Individual Final Report

Ya Liu

1. Introduction

Our project is mainly about detecting malaria cells. The dataset we use is the one used in exam 2, and the algorithm we implement is Faster R-CNN. Chao Zhou and I are mainly responsible for the coding part, and Danlei Qian is mainly responsible for illustration of the theory of Faster R-CNN.

2. Individual Work

Here is what I did following the process of the project. After we finalized the topic, I found an implementation online and transformed our data into the required format of input. Working together with Chao Zhou, we found out the setup of python packages required but not explained in the Github, and fixed the bugs when training the model. Since the runtime was too long(about 3 hours per epoch), we chose only to detect cells except red blood cells, which significantly reduced the runtime to 20 mins per epoch. During this process, I also read the Faster R-CNN theory.

After we got our model training, I prepared the part 1- 4 of the presentation, and organized the ppt. Besides, I wrote the part 3 and 4.1 of the report, illustrating the theory of the Faster R-CNN.

3. Results

There are some results of my work.

Figure 1 shows the input that I got.

```
/cells_1017.png,711,1105,844,1225,trophozoite
/cells_1017.png,761,972,891,1101,trophozoite
/cells_1017.png,907,53,1004,179,trophozoite
/cells_1017.png,195,176,315,292,trophozoite
/cells_1003.png,384,988,547,1153,difficult
/cells_218.png,463,621,603,750,trophozoite
/cells_218.png,320,1236,475,1374,trophozoite
/cells_230.png,172,941,311,1101,difficult
/cells_224.png,882,873,965,958,trophozoite
/cells_224.png,218,282,321,373,trophozoite
/cells_224.png,229,761,326,857,trophozoite
/cells_595.png,293,501,380,635,trophozoite
```

Figure 1. Input

Figure 2 illustrates the last epoch when training the model. There are four losses because we have one classifier and one regression in both RPN and the last part of the network.

```
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
```

Figure 2. Training the Model

4. Summary and Conclusion

In this project, we applied Faster R-CNN to the cells dataset. We used the network implemented online, but there were many bugs coming up when training the model. To fix the bug, it depends on how well you know the theory and the code so that you would know how to fix it. We spent a lot of time training the model, and diving into the theory. It deepens my understanding that these two things, theory and coding, are crucial and dependent with each other.

5. Percentage of Code Calculation

The total number of code is about 100, and the number of code that I found online and modified to meet our needs is 80. So the percentage of code is $\frac{100-80}{100} = 20\%$.

6. References

R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1440-1448.
Ren, S.; He, K.; Girshick, R. B. & Sun, J. (2015), Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks., in Corinna Cortes; Neil D. Lawrence; Daniel D. Lee; Masashi Sugiyama & Roman Garnett, ed., 'NIPS' , pp. 91-99.

Some online resources:
https://zhuanlan.zhihu.com/p/31426458
https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/
https://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/
https://github.com/kbardool/keras-frcnn.git