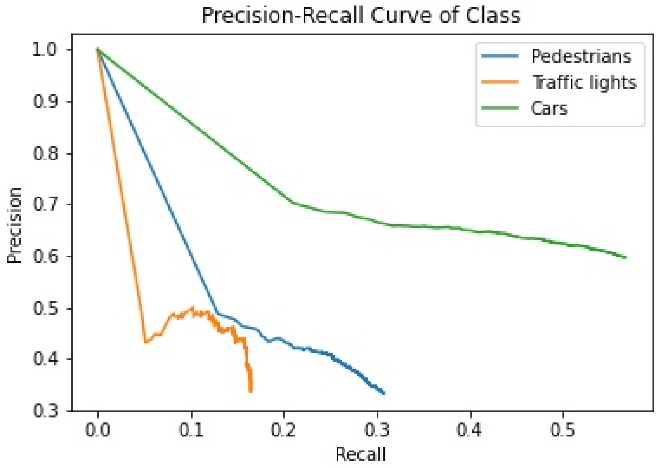


Figure 4: precision/recall curves

The achieved mean Average Precision for my inference stage is: 0.2237523328559753

7.8 Issues or challenges

a. One issue I encountered is the mismatch between the decrease of the test loss and the worse behavior of object detection. This initially doesn't make sense to me since the reduction of test loss should bring along good performance. However, after I separately plot the test loss of confidence part, classification part and the localization part, I found that the learning speed of these different loss parts are different. Although the total test loss may still decrease, the classification and confidence part have already been overfitted while the localization part is still learning. Therefore, in order to get a better detection performance of small objects such as traffic lights, I should select to use the model checkpoint when the confidence score loss is well-trained and is not overfitted.

b. While training my network, I observed that my network can only detect very sparse objects and usually ignore those tiny objects like traffic lights in the figures. Therefore, to eliminate the impact of the small size on detection performance, I set the Ctarget as 1 instead of using confidence score from iou. This enabled my network to detect more small objects and from the figures from the 7.7, the network can now succeed in detecting small objects.

c. If more time allowed, I would definitely spend more time on training a more well-behaved network. For example, use more training epochs and introduce a different learning rates during the training process. In addition, instead of setting Ctarget as 1, I can manage to find an appropriate coefficient of the confidence loss part.

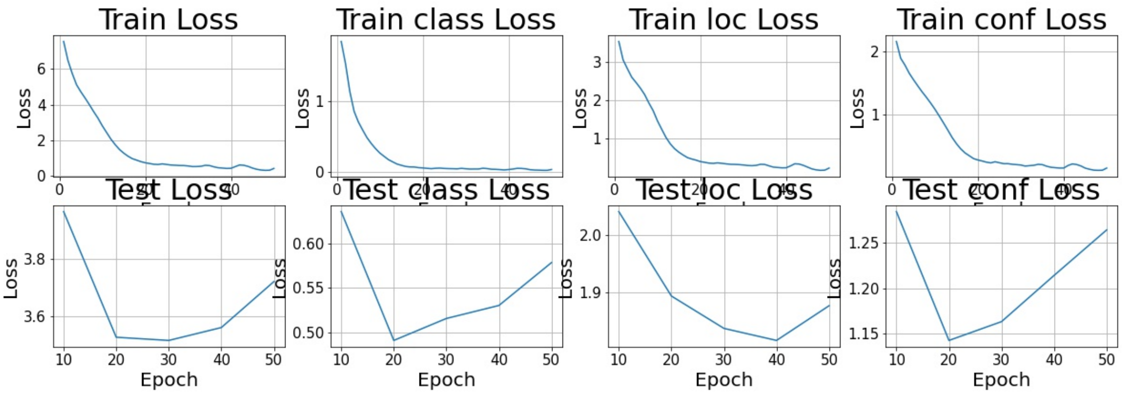


Figure 5: different loss parts have different learning rates