CS 8803 : Project Interim Report
3-D Object Detection for Autonomous Vehicles

Cao, Zhehan caozhehan@gatech.edu, Hu Hu huhu@gatech.edu
Kapil Vuthoo kapilv@gatech.edu: Chen, Tianrong tianrong.chen@gatech.edu
November 8, 2019

***Abstract*—Nowadays, self-driving technology presents a rare opportunity to improve the quality of life in many of our communities. With the combination of cameras and sensors technology in a driving car, 3-D object detection is an import task for autonomous vehicles. In this project, we combine the LIDAR data, which is collected by sensors, and RGB data, which is collected by cameras, to build an end-to-end system for 3-D object detection. We research and implement the advanced models, including UNet, Point RCNN and F-ConvNet to solve this problem. The intersection over union (IoU) is used for our model evaluation. In the meantime, this project will join the Lyft self-driving competition on Kaggle. For interim report, we've finished the training and testing of UNet based method, which achieves 0.041 score on the Kaggle leaderboard. The Point RCNN based method is implemented, the relative experiments are ongoing.**

***Index Terms*—Three dimension, object detection, LaserNet, autonomous vehicles, self-driving.**

I. Introduction

3D shape models are becoming widely available and easier to capture, making available 3D information crucial for progress in object classification. In this project we shall focus on problem of 3D object detection over semantic maps. The input we have is 3D LIDAR point cloud data which is volumetric representations to the 3D object classification problem. This is primarily inspired by recent advances in real-time scanning technology, which use volumetric data representations.

The approach we are taking to address this problem is by checking various SOTA models that are available for similar kind of problems to this particular dataset and output type that not only needs the bounding boxes and object class but also yaw, as this real time output has to be fead to self driving control system. Further, due to training computation constraints we are looking for pre-trained models that we can fine tune.

Our investigation starts from the UNet based method. Which is a image segmentation based method. Each 3D image and Lidar data is transformed to 2D Bird Eye View image. Then we segment objects from background with a global background pixel threshold which is a adjustable parameter. Our UNet based method achieves 0.041 score on the Kaggle leaderboard. It's still a gap between the best results so far.

Unet-based method highly depends on height information and background pixel threshold, it can not achieve very good result. So then we move to another method, Point RCNN method. The Point RCNN is implemented and the experiments are running. It's expected to get better results than UNet.

II. Motivation

Self-driving is a very hot topic for nowadays companies and communities. This technology presents a rare opportunity to improve the quality of life in many of our communities. Avoidable collisions, single-occupant commuters, and vehicle emissions are choking cities, while infrastructure strains under rapid urban growth.

From a technical standpoint, however, the bar to unlock technical research and development on higher-level autonomy functions like perception, prediction, and planning is extremely high. This implies technical R&D on self-driving cars has traditionally been inaccessible to the broader research community.

Using Lyft as the example, Lyft is investing in the future of self-driving vehicles. Level 5, their self-driving division, is working on a fleet of autonomous vehicles, and currently has a team of 450+ across Palo Alto, London, and Munich working to build a leading self-driving system. Their goal is to democratize access to self-driving technology for hundreds of millions of passengers.

3-D object detection is one of the most important parts for self-driving. A good 3-D object detection will ensure the safety of the passengers and the right track of the vehicles. Hence, it's import for the self-driving companies and researchers to further improve the performance of 3-D object detection systems.

The solution would be useful for automatic driving companies like Tesla, Lyft, Uber etc. We saw this problem in Kaggle competition and that motivated to understand this in depth. The platform shall give us a way to benchmark our work.

III. Solution

We have experimented with 2 kinds of methods:

1) Uet-Based Method [2] , UNet is proposed for semantic segmentation, the architecture of UNet is as Figure 1. In this task, after we project 3D LIDAR data into 2D Bird Eye View data, we view bounding box as the object ground truth that we'd like to segment as Figure 2 and Figure 3 shown. Then we use a transformation matrix to transform predicted boxes back to world space.
2) PointRCNN [1] based method : For PointRCNN based method, only 3D LIDAR point cloud data will be used. We'll use a stage-1 sub-network to generate a small number of high-quality 3D proposals from point cloud. Then we use a stage-2 sub-network to transform the

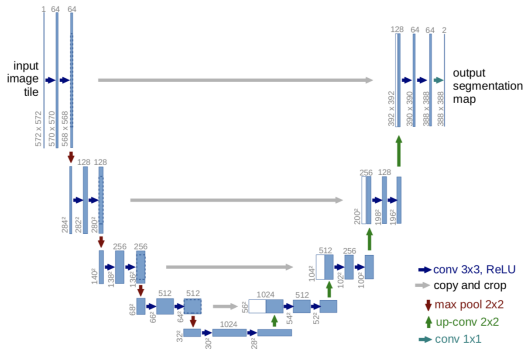

Figure 1: UNet architecture

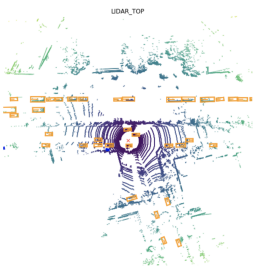

Figure 2: Bird Eye View with bounding box

Figure 3: Ground truth for segmentation

pooled points of each proposal to canonical coordinates to learn better local spatial features, which is combined with global semantic features of each point learned in stage-1 for accurate box refinement and confidence prediction.

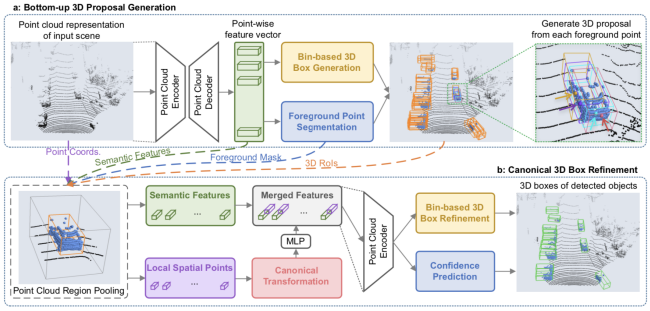

Figure 4: The PointRCNN architecture for 3D object detection from point cloud. The whole network consists of two parts: (a) for generating 3D proposals from raw point cloud in a bottom-up manner. (b) for refining the 3D proposals in canonical coordinate.

IV. Validation and Evaluation

The dataset is provided by Lyft. For each training data, the dataset provides three bird eye view Lidar point cloud from top, front left and front right respectively. The Lidar point cloud and camera images are provided together for six direction: front, front right, front left, back, back left, back right. Meanwhile, some features of the the volumes labeled in the pictures above are also given, such as center x, center y (the relative distance between the object volumes and the sensor), width, height, length, yaw and the class name. For the test, we are going to predict not only the volumes of the objects, but also the label class for them. The confidence bound will also be calculated, but it will not be included in the criteria of evaluating the model. The confidence bound can be used as a tool to help us to understand the prediction. The evaluation criteria of this project is based in the mean average precision at different intersection over union (IoU) thresholds. Intersection over Union is calculated by the size of the overlap between two objects over total area of the two objects combined. Graphical illustration provided by Kaggle is shown as following: While sweeping over IoU

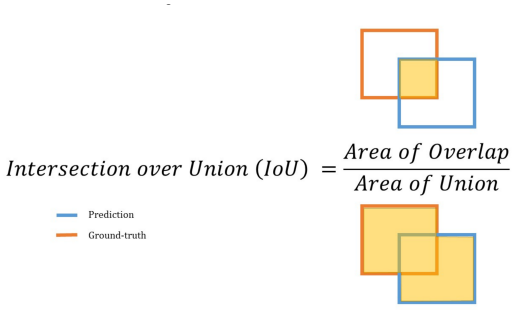

Figure 5: Approach to calculate Intersection over union

thresholds, the average precision value will be calculated with threshold value range from 0.5 to 0.95, step size of 0.05. With threshold value t, a precision value is calculated by:

$$accuracy = \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \quad (1)$$

where TP, FN, FP represent for number of true positives, false negatives and false positives respectively. If there do not exist ground truth objects for given image, zero score will be received by predicting false positives, which will be included in the mean average precision.
The average precision of a single image can be calculated as following with each IoU threshold is provided:

$$accuracy = \frac{1}{|\delta|} \sum_{t} \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \quad (2)$$

where δ represents for the thresholds. Evaluation of the results would be based on the test samples, we're predicting the bounding volumes and classes of all of the objects in a given scene.

V. Experiments

So far we implemented the training and testing of UNet based method. We have first transformed lyft dataset from Nuscene format into Bird-eye-view format, then we could have 2D ground truth for image segmentation, as shown in Figure 6

Based on UNet with EfficientNet-b3 and seresnext101 backbone, for every projected 2D Bird Eye View image, we segment objects from background with a global background

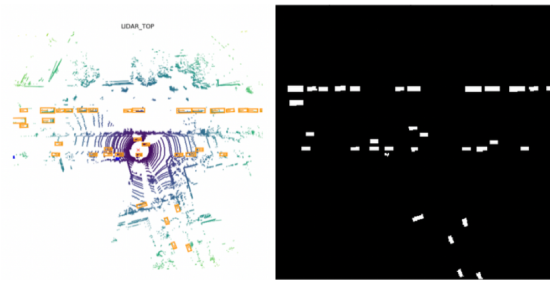

Figure 6: Illustration of 2D bird view data and it's segments

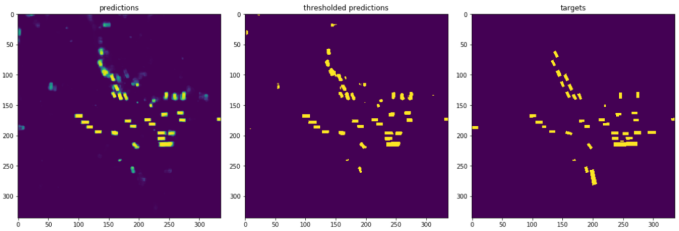

Figure 7: UNet output predictions. Predictions with threshold. Ground truth.

pixel threshold which is a adjustable parameter. The results and its groud truth are shown in Figure 7.

Based on thresholded 2D predictions, we get bounding boxes for each predicted objects, as shown in Figure 8. Then we use a transformation matrix to transform predicted bounding boxes into voxel space. However we lost the height information with Bird Eye View, we used average height of each 9 classes as the predicted bounding boxes height.

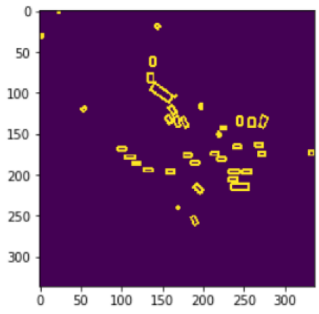

Figure 8: UNet segmentation results.

Our best result is 0.041. It's mean average precision at different intersection over union (IoU) thresholds.

Finally, we visualize our object detection result on the real video image, which is shown in Figure 9. From the visualization result, we can see that most of the cars are detected correctly. However, other classes such as human and bikes are not detected correctly. That's due to Unet-based method highly depends on height information and background pixel threshold, it can not achieve very good result. So we decide to move to another method, Point RCNN method, which is described in Section III.

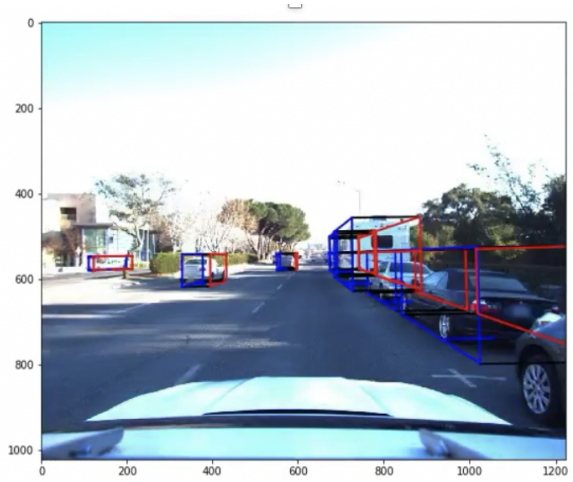

Figure 9: UNet result from the inference engine.

VI. Resources

We intend to use the following resources for development and test of this project.

- HARDWARE :
  Personal computers, Google Cloud , Kaggle platforms, College server cluster , CPUs, GPUs (2 x 1080TI, 1 x 2080TI)
- SOFTWARE :
  Standard s/w packages and free source libraries like Python, PyTorch, OpenCV, Jupyter Notebook
- DATA :
  The data is provided by Kaggle - Lyft dataset

VII. Goal

- 75% stage goal:
  - data Implement the data pre-processing (Image+Lidar) **Done**
  - Research and implement the whole pipline for 3-D objection detection **Done**
- 100% stage goal:
  - Research and implement advanced models for 3-D objection detection **Done**
  - Research and investigate segmentation based methods **Done**
  - Research and investigate detection based methods **Ongoing**
  - Evaluate our results on Kaggle competition platform **Ongoing**
  - Research and implement the database for the 3-D data saving and loading **Ongoing**
- 125% stage goal: Get a medal in the Kaggle competition. **Ongoing**

REFERENCES

[1] Alex H Lang et al. “PointPillars: Fast encoders for object detection from point clouds”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12697–12705.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.