3D Vehicle Detection Using Cheap LiDAR and Camera Sensors

Sabeeha Mehtab, Wei Qi Yan, Ajit Narayanan
Auckland University of Technology, Auckland 1010 New Zeealand

Abstract—**Autonomous Vehicles (AVs) are expected to be intelligent enough to perceive the world accurately in terms of avoiding road obstacles. Remarkable progress has been made in 3D road scene perception of AVs through machine learning and computer vision methods, but existing solutions rely on expensive 64 beams LiDAR point clouds for the 3D positioning of objects. In this paper, we propose a simple yet effective approach that is based on the success of 2D object detection to estimate 3D positions of the vehicles in front of AVs. Our approach relies on camera RGB images for predicting size and orientation of 3D bounding boxes of AVs by using a novel deep neural network (DNN) and LiDAR 3D point clouds for distance estimation. For testing and training, KITTI and Waymo datasets are employed. We have converted 64 beams of LiDAR point clouds into 32 and 16 beams point clouds for model performance analysis. Based on the results, the proposed method proved to be robust with sparse point clouds without compromising accuracy.**

Keywords— LiDAR, point clouds, 3D vehicle detection, autonomous vehicles, self-driving car, deep learning, fusion

I. INTRODUCTION

An AV system is composed of three major components: Sensing and perception, localizing and mapping, applications for driving policy [1]. In this paper, we consider the sensing and perception aspects that are responsible for positioning objects in the surrounding environment of autonomous vehicles so that information can be collected for manoeuvre and obstacle avoidance [2].

While human drivers make use of visual, auditory, and cognitive senses, AVs perceive the world with multiple sensors to overcome the shortcomings of using only one. There are several types of sensor mounted on autonomous vehicles for visual object detection: Passive sensors, such as monocular, stereo cameras, and infrared cameras; active sensors including GPS, LiDAR, IMU, radar and sonar [3]. In recent years, most 3D object detection algorithms have preferred to use monocular cameras and LiDAR sensors. A monocular camera outputs digital images in texture and reveals the shape of the objects in the form of pixel values, such as RGB, YUV, or other colour systems [4]. On the other hand, LiDARs detect object location accurately in the range of meters but have radiation limitation [5]. Unlike digital cameras, LiDARs cannot discriminate objects based on texture and colour [5]. LiDAR data has been utilized in multiple ways in AV systems, for example, front-view 2D projected form [6], top-view or BEV projected form [7], voxel form (ie.e, 3D grid cell) [8], or raw form only [9][10]. Methods need to be found for integrating the information from LiDAR and cameras to work together for visual object detection.

With the advent of low-cost GPUs and DNNs, there has been remarkable progress made in the field of 2D object detection based on digital images by using the algorithms like Faster R-CNN [11], YOLOv4 [12], and YOLOv5 [13]. However, in order to make successful manoeuvre of AVs, fine estimate of distance and shape of the front lying obstacles is necessary [14]. In this paper, we propose a cost-effective solution for 3D vehicle detection in the context of AVs. A novel framework to estimate 3D bounding boxes and orientations of the front lying vehicles is proposed based on the success of 2D vehicle detection by using deep neural networks (DNNs). The contribution of this paper is summarized as follows:

- An specialized DNN is proposed that exploits MobilNetV2 [33] for feature extraction with three detection branches pertaining to 3D box size, orientation and confidence score.
- The algorithm works on the fact that the 3D centre of a vehicle is the translation of its 2D centre in the form of world coordinates.
- Relying on actual information of LiDAR beams, vehicle distance from AVs is estimated by using trigonometry concepts.
- In the model, a specialized algorithm is proposed to mitigate occlusion problem based on top-mounted LiDAR.
- For experiments and results, the benchmark KITTI dataset [15] is employed, which is further enhanced by using the Waymo dataset. The model performance is tested over multiple beams LiDAR point clouds to verify the robustness of the results.

A comprehensive literature review is carried out in Section II for understanding background knowledge of this topic and identifying the research gap in existing work. In Section III, our proposed methods are described. In Section IV, the result analysis is presented, and in Section V, the conclusion is drawn and future work is proposed.

II. RELATED WORK

In this section, we discus some of the previous research work related to our domain and scope.

A. Image-Based 3D Object Detection

A myriad of algorithms are proposed to detect 3D bounding boxes from camera images via exploring into 2D object detection windows. In [16], the projection of a 3D bounding box fits tightly within its 2D window. A hybrid continuous loss was proposed by using multiple bins for object orientation prediction; the distance was poorly estimated. From monocular images and depth maps [17], a 3D object proposal was put forward for generating class independent proposals. The proposals were re-ranked based on monocular images before passing into the detection network; however, the heavy calculations slowed down the speed of visual object detection. In [18], a dictionary of 3D voxel patterns (3DVP) with various classes of cars were employed for training the classifiers.

B. Point Clouds-Based 3D Object Detection

LiDAR point clouds have been exploited in three major ways for 3D bounding box estimation: Projection-based, voxel-based, and raw point clouds based. Cylindrical projection of point clouds that gives a front view of AV was fed into a fully CNN to detect the positions of 3D objects [19]. BirdNet [20] was applied to generate BEV maps based on height, intensity, and density of point clouds for feeding into a DNN after density normalization; however, the intensity led

978-1-6654-0645-1/21/$31.00 ©2021 IEEE

to poor results. In complex YOLO [21] faster detection method was proposed by using BEV point clouds based on Euler-RPN to predict five 3D anchor boxes per grid cell.

In VoxelNet [8], vehicle detection was introduced by using point clouds with a voxel feature encoding (VFE) layer to generate point-wise concatenated features of vehicles. The information was passed into a region proposal network (RPN) to predict 3D localization. With [8], in SECOND [22] and SARPNET [23], voxel-based 3D road second perception was proposed; however, computational efficiency and memory consumption of voxels remain the major bottleneck for these algorithms [24].

The raw point clouds are exploited for treating every point in the cloud independently based on PointNet [9]. FVNet [24] firstly generates a 2D region proposal based on front view projection of point clouds and extends 2D bounding boxes into 3D point cloud frustum with truncated radial distances. F-ConvNet [25] categorized the point clouds frustum into multiple groups based on distance ranges. These groups of spatial point clouds pass through parallel PointNets to aggregate local point-wise features. PointRCNN [26] segments the whole point clouds into foreground and background parts to generate high-quality 3D proposals along with semantic features.

C. Sensors Fusion-Based 3D Object Detection

In MV3D, a fusion of point clouds BEV maps, front view maps, and camera images was proposed to detect 3D bounding boxes by using a multistage fusion approach in CNN. MV3D did not perform well in detecting small objects. AVOD followed MV3D by using encoder-decoder for multilevel features extraction. Using PointNets [9], FrustumNet [27] detects 3D objects in the 2D detection window based on point clouds frustum.

The general trend in current research, as identified in the literature survey, is that image-based 3D detection algorithms rely on patterns and key features, with this approach failing to achieve the desired results in complex environmental conditions. On the other hand, 3D point clouds based approaches depend on costly and dense LiDARs point clouds to achieve desirable results. Given current research literature, there is a scope to investigate cheaper and more effective 3D object detection using camera images and less costly LiDAR point clouds.

III. Our Methodology

In this paper, we propose a comparatively robust yet straightforward approach to 3D vehicle detection. The proposed method builts on top of existing work in 2D vehicle detection. We assume 2D bounding boxes of front lying cars are already in place. The proposed model is split into two parts. In the first part, the size and orientation of 3D bounding boxes of front lying vehicles are predicted along with the confidence score. In the second part, LiDAR point clouds are projected onto image coordinates to estimate the distance of each detected car. In the process of projection, the distance information of point clouds is preserved as a third channel.

A. Size and Orientation Estimation for 3D Bounding Boxes

Our aim in 3D object detection of cars is not only to solve the problem of correctly predicting the 3D coordinates of bounding boxes but also to give real-time performance. MobileNetV2 [28] is 11.7 times smaller in size than VGG-16 net [29] with comparable performance, and may be used as a feature extractor [30, 38, 39]. Fig. 1 shows the proposed DNN architecture consisting of 17 bottleneck residual blocks, which has three operators: A 1×1 linear transformation layer, a depth-wise separable convolution that does lightweight filtering by applying a single convolutional filter per input channel, the last 1×1 convolution layer with ReLU6 activation function that performs dimensionality reduction and combines the filtered data creates new features [31]. The proposed network finishes with three detection branches based on fully connected layers.

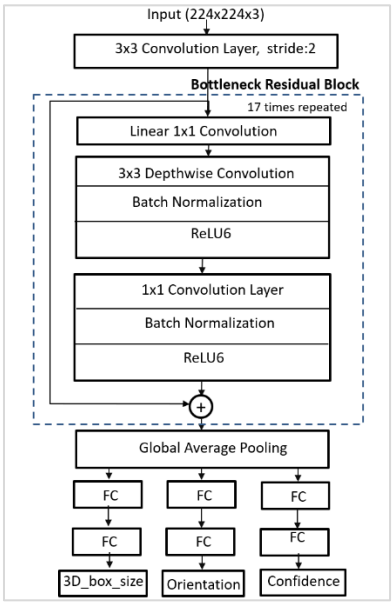


Fig. 1. The architecture of MobilNetV2-based DNN to predict the orientation and confidence of a 3D bounding box

The first branch is responsible for estimating the size of 3D bounding box by using mean squared error (MSE). The second branch conducts orientation prediction by using L2 loss, which plays a crucial role in final vehicle position detection. The third branch regresses the confidence of car orientations by using the softmax loss function. In the literature reviewed, it is found that existing methods remain susceptible to the orientation difference of 180° rotated cars [7]. Our architecture remedies this problem by considering two proposals in the intervals (0, -180°) and (0, 180°) to predict the car orientation and confidence; the one with the higher confidence is selected, which is found to be much robust. The net loss of the network is calculated by using the weighted sum of all branches as given in Eq. (1), where α and β are multiple coefficients of orientation and confidence loss respectively.

$$L_{\text{size}} = L_{\text{direct}} + \alpha \cdot L_{\text{orient}} + \beta \cdot L_{\text{conf}} \quad (1)$$

B. The Estimation of Centre Coordinates of 3D Bounding Boxes

Once the bounding box size and orientation information of vehicles are predicted, the second task is to estimate the centre distance between AVs. The algorithm works based on the fact that the 2D centre of a vehicle is directly related to its 3D centre by projecting world information on 2D image coordinates. In order to implement this approach, LiDAR point clouds are projected based on the 2D image, however with the following constraint conditions: 1) Ground points are

removed based on the LiDAR position on the AV; 2) Only points in the 2D detection windows are utilized. Point clouds are projected onto the image coordinate by using the calibration parameters while preserving the distance information intact in the form of an additional channel.

In order to convert 3D point $\boldsymbol{X} = (x, y, z)^{\mathrm{T}}$ into corresponding camera coordinate $\boldsymbol{Y} = (p, q, r)^{\mathrm{T}}$, operations such as translation, rotation and projection are carried out to execute Affine transformation [32] as given in Eq.(2):

$$Y = P_{rect} \cdot R' \cdot X, \tag{2}$$

$$P_{rect} = \begin{pmatrix} f_u & 0 & c_u & -f_u b_x \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

and

$$R' = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where P_{rect}, (f_u, f_v) and (c_u, c_v) denote focal length and optical parameters of the camera across x-axis and y-axis, respectively; b_x stands for the baseline with respect to the reference camera [15]. R' and r_{ij} represent rotation parameters and (t_x, t_y, t_z) is translation across x, y and z axes. Furthermore, Y is converted to 2D image coordinate (u, v) as Eq. (3),

$$\begin{cases} u = p/r \\ v = q/r \end{cases} \tag{3}$$

As shown in Fig. 2, the pose of vehicles is first detected by using the predicted size and orientation of 3D bounding boxes. In the case of the central vehicle, if $x_1 > (x_1+x_2)/2$, then the vehicle pose is a longitudinal side, else it orients to the front side. Tthe same principle is applied to all directions. In Fig. 3, the vehicle heading in different, forward directions is illustrated. The blue dots represent the predicted 2D centre (t_x, t_y) of the vehicles, whereas the yellow dots stand for the outermost 3D point (l_x, l_y, l_z) across 2D centres. Finally, (t_x, t_y, l_z) is considered the projected centre point on the 3D bounding box surface. However, to get the exact depth estimation, we need further analysis.

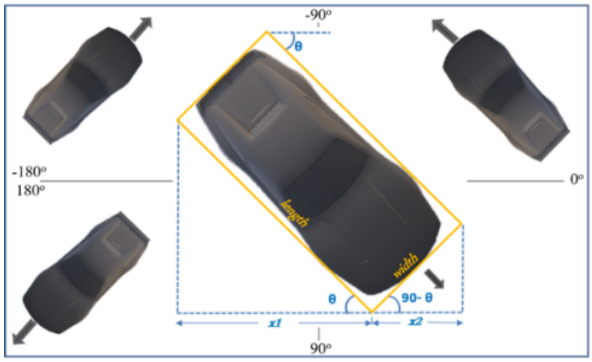


Fig. 2. The pose estimation of the size and orientation of 3D bounding boxes of vehicles. The vehicle pose of longitudinal or front/back sides depends on its orientation and size.

Fig. 3. The blue dots represent 2D centres of predicted 2D bounding boxes, whilst the yellow dots refer to the 3D outermost point on the central vertical axis on the car's surface. The right arrow shows the reference direction.

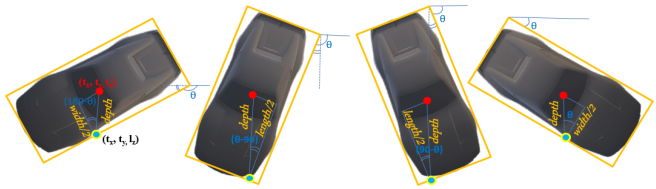
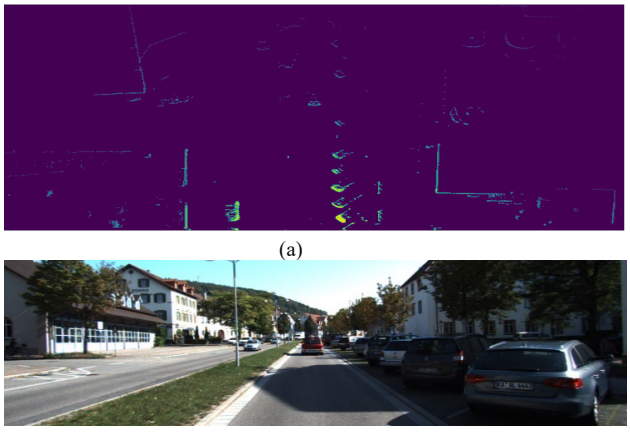


Fig. 4. The top view of cars oriented in different directions with the 3D centres represented in red circles. It shows the estimation of 3D vehicle centres through predicted 2D centres. The depth value of 3D centre from the surface point is based on estimated pose, orientation, size of the 3D boubnding box.

TABLE I. THE CALCULATION OF DEPTH ESTIMATION OF 3D CAR CENTRES BASED ON THE ORIENTATION, SIZE AND POSE OF 3D BOUNDING BOXES OF CARS

	Pose	Depth calculations
Orientation |90o|	Longitudinal Side	$\cos(\theta) = \frac{width/2}{depth}$ $\Rightarrow$ depth = $\text{width}/2.\cos(\theta)$
	Front/Back	$\cos(90-\theta) = \frac{length/2}{depth}$ $\Rightarrow$ depth = $\text{length}/2.\sin(\theta)$
Orientation |90o|	Front/Back	$\cos(\theta-90) = \frac{length/2}{depth}$ $\Rightarrow$ depth = $-\text{length}/2.\sin(\theta)$
	Longitudinal Side	$\cos(180-\theta) = \frac{width/2}{depth}$ $\Rightarrow$ depth = $-\text{width}/2.\cos(\theta)$



(a)

(b)

Fig. 5. A sample frame from KITTI dataset (a) BEV view of LiDAR point clouds after removed the ground points. (b) Image of the same frame. The occluded vehicles that cannot be seen in images are detectable in point clouds.

Fig. 4 illustrates the 2D centres (blue circle with yellow line) projection on 3D centre (red circle) of cars for different poses by using the predicted size of the 3D bounding box and orientation with respect to the reference direction. The trigonometry for calculating the depth of car centre is given in Table I, the final distance estimation of the car from AV is deduced by using Eq. (4).

$$t_z = l_z + abs(depth) \quad (4)$$

C. Occlusion

In many road scenarios, there exists a high degree of occlusion among cars that is tackled by an extension using LiDAR point clouds. In Fig. 5, we retrieve more information of occluded cars in point clouds BEV view rather than from image view. In order to retrieve relative information of occluded cars, we identify the point clusters in each 2D detection window based on points depth gap threshold after removing outliers.

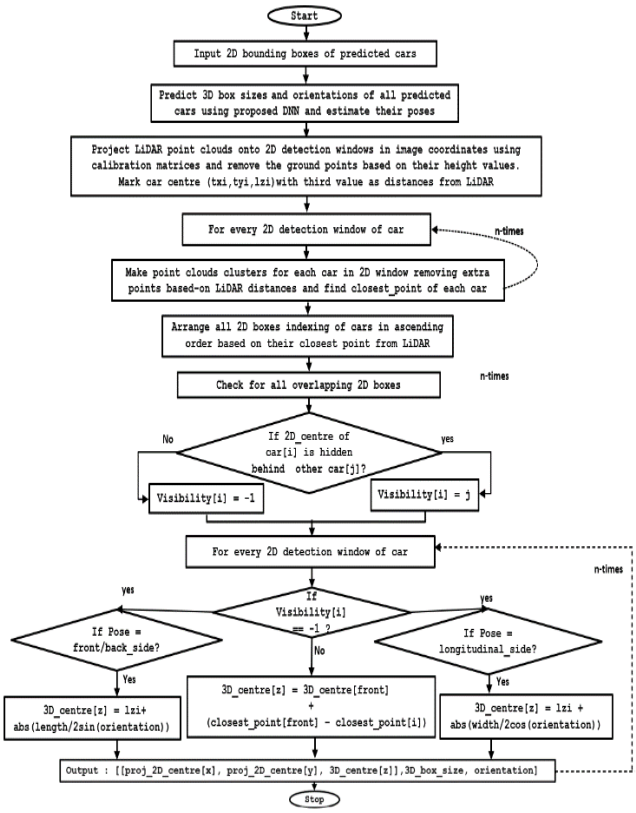


Fig. 6. The flowchart of proposed algorithm for finding the 3D Box information of visible or partly visible cars based on camera RGB images and LiDAR 3D point clouds.

To handle the occlusion problem, we primarily focus on the average distance and closest points of each car cluster from AV. Furthermore, we arrange all car indexing in ascending order of the closest points and identify the immediate front cars for all the occluded ones. From this viewpoint, we only consider occluded cars related to the ones whose centres are hidden behind others, since we cannot retrieve the direct laser distance of their centre points on the surface. To calculate the 3D centre of occluded vechiles, the gap between the closest points from the immediate front to the vehicle under consideration is utilized for relative distancing. Fig. 6 shows the flowchart of the proposed algorithm for finding the 3D box information of fully visible as well as occluded cars.

IV. Our Experiments

To test the proposed method, we take use of the benchmarks KITTI dataset [15] and Waymo dataset [33] that contain a high degree of occlusion, truncation and complex environmental conditions. For each dataset, 2,400 images were used in our experiments that were split into the ratio of 8:2:2 for training, validation, and test purposes. We retrieved 2D detection results based on our previous work [34], given 94.5%mAP@0.5IoU and 1.5% loss based on the KITTI dataset while 97.2%mAP@0.5IoU on Waymo dataset. We have trained MobileNetV2 as a feature extractor from scratch by replacing the final fully connected layers. 2D object detection results are cropped into 224×224 windows so as to train the proposed DNN based on Tesla P100-PCIE-GPU with 16.2GB memory. We investigated SGD with momentum [35] and Adam [36] gradient descent optimizers with 1.0×10^{-3} learning rate and 0.9 momentum during the experiments, which result in better performance of SGD than Adam.

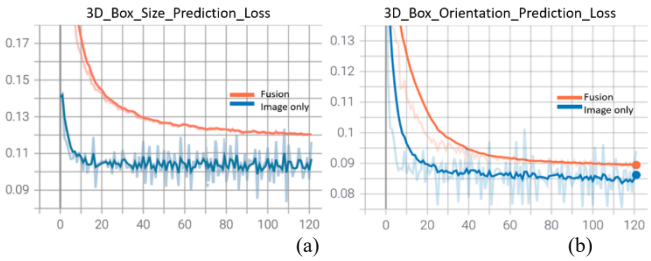
A. DNN performance

1) Result analysis based on the KITTI dataset

In order to analyse the 3D box size and orientation results of cars, we tested the proposed DNN on image datasets as well as early fused datasets (point clouds projected on images). Fig. 7(a) shows an example of LiDAR BEV point clouds of the KITTI dataset. Fig. 7(b) displays the RGB image of the same frame. On the other hand, Fig. 7(c) depicts the early-stage fusion of image and point clouds. Our experimental results based on the KITTI validation dataset are shown in Fig. 8. We see that the images are better for feature extraction using the proposed DNN, without extra computations, than the early fusion dataset. Fig. 9 represents detection results with respect to distances from AV. Based on the experiments, the neural network performance is promising in the 20~50 meters range using camera images, achieving 85.7% size accuracy and 79.7% orientation accuracy of the bounding boxes.

(a) (b) (c)

Fig. 7. Single frames in KITTI dataset (a) LiDAR point cloud (b) Image of the same scene (c) Early fusion of point clouds and image by projecting point clouds onto the image by using calibration parameters.



(a) (b)

Fig. 8. The comparison of validation loss results of the proposed DNN for early fused *vs* image only data formats based on KITTI datasets (a) dimension loss curves (b) orientation loss curves of 3D bounding boxes

2) Result analysis based on Waymo dataset

For further verification, we checked the net performance based on Waymo dataset [33] by taking into account night and rainy scenes. Fig. 10 shows the validation results obtained through start and end points of the network with the Waymo dataset. Compared to the KITTI dataset, using the Waymo dataset, we achieved 99.5% accuracy in terms of orientation as shown in Fig. 10(a); however, the prediction loss of the 3D bounding box size converged at 15.0% is shown in Fig. 10(b).

B. Estimation Results of 3D Bounding Boxes

Fig. 11 illustrates the outcomes of intermediate steps of the proposed algorithm. Fig. 11(a) shows 2D bounding boxes obtained with their confidence scores by using our previous

2D vehicle detection work [35]. Each proposal was fed into the proposed DNN net to yield the size and orientation of 3D bounding boxes around car objects. Fig.11(b) displays the projected point clouds on the image in the predicted 2D detection windows after removing ground points. Fig. 11(c) shows the 2D centre of detected cars by using small circles (i.e., p_1) with their depths estimated by using projected point clouds, whereas the outermost 3D points on the vehicle surface using 2D central y-axis are represented with big circles (e.g., p_2). After merging p_1 and p_2, we estimated 3D car centres projection on the 3D bounding box surfaces, which is further extended to the inner centre by using the orientation and dimensions values based on their pose. Furthermore, these 3D centres were converted into world coordinates by using inverse projection and inverse rotation-translation matrices. The final positions of 3D bounding boxes are presented in Fig. 11(d). Fig. 12 shows the images of detected 3D bounding boxes by using Waymo dataset based on proposed network.

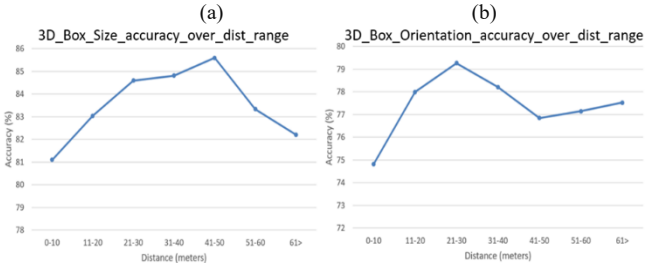


Fig. 9. The detection accuracy of the proposed DNN over distance range based on the KITTI dataset (a) prediction accuracy of 3D bounding box w.r.t distances (b) accuracy of car orientation w.r.t distances.

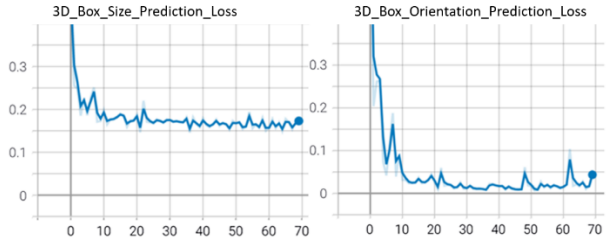


Fig. 10. The validation loss results of the proposed DNN based on the Waymo datasets (a) prediction loss curve of 3D bounding boxes (b) loss curve of car orientation prediction

Fig. 11. (a) The example of predicted 2D bounding boxes based on the KITTI dataset (b) The projected LiDAR point clouds onto 2D detection windows of the image after ground points removal (c) The small dots show the centres of 2D bounding boxes whilst the big dots depict the maximum bulged out 3D surface points across y-axis of the 2D centres. (d) Based on the estimated 3D centres, orientations and poses of 3D bounding boxes of cars

In order to analyze the proposed method based on sparse point clouds, we converted the KITTI point clouds into 32 and 16 beams based on the number of points in a 360° rotation of LiDAR, as shown in Fig. 13. Fig. 14 represents the evaluation results of 3D box centre accuracy with 64, 32, and 16 beam point clouds over distances based on the KITTI dataset. Our results show that sparse point clouds have a noteworthy performance based on our algorithm, up to 40 meters of range. This is due to the fact that the proposed solution does not rely on the density of point clouds. In fact, on the horizontal stream of points, cheap LiDAR point clouds provide information needed for our algorithm, i.e., the closest point and distance from the 2D centres based on surface value. Table II represents the overall inference time of the proposed model using different density point clouds.

Fig. 12. The test results of 3D car detection based on the Waymo dataset by using the proposed model.

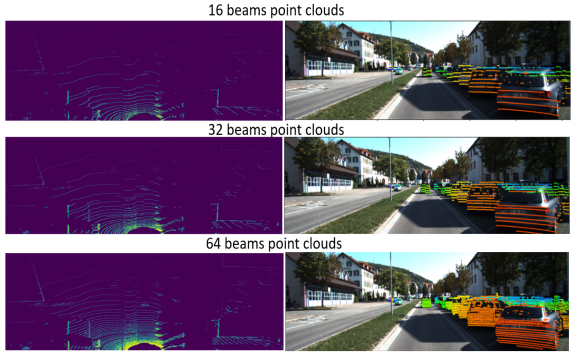


Fig. 13. KITTI point clouds converted into 64 beams, 32 beams and 16 beam LiDAR forms for model testing. Left images are the raw BEV of point clouds, the right images are projected point clouds onto image coordinates in 2D detection windows with ground points removed.

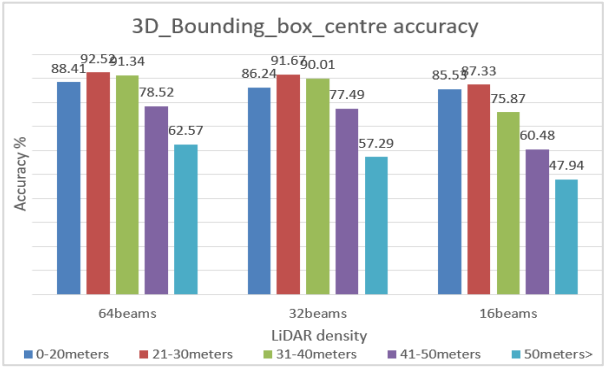


Fig. 14. The evaluations of the proposed model performance with 64, 32, and 16 beams point clouds over distance range based on the KITTI dataset.

TABLE II. THE EVALUATIONS OF THE INFERENCE SPEED USING THE PROPOSED MODEL WITH 64, 32, 16 BEAM POINT CLOUDS BASED ON THE KITTI DATASET.

Beams density	Inference time (sec)
64	.224
32	.212
16	.206

V. CONCLUSION

In this paper, we propose a novel solution for the estimation of 3D bounding boxes to detect car positions on the road in the 3D world. In the proposed solution, we leverage the mature vehicle detection in 2D to position the 3D

bounding boxes. The model extracts the 3D world coordinate of cars in 2D detection windows of the image plane using LiDAR point clouds. We firstly regressed the size and orientations of 3D bounding boxes of cars using MobileNetV2-based DNN. Secondly, LiDAR point clouds were exploited to get 3D centre coordinates of cars based on 2D centres. The performance of the proposed algorithm is evaluated over 16, 32, and 64 beams point clouds. Our results prove that the proposed method provides a cost-effective solution for 3D vehicle detection, generating desired accuracy and speed. The model performance has been verified by using the Waymo dataset.

In future, we will make the proposed solution more robust, especially for long vehicles by embedding perspective transformation. The performance of the proposed solution will be improved by using the latest low-priced solid-state LiDARs that hold sparse point clouds but can give accuracy up to a 250-meter range in fixed 120^{o} horizontal and 30^{o} vertical field of view. The use of specialized deep learning neural networks (DNNs) [37,40] has led to potentially novel architectures for future vehicle detection. Exploring the features that these DNNs are actually extracting layer by layer from the sensor information could lead to further enhancements to both sensor technology as well as faster control systems for vehicle avoidance.

REFERENCES

[1] Y. Zakaria et al., "A novel vehicle detection system," in International Conference on Computer Engineering and Systems, 2018, pp. 127–131

[2] V. Silva, J. Roche, and A. Kondoz, "Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots," Sensors, 2018, 18(8)

[3] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. Mccullough, and A. Mouzakitis, "A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications," IEEE Internet of Things Journal, 2018, 5(2), 829–846, 2018

[4] M. Podpora, G. P. Korbaś, and A. Kawala-Janik, "YUV vs RGB—choosing a color space for human-machine interaction," In FedCSIS, pp. 29-34, 2014

[5] P. Wei, L. Cagle, T. Reza, J. Ball, and J. Gafford, "LiDAR and camera detection fusion in a real-time industrial multi-sensor collision avoidance system," In Electronics, 2018, 7(6): 88

[6] Z. Wang, W. Zhan, and M. Tomizuka, "Fusing bird's eye view LIDAR point cloud and front view camera image for 3D object detection," In IEEE Intelligent Vehicles Symposium, pp. 834–839, 2018

[7] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," In International Conference on Intelligent Robots and Systems, pp. 5750–5757, 2018

[8] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," In IEEE CVPR, pp. 4490-4499, 2018

[9] C. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," In IEEE CVPR, pp. 652-660, 2017

[10] C. Qi, L. Yi, H. Su, and L. Guibas, "PointNet++: Deep hierarchical feature learning on," In International Conference of Neural Information Process Systems, 2017, 12, 5105–5114

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," In Advances in Neural Information Processing Systems, pp.91-99, 2015

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," In IEEE CVPR, 2020.

[13] W. Yan, Computational Methods for Deep Learning, Springer, 2021.

[14] E. Arnold, O. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," In IEEE Transactions on Intelligent Transportation Systems, 2019, 20(10), 3782-3795

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," International Journal of Robotics Research, 2013, 32(11), 1231-1237

[16] A. Mousavian, D. Anguelov, J. Košecká, and J. Flynn, "3D bounding box estimation using deep learning and geometry," In IEEE CVPR, pp. 7074-7082, 2017

[17] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D object proposals using stereo imagery for accurate object class detection," IEEE Transactions on PAMI, 2018, 40(5), pp. 1259–1272

[18] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D voxel patterns for object category recognition," In IEEE CVPR, pp. 1903–1911, 2015

[19] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D Lidar using fully convolutional network," In arXiv:1608.07916. 2016

[20] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "BirdNet: A 3D object detection framework from LiDAR information," In International Conference on Intelligent Transportation Systems, pp. 3517–3523, 2018

[21] M. Simon, S. Milz, K. Amende, and H. M. Gross, "Complex-YOLO: An euler-region-proposal for real-time 3D object detection on point clouds," In ECCV Workshops, pp. 197–209, 2019

[22] Y. Yan, Y. Mao, B. Li, "Second: Sparsely embedded convolutional detection." Sensors, 2018, 18(10), 3337

[23] Y. Ye, H. Chen, C. Zhang, X. Hao, Z. Zhang, "SARPNet: Shape attention regional proposal network for lidar-based 3D object detection," In Neurocomputing, 2020, pp.53-63

[24] J. Zhou, X. Tan, Z. Shao, L. Ma, "FVNet: 3D front-view proposal generation for real-time object detection from point clouds," In International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, 2019, pp. 1-8

[25] Z. Wang, K. Jia, "Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection," In International Conference on Intelligent Robots and Systems, 2019, pp. 1742-1749

[26] S. Shi, X. Wang, H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," In IEEE CVPR, pp. 770-779, 2019

[27] R. Qi, et al. "Frustum pointnets for 3D object detection from RGB-D data." In IEEE CVPR, pp. 918–27, 2018

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," In IEEE CVPR, pp. 4510–4520, 2018

[29] X. Liu, W. Yan, N. Kasabov, "Vehicle-related scene segmentation using CapsNets," IEEE IVCNZ, 2020

[30] X. Wang, W. Yan, "Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term msemory," International Journal of Neural System, 2020, 30(1): 1950027:1-1950027:12

[31] N. An, W.Yan, "Multitarget tracking using Siamese neural networks," ACM Transactions on Multimedia Computing, Communications, and Applications, 2021, 17(2):1-16

[32] E. Weisstein, Affine Transformation, Mathworld, 2004.

[33] P. Sun, K. Henrik, and D. Xerxes, "Scalability in perception for autonomous driving: Waymo open dataset," in IEEE CVPR, pp. 2443–245, 2020

[34] S. Mehtab and W. Yan, "FlexiNet: Fast and accurate vehicle detection for autonomous vehicles," In ACM ICCCV, 2021

[35] C. Pan, J. Liu, W. Yan, F. Cao, W. He, Y. Zhou, "Salient object detection based on visual perceptual saturation and two-stream hybrid networks," IEEE Transactions on Image Process, 2021, 30: 4773-4787

[36] N. Gowdra, R. Sinha, S. MacDonell, W. Yan, "Mitigating severe over-parameterization in deep convolutional neural networks through forced feature abstraction and compression with an entropy-based heuristic," Pattern Recognit, 2021, 119: 108057

[37] W. Yan. Introduction to Intelligent Surveillance Surveillance Data Capture, Transmission, and Analytics, Springer, 2019

[38] X. Wang, W. Yan, "Cross-view gait recognition through ensemble learning. Neural Comput. Appl. 2020, 32(11): 7275-7287

[39] X. Wang, J. Zhang, W. Yan, "Gait recognition using multichannel convolution neural networks," Neural Computing and Applications, 2020, 32(18): 14275-14285

[40] Y. Shen, W. Yan, "Blind spot monitoring using deep learning," IEEE IVCNZ 2018