2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

3D Fully Convolutional Network for Vehicle Detection in Point Cloud

Bo Li*

Abstract— **2D fully convolutional network has been recently successfully applied to the object detection problem on images. In this paper, we extend the fully convolutional network based detection techniques to 3D and apply it to point cloud data. The proposed approach is verified on the task of vehicle detection from lidar point cloud for autonomous driving. Experiments on the KITTI dataset shows significant performance improvement over the previous point cloud based detection approaches.**

I. INTRODUCTION

Understanding point cloud data has been recognized as a key task for many robotic applications. In the application of autonomous driving for example, this task generally includes the detection of on-road obstacles, e.g. cars and pedestrian, and the segmentation of drivable free regions, from the point cloud generated by lidars. In the application of robotic manipulation, estimating the position and orientation of the target object is the prerequisite for the consecutive grasping manipulation. Compared to image data, point cloud data naturally persists 3D metric information. Thus it is usually admitted to provide better estimation of object orientation and position than image data.

Recently, the deep learning based methods have achieved significant performance in the tasks of object recognition, detection and segmentation on image data. Deep neural network, especially the convolutional neural network (CNN), embodies impressive capacity of complex feature representation on images. It is believed that such capacity can be generalized to data from other forms including point cloud data. As will be mentioned in the following sections, several previous works have applied neural network for feature extraction in point cloud processing and achieved significant performance improvement in in tasks of object recognition and detection.

In this paper, we propose to use 3D fully convolutional network (FCN) to further enhance the performance of object detection in point cloud. The 2D FCN [22] has achieved notable performance in image based detection tasks. The proposed approach extends FCN to 3D. It detects objects and estimates oriented object bounding boxes in an end-to-end manner. We apply the proposed method in 3D vehicle detection for an autonomous driving system, using a Velodyne 64E lidar. Meanwhile, the approach can be generalized to other object detection tasks on point cloud captured by Kinect, stereo or monocular structure from motion.

*Bo Li is a researcher at Trunk Inc. Contact: prclibo.github.io or libo@trunk.tech

II. RELATED WORKS

The proposed method transplants the recently popular deep learning based detection framework to point cloud data. Thus we spend some space in this section to respectively introduce three related research areas with the proposed work. We first introduce previous works of the general point cloud based detection. Secondly, we introduce how the point cloud data can be fed into CNNs, which is the basic operation the proposed approach. Thirdly, we introduce the well-performing FCN based object detection framework, which the proposed method follows but with the 3D CNN operation.

A. 3D Object Detection in Point Cloud

A majority of 3D detection algorithms can be summarized as two stages, i.e. candidate proposal and classification. In the scenario of on-road obstacle detection from point cloud, a simple category of works propose candidates by some rule-based segmentation algorithms. For example, [8], [14], [25] remove the ground plane and cluster the remained segments as potential object proposals. More delicate segmentation methods are also proposed for better robustness and performance. [17], [34] construct graph on point cloud and exploit nearest neighbor alike clustering techniques to generate object proposals. [26] over-segments point cloud as super-voxels and consecutively classifies super-voxels as object parts. A drawback of these segmentation based candidate proposal is that defect segments might be produced cross object instances, which makes the consecutive recognition unable to recognize. To overcome this drawback, [3] segments the scene in a hierarchical manner. In scenarios where the rough size of targeted objects is known, e.g. vehicle detection on road [35] or workpiece detection for manipulation [16], overlapping candidates can be enumerated over the space for classification. Deep learning based Region Proposal Network (RPN) is also suggested to generate candidate proposals [5], [31].

For the classification stage, a variety of hand-crafted features have been researched. [10], [16] directly match CAD shape model with the percepted point cloud. [33], [34], [36] combines shape spin images, shape factors and shape distribution as features. Other hand-crafted features include FPFH [26], raw voxel occupancy [35], normal distribution histogram and etc. A comparison of these tranditional hand-crafted features can be found in [2]. Higher level feature representation techniques, e.g. sparse coding [7], [18] and deep learning [5], [9] is also deployed. In Section II-B, more details about the deep learning based feature representation are introduced.

978-1-5386-2682-5/17/$31.00 ©2017 IEEE

Note that the above methods mostly operate point cloud data in the 3D space. As in many scenarios point cloud can be represented as 2D depthmaps or range scans. 2D detection methods are also proposed in some previous works. [4], [20] propose to detect objects from the RGBD representation captured by Kinect alike sensor. [19] proposes to use convolutional neural network in detect vehicles from range scan converted from point cloud captured by Velodyne lidar. The projection inevitably loses or distorts useful 3D spatial information but can benefit from the well developed image based 2D detection algorithms.

B. Convolutional Neural Network and 3D Object Detection

CNN based 3D object detection is recently drawing a growing attention in computer vision and robotics. Early works in this direction come in the object detection task using RGBD data. Since the RGBD data are represented as 2D images, it is straightforward to exploit the well-developed 2D CNN techniques on such data for object detection or recognition tasks. For example, [28], [30] interprets depthmap as image channels and feed to image recognition CNN. [13] uses CNN to produce feature description from RGBD images for a sliding window based detection framework. [4] proposes object candidates from the depthmap and verifies candidates under the Fast-RCNN framework. Similar to RGBD data, point cloud data captured by LiDAR can also be projected as depthmaps or range scans. As is first proposed in [19], such data can also be feed to 2D CNN for object detection. In addition, different from the previous works which simply interpret depth data as a feature channel, [19] find it possible to predict the 3D object location based on the depth data.. In applications like vehicle detection for autonomous driving, this location prediction is an important advantage over the traditional 2D image based detection methods. Besides using range scans, previous works also project point cloud onto planes of multiple direction and take the projection as input data. For example, [6] uses bird eye view and the front view projection of point cloud as input data. [32] combine even more projection from uniformly distributed viewing directions.

A drawback of the depthmap based detection approaches is that 2D CNN does not embed spatial information carried by point cloud data, but rather interpret depth as common features. In addition, the projection from 3D points to 2D depthmap inevitably loses information due to occlusion. To overcome these drawbacks, 3D CNN is firstly proposed in several previous works for object recognition [12], [23], [37] and then also used to process volumetric medical data [24]. 3D CNN is a natural extension for 2D CNN. These methods discretize point cloud as voxel data and feed to 3D CNN. [12] also notices that the sparsity of the voxel data can be exploited to accelerate the convolution. [31] proposes 3D R-CNN for indoor object detection combinig Kinect image and point cloud. [9] uses 3D CNN to produce feature on voxel data and fed features to a multiple directional sliding window for detection.

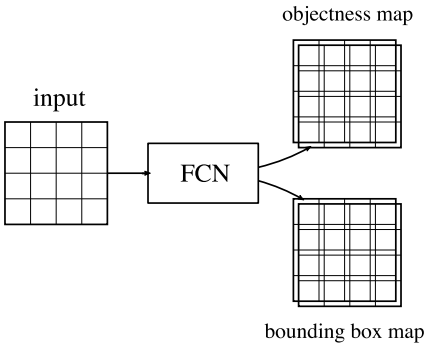


Fig. 1. A sample illustration of the structure of the FCN.

C. Fully Convolutional Network (FCN)

The concept of FCN is originally proposed by [22], aiming to solve the image segmentation task in an end-to-end deep learning framework. Each element in the output map of a FCN predicts the segmentation label of its corresponding input pixel or region. In the recent development of object detection, it is shown that FCN can be extended for end-to-end object detection by augmenting the output channels with object bounding box parameters. This framework does not require any object candidate to be proposed but implicitly detects objectness over the whole input image. In addition, variations of the FCN framework are able to simultaneously predict extra object properties such as bounding box size, aspect ratio, landmark offsets and etc.

Following this framework, several well-performing deep learning approaches have been proposed and achieved good performance in tasks including KITTI vehicle detection, OCR and face detection. These methods include OverFeat [29], DenseBox [15], YOLO [27] and SSD [21]. These methods shares similar formulation and differs in the aspects of data encoding, network structure or post-processing. For example, YOLO divides input map as fixed grid and predicts object category and bounding box for each grid cell. SSD predicts the offsets of bounding boxes with different scales and aspect ratios respectively. Another representative work VeloFCN [19] applies the FCN based detection framework on depthmap data.

In Section III-A, the detailed procedure of the FCN detection framework is further described.

The proposed work in this paper inherits the idea of FCN based detection framework but transplants it to 3D convolution operation. The proposed method takes voxelized 3D data from lidars or RGBD cameras as input and directly predicts the objectness and object shape in the 3D space. To the best of our knowledge, the proposed framework is the first to deploy 3D FCN for the object detection task. Experiments show significant performance improvement of this work over previous methods.

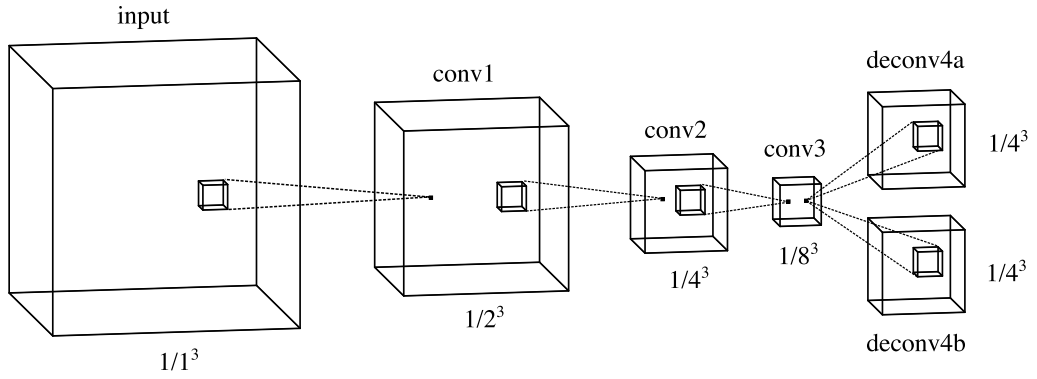


Fig. 2. A sample illustration of the 3D FCN structure used in this paper. Feature maps are first down-sampled by three convolution operation with the stride of $1/2^3$ and then up-samped by the deconvolution operation of the same stride. The output objectness map ($\mathbf{o}^a$) and bounding box map ($\mathbf{o}^b$) are collected from the deconv4a and deconv4b layers respectively.

III. APPROACH

A. FCN Based Detection Revisited

The procedure of FCN based detection frameworks can be summarized as two tasks, i.e. objectness prediction and bounding box prediction. As illustrated in Figure 1, a FCN is formed with two output maps corresponding to the two tasks respectively. The objectness map predicts if a region belongs to an object and the bounding box map predicts the coordinates of the object bounding box. We follow the denotion of [19]. Denote $\mathbf{o}^a_{\mathbf{p}}$ as the output at region $\mathbf{p}$ of the objectness map, which can be encoded by softmax or hinge loss. Denote $\mathbf{o}^b_{\mathbf{p}}$ as the output of the bounding box map, which is encoded by the coordinate offsets of the bounding box.

Denote the groundtruth objectness label at region $\mathbf{p}$ as $\ell_{\mathbf{p}}$. For simplicity each class corresponds to one label in this paper. In some works, e.g. SSD or DenseBox, the network can have multiple objectness labels for one class, corresponding to multiple scales or aspect ratios. The objectness loss at $\mathbf{p}$ is denoted as

$$\begin{aligned} \mathcal{L}_{\text{obj}}(\mathbf{p}) &= -\log(p_{\mathbf{p}}) \\ p_{\mathbf{p}} &= \frac{\exp(-\mathbf{o}^a_{\mathbf{p},\ell_{\mathbf{p}}})}{\sum_{\ell\in\{0,1\}}\exp(-\mathbf{o}^a_{\mathbf{p},\ell})} \end{aligned} \tag{1}$$

Denote the groundtruth bounding box coordinates offsets at region $\mathbf{p}$ as $\mathbf{b}_{\mathbf{p}}$. For simplicity, in this paper we assume only one bounding box map is produced, though a more sophisticated network, e.g. SSD, can have multiple bounding box offsets predicted for one class, corresponding to multiple scales or aspect ratios. Each bounding box loss is denoted as

$$\mathcal{L}_{\text{box}}(\mathbf{p}) = \|\mathbf{o}^b_{\mathbf{p}} - \mathbf{b}_{\mathbf{p}}\|^2 \tag{2}$$

The overall loss of the network is thus denoted as

$$\mathcal{L} = \sum_{\mathbf{p}\in\mathcal{P}} \mathcal{L}_{\text{obj}}(\mathbf{p}) + w\sum_{\mathbf{p}\in\mathcal{V}} \mathcal{L}_{\text{box}}(\mathbf{p}) \tag{3}$$

with w used to balance the objectness loss and the bounding box loss. $\mathcal{P}$ denotes all regions in the objectness map and $\mathcal{V}\in\mathcal{P}$ denotes all object regions. In the deployment phase, the regions with postive objectness prediction are selected. Then the bounding box predictions corresponding to these regions are collected and clustered as the detection results.

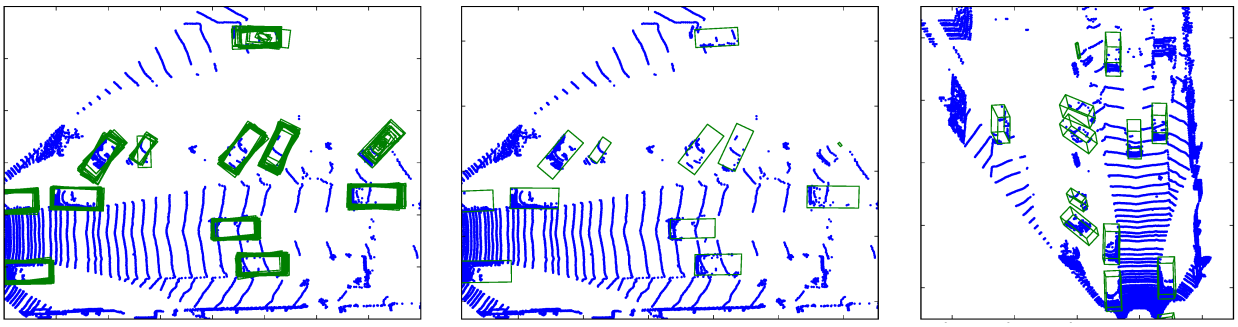
B. 3D FCN Detection Network for Point Cloud

1) Network Structure: The mechanism of 2D CNN naturally extends to 3D on the square grids. Figure 2 shows an example of the network structure used in this paper. The network follows and simplifies the hourglass-shape structure from [22]. Layer conv1, conv2 and conv3 downsample the input map by $1/2^3$ sequentially. Layer deconv4a and deconv4b upsample the incoming map by 2^3 respectively. The ReLU activation is deployed after each layer. The output objectness map ($\mathbf{o}^a$) and bounding box map ($\mathbf{o}^b$) are collected from the deconv4a and deconv4b layers respectively.

The proposed work uses convolution kernel with same size f on three dimensions. Thus for input feature of m dimensions and output feature out n dimensions, this involves f^3mn multiplication to compute one convolution output element, while for 2D case this complexity is f^2mn. In our experiments, we find $f = 4$ a good compromise between kernel capacity and computational complexity.

2) Data Encoding: Although a variety of discretization embedding have been introduced for high-dimensional convolution [1], [12], for simplicity we discretize the point cloud on square grids. The discretized data can be represented by a 4D array with dimensions of length, width, height and channels. For the simplest case, only one channel of binary value $\{0, 1\}$ is used to present whether there is any points observed at the corresponding grid elements. Some more sophisticated features have also been introduced in the previous works, e.g. [23]. Take the vehicle detection task in autonomous driving for example, we use grids of size 10cm to discretize the input point cloud.

Similar to DenseBox [15], we denote the objectness region $\mathcal{V}$ as the center region of the object. For the proposed 3D case, a 3D sphere located at the object center is used. Points inside the sphere are labeled as positive / foreground label. The bounding box prediction at point $\mathbf{p}$ is encoded by the



(a) (b) (c)

Fig. 3. Intermediate results of the 3D FCN detection procedure. (a) Bounding box predictions are collected from regions with high objectness confidence and are plotted as green boxes. (b) Bounding boxes after clustering plotted with the blue original point cloud. (c) Detection in 3D since (a) and (b) are visualized in the bird's eye view.

coordinate offsets, defined as:

$$\mathbf{b}_{\mathbf{p}} = (\mathbf{c}_{\mathbf{p},1}^{\top}, \mathbf{c}_{\mathbf{p},2}^{\top}, \ldots, \mathbf{c}_{\mathbf{p},8}^{\top})^{\top} - (\mathbf{p}^{\top}, \ldots, \mathbf{p}^{\top}) \qquad (4)$$

where $\mathbf{c}_{\mathbf{p},\star}$ define the 3D coordinates of 8 corners of the object bounding box corresponding to the region $\mathbf{p}$, analogous to [19].

3) Training: The training phase of the 3D CNN follows the general idea from FCN based frameworks like [15], [19] by minimizing (3). Similar to previous related methods, since the negative samples account for the dominant majority of training samples, the sample ratio between positive and negative samples should be re-balanced to avoid overfitting on negative samples. In the training phase, $\mathcal{L}(\mathbf{p})$ for negative samples are randomly discarded such that the kept portion is $1 \sim 10$ times the number of the positive samples.

With limited training samples, network models sometimes mistakenly overfit on the correlation between objects that exist in the common sample scene. To remove such correlation and increase the sample diversity in each batch, sample point cloud is randomly cropped such that only a part of the point cloud is put in the batch.

4) Testing: For the testing phase, candidate bounding boxes are extracted from regions predicted as objects and scored by counting its neighbors from all candidate bounding boxes. Bounding boxes are selected from the highest score and candidates overlapping with selected boxes are suppressed.

Figure 3 shows an example of the detection intermediate results. Bounding box predictions from objectness points are plotted as green boxes. Note that for severely occluded vehicles, the bounding boxes shape are distorted and not clustered. This is mainly due to the lack of similar samples in the training phase.

IV. EXPERIMENTS

We evaluate the proposed 3D CNN on the vehicle detection task from the KITTI benchmark [11]. The KITTI dataset contains images aligned with point cloud and object info labeled by both 3D and 2D bounding boxes.

The experiments mainly focus on detection of the *Car* category for simplicity. Regions within the 3D center sphere of a *Car* are labeled as positive samples, i.e. in $\mathcal{V}$. *Van* and *Truck* are labeled to be ignored. *Pedestrian*, *Bicycle* and the rest of the environment are labeled as negative background, i.e. $\mathcal{P} - \mathcal{V}$.

The KITTI training dataset contains 7500+ frames of data, of which 6000 frames are randomly selected for training in the experiments. The rest 1500 frames are used for offline validation, which evaluates the detection bounding box by its overlap with groundtruth on the image plane and the ground plane. The detection results are also compared on the KITTI online evaluation, where only the image space overlap are evaluated.

The KITTI benchmark divides object samples into three difficulty levels. Though this is is originally designed for the image based detection, we find that these difficulty levels can also be approximately used in difficulty division for detection and evaluation in 3D. The minimum height of 40px for the easy level approximately corresponds to objects within 28m and the minimum height of 25px for the moderate and hard levels approximately corresponds to object within 47m.

A. Performance Analysis

The original KITTI benchmark assumes that detections are represented as 2D bounding boxes on the image plane. Then the overlap area of the image plane bounding box with its ground truth is measured to evaluate the detection. However, from the perspective of building a complete autonomous driving system, evaluation in the 2D image space does not well reflect the demand of the consecutive modules including planning and control, which usually operates in world space, e.g. in the full 3D space or on the ground plane. Therefore, in the offline evaluation, we validate the proposed approach in both the image space and the world space, using the following metrics:

- **Bounding box overlap on the image plane**. This is the original metric of the KITTI benchmark. The 3D bounding box detection is projected back to the image plane and the minimum rectangle hull of the projection is taken as the 2D bounding boxes. Some previous point cloud based detection methods [3], [9], [19], [35] also use this metric for evaluation. A detection is accepted

TABLE I

PERFORMANCE IN AVERAGE PRECISION AND AVERAGE ORIENTATION SIMILARITY FOR THE OFFLINE EVALUATION

		Easy	Moderate	Hard
Image Plane (AP)	Proposed	93.7%	81.9%	79.2%
	VeloFCN	74.1%	71.0%	70.0%
Image Plane (AOS)	Proposed	93.7%	81.8%	79.1%
	VeloFCN	73.9%	70.9%	69.9%
Ground Plane (AP)	Proposed	88.9%	77.3%	72.7%
	VeloFCN	77.3%	72.4%	69.4%
Ground Plane (AOS)	Proposed	88.9%	77.3%	72.7%
	VeloFCN	77.2%	72.3%	69.4%

TABLE II

PERFORMANCE COMPARISON IN AVERAGE PRECISION AND AVERAGE ORIENTATION SIMILARITY FOR THE KITTI ONLINE EVALUATION

		Easy	Moderate	Hard
Image Plane (AP)	Proposed	84.2%	75.3%	68.0%
	Vote3Deep [9]	76.8%	68.2%	63.2%
	VeloFCN [19]	71.1%	53.6%	46.9%
	Vote3D [35]	56.8%	48.0%	42.6%
	mBoW [3]	36.0%	23.8%	18.4%
Image Plane (AOS)	Proposed	84.1%	75.2%	67.9%
	VeloFCN [19]	70.6%	52.8%	46.1%
	CSoR	34.0%	25.4%	22.0%

if the overlap area IoU with the groundtruth is larger than 0.7.

- **Bounding box overlap on the ground plane**. The 3D bounding box detection is projected onto the 2D ground plane orthogonally. A detection is accepted if the overlap area IoU with the groundtruth is larger than 0.7. This metric reflects the demand of the autonomous driving system naturally, in which the vertical localization of the vehicle is less important than the horizontal. This metric has been also used in [19].

For the above metrics, the naive Average Precision (AP) and the Average Orientation Similarity (AOS) [11] are both evaluated.

The performance of the proposed approach and [19] is listed in Table I. The proposed approach uses less layers and connections compared with [19] but achieves much better detection accuracy. This is mainly because objects have less scale variation and occlusion in 3D embedding. More detection results are visualized in Figure 4. For our unoptimized GPU implementation, the average running time of network forwarding is 0.5s for VeloFCN and 1s for the proposed 3D FCN. With further optimization, the running time can be reduced. In addition, complicated network structure with more layers can be used to enhance the precision.

B. KITTI Online Evaluation

The proposed approach is also evaluated on the KITTI online system. Note that on the current KITTI object detection benchmark image based detection algorithms outperforms previous point cloud based detection algorithms by a significant gap. This is mainly due to that images have much higher resolution than point cloud (range scan), which enhances the detection of far or occluded objects.

The proposed approach is compared with previous point cloud based detection algorithms and the results are listed in Table II. Deep learning based detection frameworks [9], [19] outperforms the traditional detection approaches by a significant gap. VeloFCN detects 3D objects in a 2D CNN and uses the most complex network to handle perspective scale variation. Vote3Deep operates directly in 3D embedding, similar with the proposed method, but uses a sliding window strategy for the final prediction. Such strategy sometimes does not handle objects with varying sizes and directions suitably. The proposed method naturally predicts the size and direction of object and achievess the best average precision comparing with the related methods.

V. CONCLUSIONS

Recent study in deploying deep learning techniques in point cloud have shown the promising ability of 3D CNN to interpret shape features. This paper attempts to further push this research. To the best of our knowledge, this paper proposes the first 3D FCN framework for end-to-end 3D object detection. The performance improvement of this method is significant compared to previous point cloud based detection approaches. While in this paper the framework are experimented on the point cloud collected by Velodyne 64E under the scenario of autonomous driving, it naturally applies to point cloud created by other sensors or reconstruction algorithms.

ACKNOWLEDGMENT

The author would like to acknowledge the help from Xiaohui Li and Songze Li. Thanks also goes to Ji Wan and Tian Xia.

REFERENCES

[1] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, 29(2):753–762, 2010.

[2] Jens Behley, Volker Steinhage, and Armin B Cremers. Performance of Histogram Descriptors for the Classification of 3D Laser Range Data in Urban Environments. *2012 IEEE International Conference on Robotics and Automation*, pages 4391–4398, 2012.

[3] Jens Behley, Volker Steinhage, and Armin B. Cremers. Laser-based segment classification using a mixture of bag-of-words. *IEEE International Conference on Intelligent Robots and Systems*, (1):4195–4200, 2013.

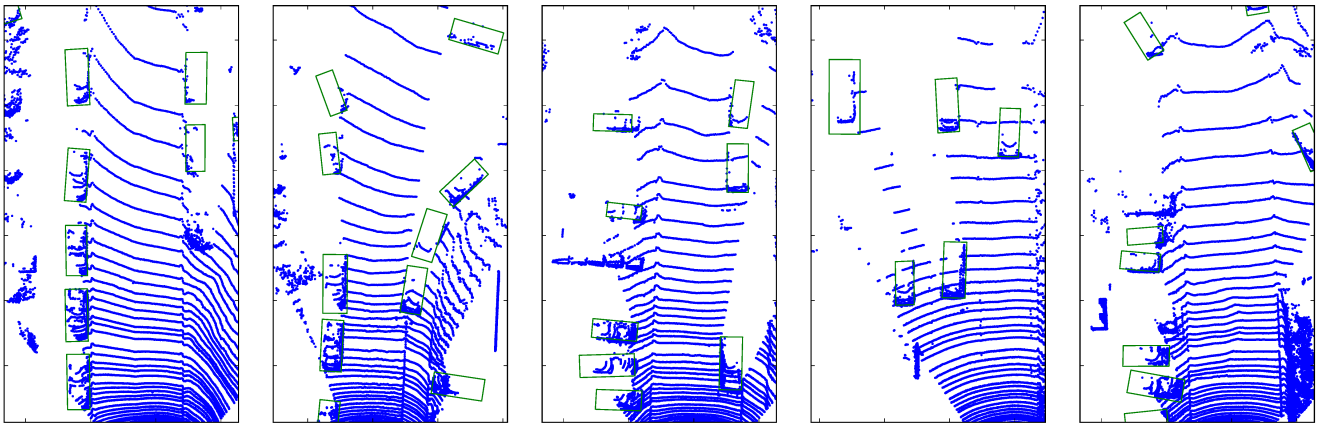
[4] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. *Advances in Neural Information Processing Systems*, pages 424–432, 2015.

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. *arXiv preprint arXiv:1611.07759*, 2016.

[6] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, 2017.

[7] Mark De Deuge, A Quadros, C Hung, and B Douillard. Unsupervised feature learning for classification of outdoor 3d scans. In *Australasian Conference on Robitics and Automation*, volume 2, page 1, 2013.

[8] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, a. Quadros, P. Morton, and a. Frenkel. On the segmentation of 3D lidar point clouds. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2798–2805, 2011.



(a) (b) (c) (d) (e)

Fig. 4. More detection results on the KITTI dataset using 3D FCN.

[9] 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. *arXiv preprint arXiv:1609.06666*, 2016.

[10] O.D. Faugeras and M. Hebert. The Representation, Recognition, and Locating of 3-D Objects. *The International Journal of Robotics Research*, 5(3):27–52, 1986.

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[12] Ben Graham. Sparse 3D convolutional neural networks. *Bmvc*, pages 1–11, 2015.

[13] S Gupta, R Girshick, P Arbeláez, and J Malik. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. *arXiv preprint arXiv:1407.5736*, pages 1–16, 2014.

[14] Michael Himmelsbach, Felix V Hundelshausen, and Hans-Joachim Wünsche. Fast segmentation of 3d point clouds for ground vehicles. *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 560–565, 2010.

[15] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. DenseBox: Unifying Landmark Localization with End to End Object Detection. pages 1–13, 2015.

[16] Andrew E Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[17] Klaas Klasing, Dirk Wollherr, and Martin Buss. A clustering method for efficient segmentation of 3D laser data. *Conference on Robotics and Automation, ICRA 2008. IEEE International*, pages 4043–4048, 2008.

[18] Kevin Lai, Liefeng Bo, and Dieter Fox. Unsupervised Feature Learning for 3D Scene Labeling. *IEEE International Conference on Robotics and Automation (ICRA 2014)*, pages 3050–3057, 2014.

[19] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *Proceedings of Robotics: Science and Systems*, 2016.

[20] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1417–1424, 2013.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2014.

[23] Daniel Maturana and Sebastian Scherer. VoxNet : A 3D Convolutional Neural Network for Real-Time Object Recognition. pages 922–928, 2015.

[24] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.

[25] Frank Moosmann, Oliver Pink, and Christoph Stiller. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 215–220, 2009.

[26] Jeremie Papon, Alexey Abramov, Markus Schoeler, and Florentin Worgotter. Voxel cloud connectivity segmentation - Supervoxels for point clouds. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2027–2034, 2013.

[27] Joseph Redmon, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *arXiv*, 2015.

[28] Max Schwarz, Hannes Schulz, and Sven Behnke. RGB-D Object Recognition and Pose Estimation based on Pre-trained Convolutional Neural Network Features. *IEEE International Conference on Robotics and Automation (ICRA)*, (May), 2015.

[29] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat : Integrated Recognition , Localization and Detection using Convolutional Networks. *arXiv preprint arXiv:1312.6229*, pages 1–15, 2013.

[30] Richard Socher, Brody Huval, Bharath Bath, Christopher D Manning, and Andrew Y Ng. Convolutional-recursive deep learning for 3d object classification. *Advances in Neural Information Processing Systems*, pages 665–673, 2012.

[31] Shuran Song and Jianxiong Xiao. Sliding shapes for 3d object detection in depth images. pages 634–651, 2014.

[32] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*, 2015.

[33] Alex Teichman, Jesse Levinson, and Sebastian Thrun. Towards 3D object recognition via classification of arbitrary object tracks. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4034–4041, 2011.

[34] Rudolph Triebel, Jiwon Shin, and Roland Siegwart. Segmentation and Unsupervised Part-based Discovery of Repetitive Objects. *Robotics: Science and Systems*, 2006.

[35] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. *Proceedings of Robotics: Science and Systems, Rome, Italy*, 2015.

[36] Dominic Zeng Wang, Ingmar Posner, and Paul Newman. What could move? Finding cars, pedestrians and bicyclists in 3D laser data. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4038–4044, 2012.

[37] Zhirong Wu and Shuran Song. 3D ShapeNets : A Deep Representation for Volumetric Shapes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*, pages 1–9, 2015.