3DYOLO: Real-time 3D Object Detection in 3D Point Clouds for Autonomous Driving

Priya M V[1], Dhanya S Pankaj[2]
[1,2]*Department of Computer Science and Engineering, College of Engineering Trivandrum, Trivandrum, India*
[1]*priyamv1234@gmail.com* [2]*dhanyaspankaj@cet.ac.in*

2021 IEEE International India Geoscience and Remote Sensing Symposium (InGARSS) | 978-1-6654-4249-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/InGARSS51564.2021.9791912

***Abstract*—In the recent era, a lot of interest is attracted by the autonomous vehicles which can sense surroundings and navigate without human intervention. Object detection and recognition form a major part of autonomous driving systems. Lidar sensors can be used to capture point clouds of driving environment. Detecting multiple 3D objects in point clouds in real time and defining their boundaries with the help of 3D bounding boxes are critical in motion planning by self driving systems. This paper proposes a LiDAR-based 3D object detection system that operates in real-time, with emphasis on autonomous driving scenarios. A state-of-the-art 2D standard object detector for RGB images, YOLOv4, is used as the base for object detection. The multi-class 3D bounding boxes are generated using a complex regression approach. An Euler-Region-Proposal Network (E-RPN) is used to predict the pose of the object. The proposed model receives point cloud data as input and outputs 3D bounding boxes with classes in real-time. The experiments done on the KITTI benchmark dataset proves that the proposed system outperforms existing methods in terms of accuracy and performance.**

***Index Terms*—3D Object Detection, Point Cloud processing, Yolo, Autonomous Driving.**

I. Introduction

With the quick development of 3D acquisition technologies, 3D sensors like LiDAR and RGB-D cameras have become frequently convenient and affordable. 3D point cloud data is universal in mobile robotics purposes such as autonomous driving, where efficient and robust object detection is crucial for planning and decision [1]. The set of applications using 3D point clouds is growing as a result of the growing availability of sensing devices such as LiDAR which allow easy acquisition of the 3D world in the 3D point clouds [2]. In autonomous driving systems, the object detection subtask is one of the most important requirement that allows the vehicle controller to estimate distance to obstacles [3]. Environmental awareness is a central component of autonomous driving function. Light and Range Detection (LiDAR) sensors are less susceptible to weather conditions and can operate under poor lighting conditions [3]. LiDAR is one of the leading sensors to gives the 3D characteristics of the object in terms of the point cloud to localize or detect the objects and the shapes.

A number of deep learning methods are proposed in literature to address various problems associated with point cloud processing, including 3D shape classification and 3D object detection [4]. Many high-quality object detectors [4], [11], [18] have been developed in recent years. This paper proposes a network that operates in real-time. The main goal of this work is to propose a real time 3D object detector which operates directly on the 3D point clouds.

The rest of the paper is organized as follows. Section 2 provides a brief description of related work. Section 3 explains the methodology of the proposed system. Section 4 deals with experiments and section 5 deals with the results and discussion. Section 6 provides the conclusion.

II. Related Work

Object detection directs at positioning and classifying objects in an image, and identifying them with rectangular bounding boxes to describe the confidences of existence and class. The frameworks of object detection methods can mainly be classified into two types: region proposal-based methods and single-shot methods. The former methods follow a conventional object detection pipeline, generating region proposals at first and then classifying each proposal into different object classes. The latter methods directly predict class probabilities and retreat 3D bounding boxes of objects using a single-stage network. The region proposal based methods mainly include R-CNN [6], Fast R-CNN [7], Faster R-CNN [8], SPP-net [9], R-FCN [10]. The single shot methods mainly includes YOLO, SSD [11], YOLOv2 [12] YOLOv3 [13] and YOLOv4 [21]. They can run at a high speed and are highly suitable for real-time applications.

The methods before deep learning era uses hand-crafted point cloud features. Many works [15], [16] make use of voxel grid representation with hand-crafted features. Besides the voxel grid representation, [16] uses a 3D sliding window approach for 3D object detection. With the advent of deep learning networks for feature extraction, 3D object detectors using convolutional neural networks (CNN) are proposed [17]. Some works consider a multimodal approach which uses images and connects them with the point cloud data. For example, MV3D [18] proposed a framework using information from multiple view points (LiDAR front view, LiDAR bird eye view, and camera) to build a 3D object detection network. The limitations of the above methods are that they have low detection or localization accuracy, poor detection in smaller images and single object detection only at a time.

3D object detection systems that use LiDAR data only are also gaining attention. The advantage of using point cloud data is a direct measurement of the distance of enveloping objects. This allows us to develop object detection algorithms for

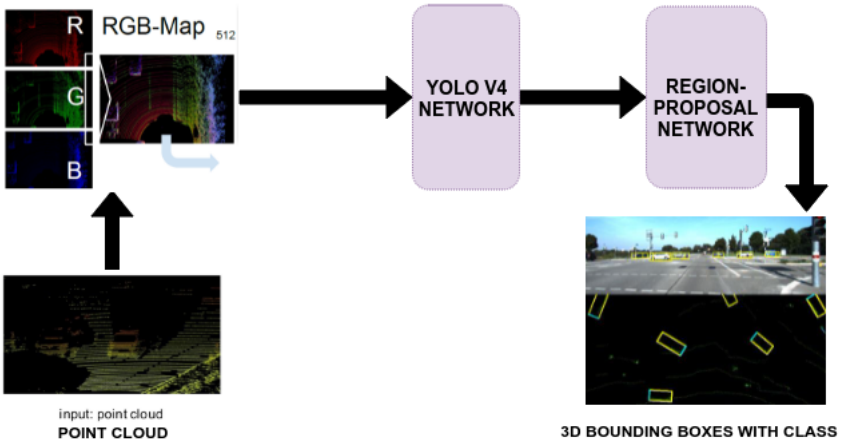

Fig. 1. 3DYOLO - The Architecture Of The Proposed System

autonomous driving that estimate the position of different objects accurately in 3D. However, it also has some limitations. Compared to images, Lidar point clouds are scattered with varying density distributed all over the area. Object detection methods on point clouds capable of predicting 3D bounding boxes are still limited, but more and more prominent. Some works make use of the advanced 2D object detectors and adapt them to 3D using various techniques. One of them [23] make use of the YOLOv2 [12] 2D detector and adapts it to find the 3D bounding boxes. However, the YOLOv2 network has many limitations including less accuracy and less fps. In the present work, we propose a 3D object detection model by preprocessing the point clouds to an image representation and feeding to YOLOv4, an advanced 2D object detector. Then 3D bounding boxes are generated using a region proposal network.

III. Proposed System

The section explains the proposed model for real time 3D object detection on point clouds. The Fig.1 shows the architecture of the proposed method. The proposed model will be denoted 3DYOLO. It presents a 3D version of YOLOv4, which is one of the fastest methods for 2D object detectors. The ERPN method in Complex-YOLO [23], that estimates the orientation of objects, is employed in our architecture to predict exact 3D bounding boxes. This enables the localization and detection of the objects in real-time on the point cloud. Here anchor-boxes based object detection is employed.

The proposed 3DYOLO object detection method contains mainly three modules: Point cloud preprocessing, Network architecture for 2D detection, RPN for 3D bounding boxes.

The 3DYOLO model takes a point cloud as input and converts it into birds-eye-view RGB-map.The RGB map is then fed to a simplified YOLOv4 [21] CNN architecture, extended by a complex angle regression and RPN [18], to detect accurate multi-class 3D objects operating in real-time. The complex angle regression and RPN does the conversion from 2D to 3D bounding boxes using a predefined height based on each object class.

A. Point Cloud Preprocessing

The first step is preprocessing which converts the 3D point cloud into a BEV (Bird's Eye View) map. The 3D point cloud of a frame, obtained by Velodyne HDL64 laser scanner [20], is converted into a single birds-eye-view RGB-map, covering an area of $80m \times 40m$. The three channels used in a BEV RGB map of a point cloud is formed of height, intensity, and density. Consequently, all three feature channels (z_r, z_g, z_b) are determined for the whole point cloud P inside the covering area Ω . Consider the Velodyne at the origin of P_Ω.

Next consider z in range $[2m, 1.25m]$, and Lidar sensor z position of $1.73m$ [20], to cover an area above the ground to about $3m$ height. Then establish a mapping function S_j and map each point with index i into a specific grid cell S_j of our RGB-map. A set represents all points mapped into a specific grid cell [23] as given by eqn.1.

$$P_{\Omega i \to j} = \{P_{\Omega i} = [\ x, y, z]\}^T \mid S_j = f_{ps}(\ P_{\Omega I}, g)\ \} \quad (1)$$

Then calculate the channel of each pixel, regarding the Velodyne intensity as I(P_Ω). This is shown in eqn. 2.

$$\begin{aligned} z_g(\ S_j)\ &= max(\ P_{\Omega i \to j}.[\ 0,0,1]\ ^T) \\ z_g(\ S_j)\ &= max(\ I(\ P_{\Omega i \to j})\ ) \\ z_r(\ S_j)\ &= min(\ 1.0, log(\ N+1)\ /64)\ \mid \end{aligned} \quad (2)$$

Here, N represents the number of points mapped from $P_{\Omega i}$ to s_j, and g is the parameter for the grid cell size. The z_g indicates the maximum height, z_b the maximum intensity, and z_r the normalized density of all points mapped into S_j.

B. CNN Architecture

A simplified version of YOLOv4 [21], which is one of the latest object detectors in YOLO family, is used as the 2D

object detector operating on BEV RGB Map. It is a real-time object recognition system that can recognize multiple objects in a single frame. It also marks boundary boxes around the objects. The YoloV4 architecture is composed of three parts.

- Backbones: CSPDarknet53
- Neck: PANet+SPP
- Head: YOLOv3

Backbone is a deep neural network composed mainly of convolution layers. YOLOv4 network implements CSPDarknet53 as backbone network for features extraction. YOLOv4 uses Bag of Freebies (BoF) for backbone and detector, Bag of Specials(BoS) for backbone and detector in training for improve accuracy. YOLOv4 chooses a Path aggregation Network (PANet) with a spatial pyramid pooling layer (SPP) module for the feature aggregation of the network and collects feature maps from different stages. The feature map from the neck is fed into the head. The role of the head in the case of a one-stage detector is to make dense predictions. YOLOv4 uses the same YOLO head as YOLOv3 [13] (anchor based) for detection with the anchor-based detection steps. The YOLO Network divides the image into a grid. The 3DYOLO perform following steps:

- YOLO predicts a fixed set of boxes, in this case, 5 per grid cell.
- Each cell in the feature map grid predicts B bounding boxes, confidence scores for each of them and K class scores p_1,...p_k.
- Predicted bounding boxes are parameterized as refined anchors. A refined anchor is a vector (t_x,t_y,t_z,t_w,t_h,t_l,t_θ), where t_x,t_y,t_z are the offset center coordinates, t_w,t_h,t_l are the offset dimensions, and t_θ is the offset rotation angle.
- For each box, the box dimensions, and angles (real and imaginary parts, (t_x,t_y,t_w,t_l,t_{im},t_{re}) where t_x,t_y,t_w,t_l are the x, y point of the center, width, and length of the bounding box. t_{im},t_{re} represent the real and imaginary parts of the angle of bounding box orientation, are calculated.
- An objectness probability, i.e. probability of the predicted bounding box holding an object, is found out.

C. Region Proposal Network (E-RPN)

A complex angle regression and RPN network [23] is employed after the YOLOv4 [21] CNN architecture to detect multi-class oriented 3D objects in real-time. RPN parses the 3D position $b_{x,y}$, object dimensions (width b_w and length b_l) as well as a probability p_0 , class scores p_1 ...p_n and finally its orientation b_ϕ from the incoming feature map. This help to estimate the accurate object orientations based on an imaginary and real fraction directly implanted into the network. The design of RPN is to parse the object dimension from the incoming feature map and determine actual object orientations and dimensions of the bounding boxes. The bounding boxes coordinates are computed as in Eqn 3. More details on the calculations can be found in [23].

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_l &= p_l e^{t_l} \\ b_\phi &= arctan_2(t_{im}, t_{re}) \end{aligned} \tag{3}$$

D. Complex Angle Regression

The orientation angle for each object t_ϕ can be computed from the effective regression parameters t_{im} and t_{re}, which resemble to the phase of a complex number [23]. The angle is provided simply by doing $arctan_2$ (t_{im},t_{re}).

TABLE I
LIST OF USAGE FOR BOF AND BOS IN THIS IMPLEMENTATION

	Backbone	Detector
BoF	1.Dropblock 2.Mosaic 3.Random rescale	1.Cross stage partial network 2.Dropblock 3.Random training shapes
BoS	1.Mish activation 2.Cross stage partial network 3.Multi-input weighted residual connection	1.Mish activation 2.SPP block 3.PAN 3.GloU loss

IV. EXPERIMENTS

The proposed model is trained and evaluated using a dataset generated from KITTI Vision Benchmarking Suite [20], which is subdivided into three categories - Cars, Pedestrians, and Cyclists. Each class is divided into three difficulty levels - easy, moderate and hard - considering the object size, distance, and truncation. This public dataset presents $7,481$ samples for training including annotated ground truth and $7,518$ test samples with point clouds from a Velodyne laser scanner, where annotation data is private. Note that it concentrated on birds-eye-view and does not run the 2D object detection benchmark, since our input is Lidar based only.

The model was trained via stochastic gradient descent with a weight decay of 0.0005 and momentum 0.9. The implementation is based on a modified variant of the Darknet neural network framework. The training and testing were run on a DGX Station with a NVIDIA Tesla P100 GPU 16 GB. In the experiment the training steps are 4000 and the epoch is 300.

A. Different features used for Classifier training

Bag of freebies (Bof) and Bag of specials (BoS) is the developments that can be made in the training process (like data augmentation) to improve accuracy. In the present model, Bof and BoS are applied in both the backbone and detector of the architecture (Please refer to Table I for details).

TABLE II
A COMPARISON OF THE 3D OBJECT DETECTION PERFORMANCE OF 3DYOLO WITH THE STATE OF THE ART 3D OBJECT DETECTORS ON KITTI VALIDATION SET. THE EVALUATION METRIC IS AVERAGE PRECISION (IN %) OF 3D BOUNDING BOXES.

			Car			**Pedestrian**			**Cyclist**		
Method	**Modality**	**FPS**	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
VoxelNet [24]	Lidar	4.3	77.47	65.11	57.73	39.48	33.69	31.51	61.22	48.36	44.37
Complex-YOLO [23]	Lidar	50.4	67.72	64.00	**63.01**	41.79	**39.70**	35.92	68.17	58.32	54.30
3DYOLO	**Lidar**	**62**	**95.50**	**82.54**	48.37	**76.27**	32.24	**55.08**	**88.88**	**63.51**	**57.67**

V. RESULTS AND DISCUSSION

The proposed 3DYOLO model is evaluated on KITTI object detection benchmark [20], based on the official KITTI evaluation protocol, where the IoU thresholds are 0.7 for classes. APs (in %) of 3D bounding boxes is used for evaluation. The overall mAP of proposed work is 88%.

The results for 3DYOLO are compared with the state-of-the-art 3D object detectors VoxelNet [24] and Complex-YOLO [21] as shown in Table II. It can be observed that compared to the existing systems, our system achieves better results in terms of accuracy and speed. The inference time of 3DYOLO is 62 frames per seconds (FPS), which satisfies the real-time requirements and is better than the compared methods. The overall results show that 3DYOLO is significantly better than the other methods in terms of accuracy. This can be attributed to the superior performance of the YOLOv4 2D detection stage, which offers real time 2D detection. By adapting it to the case of point clouds, 3DYOLO achieves real time 3D detection.

VI. CONCLUSION

This paper proposes a new model of real time 3D object detection in Lidar-based point clouds, for autonomous driving. The most recent 3D object detection method using YOLO detectors employs YOLOv2 which has less accuracy and efficiency. The proposed method employs the latest 2D object detector YOLOv4, which is a significant improvement over YOLOv2. This method is able to detect objects of multiple classes (e.g. cars, pedestrians, cyclists) simultaneously in one forward path. This enables deployment of the detection model in real time systems like self-driving cars. This system does not need additional sensors, e.g. 2D camera, unlike most of the leading approaches. The proposed 3DYOLO model is a real-time end-to-end model, based on LiDAR data only, that outperforms existing approaches in terms of accuracy and efficiency.

REFERENCES

[1] Dong Ho Yun, Sung In Choi, Sung Han Kim, and Kwang Hee Ko.*Registration of multiview point clouds for application to ship fabrication. Graphical Models*. 90(Supplement C):1 – 12, 2017.

[2] *MathWorks 3-d point cloud processing*

[3] Hong Cheng. *Autonomous intelligent vehicles: theory, algorithms, and implementation*.Springer Science and Business Media, 2011.

[4] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi.*You Only Look Once: Unified, Real-Time Object Detection*. In: CoRR abs/1506.02640 (2015). arXiv: 1506.02640.

[5] B. Li, T. Zhang, and T. Xia.*Vehicle Detection from 3D Lidar Using Fully Convolutional Network*. arXiv preprint arXiv:1608.07916, 2016

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik .*Rich feature hierarchies for accurate object detection and semantic segmentation* .In CVPR, 2014

[7] R. Girshick.*Fast r-cnn* In ICCV, 2015.

[8] S. Ren, K. He, R. Girshick, and J. Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks*In NIPS, 2015,pp. 91–99.

[9] K. He, X. Zhang, S. Ren, and J. Sun.*Spatial pyramid pooling in deep convolutional networks for visual recognition*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, 2015.

[10] Y. Li, K. He, J. Sun et al.*R-fcn: Object detection via region-based fully convolutional networks*.In NIPS, 2016, pp. 379–387.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg.*Ssd: Single shot multibox detector*.In ECCV, 2016.

[12] J. Redmon and A. Farhad.*Yolo9000: better, faster, stronger*. arXiv:1612.08242, 2016.

[13] J. Redmon and A. Farhad.*YOLOv3: An Incremental Improvement*. arXiv:1804.02767, 2018.

[14] Dominic Zeng Wang, I. Posner, and P. Newman.*What could move? Finding cars, pedestrians and bicyclists in 3D laser data*.In: 2012 IEEE International Conference on Robotics and Automation. 2012

[15] Dominic Zeng Wang and Ingmar Posner.*Voting for Voting in Online Point Cloud Object Detection*. In: Robotics: Science and Systems. Vol. 1. 2015, p. 5.

[16] Dominic Zeng Wang and Ingmar Posner.*Voting for Voting in Online Point Cloud Object Detection*. In: Robotics: Science and Systems. Vol. 1. 2015, p. 5..

[17] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner.*Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks*. In: CoRR abs/1609.06666 (2016).

[18] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia.*Multi-View 3D Object Detection Network for Autonomous Driving* In: CoRR abs/1611.07759 (2016).

[19] *PointClouds.org. The pcd (point cloud data) file format.*.

[20] Geiger, A.*Are we ready for autonomous driving? the kitti vision benchmark suite.* In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). CVPR '12, Washington, DC, USA, IEEE Computer Society (2012) 3354–3361.

[21] Alexey Bochkovskiy, Chien Yao Wang and Hong.*YOLOv4: Optimal Speed and Accuracy of Object Detection*.journals/corr/abs-2004-10934.

[22] Beyer, L., Hermans, A., Leibe.*Biternion nets: Continuous head pose regression from discrete training labels.* Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9358 (2015) 157–168.

[23] Martin Simon , Stefan Milz , Karl Amende, Horst-Michael Gross .*Complex-YOLO: Real-time 3D Object Detection on Point Clouds*.International Journal of Computer Vision,2018.

[24] Zhou, Y., Tuzel, O.*Voxelnet: End-to-end learning for point cloud based 3d object detection.*CoRR abs/1711.06396 (2017)