3D Object Detection Midterm Project

Report by: **Nolan Lunscher**
Report Submitted: **2021-10-03**
Repo: https://github.com/nlunscher/nd013-c2-fusion-starter (branch: devel-nl)

Introduction **1**

Data Visualization **2**
ID_S1_EX1 2
ID_S1_EX2 2
Viewing Vehicles with Varying Visibility 2

Converting from Point cloud to BEV Representation **5**
ID_S2_EX1 5
ID_S2_EX2 5
ID_S2_EX3 6

Running Inference **7**
ID_S3_EX1 and ID_S3_EX2 7

Measuring Performance **8**
ID_S4_EX1, ID_S4_EX2 and ID_S4_EX3 8
Darknet 8
Fpn Resnet 9

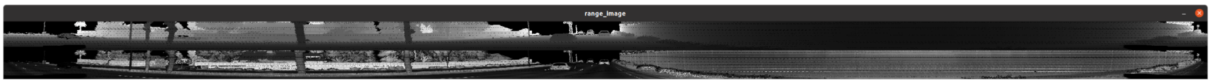
Introduction

In this project, we explore the use of BEV based 3D object detection for use with 3D lidar sensor data. We utilize algorithms adapted from image based object detection, and show how they can be applied to 3D data.

Data Visualization

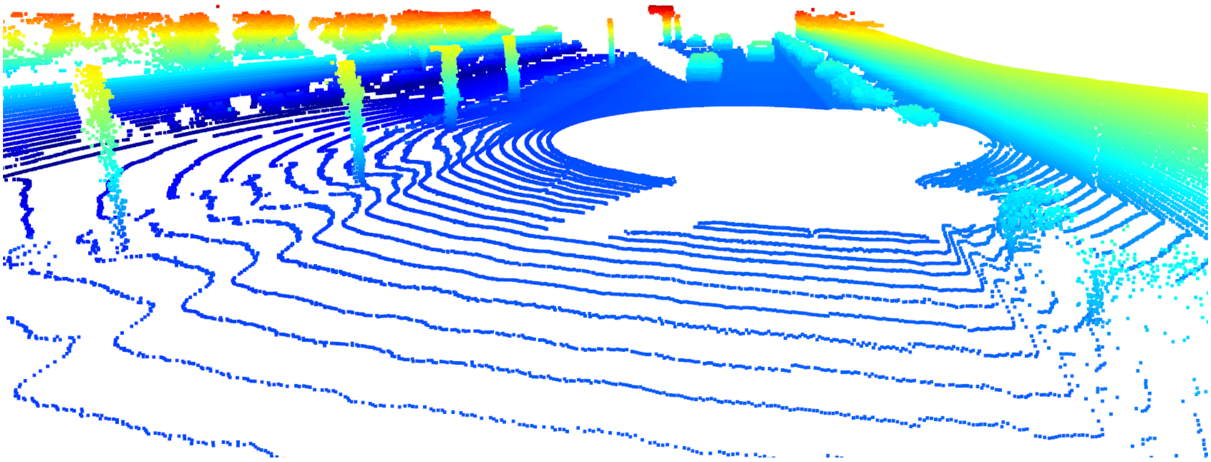
ID_S1_EX1

The output image created using show_range_image:



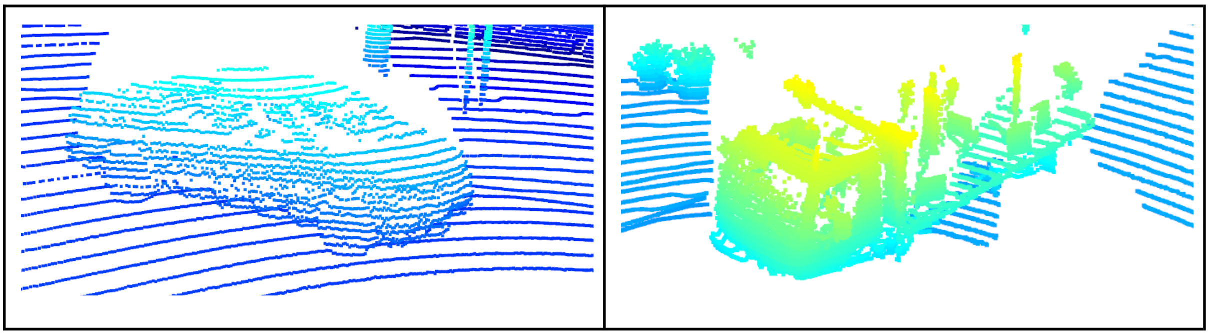
ID_S1_EX2

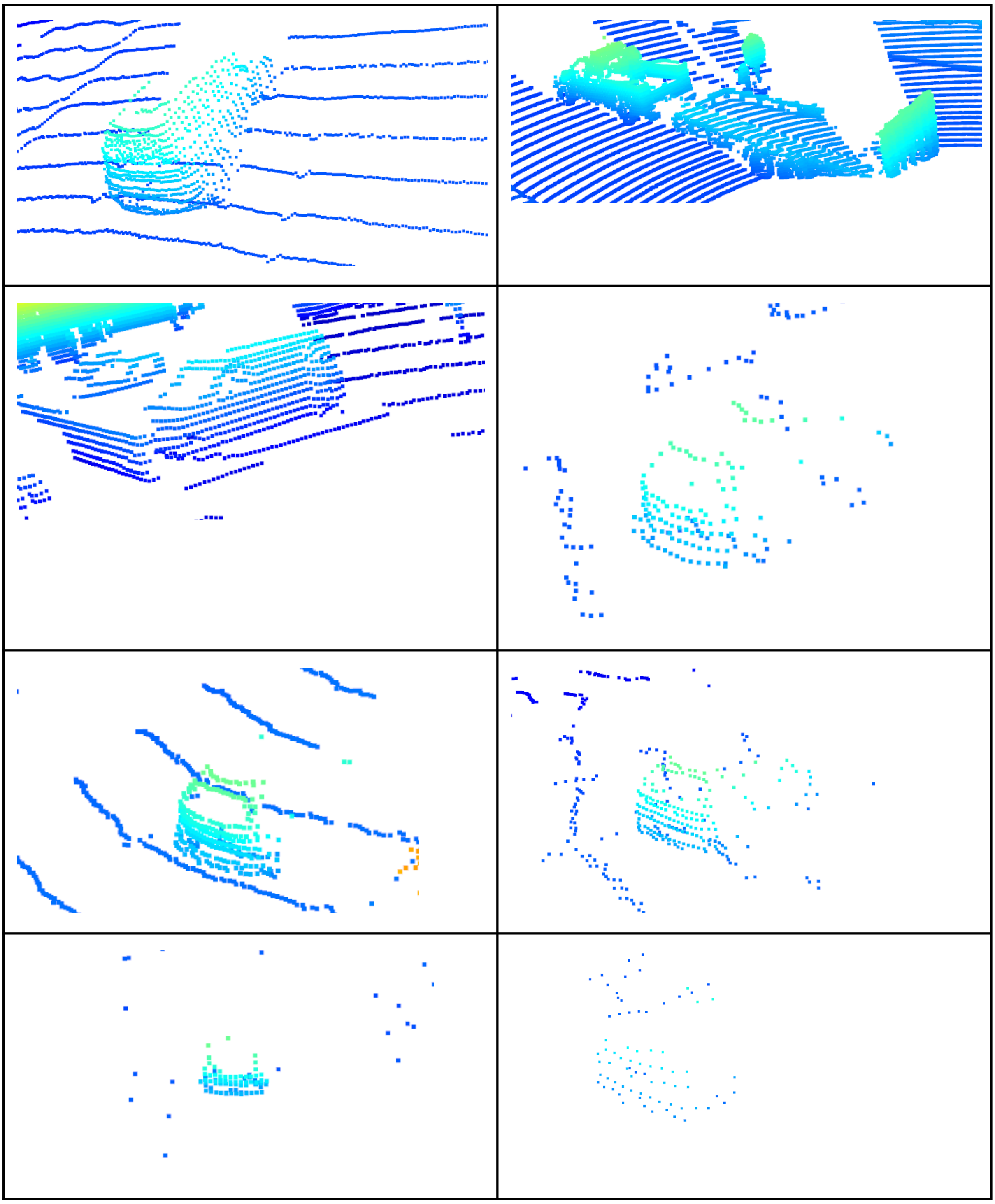
The output point cloud using show_pcl:



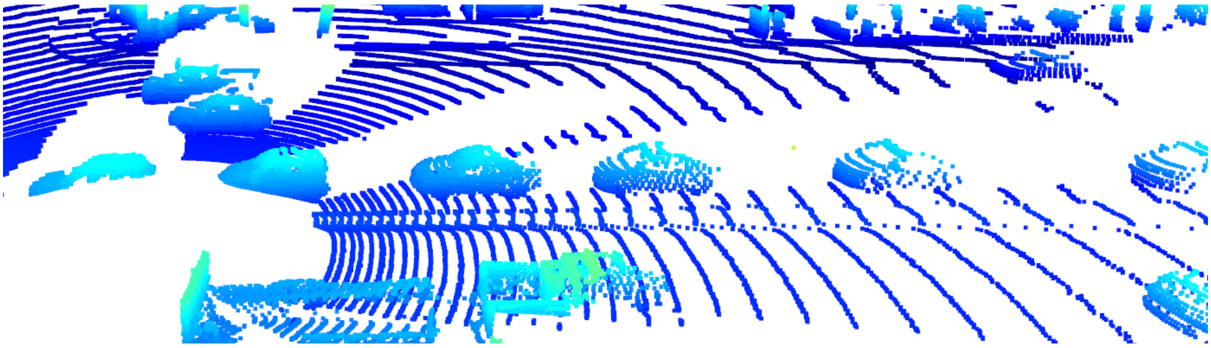
Viewing Vehicles with Varying Visibility

Below are 10 examples of vehicles that have varying degrees of visibility. Images are ordered roughly in descending order from most distinctly visible to least. Visibility largely seems to be a function of distance to the sensor, where the point cloud becomes more and more sparse. In addition to this, visibility can be affected by occluding objects creating shadows in the [point cloud.





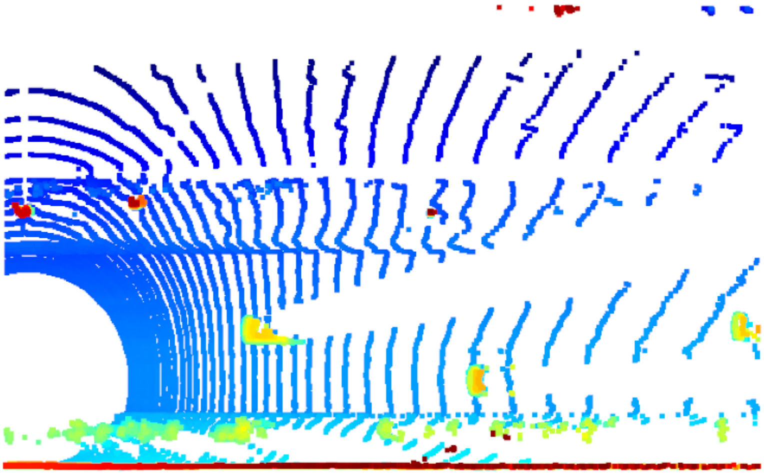
This example from sequence 3 shows a variety of vehicles with visibility decreasing with distance.



Converting from Point cloud to BEV Representation

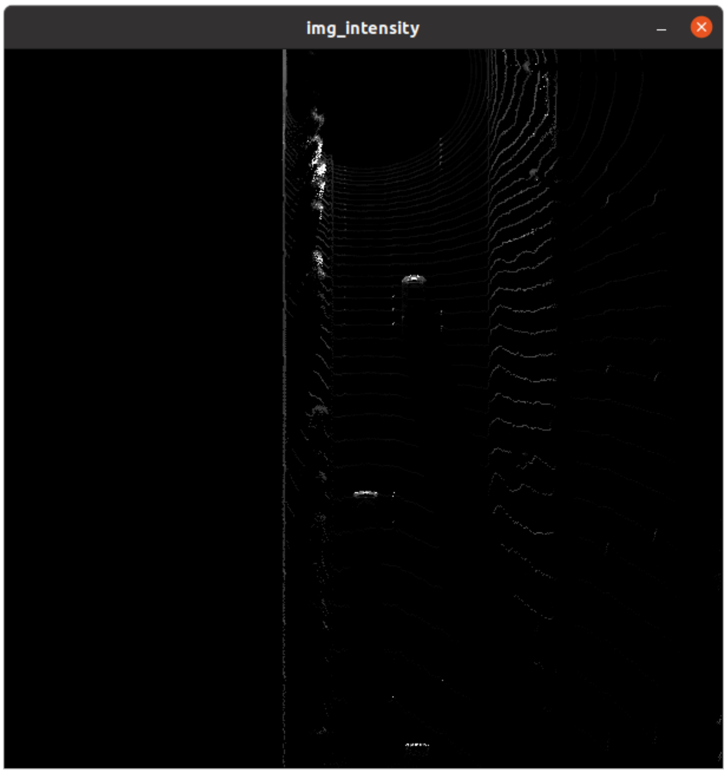
ID_S2_EX1

The output BEV point cloud using bev_from_pcl:



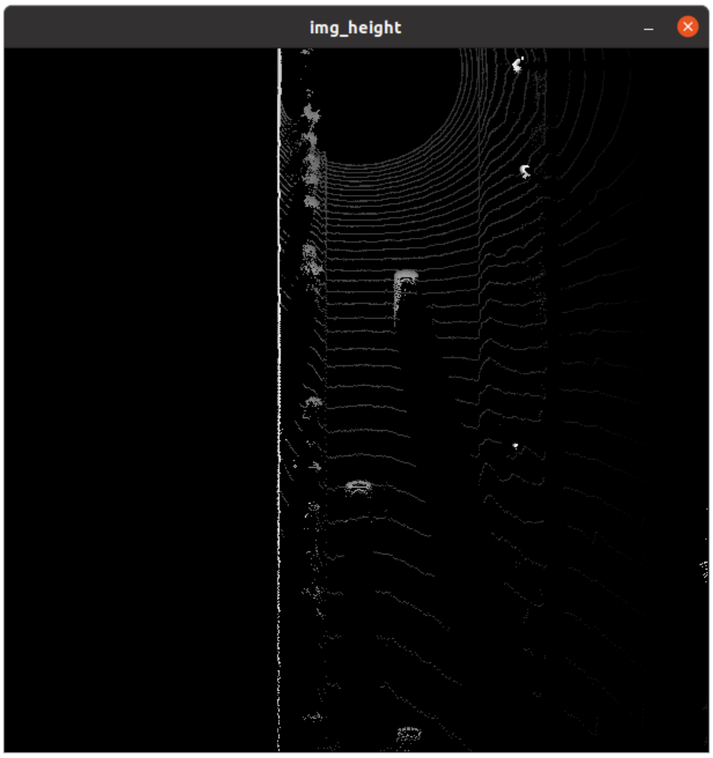
ID_S2_EX2

The output intensity map using bev_from_pcl:

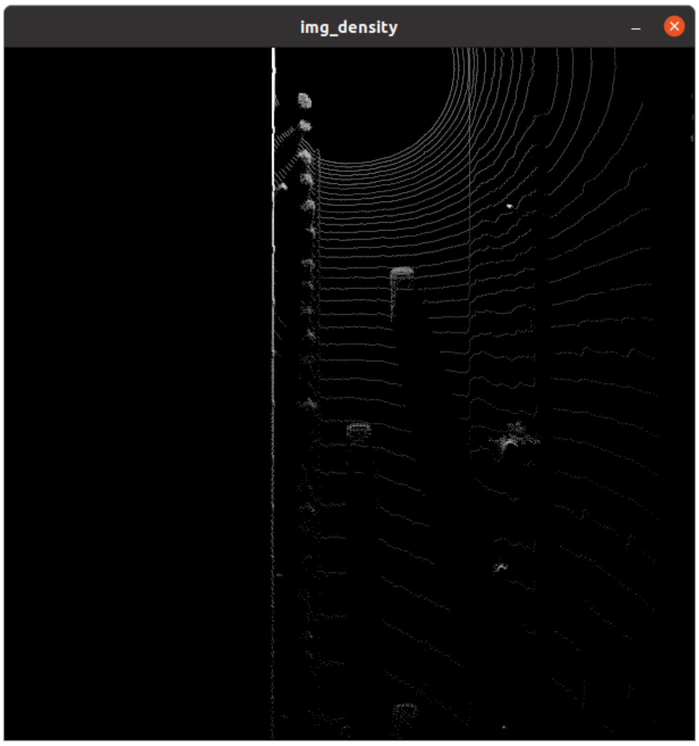


ID_S2_EX3

The output height map using bev_from_pcl:



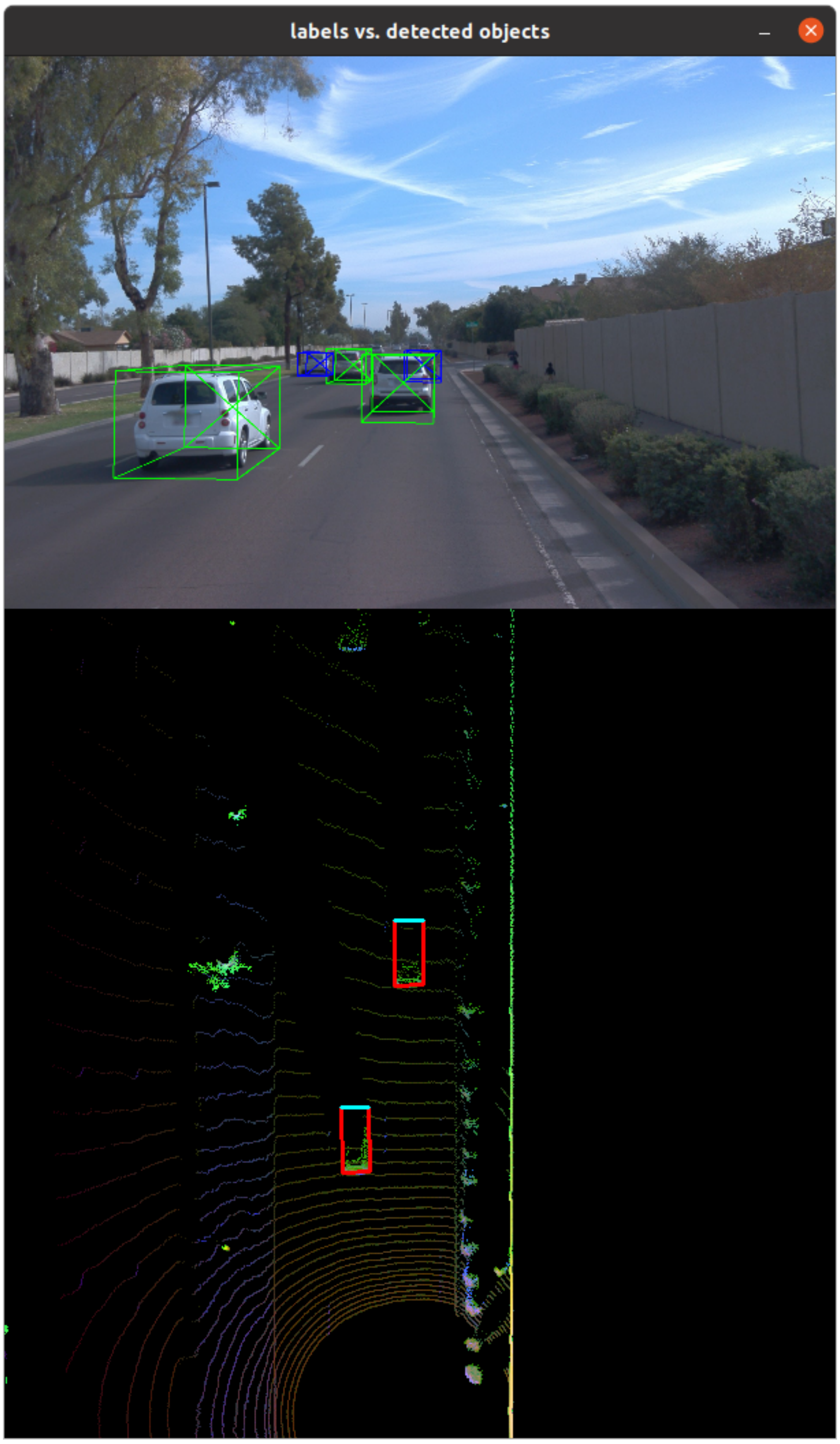
The output density map using bev_from_pcl:



Running Inference

ID_S3_EX1 and ID_S3_EX2

The fpn_resnet detections from detect_objects visualized:

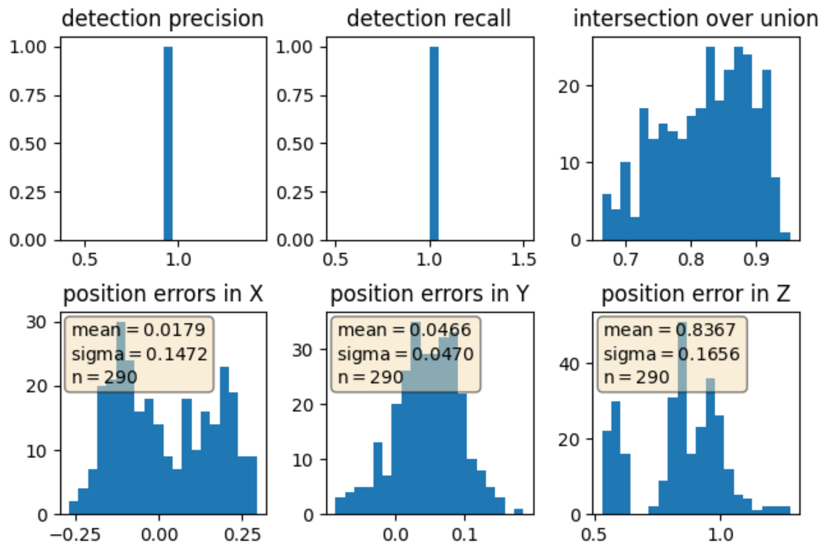


Measuring Performance

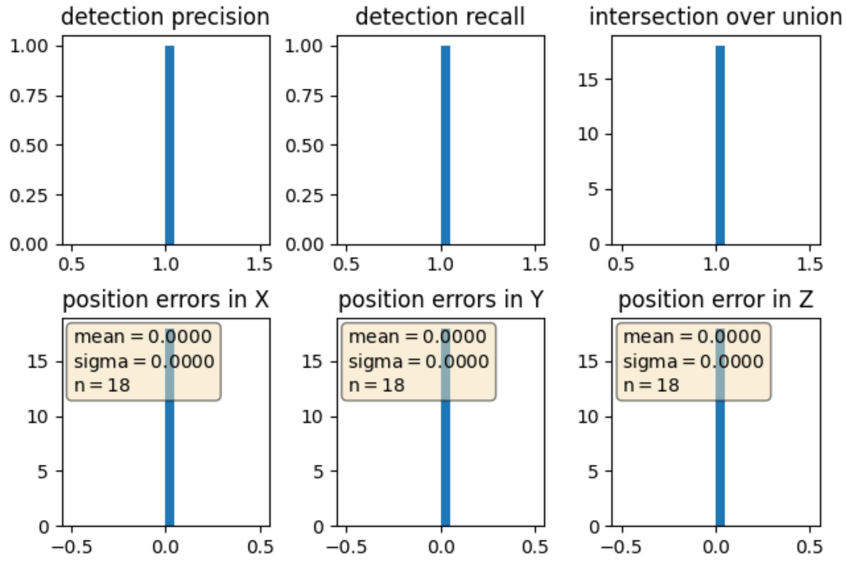
ID_S4_EX1, ID_S4_EX2 and ID_S4_EX3

Darknet

The performance metric plots when using the Darknet model:
precision = 0.9206349206349206, recall = 0.9477124183006536

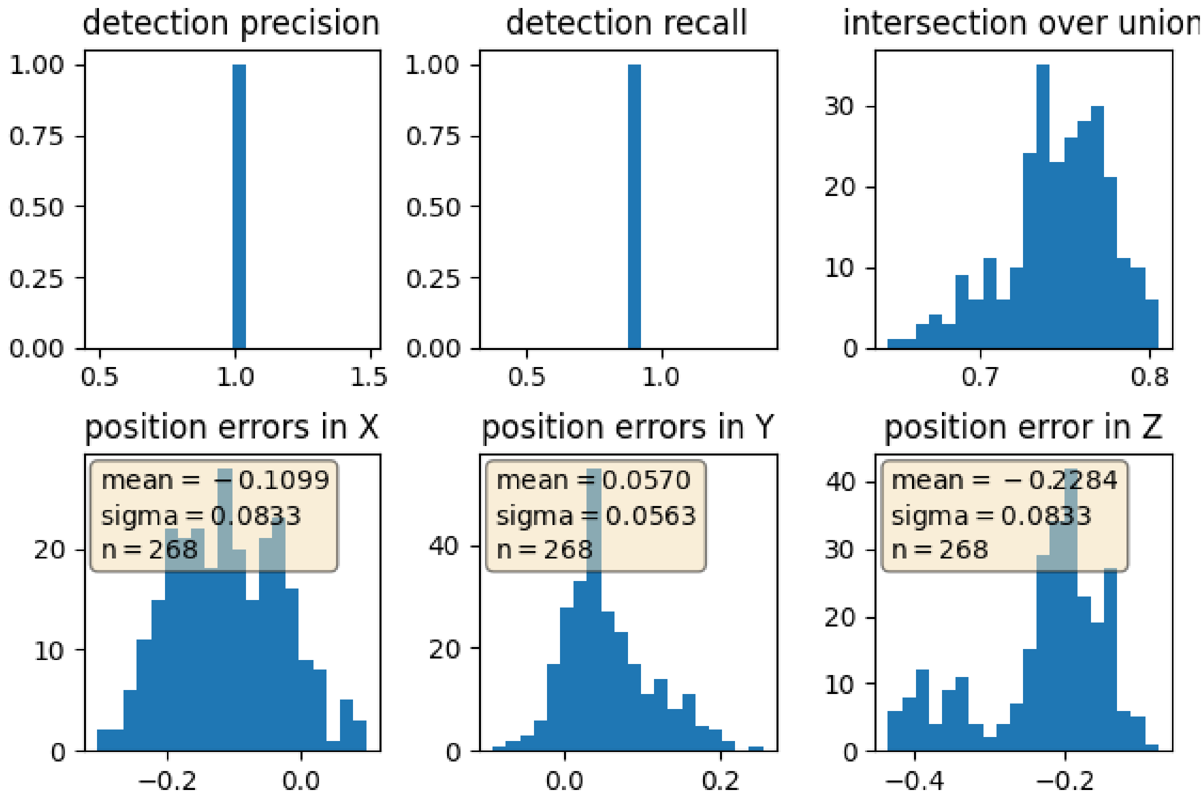


The performance metrics plots when using the labels as the detected objects (perfect detector):
precision = 1.0, recall = 1.0



Fpn Resnet

The performance metric plots when using the Fpn_resnet model:
precision = 0.988929889298893, recall = 0.8758169934640523


detection precision
detection recall
intersection over union
position errors in X
mean = −0.1099
sigma = 0.0833
n = 268
position errors in Y
mean = 0.0570
sigma = 0.0563
n = 268
position error in Z
mean = −0.2284
sigma = 0.0833
n = 268