CS230

Frustum Votenet and its Application to the Waymo Lidar Dataset

Brian Johnson
briancj@stanford.edu

Abstract

Votenet [5] is a novel approach to the 3D point cloud object detection problem. It achieved SOTA results in multiple 3D object detection tasks for indoor scenes. However, other work [3] found that it performs less well on outdoor scenes without modifications. Our contributions in this work are twofold: first, we will benchmark Votenet on (a subset of) the Waymo lidar dataset [1], (the first time this has been done to our knowledge). Second, we will build on the ideas of [6] by reducing the problem of whole outdoor scene 3D object detection to 3D object detection within a pre-defined viewing frustum obtained from 2D image recognition data. While our results still require additional verification, we observed that Frustum VoteNet outperformed status quo VoteNet by a factor of 6 in terms of mAP.

1 Introduction

As self driving cars become more prevalent on our roads, the problem of outdoor 3D object detection (in particular, 3D object detection based on lidar data) has become increasingly important. Many self driving car companies (Waymo, Nuro, Cruise etc.) have invested significance engineering resources in training models for this task. Many of these approaches, however, don't operate on raw point clouds [5] as this approach is only starting to get traction within the 3D object detection community. That being said, given the lidar data that these cars are constantly collecting point clouds are a fairly natural data representation for these sorts of problems.

Votenet [5] is a deep neural network architecture specifically designed to operate on point cloud data. It has achieved SOTA results on multiple indoor 3D point cloud object detection tasks. Thus, it's a natural choice for an investigation into point cloud object detection. Unfortunately, some authors [3] have noted that its performance out of the box isn't as spectacular on outdoor scenes. This is to some extent expected, as many have noted that the indoor and outdoor 3D object detection tasks are quite different (i.e. the scale of the outdoor problem is much larger, there are often many more objects, etc.). To that end, we aim to improve the performance of VoteNet on on the Waymo dataset by drawing on ideas from [6]. In particular, we aim to extend their idea of decreasing the scale / scope of the outdoor 3D object detection problem from an entire scene to a single viewing frustum (generated from lifting a 2D bounding box into 3D) that contains an object of interest.

To be very explicit: The input to our algorithm is 3D points generated from Lidar data on Waymo cars. In our baseline, the input data is an entire scene. In our experiment, the input is a subset of the points in an entire scene that fall within a viewing frustum that represents a 2D bounding box projected into 3D. **The output of our algorithm** are 3D bounding boxes and corresponding semantic labels spanning the following classes: pedestrian, car, sign, cyclist.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

As there are two novel aspects to our investigation we will give brief overviews of related work in both of these areas.

2.1 Deep learning on point clouds

There have been a number of papers describing architectures similar to VoteNet [5]. Indeed, VoteNet is a direct descendent of many of these works.

The core ideas upon which VoteNet is based were introduced in PointNet [7]. In this paper, the authors (incidentally, some of whom went on to be authors of the VoteNet paper), introduced the idea that the core abstraction one should use to train point cloud based networks is the idea of a set (as point clouds have no inherent ordering). This led them to an architecture where each component was required to be order invariant, thus preserving the set abstraction. The authors showed that with this architecture they could achieve results at least as good as SOTA on indoor 3D object recognition tasks.

This work was extended by a follow up paper PointNet++ [8]. The main contribution of this paper was that it overcame a previous limitation of PointNet: " by design PointNet does not capture local structures induced by the metric space points live in, limiting its ability to recognize fine-grained patterns and generalizability to complex scenes." PointNet++ fixed this problem by applying PointNet recursively. This allowed the network to learn local features with increasingly complexity across scales.

VoteNet can be seen as an application of PointNet++. VoteNet uses PointNet++ as a feature extractor network trained jointly with a downstream network that computes and aggregates "votes" [1].

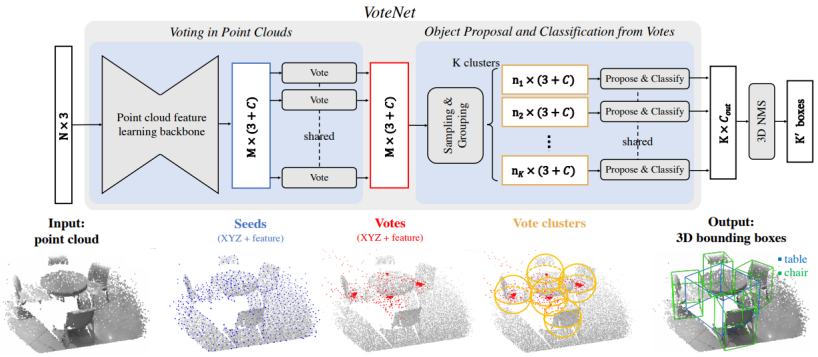

Figure 1: VoteNet model architecture. Image taken from [5]. The "point cloud feature learning backbone" used here is PointNet++

The key insight of VoteNet was that we could combine more traditional computer vision techniques like Hough Voting with features learning via deep learning (PointNet++ in particular) to achieve SOTA results on object detection tasks.

The shortcoming of this approach, which we aim to make progress toward addressing in this work, is that the network does not perform as well on outdoor scenes as indoor scene. This was found in [3] who applied this architecture to the KTTI lidar self driving car dataset.

To that end, we refer to Frustum PointNet [6]. In this work, authors had the clever idea of using 2D object detection (a much more solved problem) to bootstrap what one can think of as a "strong prior" for the 3D object detection problem.

[1] Here votes mean Hough Votes

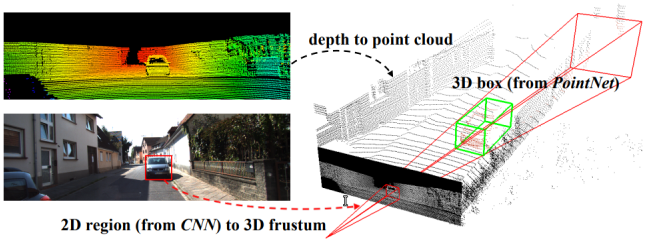

Figure 2: Visualization of Frustum PointNet's 2D -> 3D frustum idea. Image taken from [6]

In their pipeline, they would first get a 2D bounding box from a 2D object detector (indeed, in their experiments they used the ground truth 2D bounding box to isolate the quality of the pipeline from the quality of the 2D object detector ... an idea which we will also adopt) and use this to generate a frustum in 3D going through that 2D bounding box. The thinking being that within that frustum lies the object of interest, however, we have now greatly reduced the scope of the outdoor problem to a scale that is more similar to the indoor problem. The main shortcoming of this work, simply as a function of when the paper was written, is that it only uses PointNet. In the years since, the options for a base 3D object detection network have improved. We will thus, extend their results by replacing PointNet with VoteNet.

2.2 3D object detection on the Waymo dataset

We also examined a few of the currently top ranked networks on the leaderboard for this data set.

In [10] the authors used a Feature Pyramid Network for 2D object recognition. This allowed them to achieve first place results on the 2D image recognition task. However, in their paper they do not discuss any attempts made at the 3D object recognition task. However, this paper is worth noting because our architecture (if it were to be applied in production) would require a strong 2D object detector network. This would make a strong candidate.

In [9] the authors used an approach that combined PointNet based networks with 3DVoxel CNN based networks. While this approach is interesting in that it builds on PointNet, we believe that the solution of this problem should operate solely at the point cloud (not voxel) level, thus, we did not pursue more investigation in this direction.

3 Dataset and Features

The dataset we used was a processed sub-sample of the Waymo Open Dataset [1]. In particular, we selected 1500 frames frames from this dataset where each frame consisted of a 3D point cloud generated from lidar data off a Waymo car. 1500 was selected as this is the same number of scans as in the Scannet dataset. Each frame had aprox. 150-200k points out of the box. We sub-sampled each frame to 150k randomly selected points. For each frame, we had annotated 3D bounding boxes (in the same frame) with labels of pedestrian, cyclist, car, and sign. We did not do any data augmentation as we had terabytes more data available to us than we trained on for this experiment.

4 Methods

4.0.1 Model + Pipeline

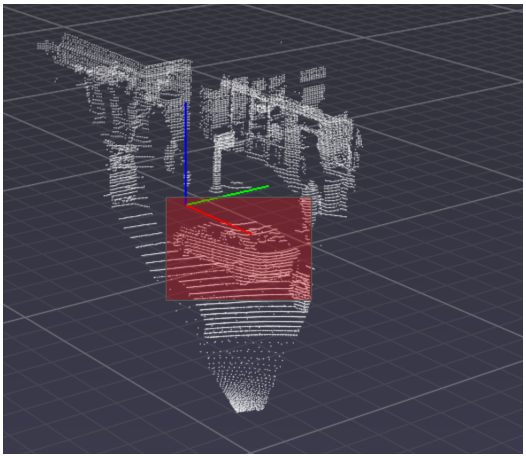

Figure 3: One frustum from one frame of the Waymo dataset

We will use the standard VoteNet model for both our baseline and our experiment. In the baseline, we will train on a dataset where each data point is one frame from the Waymo dataset (down-sampled to 150k points). An example of this (from the actual dataset) is produced below.

In the experiment we will train on the same dataset, however, each sample will not be of the entire scene but only of one frustum of potentially interesting point clouds as determined by a 2D image recognition task + 3D project. Following the Frustum PointNet paper, we will use ground truth 2D bounding boxes to isolate the our results from the quality of the 2D object detector.

4.0.2 Loss function

We will follow both [6] and [5] and use the following loss function

$$\begin{aligned} L_{multi-task} =& L_{seg} + \lambda(L_{c1-reg} + L_{c2-reg} + L_{h-cls} + \\ & L_{h-reg} + L_{s-cls} + L_{s-reg} + \gamma L_{corner}) \end{aligned}$$

Figure 4: Training loss function (taken from [6])

Here all classification tasks use softmax and all regression tasks use smooth-l1 (huber) loss. For a complete overview of the loss function we refer to section 4.4 of [6].

5 Experiments/Results/Discussion

5.1 Experiment Setup

We followed VoteNet and trained both our baseline and experiment with an Adam optimizer, batch size 8 and an initial learning rate of 0.001. The learning rate is decreased by 10× after 80 epochs and then decreased by another 10× after 120 epochs. Both networks were trained end to end to convergence (as determined by loss function plateau). This took 180 epochs in the case of status quo, and 80 in the case of the experimental group.

For scoring, the proposals are post-processed by a 3D NMS module with an IoU threshold of 0.25. These proposals are then used to calculate AP / mAP.

5.2 Results

5.2.1 Quantitative

Model	Vehicle AP	Pedestrian AP	Sign AP	Cyclist Ap	mAP
Frustum VoteNet	0	.000005	.000250	.000009	.000066
VoteNet	.00041	0	0	0	.00001

Figure 5: AP per class and mAP for VoteNet and Frustum VoteNet on the Waymo Dataset

5.2.2 Qualitative

Visualizations of some results are in the appendix. There are a few things we can learn from these photos. The first image is from the status quo group. It is of predicted votes overlayed over ground truth bounding boxes. We note that in status quo the network seems to do an ok job of determining where objects are, i.e. its votes correspond to ground truth bounding boxes. However, these votes are no where near as localized / clustered as we'd like them to be (especially if you compare them with the results achieved on indoor scenes).

The next two photos are from the experimental group. Here we visualize predicted votes overlayed on top of point clouds. We include an example from a class where the network does well (people) and one where it does not do well (cars). Interestingly, there does not seem to be a huge difference in the distribution of votes between these two cases. We had expected the votes to be more localized in the person case, due to its higher performance. This results warrants follow up in future work.

5.3 Discussion

We note from a qualitative investigation that both status quo and experiment had some trouble clustering their votes tightly. However, the experimental group likely benefited from a decreased search space which lessened these effects. This suggests to me that perhaps training on more data and for more iterations could be useful to see if we can get the vote to cluster more rightly.

We note that in terms of AP / mAp Frustum VoteNet outperforms VoteNet by a significant margin. However, it's interesting that it still does poorly on vehicles. Qualitative investigation of the frustums generated for vehicles suggest that often times the scan (and thus the frustum) only captures one part of the vehicle (say the back). Perhaps without the context of the larger scene this makes it harder to capture vehicles, whereas other objects are more often totally and clearly represented within the frustum.

6 Conclusion/Future Work

6.1 Conclusion

We investigated how Frustum VoteNet would compare to a standard VoteNet on a subset of the Waymo lidar dataset. While our results are still early, we found that introduction of frustums into the training pipeline resulted in a 6x increase in mAP. This validates our initial assumption that reducing the problem of 3D object recognition to finding a single object within a single frustum (as opposed to all objects in a scene) can improve performance of VoteNet on outdoor scenes.

6.2 Future Work

In terms of future work there are a few interesting avenues to explore. First, training on more data. We were limited to training on only a fraction of the entire Waymo dataset by time and computational resources. A reasonable next step would be to train on the entire 2TB of data and see how that effects performance. Additionally, we didn't do any data augmentation (because we had ample training data) thus experimenting with various data augmentation strategies could also be an interesting next step.

Finally, our system, did not use a trained model for 2D bounding box detection but instead used 2D bounding boxes generated from ground truth data. Thus, a reasonable next step would be to create a end to end system that uses a model to do the 2D object detection step of our pipeline.

7 Contributions

All work done for this project was my own. That being said, my work greatly benefited from the work [3] and [5]. In addition, I'd be remiss if I didn't thank Div for his useful comments and for pointing me in the direction of [6].

References

[1] Waymo open dataset: An autonomous driving dataset, 2019.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Alexander Arzhanov. 3d object detection from point cloud, 2019.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[5] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[6] Charles Ruizhongtai Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. *CoRR*, abs/1711.08488, 2017.

[7] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016.

[8] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *CoRR*, abs/1706.02413, 2017.

[9] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[10] Yu Wang, Sijia Chen, Li Huang, Runzhou Ge, Yihan Hu, Zhuangzhuang Ding, and Jie Liao. 1st place solutions for waymo open dataset challenges – 2d and 3d tracking, 2020.

8 Appendix

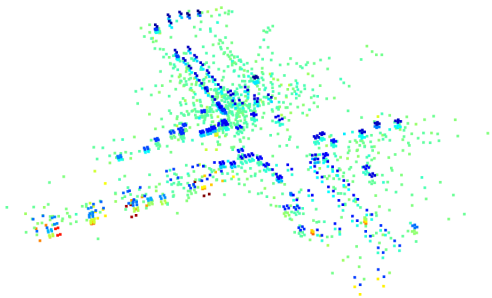

Figure 6: Status Quo: Full scene VoteNet. (Legend: green dots are votes generated by VoteNet; colored boxes are GT bounding boxes

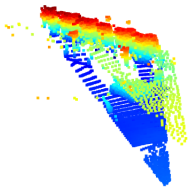

Figure 7: Experiment: Frustum VoteNet car example (performs poorly). (Legend: green dots are votes generated by VoteNet; colored boxes are GT bounding boxes

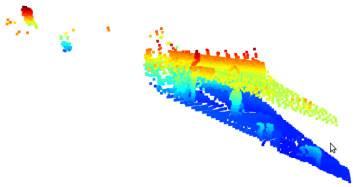

Figure 8: Experiment: Frustum VoteNet pedestrian example (performs better). (Legend: green dots are votes generated by VoteNet; colored boxes are GT bounding boxes