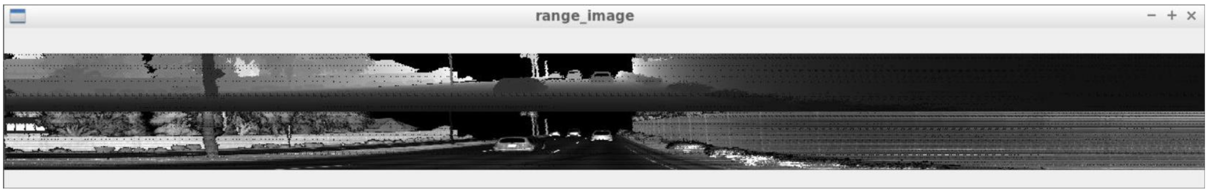
3D Object Detection Mid-term

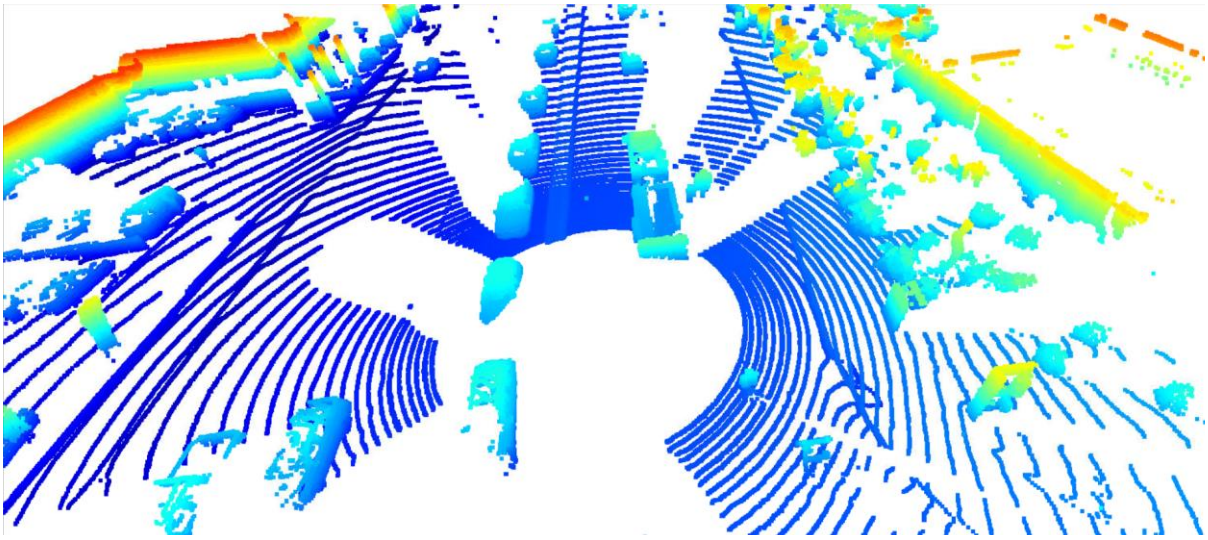
https://github.com/LimHaeryong/3D-Object-Detection

Project Instructions Step 1

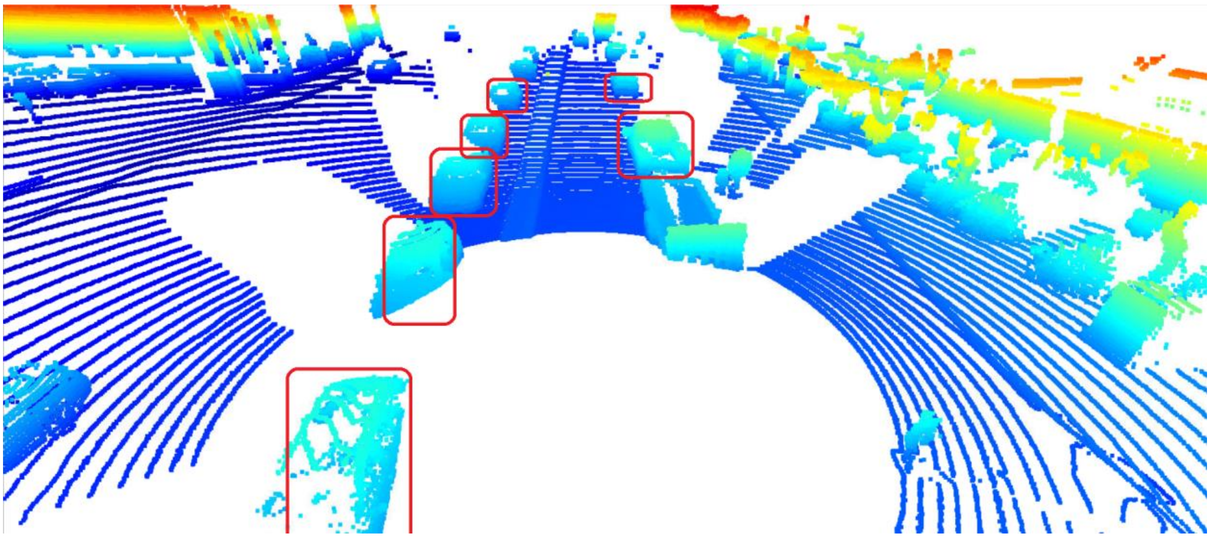
Visualize range image channels(ID_S1_EX1).

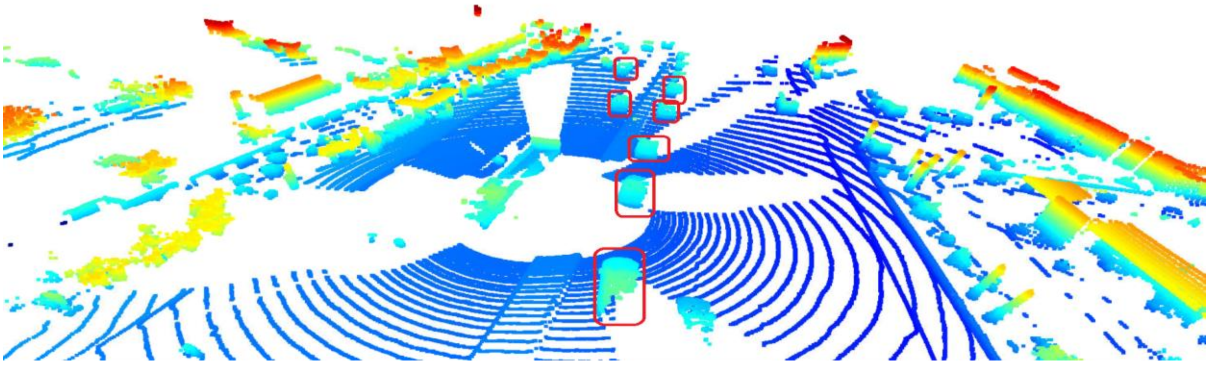


Visualize point-cloud(ID_S1_EX2)



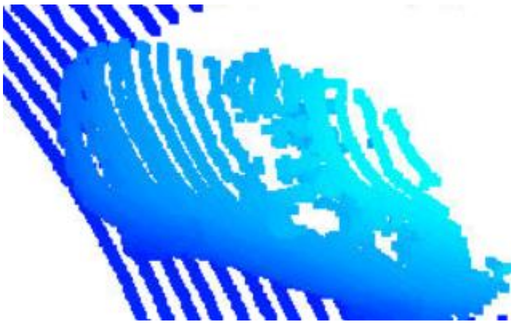
-Find 10 examples of vehicles with varying degrees of visibility in the point-cloud.





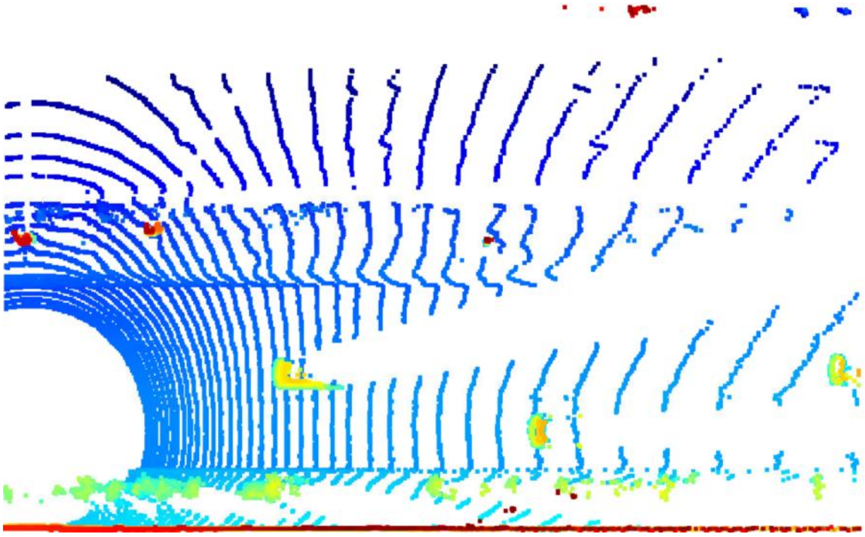
-Try to identify vehicle features that appear stable in most of the inspected examples and describe them.

We can identify vehicle features like front-bumper, wheels and windshield from the point cloud image.

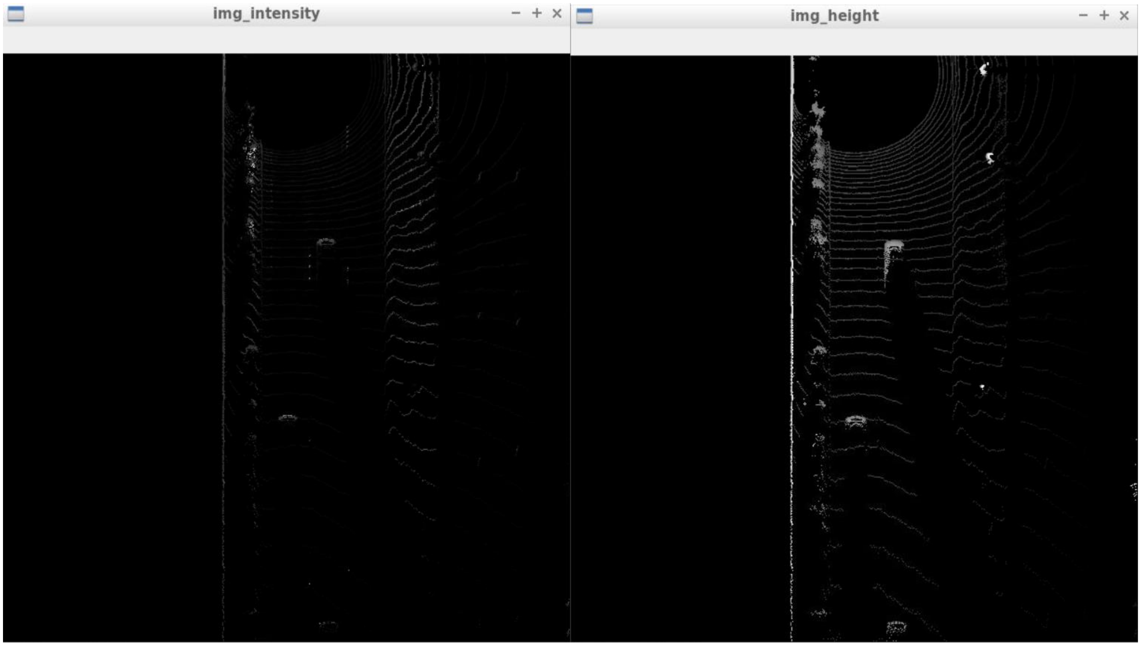


Project Instructions Step 2

Convert sensor coordinates to bev-map coordinates(ID_S2_EX1)

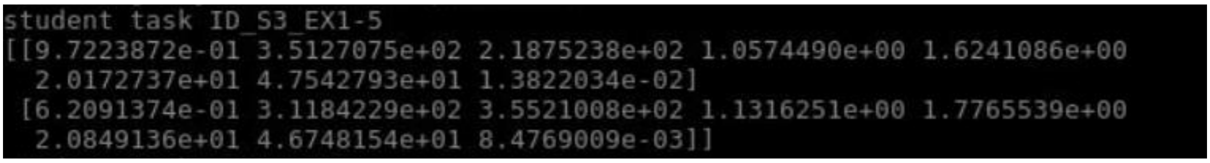


Compute intensity layer and height layer of bev-map(ID_S2_EX2, 3).



Project Instructions Step 3

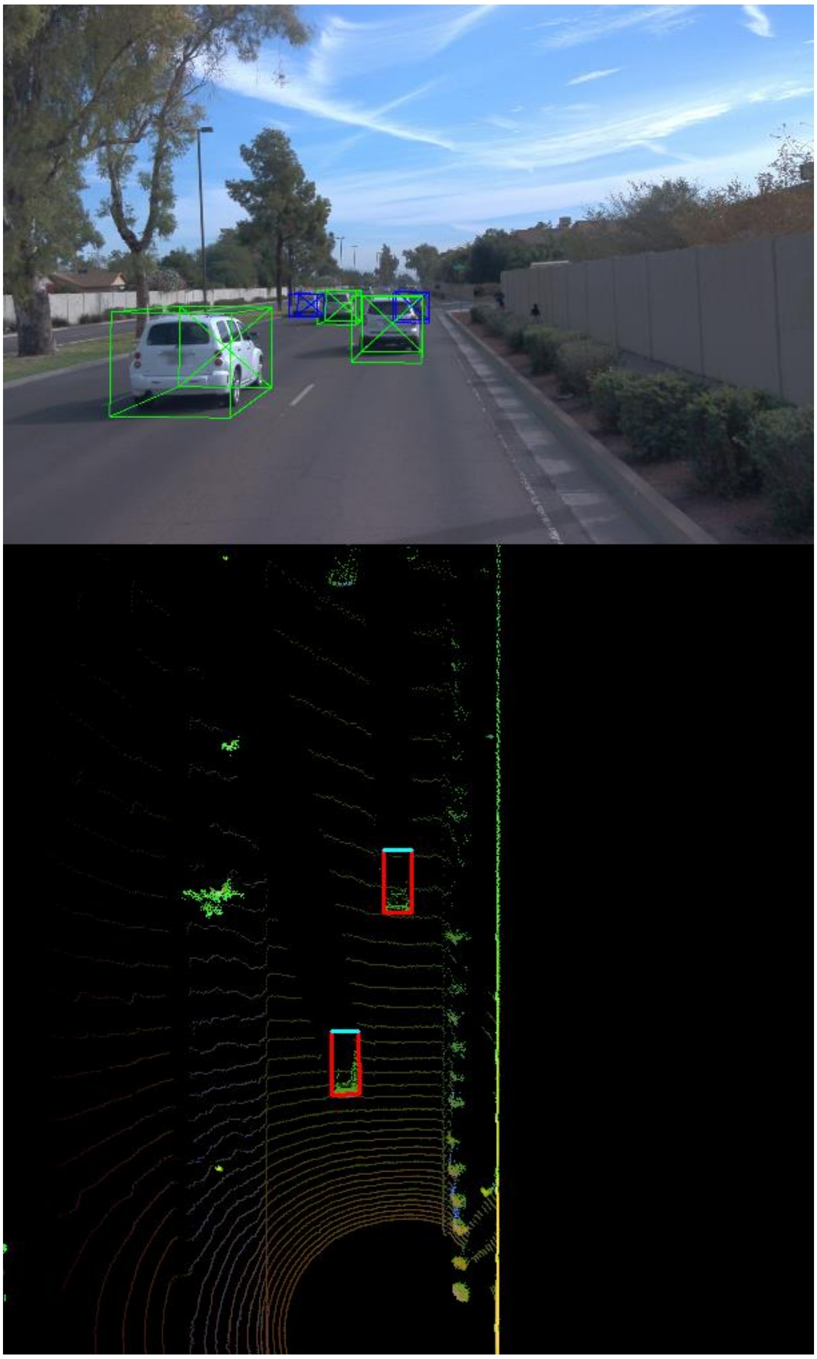
Add a second model from a GitHub repo(ID_S3_EX1)



```
student task ID_S3_EX1-5
[[9.7223872e-01 3.5127075e+02 2.1875238e+02 1.0574490e+00 1.6241086e+00
  2.0172737e+01 4.7542793e+01 1.3822034e-02]
 [6.2091374e-01 3.1184229e+02 3.5521008e+02 1.1316251e+00 1.7765539e+00
  2.0849136e+01 4.6748154e+01 8.4769009e-03]]
```

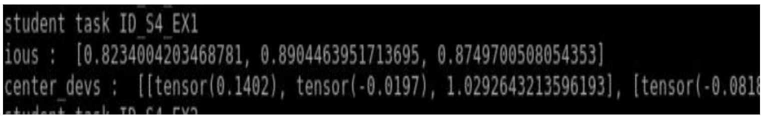
Output of detections.

Extract 3D bounding boxes from model response(ID_S3_EX2)



Project Instructions Step 4

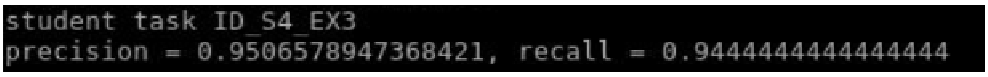
Compute IOU between labels and detections(ID_S4_EX1)



```
student task ID_S4_EX1
ious :  [0.8234004203468781, 0.8904463951713695, 0.8749700508054353]
center_devs :  [[tensor(0.1402), tensor(-0.0197), 1.0292643213596193], [tensor(-0.0818
```

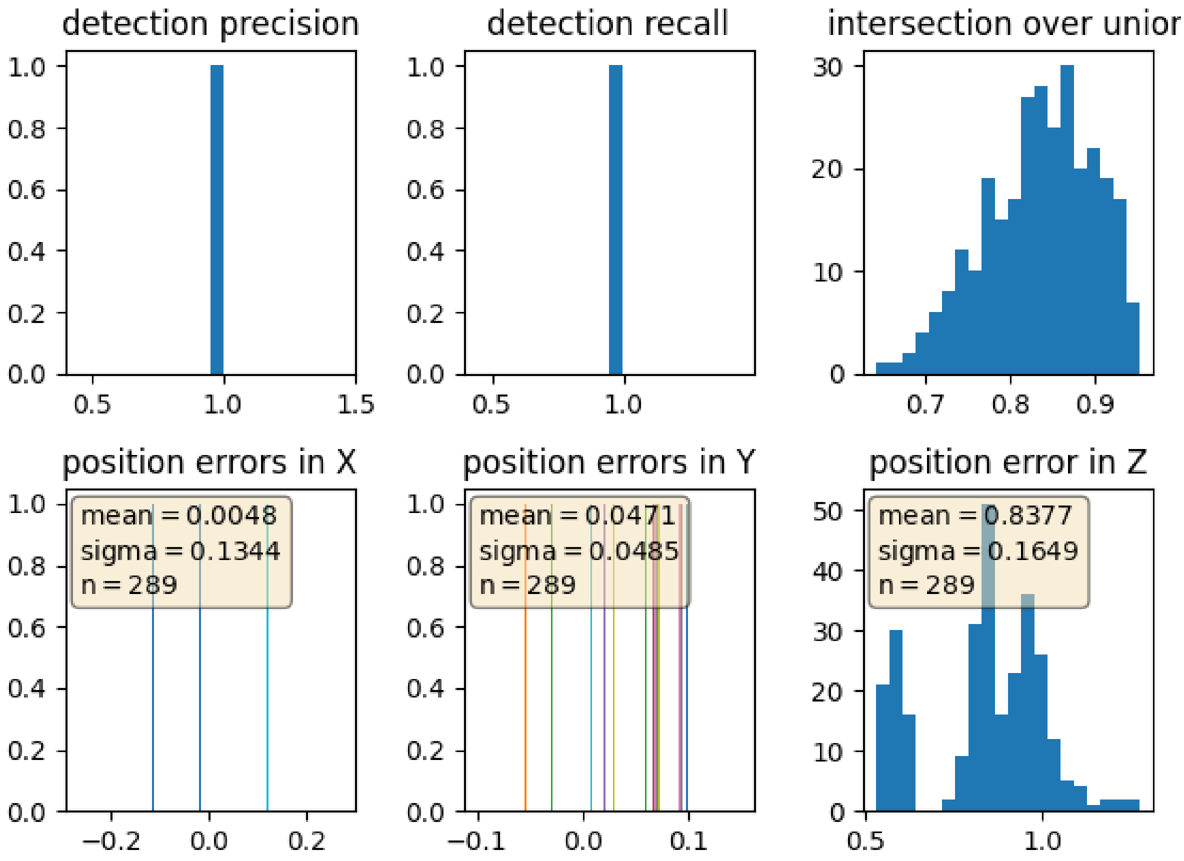
ious and center_devs

Compute precision and recall(ID_S4_EX3)



```
student task ID_S4_EX3
precision = 0.9506578947368421, recall = 0.9444444444444444
```

Precision and recall


detection precision
detection recall
intersection over unior
position errors in X
mean = 0.0048
sigma = 0.1344
n = 289
position errors in Y
mean = 0.0471
sigma = 0.0485
n = 289
position error in Z
mean = 0.8377
sigma = 0.1649
n = 289