Part1:
For this part, I just read the disparity map for each image and importing kitti_datahandler to read stereo calibration. Then I used the formula z = f*B/disp to get the depth map and assigned depth 0 to those with depth <0.1 m and > 80m. The result is in est_depth.
Running command would be “python part1_estimate_depth.py test”.

Part2:
After understanding and experimenting the parameter “confidence_th” and “threshold”. I finally chose “confidence_th” = 0.1 and “threshold” = 0.6, which can recognize most objects accurately but don't have many duplicated bounding box for the same object identified. Having a lower “confidence_th” allows me to recognize more object at the beginning and then using proper threshold 0.6 to remove redundant bounding boxes for the same recognized object during non-maximum suppression.
Running command is “python part2_yolo.py test”.

Following are estimated bounding box in the test set
000011:

000012:

000013:

000014:

000015:

Part3:

After numerous experiments, I finally chose distance range as 10. Which means I will mask the pixels inside car bounding box with the depth in (average_depth-10, average_depth+10). For the calculation of average_depth, I remove the 0 value's weights so it won't be affected by 0 depth a lot. The measurement of accuracy is precision and recall and also F score that combines those two together. The code of measurements is in part3_segmentation.py. In this way, it achieves overall above 80% precision and recall.
The result is in est_seg.
Running command is "python part3_segmentation.py test".