Exam Report

Yuan Le

In this exam, I started with a simple CNN with 64 kernels as the first COV layer, and 256 kernels as the second layer. The output of the cov layer is flattened and feeded into a fully connected layer, with 7 output. The activation function for the cov layers are RELU, and since I used BCEWithLogitsLoss as the loss function, a linear activation function is used in the last fc layer.

The results of the simple CCN are not very well, and the model overfits pretty quickly. So I had several attempts, including use FocalLoss and down-weight the red blood cell label, rotate and flip the input pictures for data augmentation, normalize the picture input by using the mean and variance from ImageNet dataset. The normalization and data augmentation does reduce the overfitting problem, but the loss on my validation set is still high.

I then changed my network structure and tried to use a pre-trained network with some tuning. I chose the Resnet50 which is included in the torchvision package, and I had two attempts. I tried to load the weights from the pre-trained weight, and freeze those parameters in hidden layers, so only the parameters in the last fully connected layer are trained from our input. I also tried to train the whole network with our input, and in this particular problem, it seems that the pre-trained weights are not well suitable. A guess for the reason is that the original dataset used for the pre-trained weights are the ImageNet dataset, which is not similar to our input.

In the last few days, I also tried to tackle the problem as a bounding box detection problem, and to predict the class for each detected bounding box. I tried to use the faster-rcnn which is also included in the torchvision package, but I had a difficult time in debugging it, and I failed to fix the code to have a prediction. Then I spent some time on reading papers and started to implemented a network called DarkNet, which is the core network of an object detection algorithm called YOLO (You Only Look Once), by following some tutorials I found online. Up to the last day, the network worked partially, as I was able to convert the training labels into YOLO style labels, and train the network with our training set, and make predictions about the bounding boxes and classes, but the performance of the prediction varies among different pictures. More specific, when the objects are not too dense, my output looks accurate enough, but when the cells are crowded and overlapping with each other, the results are poor. Due to lack of time, I was not able to finish this part to get solid results, but below is a demonstration of the predicted bounding boxes and the corresponding labels and confidential probabilities.

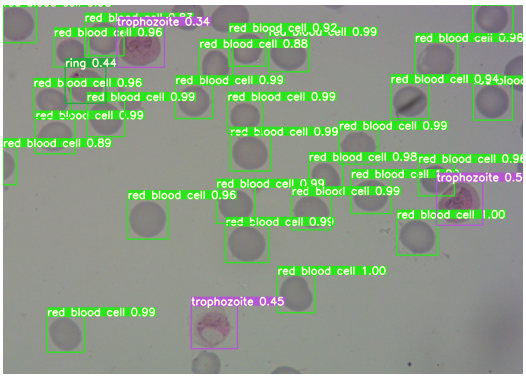

Fig. 1: Predicted Bounding-box.