Individual Report

Danlei Qian

1. Introduction

Object detection is a computer vision technique aiming to detect objects of different classes, which locates the presence of an object in an image and draw a bounding box around that object. It is also an interdisciplinary technique which has been applied widely in many areas. In this project, we would like to apply one of the Object detection algorithm--Faster R-CNN to detecting the state of malaria infection from the blood cells images.

2. Description of your individual work

I'm mainly responsible for the exploring and explaining the Faster R-CNN principles. And I was intended to train VGG16 as pretrained model (we use Resnet50 in report) and make a comparison between these two pretrain models, but finally gave up since our model is too time-consuming.

Since our codes are mostly learnt from GitHub, there are not much work on writing the original codes. But there are many problems appeared when we wanted to implement the prebuilt repository. Thus, I spent most of my time understating the logic of codes and dealing with the environment configuration problems.

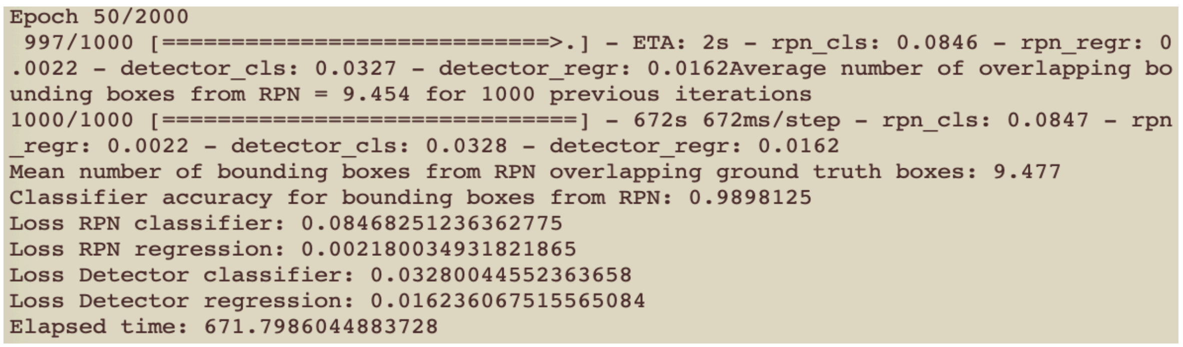
3. Describe the portion of the work that you did on the project in detail

In model training work, I successfully implement the model with pre-trained Resnet50 weights and learnt logics of the codes. And as mentioned before, I was intended to train VGG16 as pretrained model, but VGG16 cannot be loaded for some unknown reasons. Since each epoch needs over 20 minutes and I gave up this after many attempts.

In presentation, to explain the theory well, I perused the whole original paper for several times and did some 'deep learning' on this theory.

Besides, I'm responsible for writing part 1, 2 and 5 in the final report and organizing the report.

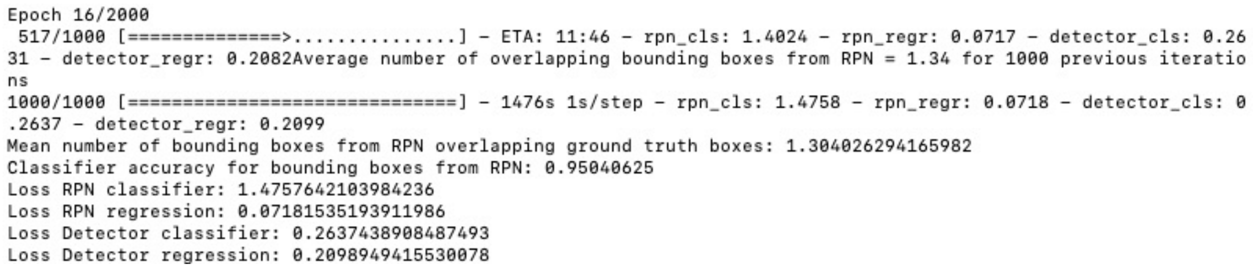
4. Results



```
Epoch 50/2000
 997/1000 [============================>.] - ETA: 2s - rpn_cls: 0.0846 - rpn_regr: 0
.0022 - detector_cls: 0.0327 - detector_regr: 0.0162Average number of overlapping bo
unding boxes from RPN = 9.454 for 1000 previous iterations
1000/1000 [==============================] - 672s 672ms/step - rpn_cls: 0.0847 - rpn
_regr: 0.0022 - detector_cls: 0.0328 - detector_regr: 0.0162
Mean number of bounding boxes from RPN overlapping ground truth boxes: 9.477
Classifier accuracy for bounding boxes from RPN: 0.9898125
Loss RPN classifier: 0.08468251236362775
Loss RPN regression: 0.002180034931821865
Loss Detector classifier: 0.03280044552363658
Loss Detector regression: 0.016236067515565084
Elapsed time: 671.7986044883728
```

Figure1

Figure 1 shows the result of our final model with pretrained Resnet50 model weights and 50 epochs.



```
Epoch 16/2000
 517/1000 [==============>...............] - ETA: 11:46 - rpn_cls: 1.4024 - rpn_regr: 0.0717 - detector_cls: 0.26
31 - detector_regr: 0.2082Average number of overlapping bounding boxes from RPN = 1.34 for 1000 previous iteratio
ns
1000/1000 [==============================] - 1476s 1s/step - rpn_cls: 1.4758 - rpn_regr: 0.0718 - detector_cls: 0
.2637 - detector_regr: 0.2099
Mean number of bounding boxes from RPN overlapping ground truth boxes: 1.304026294165982
Classifier accuracy for bounding boxes from RPN: 0.95040625
Loss RPN classifier: 1.4757642103984236
Loss RPN regression: 0.07181535193911986
Loss Detector classifier: 0.2637438908487493
Loss Detector regression: 0.2098949415530078
```

Figure2

And Figure 2 shows the result of model without loading pretrained model weights. Although it is the result of only 16 epochs, it seems that the result won't be improved. Thus, we can still learn that the pretrained model will make a big difference.

5. Summary and conclusions

In this project, we dig deeper in one of the applications of deep learning. Object detection is a big topic and it takes a lot of work to understand one object detection algorithms thoroughly. Without understanding the principle, it hard to implement the codes. And environment configuration is also important which can result in many unnecessary troubles. And I learnt a lot from how to realize the theory into codes and how to set up the environment.

6. Calculate the percentage of the code that you found or copied from the internet

Since our codes are mostly learnt from GitHub, I don't calculate the percentage.

7. References

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in NIPS, 2015.[2] https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/