DepthCN: Vehicle Detection Using 3D-LIDAR and ConvNet

Alireza Asvadi, Luis Garrote, Cristiano Premebida, Paulo Peixoto and Urbano J. Nunes

***Abstract*—This paper addresses the problem of vehicle detection using Deep Convolutional Neural Network (ConvNet) and 3D-LIDAR data with application in advanced driver assistance systems and autonomous driving. A vehicle detection system based on the Hypothesis Generation (HG) and Verification (HV) paradigms is proposed. The data inputted to the system is a point cloud obtained from a 3D-LIDAR mounted on board an instrumented vehicle, which is transformed to a Dense-depth Map (DM). The proposed solution starts by removing ground points followed by point cloud segmentation. Then, segmented obstacles (object hypotheses) are projected onto the DM. Bounding boxes are fitted to the segmented objects as vehicle hypotheses (the HG step). Finally, the bounding boxes are used as inputs to a ConvNet to classify/verify the hypotheses of belonging to the category 'vehicle' (the HV step). In this paper, we present an evaluation of ConvNet using LIDAR-based DMs and also the impact of domain-specific data augmentation on vehicle detection performance. To train and to evaluate the proposed vehicle detection system, the KITTI Benchmark Suite was used.**

***Index Terms*—Vehicle Detection, 3D-LIDAR, ConvNet**

I. Introduction

Object detection is a crucial component of sensor-based perception systems for advanced driver assistance systems (ADAS) and for autonomous driving. Despite remarkable advancements in object detection using data from cameras, designing a LIDAR-based object detection system for real-world driving applications is a very challenging problem.

Monocular cameras have been the most common sensor technology for object detection. Specifically, high-resolution color cameras are the primary choice to detect traffic signs, license plates, pedestrians, cars, and so on. However, passive vision systems suffer from disadvantages such as night vision incapability, inability of direct depth perception and illumination variations, which limits their use in realistic driving scenarios. By contrast, 3D-LIDARs are robust against the aforementioned problems at the cost of having a higher price and having moving parts. High-definition 3D-LIDARs (with perception range upward 80 m) proved to be a very prominent technology for obstacle detection (*e.g.* [1], [2]). Recently, 3D-LIDARs also started to become used for high-level perception tasks, like object detection [3] and tracking [4].

Influenced by successful applications of 3D-LIDARs in obstacle detection, this paper proposes a vehicle detection system based on obstacle detection as hypothesis generation step and classification (the verification step). The proposed vehicle detection system (herein called DepthCN) depends solely on range data obtained from a LIDAR and has the advantage of making the perception of vehicles' shapes and locations robust to scene illumination changes. The DepthCN finds applications in collision avoidance system, autonomous cruise control, ADAS, and autonomous driving. In this paper, where a sensory perception solution for vehicle detection using 3D-LIDAR data and ConvNets [5], [6] is proposed, the main contributions are:

– Vehicle HG and HV using only 3D-LIDAR data (to explore how far we can go using a single 3D-LIDAR point cloud): HG step is achieved by clustering 3D-LIDAR points (after removing ground points) using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [7], [8]. The resulting clusters are projected onto the DM to generate a set of candidate vehicle hypotheses. HV step is performed by a ConvNet-based vehicle classification.

– Domain-specific data augmentation: Deep ConvNets are known to be data-hungry. Extensive domain-specific augmentation is used to satisfy the data-hungry nature of Deep ConvNets and to deal with the problem of imbalanced datasets.

– Vehicle detection using DM-based ConvNet: unlike 3D sparse point cloud, DM is a dense/up-sampled representation which makes it suitable for object detection. Highly accurate DMs and computed vehicle hypotheses run through the DM-based ConvNet pipeline to achieve vehicle detection.

The DepthCN has shown a promising performance in KITTI Benchmark [9]. Moreover, DepthCN can be extended to detect other object classes (*e.g.*, pedestrian and cyclist), and it is composed of simple yet efficient modules which makes it easy to implement and suitable for ADAS and autonomous driving.

The paper is organized as follows. Section II summarizes the trends and recent related works in object detection. Section III details DepthCN. Experimental results are described in Section IV, and Section V brings some concluding remarks and proposes directions for further research.

II. Object Detection in ADAS and ITS Domains

This section gives an overview of object detection methods, applied in ADAS and ITS domains. Traditional object detection approaches (before the emergence of the Deep Learning) were mainly focused on using hand-crafted features and sliding windows. In this regard, examples of non-deep learning object detection approaches are presented in [3], [10]–[13]. In [10], a dense-depth map using a low-pass upsampling method on LIDAR's point cloud is used, followed by DPM [14] applied to LIDAR-depth map and RGB-image, and

Institute of Systems and Robotics (ISR-UC), Department of Electrical and Computer Engineering (DEEC), University of Coimbra, Coimbra, Portugal {asvadi, garrote, cpremebida, peixoto, urbano}@isr.uc.pt

then a late re-scoring strategy is used to improve detection performance. This kind of method especially works well for eliminating the adverse effects of textured background in HOG feature extraction process and hence shows the value of depth perception for pedestrian detection. Yebes et al. [11] proposed DPM based on 3D-aware HOG-based features extracted from color images and disparity maps. Disparity maps are computed from each pair of left-right images of stereo cameras employing the Semi-Global Matching (SGM) [15] method. The DPM object detector is trained on 3D-aware features. Gonzalez et al. [12] proposed a multi-view classifier to take into account different views and poses of objects. In their approach, a dense-depth map is computed from Velodyne LIDAR, similar as in [10]. Next, HOG and Local Binary Pattern (LBP) descriptors are used to extract shape and texture data from RGB image and depth map. The SVM is used for holistic object modeling, and Random Forest (RF) is used as a patch based object model. They considered early and late fusions, and achieved a better result with the early fusion framework. Behley et al. [3] proposed a segmentation-based object detection using LIDAR range data. In their method, a hierarchical segmentation is used to reduce the over- and under-segmentation effects. A mixture of multiple bag-of-word classifiers is applied to classify extracted segments. Finally, a non-maximum suppression is used considering the hierarchy of segments. In the Wang and Posner [13] approach, LIDAR points together with their reflectance values are discretized into a coarse 3D voxel grid. A 3D sliding window detection approach is used to generate the feature grid. At each window location, the feature vectors contained within its bounds are stacked up into a single long vector and passed to a classifier. A linear SVM classifier scores each window location and returns a detection score.

Deep learning has been receiving special attention recently with some research focused on applying ConvNet for object detection using monocular camera vision for IV/ITS application. Some of the state-of-the-art object detectors are based on region-based ConvNets: Fast R-CNN [16] and Faster R-CNN [17]. Fast R-CNN uses Spatial Pyramid Pooling networks (SPPnets), and Faster R-CNN uses a Region Proposal Network (RPN) for region proposal generation. Xiang et al. [18] introduced a ConvNet-based region proposal network that uses subcategory information to guide the proposal generating process. In their approach Fast R-CNN [16] is modified by injecting subcategory information (using 3D Voxel Patterns as subcategories) into the network for joint detection and subcategory classification. Chabot et al. [19] introduced Deep MANTA, a framework for 2D and 3D vehicle detection in monocular images. In their method, inspired by the Region Proposal Network (RPN) [17], vehicle proposals are computed and then refined to detect vehicles. They optimized ConvNet for six tasks: region proposal, detection, 2D box regression, part localization, part visibility and 3D template prediction. Cai et al. [20] proposed a multi-scale object detection based on the concept of rescaling the image multiple times, so that the classifier can match all possible object sizes. Their approach consists of two ConvNet-based sub-networks: a proposal sub-network and a detection sub-network learned end-to-end. Some approaches try to incorporate 3D-LIDAR data for the object detection task. Chen et al. [21] proposed a framework using LIDAR point cloud and RGB images that predicts 3D Bounding Boxes (BBs) of objects. A top-view representation of point cloud is used to generate 3D proposals. Then, the 3D proposals were projected onto three feature maps (LIDAR's top and front views and RGB image). A deep region-based fusion network is used for 3D object detection. Kim and Ghosh [22] used Fast R-CNN [16] for vulnerable road user (pedestrians and cyclists) detection using RGB and LIDAR data. The main idea of their approach is extending the selective search region proposals of the Fast R-CNN to integrate LIDAR data (by providing more object proposals using data from LIDAR). Li et al. [23] used a 2D Fully-Convolutional Network (FCN) in a 2D point map (in top-view) and trained it end-to-end to build a vehicle detection framework based on only 3D-LIDAR data.

In contrast with other works, instead of doing sliding windows (which wastes a significant amount of computation time on BBs with no objects), employing features or designing a network for obtaining object hypotheses, the DepthCN method is based solely on 3D-LIDAR data to generate class-agnostic object proposals (which represent obstacles in driving environment) and class-specific (vehicles) detections. LIDAR-based DMs and DM-based ConvNet are used to decide whether or not an obstacle is a vehicle. Although DepthCN explores a solution using only LIDAR data, it can be fused with RGB image-based object detector to improve the overall performance. Moreover, due to the class-agnostic hypotheses, detection of other object classes can be included in our framework with a marginal increment of computational cost.

III. 3D-LIDAR-based Vehicle Detection Using ConvNet

A. Problem Formulation and System Overview

In this paper, we introduce the DepthCN approach which takes 3D-LIDAR Point Cloud Data (PCD) as input and predicts 2D Bounding Boxes (BBs) of vehicles. The architecture of DepthCN is presented in Fig. 1. The approach comprises two stages: 1) the offline learning stage to optimize HG and HV steps (more details on section IV-B), 2) the online vehicle detection stage. The online detection starts with removing ground points, clustering on-ground obstacles, projecting the segmented obstacles onto 3D-LIDAR-based Dense-depth Map (DM), and finally ConvNet-based hypothesis evaluation to check whether a proposal contains a vehicle or not.

B. Hypothesis Generation (HG) Using 3D-LIDAR Data

Objects in driving environment may appear in different sizes and locations. The state-of-the-art approaches speed-up the detection process using a set of object proposal/hypothesis instead of an exhaustive sliding windows search. In this paper, vehicle proposals are generated solely from 3D-LIDAR data, applying three steps.

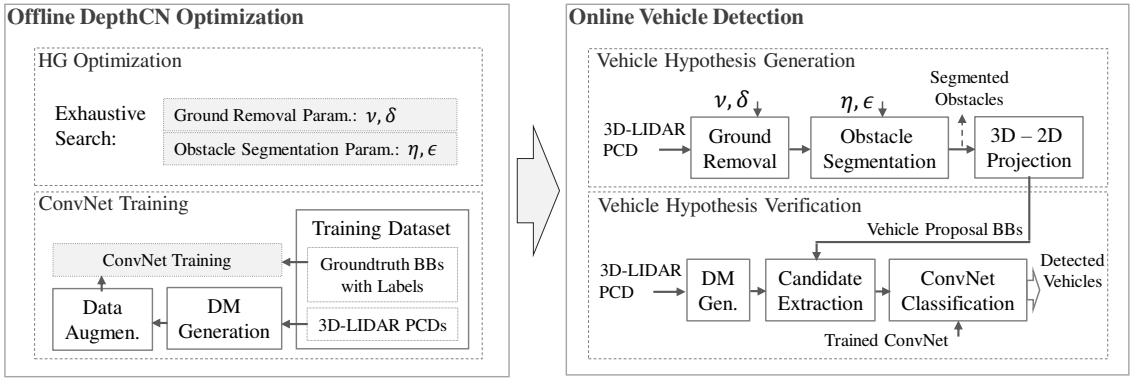

Fig. 1. The proposed 3D-LIDAR-based vehicle detection algorithm (DepthCN). The components of the system are detailed in Section III.

1) Grid-based Ground Removal: To increase the quality of object proposals and to reduce unnecessary computations, points that belong to the ground first need to be removed. In a grid-based framework, ground points are eliminated by rejecting cells containing points with low variance in Z-dimension.

2) Obstacle Segmentation for HG: 3D-LIDARs have previously shown promising performance for obstacle detection [1], [2]. Taking this into account, we explore a HG technique using data from a 3D-LIDAR. This approach generates a set of class-agnostic proposals which represent obstacles in the environment. After removing ground points, by applying DBSCAN [7], [8] on the top-view X-Y values of the remaining points, 3D-LIDAR points are segmented into separate clusters where each cluster represents an obstacle. The segmented obstacles are then projected onto the DM (using LIDAR to camera calibration matrices), and the fitted 2D BB for each cluster is assumed as an object hypothesis (see Fig. 2).

C. Vehicle Classification using DM and ConvNet

The ConvNet classifier focuses on identifying vehicles from the set of object hypotheses onto 3D-LIDAR-based DM (as illustrated in Fig. 3). At this stage, the system encompass: DM generation from 3D-LIDAR, data augmentation to improve classification accuracy, and DM-based ConvNet for vehicle Hypothesis Verification (HV).

1) DM Generation: The LIDAR Dense-depth Map (DM) generation is made by projecting sparse 3D-LIDAR's point cloud on camera coordinate system, performing interpolation and encoding. The Delaunay Triangulation is used for mesh generation from the projected sparse depth points on the camera coordinate system. In this approach, the pixel values which lie within a triangle are estimated by depth values of the surrounded triangle vertices using Nearest Neighbor interpolation. DM is converted to 8-bit integer gray-scale image format using the Range Inverse method which assigns more bits to closer depth values.

2) Imbalanced Data and Data Augmentation: Data augmentation is the process of generating a large training dataset from a small dataset using different types of transformations in a way that a balanced distribution is reached, while the new dataset still resembles the distribution that occurs in practice. In this paper, a set of augmentation operations like scaling, flipping, jittering, cropping, rotation, brightness (depth) and aspect-ratio augmentation, and shifting each line with different small random biases is performed to aggregate and balance the training dataset with two major goals and benefits: i) Balancing data for classes: reducing bias of ConvNet, ii) Increasing data: helping ConvNets to tune large number of parameters in the network.

3) ConvNet for Hypothesis Verification (HV): ConvNet is used as the HV core in DepthCN. The ConvNet input size is set as $66 \times 112 \times 1$ where 66 and 112 are the average ground-truth vehicle height and width (in pixels) in the training dataset. A vehicle proposal BB in the DM is extracted as the vehicle candidate (candidate extraction), resized to 66×112 and inputted to ConvNet for classification. The ConvNet employed in DepthCN is composed by 2 Convolutional layers, 3 Rectified Linear Units (ReLUs), 2 Pooling layers, 2 Fully Connected (FC) layers, a Softmax layer, and a Dropout layer for regularization (as illustrated in Fig. 4). Each component of ConvNet architecture is described briefly in the following.

– By applying convolution filters across input data, feature maps are computed. The first and the second convolutional layers contain 32 filters of $5 \times 5 \times 1$ and 64 filters of $5 \times 5 \times 32$ respectively, with stride 1 and padding 2. The learned convolutional filters in the first layer are shown in Fig. 5.

– ReLUs use the non-saturating activation function to increase the nonlinear properties of the network.

– Max-pooling (with stride 2 and padding 0) is used to partition the input into a set of 3×3 rectangles. It outputs the maximum value for each sub-region.

– FC layers have full connections to all activations in the previous layer. Two FC layers with 64 and 2 neurons are used to provide the classification output.

IV. Experimental Setup and Evaluation

A. KITTI Dataset for DM-based Vehicle Detection

DepthCN is evaluated using Object Detection Evaluation from KITTI Vision Benchmark Suite [9]. DepthCN is relying on Velodyne LIDAR (range data only). The maximum range

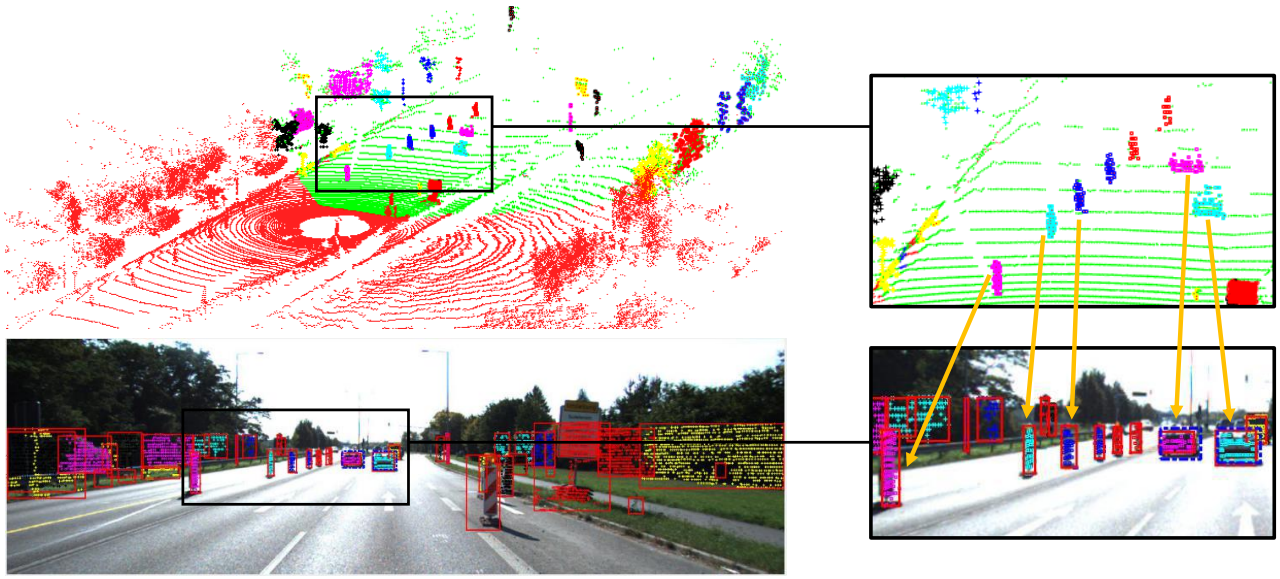

Fig. 2. HG using DBSCAN in a given point cloud. Top shows the point cloud where the detected ground points are denoted with green and LIDAR points that are out of the field of view of the camera are shown in red. The segmented obstacles are shown with different colors. The bottom image shows the projected clusters and HG results in the form of 2D BB. The image frame here is used only for visualization purpose. The right-side shows the zoomed view, and the vertical orange arrows indicate corresponding obstacles. The two vehicles shown by dashed-blue BBs indicate ground-truth.

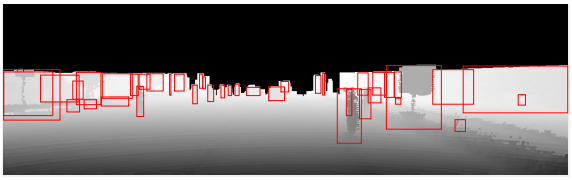

Fig. 3. The generated Dense-depth Map (DM) with the projected hypotheses (41 object proposals are depicted with red rectangles). For viewing the corresponding RGB image and 3D-LIDAR data please refer to Fig. 2.

in DepthCN algorithm is limited to 80 m. The KITTI Object Detection Evaluation contains 7481 frames in the training dataset with 51867 labels for 9 different categories: Pedestrian, Car, Cyclist, Van, Truck, Person sitting, Tram, Misc, and Don't care. In this paper, the 'Car' class is trained as the positive class to DepthCN and considered for evaluation. The training dataset of original KITTI dataset was divided into two sections: training (80%) and validation (20%) data, and DepthCN was optimized for the latter training and validation data.

B. DepthCN Optimization

DepthCN is composed by Hypothesis Generation (HG) and Hypothesis Verification (HV) modules. The optimization of these modules are performed offline as follows.

1) HG Optimization: In the grid-based ground removal, the parameters are grid cell size (υ) and variance threshold (δ). The minimum number of points (η) and the distance metric (ε) are related to DBSCAN. The optimal parameter values (in Table I) for ground removal and clustering were optimized jointly, using exhaustive search, by maximizing the overlap of generated hypotheses with ground-truth BBs (minimum overlap of 70%).

TABLE I
PARAMETERS USED IN DEPTHCN (υ, δ, AND ε IN METERS).

υ	δ	η	ε
0.5	0.01	5	0.5

2) ConvNet Training using Augmented DM Data: The ConvNet was trained on the augmented KITTI 3D-LIDAR-based DMs. The Stochastic Gradient Descent (SGD) with a momentum of 0.9, a mini-batch size of 128, and max epochs of 40 with L2 regularization was employed for the ConvNet training.

C. Evaluation of 3D-LIDAR-based Vehicle Detection

DepthCN was evaluated in terms of classification and detection accuracy and computational cost. Results are provided in the next subsections.

1) Performance Analysis of Classification: Considering an input DM of size 66×112, the accuracy of the implemented ConvNet for vehicle classification with and without using data augmentation is reported in Table II. The data augmentation improved the accuracy by more than 5 percentage points.

2) Vehicle Detection Evaluation: DepthCN was evaluated against mBoW [3] which is one of the most relevant methods and like ours operate directly on 3D-LIDAR's range data. mBoW uses hierarchical segmentation with bag-of-word classifiers whereas DepthCN uses DBSCAN with a ConvNet classifier. Results for vehicle detection are given in terms of average precision (AP) in Table III. As can be noted from the table, DepthCN surpasses by about 1.5 percentage points the

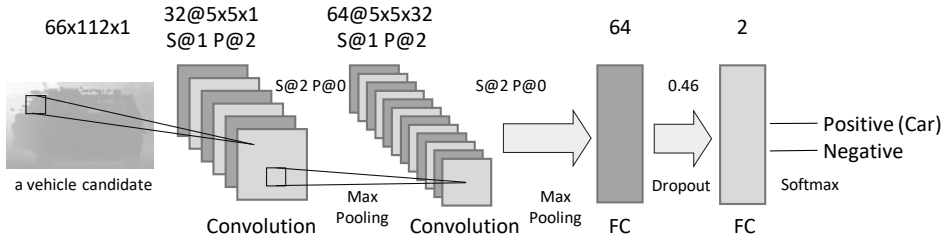

Fig. 4. The ConvNet architecture.

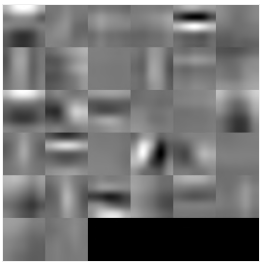

Fig. 5. The 32 convolutional filters learned in the first layer of ConvNet.

TABLE II
THE CONVNET'S VEHICLE RECOGNITION ACCURACY WITH (W) AND WITHOUT (WO) APPLYING DATA AUGMENTATION (DA).

Dataset	WO-DA	W-DA
Train set	92.83%	96.02%
Validation set	86.69%	91.93%

mBoW in Easy difficulty level. A Precision-Recall curve is shown in Fig. 6, and qualitative results are provided in Fig. 7.

3) Computational Analysis: Preliminary experiments with DepthCN were performed using a Hexa core 3.5 GHz processor powered with GTX 1080 GPU and 64 GB RAM under MATLAB R2017a. The runtime of DepthCN (unoptimized implementation) for processing a point cloud is about 2.3 seconds in comparison with 10 seconds processing time of mBoW (implemented in C/C++) under 1 core 2.5 Ghz.

V. CONCLUDING REMARKS AND FUTURE DIRECTIONS

In this paper, we introduced the DepthCN: a vehicle detection system based on HG-HV paradigm using a Deep ConvNet and range data from a 3D-LIDAR. HG is performed by applying DBSCAN on point cloud data (after removing ground points). In the HV phase, a DM is generated using 3D-LIDAR data. DM values in HG are inputted to ConvNet for vehicle detection.

As future works we plan to exploit the hypotheses, that are generated in 3D, for 3D object perception. Detection of other

TABLE III
DEPTHCN VEHICLE DETECTION EVALUATION ON KITTI TEST SET.

Benchmark	Easy	Moderate	Hard
DepthCN	37.59 %	23.21 %	18.01 %
mBoW [3]	36.02 %	23.76 %	18.44 %

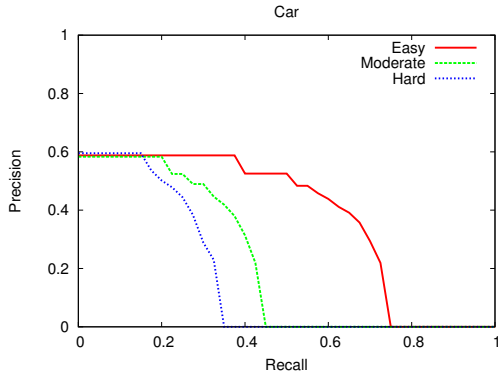

Fig. 6. Precision-Recall on the KITTI testing dataset for easy, moderate and hard Car detection difficulty levels.

object classes can be considered and explored for future works. Moreover, DepthCN can be combined with a RGB image-based detector in a fusion framework to improve the object detection performance.

ACKNOWLEDGMENTS

This work has been financially supported by the European Union, Compete 2020 and Portugal 2020 programs under grant UID/EEA/00048/2013; and by "AUTOCITS: Regulation Study for Interoperability in the Adoption of Autonomous Driving in European Urban Nodes" - Action number 2015-EU-TM-0243-S, co-financed by the European Union (INEA-CEF).

REFERENCES

[1] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[2] A. Asvadi, C. Premebida, P. Peixoto, and U. Nunes, "3d lidar-based static and moving obstacle detection in driving environments: an approach based on voxels and multi-region ground planes," *Robotics and Autonomous Systems*, vol. 83, pp. 299–311, 2016.

[3] J. Behley, V. Steinhage, and A. B. Cremers, "Laser-based segment classification using a mixture of bag-of-words," in *IROS*, 2013.

[4] A. Asvadi, P. Girão, P. Peixoto, and U. Nunes, "3d object tracking using rgb and lidar data," in *ITSC*, 2016.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.

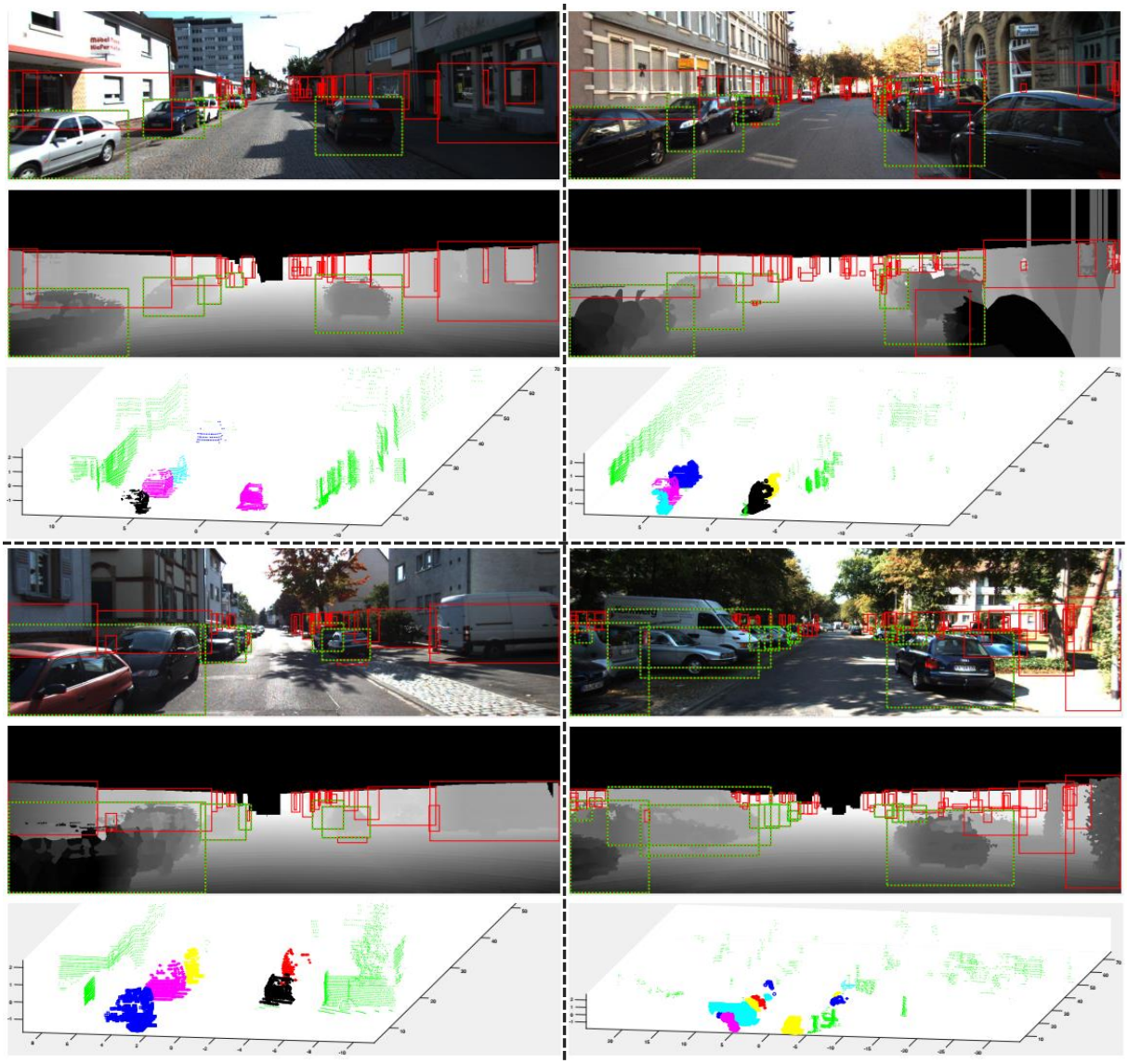

Fig. 7. Screenshot examples of DepthCN detection results (four pairs of DM and color images with the corresponding PCDs). The generated hypotheses and the detection results are shown as red and dashed-green BBs, respectively, in both DM and color images. The bottom figures show the result in the PCD, where the detected vehicles' clusters are shown in different colors, and the remaining LIDAR points are shown in green. Notice that the depicted color-images are just to make visualization and understanding easier.

[8] T. N. Tran, K. Drab, and M. Daszykowski, "Revised DBSCAN algorithm to cluster data with dense adjacent clusters," *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92–96, 2013.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[10] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian detection combining RGB and dense LIDAR data," in *IROS*, 2014.

[11] J. J. Yebes, L. M. Bergasa, and M. García-Garrido, "Visual object recognition with 3D-aware features in KITTI urban scenes," *Sensors*, vol. 15, no. 4, pp. 9228–9250, 2015.

[12] A. González, G. Villalonga, J. Xu, D. Vázquez, J. Amores, and A. M. López, "Multiview random forest of local experts combining RGB and LIDAR data for pedestrian detection," in *IV*, 2015.

[13] D. Z. Wang and I. Posner, "Voting for voting in online point cloud object detection." in *Robotics: Science and Systems*, 2015.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[15] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE TPAMI*, vol. 30, no. 2, pp. 328–341, 2008.

[16] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[18] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *WACV*, 2017.

[19] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in *CVPR*, 2017.

[20] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *ECCV*, 2016.

[21] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *CVPR*, 2017.

[22] J. G. Taewan Kim, "Robust detection of non-motorized road users using deep learning on optical and LIDAR data," in *ITSC*, 2016.

[23] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D LIDAR using fully convolutional network," in *Robotics: Science and Systems*, 2016.