Assignment 3B: Instance Segmentation with MaskRCNN

Venkata Sai Nikhil, Thodupunuri

62716136

Part 8

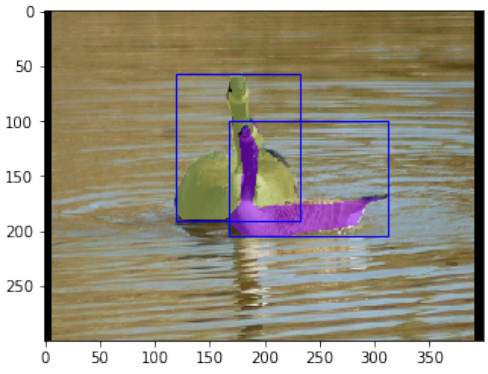
The original COCO data-set has many classes (80 excluding background). So the backbone and rpn will try to capture every class when trying to generate the region proposals. But for the new data-set, we have few classes(3).

If we use the same backbone/rpn without fine-tuning, There will be many false-positive box proposals (Proposals will try to capture all 80 classes, but we need proposals only focused to 3 classes). To get better regional proposals, The remaining 77 classes related proposals should be treated as non-objects. Having a good region proposals will ease the localization. The proposal box sizes will tend to be similar to original bounding boxes, and hence eases the complexity on regressor.

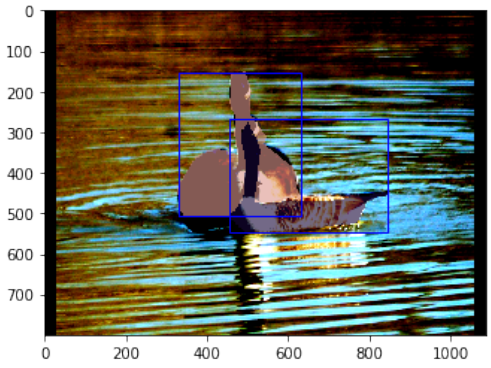
Coming to feature maps, The backbone will try to generate activation for all 80 classes. If we are working on only 3 classes, having activation for those 3 classes will help to achieve better results. Fine-tuning the backbone will help to generate precise activation focused to the 3 classes which will help in generating precise masks and good classification accuracy.

Part 9 : Image preparation

Sample 1 :

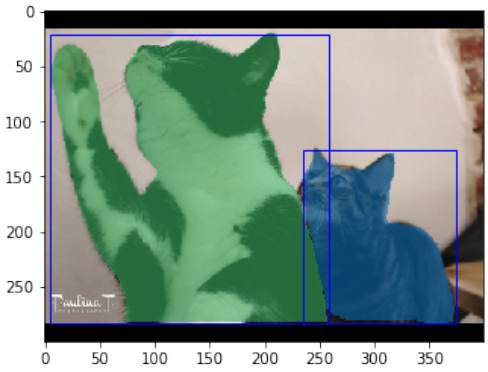


Ground Truth

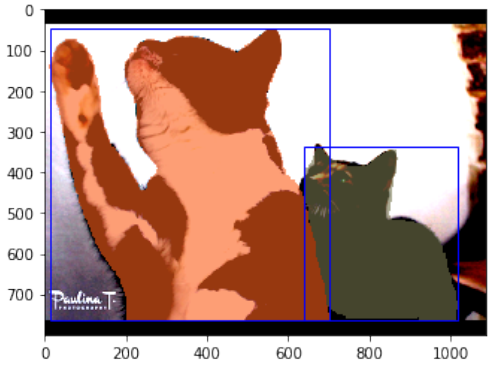


Normalized image

Sample 2 :

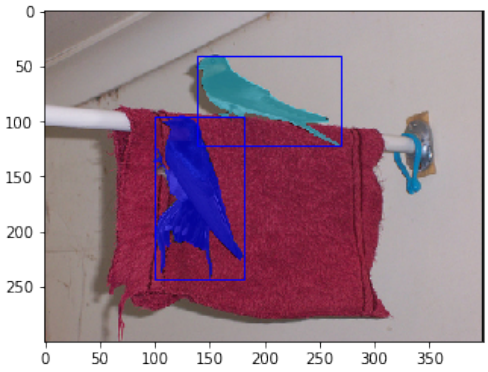


Ground Truth

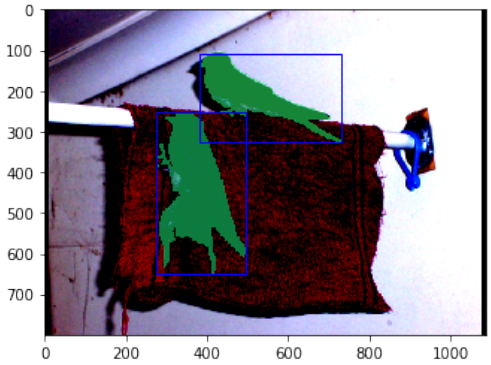


Normalized image

Sample 3 :



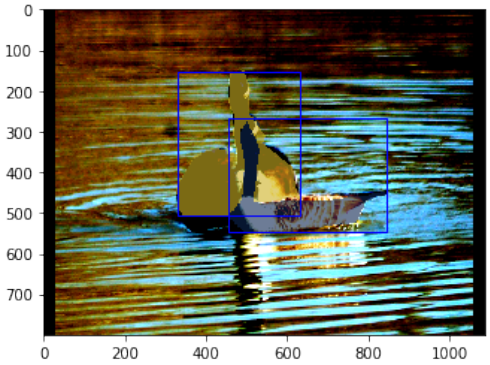
Ground Truth



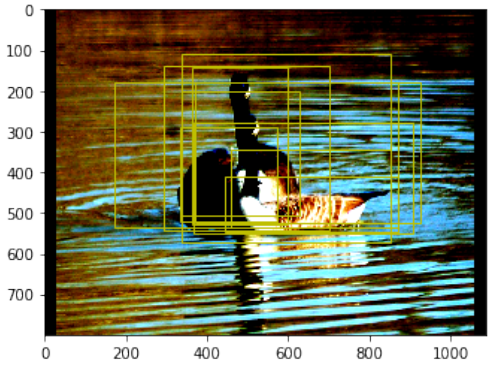
Normalized image

Part 10: RPN Out proposals

Sample 1 :

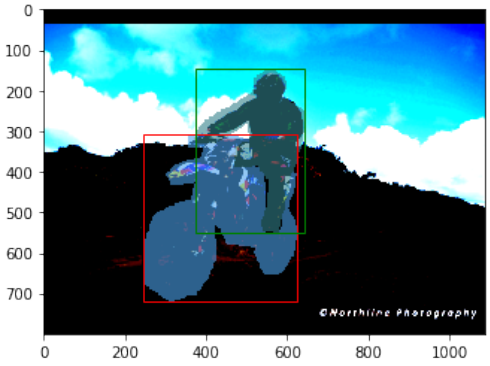


Ground Truth

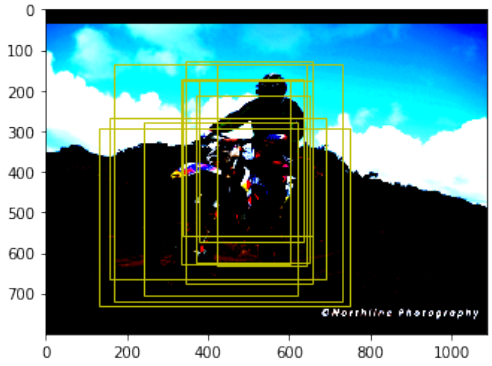


RPN proposals

Sample 2 :

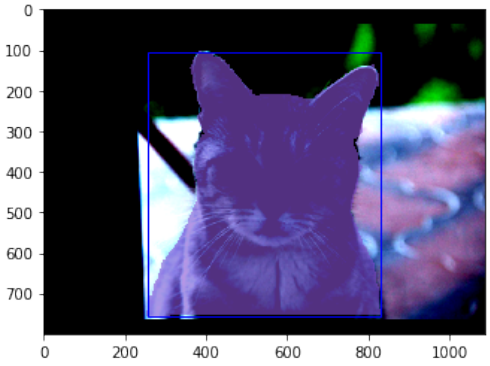


Ground Truth

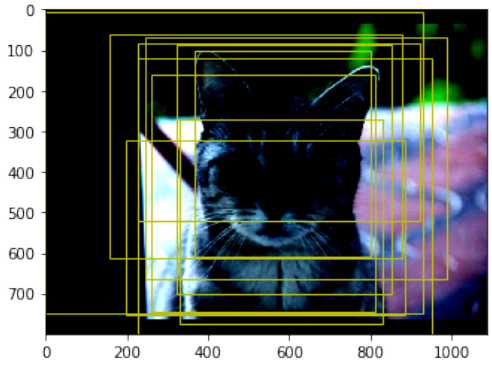


RPN proposals

Sample 3 :

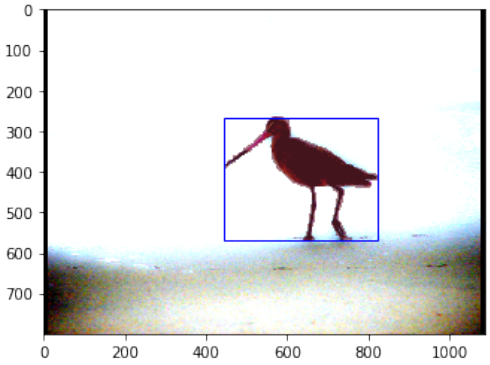


Ground Truth

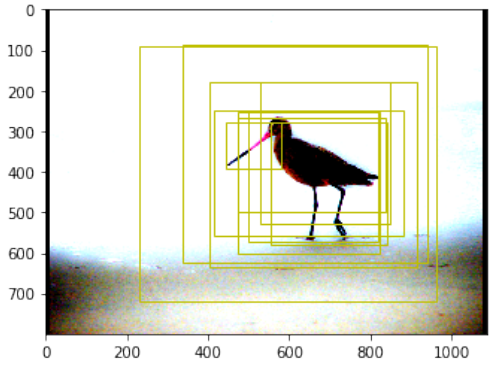


RPN proposals

Sample 4 :



Ground Truth



RPN proposals

Part 11: Advantage/Disadvantages - alternative training vs Joint training

In joint training, we combine the loss from RPN and Heads into a single loss and back-propagate it to the network. For shared network, we propagate the gradients from both RPN and heads.

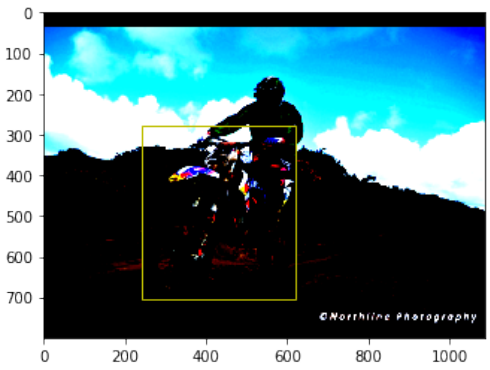
This is a approximate solution, as we ignore the gradients with respect to RPN box coordinates which are input to the heads. To overcome this, the paper proposes a "ROI wrapping" layer to involve the gradient propagation.

While Alternative training is a iterative optimization. The disadvantage is time. Joint training reduces the time training time by 25%.

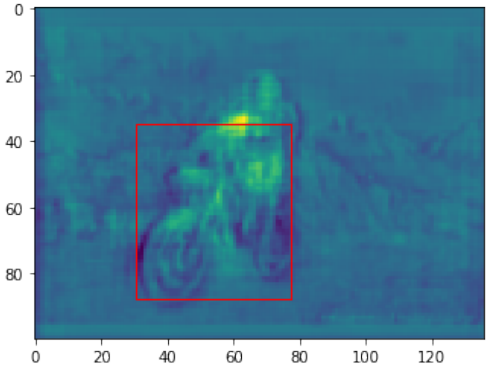
Part 12: ROI Align

Included some of the outputs of ROI align. To validate the ROI, we tried different P values to test the code. ROI align implemented from scratch.

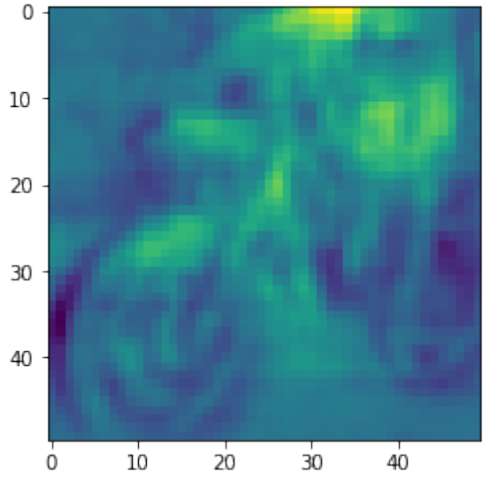
Sample 1:



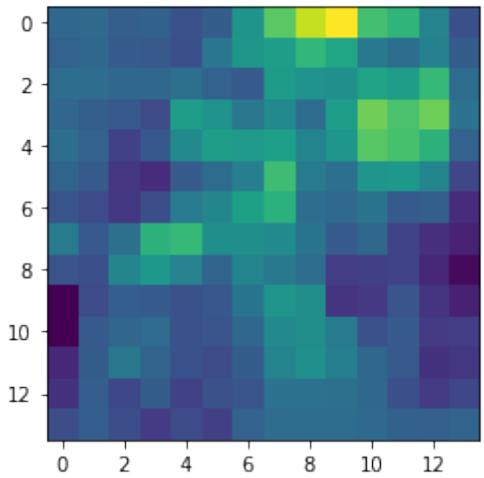
Ground Truth



Feature Map

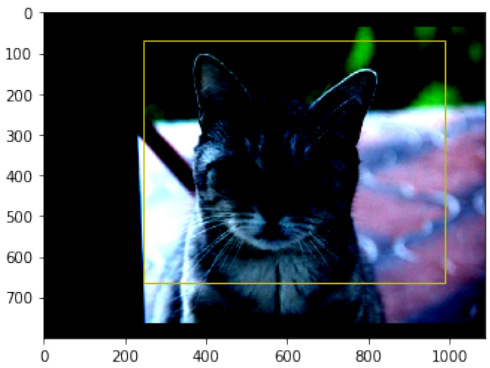


ROI 50x50

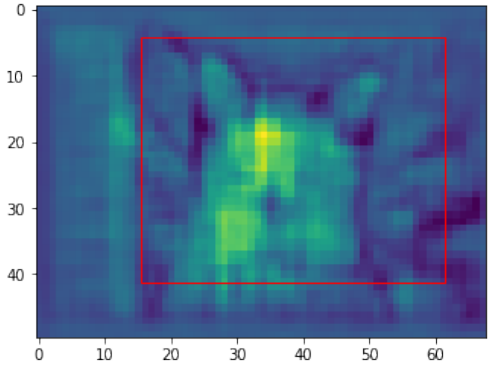


ROI 14x14

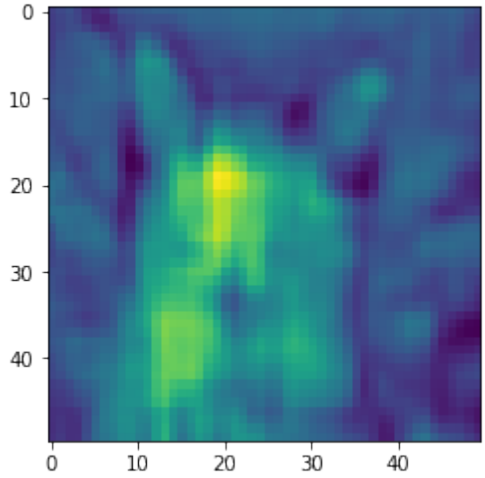
Sample 2:



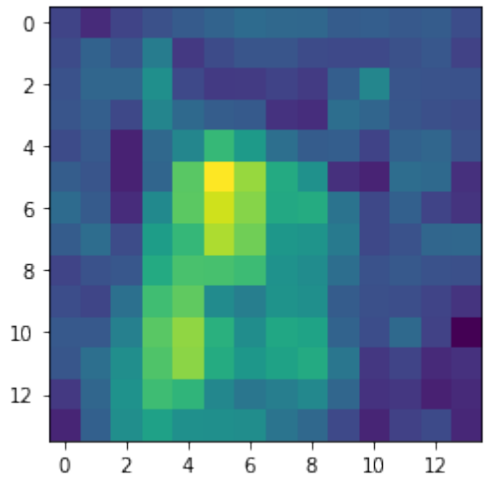
Ground Truth



Feature Map



ROI 50x50



ROI 14x14

Part 13: Two Stage versus Single Stage

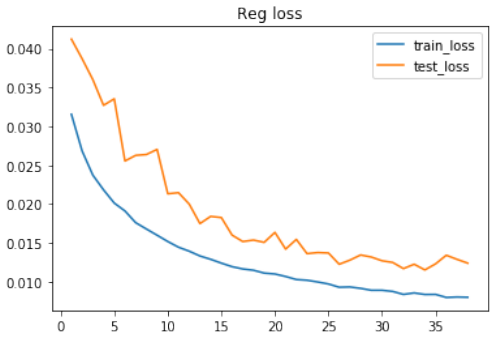
In two stage, The second network only focuses on the part of the image (not entire image). The second stage can just focus on features specific to region proposal. So second stage has access to region specific features which removes lot of noise from rest of the image. Hence two stage detectors achieve better accuracies.

But in single Stage detectors, for prediction we look at the entire image for predicting final results. With high receptive fields, the predictions are sensitive to noisy activations from rest of the receptive filed.

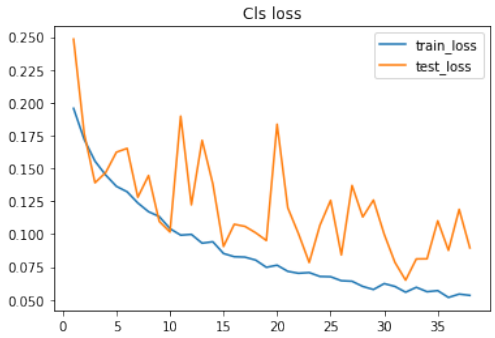
But single stage detectors are fast. (As image is passed only once through the network). But in two stage, the network has to look back again for the feature maps.

If speed is more important, then single stage detectors will be a good option to use. Otherwise use two-stage detectors.

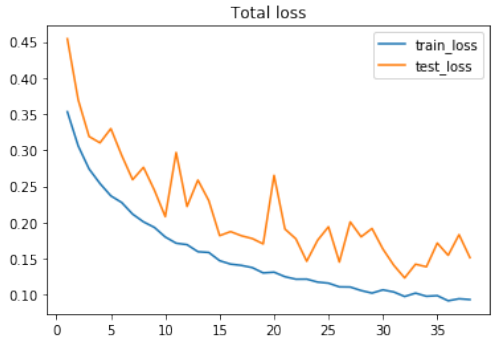
Part 14: Regression and Classification Training



Regression Loss



Classifier Loss



Total Loss

Note: Total loss is computed using 5* Reg loss + class loss. Having a lambda smaller than 5, results in classifier dominance in training.

Learning rate used: 0.0001, Image centric training

Part 15: Post processing and metrics

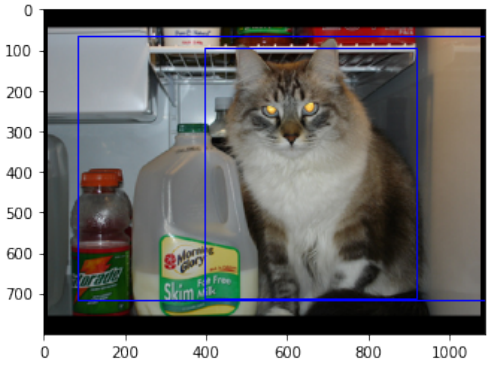
Accuracy metrics

Class	precision	Recall	F1
0	0.84	0.94	0.89
1	0.79	0.65	0.71
2	0.87	0.64	0.74
3	0.88	0.71	0.79

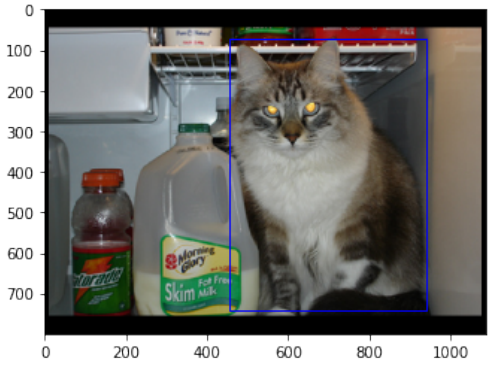
Bounding box regression output after NMS

Color Codes:

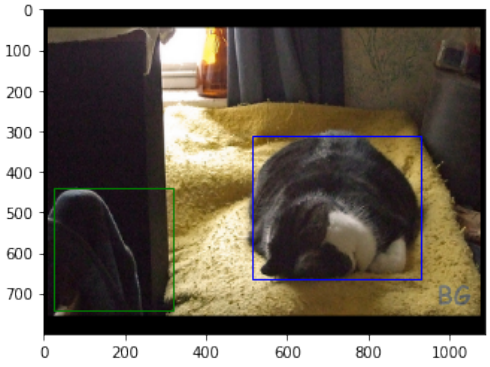
Red : Vehicle, Blue: Animals, Green: Human.



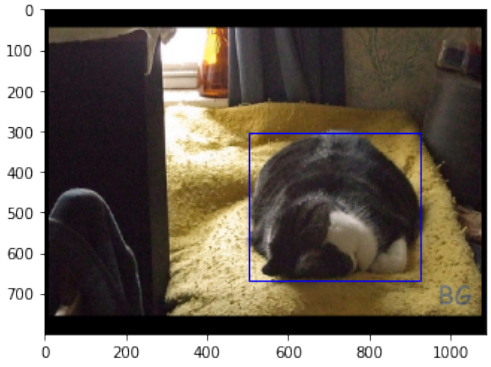
Prediction



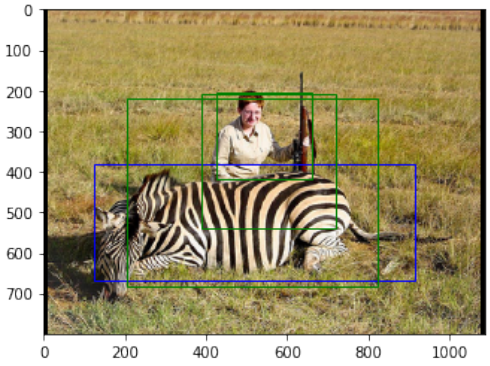
GT



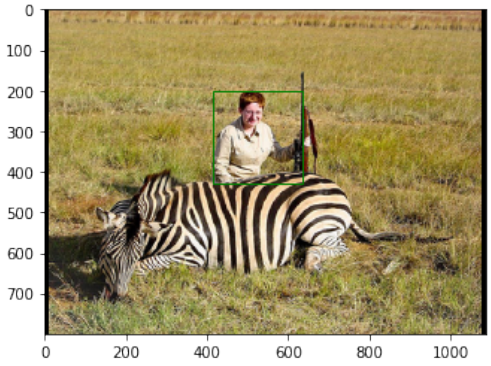
Prediction



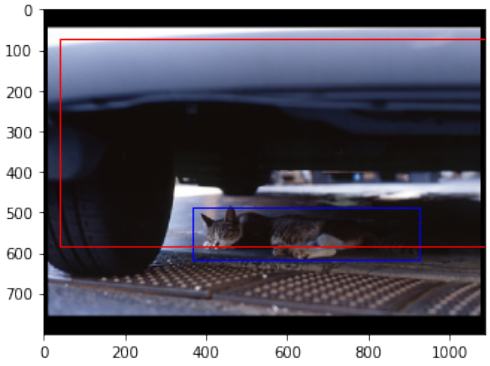
GT



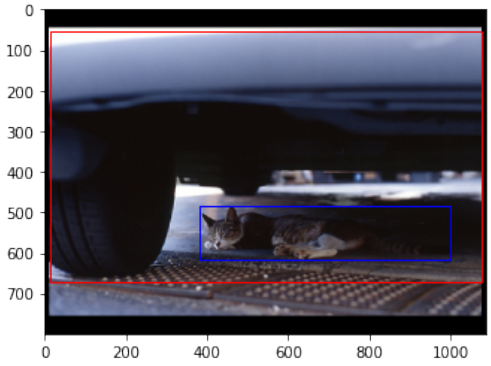
Prediction



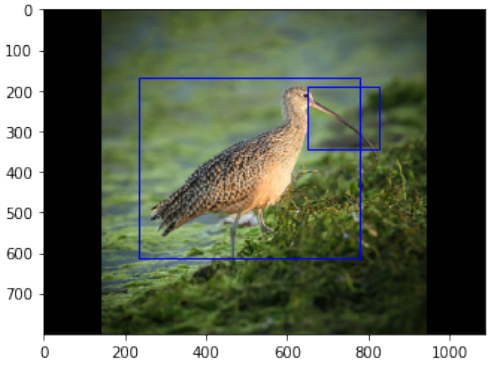
GT



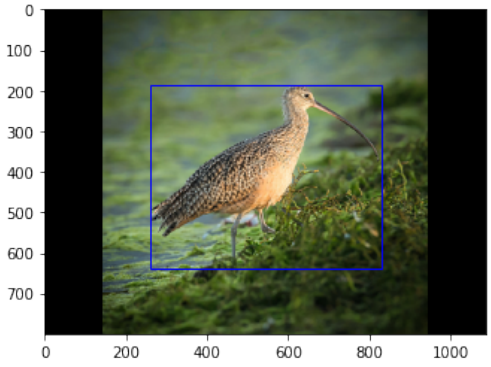
Prediction



GT



Prediction

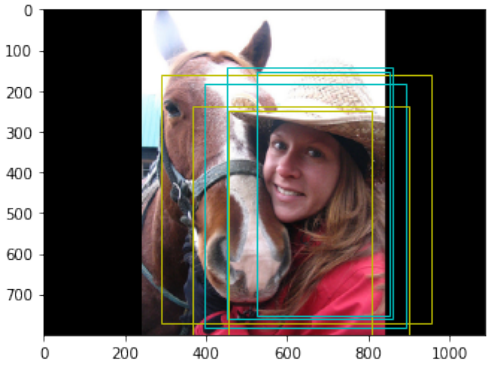


GT

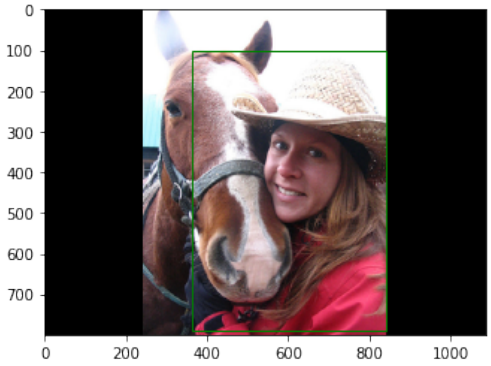
Regression output Refinement

Color Codes:

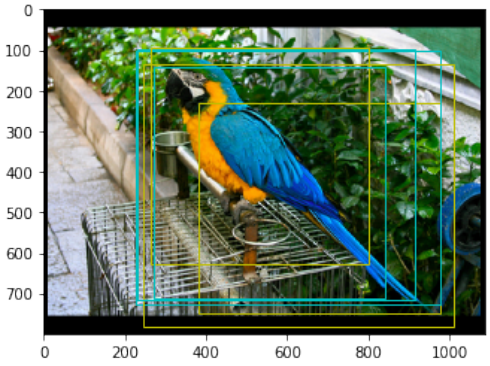
Yellow: RPN proposals, Cyan: Regressor refinement



Regressor refinement

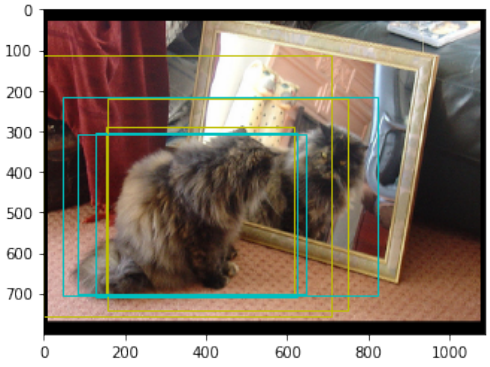


GT

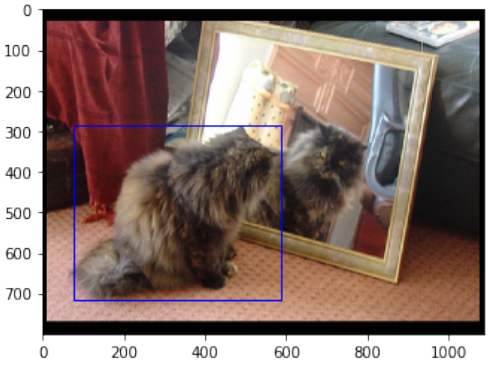


Regressor refinement

GT

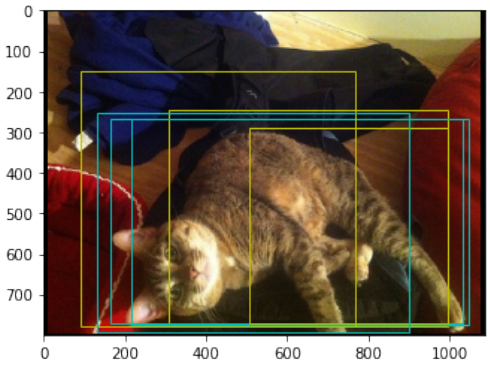


Regressor refinement

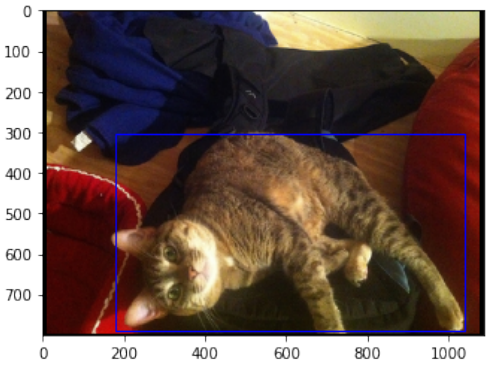


GT

Regressor finding object, underestimated by RPN

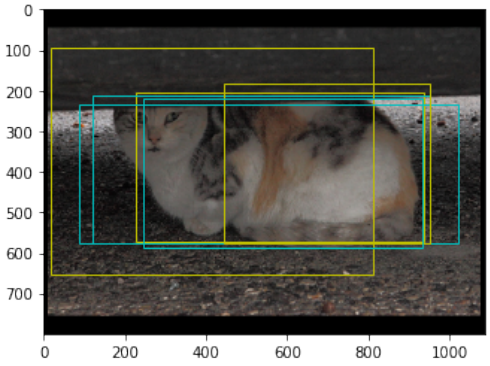


Regressor refinement

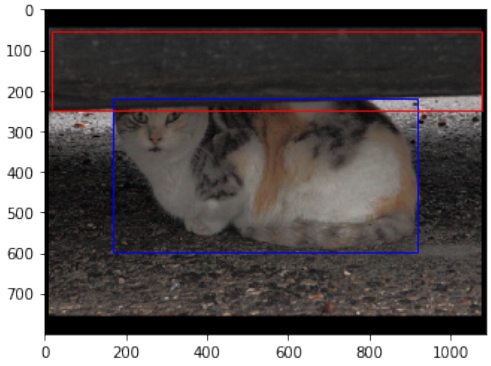


GT

If we look at the proposal box, which only covers only half of the cat (the right most box). Regressor was able to scale it to cover the entire body.



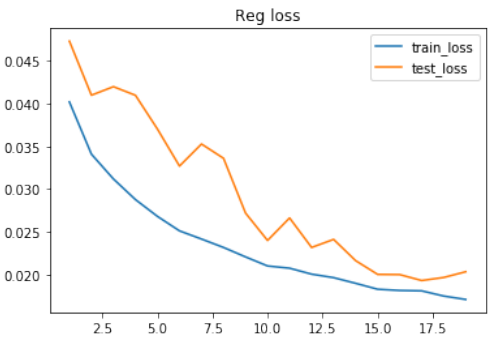
Regressor refinement



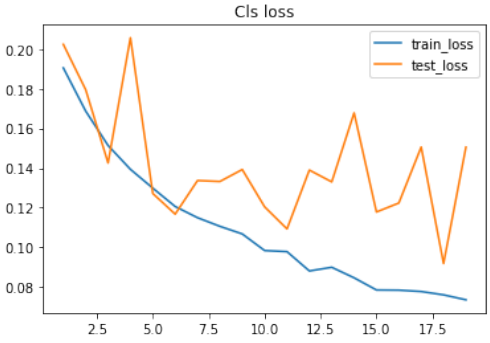
GT

Similar case. The right most yellow box (proposal) only covers half of the cat. Regressor was able to stretch it and cover the entire cat.

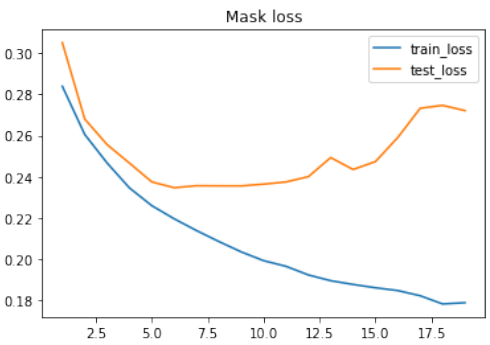
Part 16: Mask Training



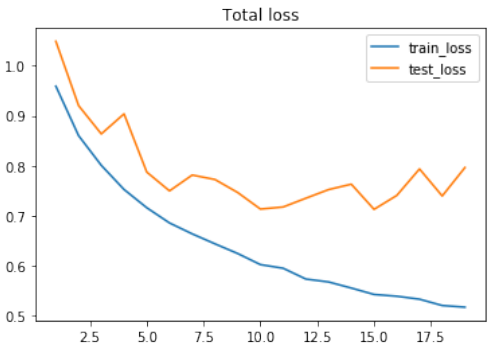
Regression Loss



Classifier Loss



Mask Loss



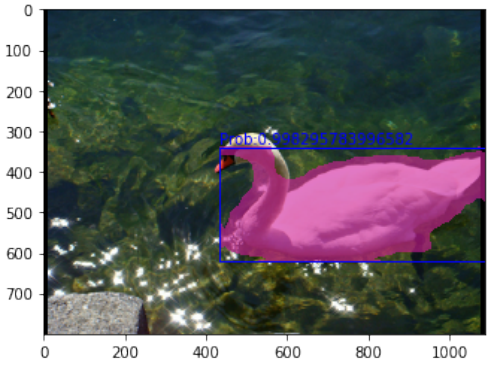
Total Loss

Learning rate used: 0.0001. Training is in batch. Waited till 128 mask outputs are obtained. All the inputs/outputs till then are then batched. Total loss is computed using 5* Reg loss + class loss + 2*Mask loss

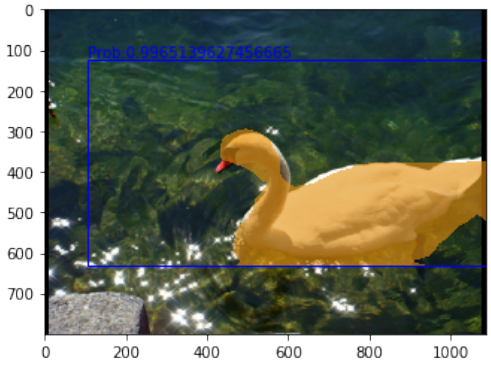
Part 17: Image reconstruction

Note: Just to avoid overlapping masks, each mask is plotted in different image for easier understanding

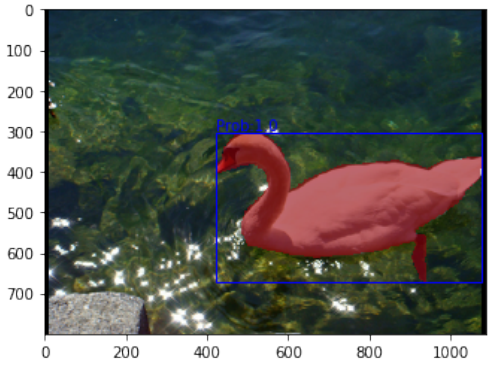
Example 1:



Mask 1

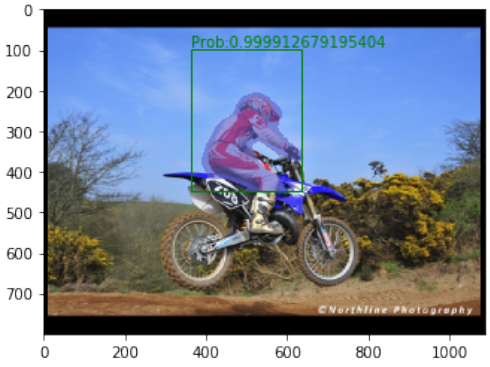


Mask 2



GT

Example 2:



Mask 1

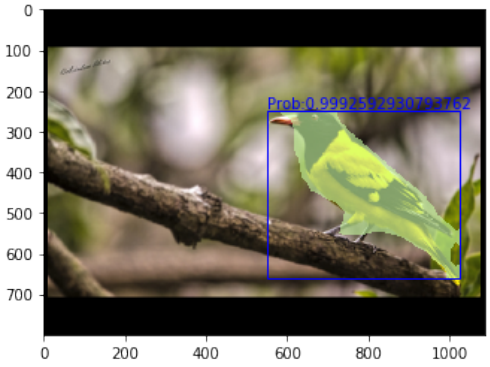
Mask 2

Mask 3

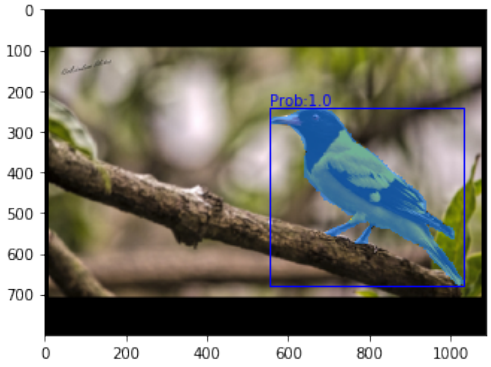
Mask 4

GT

Example 3:

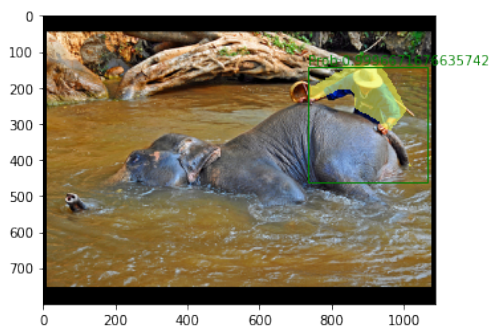


Mask 1

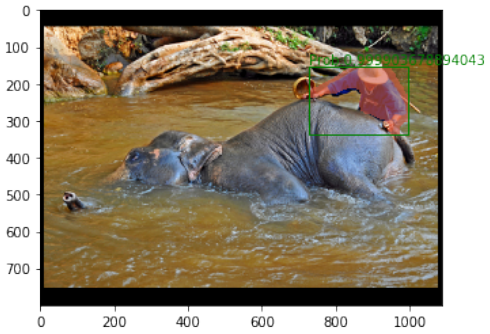


GT

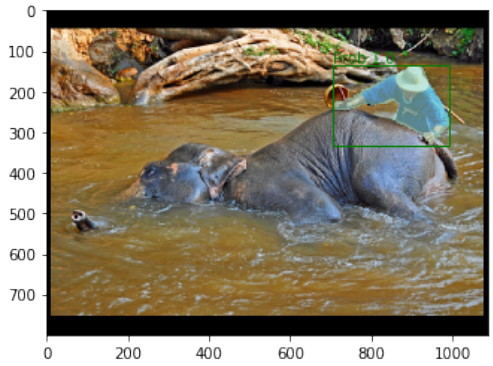
Example 4:



Mask 1

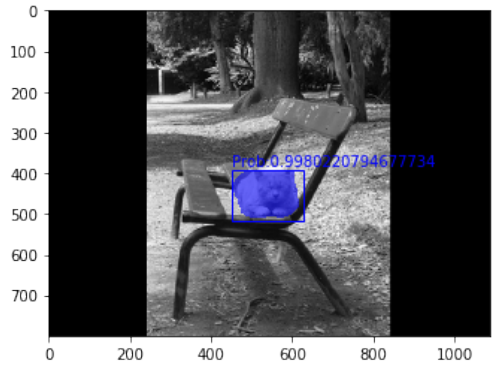


Mask 2

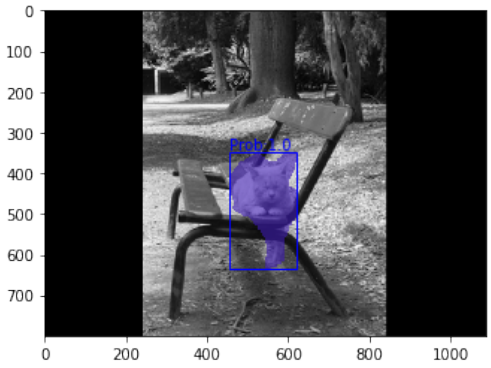


GT

Example 5:



Mask 1



GT

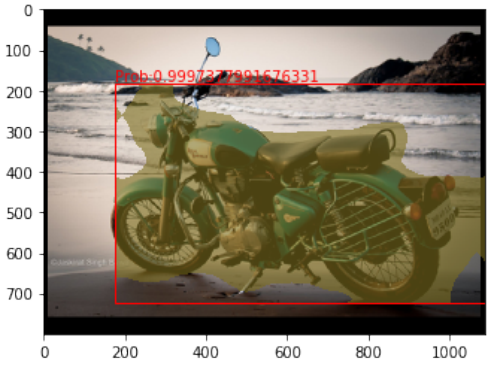
Example 6:

Mask 1

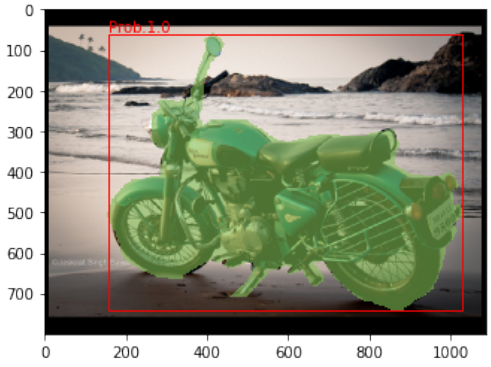
Mask 2

GT

Example 7:



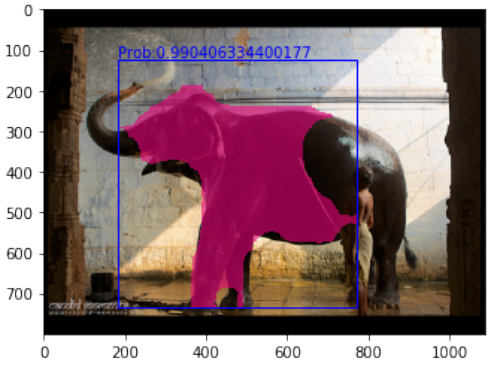
Mask 1



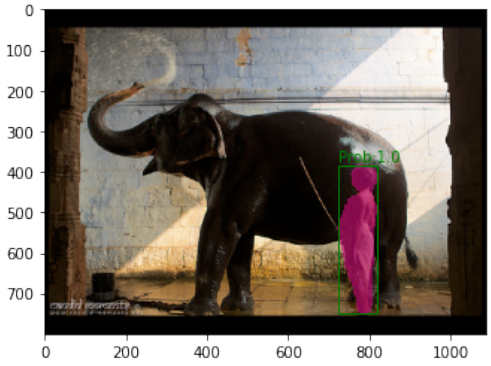
GT

Predicting objects which are not in ground truth

Example 1: Predicting Elephant

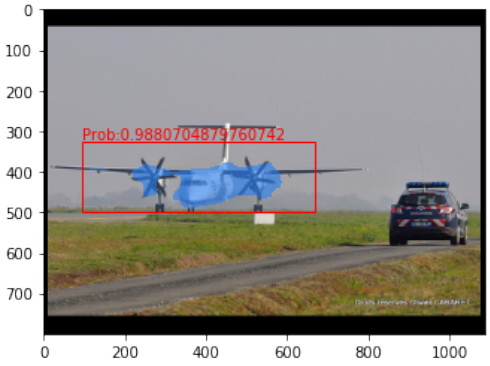


Mask

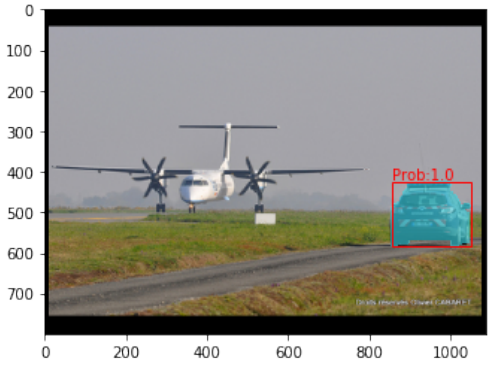


GT

Example 2: Predicting Aeroplane

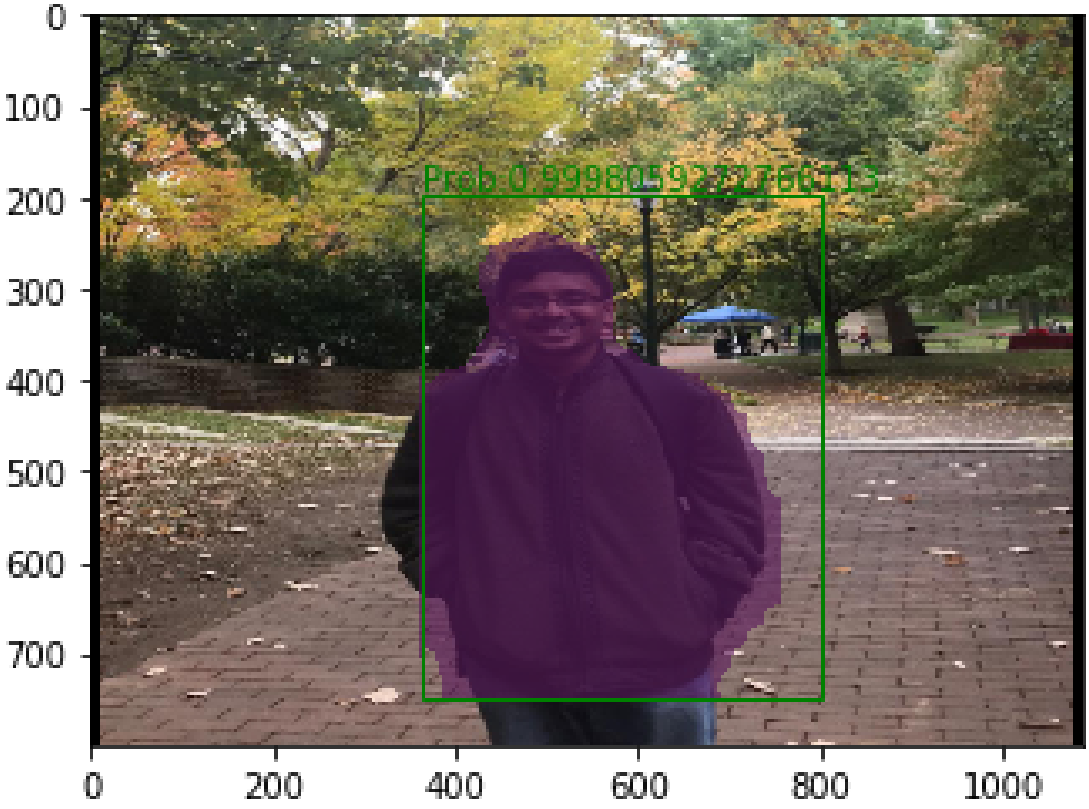


Mask



GT

Part 18: Own image



Self