UDC 621
DOI: 10.15587/1729-4061.2023.275155

DEVELOPMENT OF OBJECT DETECTION FROM POINT CLOUDS OF A 3D DATASET BY POINT-PILLARS NEURAL NETWORK

Omar I. Dallal Bashi
Doctor of Robotics and Automation Engineering
Department of Computer Engineering
Northern Technical University, Mosul, Iraq, 41002
Husamuldeen K. Hameed
Doctor of Electronic Engineering
Department of Telecommunications and Information
Higher Institute of Telecommunications and Postal
Karkh, Baghdad, Iraq, 64074
Yasir Mahmood Al Kubaiaisi
Doctor of Electrical and Control Engineering
Department of Sustainability Management
Dubai Academic Health Corporation
OudMetha, Dubai, United Arab Emirates, 4545
Ahmad H. Sabry
Corresponding author
Doctor of Control and Automation Engineering
Department of Computer Engineering
Al-Nahrain University
Al Jadriyah Bridge, Baghdad, Iraq, 64074
E-mail: ahs4771384@gmail.com

Deep learning algorithms are able to automatically handle point clouds over a broad range of 3D imaging implementations. They have applications in advanced driver assistance systems, perception and robot navigation, scene classification, surveillance, stereo vision, and depth estimation. According to prior studies, the detection of objects from point clouds of a 3D dataset with acceptable accuracy is still a challenging task. The Point-Pillars technique is used in this work to detect a 3D object employing 2D convolutional neural network (CNN) layers. Point-Pillars architecture includes a learnable encoder to use Point-Nets for learning a demonstration of point clouds structured with vertical columns (pillars). The Point-Pillars architecture operates a 2D CNN to decode the predictions, create network estimations, and create 3D envelop boxes for various object labels like pedestrians, trucks, and cars. This study aims to detect objects from point clouds of a 3D dataset by Point-Pillars neural network architecture that makes it possible to detect a 3D object by means of 2D convolutional neural network (CNN) layers. The method includes producing a sparse pseudo-image from a point cloud using a feature encoder, using a 2D convolution backbone to process the pseudo-image into high-level, and using detection heads to regress and detect 3D bounding boxes. This work utilizes an augmentation for ground truth data as well as additional augmentations of global data methods to include further diversity in the data training and associating packs. The obtained results demonstrated that the average orientation similarity (AOS) and average precision (AP) were 0.60989, 0.61157 for trucks, and 0.74377, 0.75569 for cars

Keywords: object detection, point clouds, point-pillars, deep learning convolutional neural network

Received date 04.01.2023
Accepted date 15.03.2023
Published date 28.04.2023

***How to Cite:** Bashi, O. I. D., Hameed, H. K., Al Kubaiaisi, Y. M., Sabry, A. H. (2023). Developing of object detection from point clouds of a 3d dataset by point-pillars neural network. Eastern-European Journal of Enterprise Technologies, 2 (9 (122)), 26–33. doi: https://doi.org/10.15587/1729-4061.2023.275155*

1. Introduction

Due to its numerous applications in fields like computer vision, autonomous driving, and robotics, point cloud learning has recently drawn more and more attention. Deep learning, a dominant AI technology, has been effectively applied to address a variety of 2D vision issues. Due to the particular difficulties in processing point clouds with deep neural networks, deep learning on point clouds is still in its infancy. Particularly in the past five years, deep learning on point clouds has been increasingly popular. Several widely accessible datasets are also free, such as the KITTI Vision Benchmark Suite, "ApolloCar3D", "Semantic3D", "ScanNet", "S3DIS", "PartNet", "ShapeNet, "ScanObjectNN", and "ModelNet" [1–6]. These datasets have further accelerated the study of deep learning on 3D point clouds, and an increasing number of approaches are being put forth to handle a range of issues relating to point cloud processing, such as 3D reconstruction, 6-DOF pose estimation [7], 3D point cloud segmentation [4], 3D point cloud registration, 3D object detection, and tracking, and 3D shape classification [8].

Urban deployment of autonomous vehicles (AVs) presents a difficult technological barrier. Among other things, AVs must be able to recognize and follow moving items like bicycles, cars, and pedestrians in real time. Autonomous vehicles rely on a variety of sensors to accomplish this, with the Lidar being arguably the most crucial. A lidar creates a sparse point cloud representation by using a laser scanner to measure the distance to the environment. Such point clouds are often interpreted as object detections by a Lidar robotics pipeline using a bottom-up channel that involves background subtraction, followed by spatiotemporal clustering and classification [9].

The Point-Pillars technique is used to detect a 3D object employing 2D convolutional neural network (CNN) layers. Point-Pillars architecture includes a learnable encoder to use Point-Nets for learning a demonstration of point clouds structured with vertical columns (pillars). The Point-Pillars architecture operates a 2D CNN to decode the predictions, create network estimations, and create 3D envelop boxes for various object labels like pedestrians, trucks, and cars.

A substantial body of literature has examined how far deep learning techniques for computer vision can be used for object detection from lidar point clouds [10–12]. This is a result of the enormous advancements in these techniques. The point cloud is a sparse representation, whereas an image is dense, and the point cloud is 3D, whilst the picture is 2D, even though there are many other parallels between the modalities. Consequently, traditional image convolutional algorithms do not easily lend themselves to object detection from point clouds.

Due to its numerous applications in fields like robotics, autonomous driving, and computer vision, point cloud learning has recently drawn more and more attention. Deep learning, a dominant artificial intelligent technology, has been effectively applied to address a variety of 2D vision issues. Due to the particular difficulties in processing point clouds with deep neural networks, deep learning on point clouds is still in its infancy. Therefore, it is necessary to use deep learning of Point-Pillars, to obtain high accuracy in the results and thus generalize interest and contribute to the development of human needs.

2. Literature review and problem statement

Point-Pillars is an encoder software suggested by the research [13] that learns to represent point clouds structured in vertical columns using Point-Nets, achieving a speed advantage of 2–4 times. Point-Pillars is a suitable codec for finding objects in point clouds, however, its priority is speed rather than good accuracy. Another traditional method for detecting 3D objects is called Point-Pillars-RCNN (PP-RCNN), which was used by researchers in the study [41]. To generate 3D proposals, the study used in the first stage a pillar network to encode the point cloud. Then, in the second stage, it used RoI network feature abstraction to lighten the proposals. But since their experiments on the KITTI standard and their style method lack the use of modern and high-resolution content, their work is not in line with the required speed and accuracy to some extent. The performance, development, and assessment of the Unmanned Aircraft (Lidar) system were the main topics of the study [51]. In their studies on woods, they emphasized the system's inexpensive cost and usage of a DJI Livox MID40 laser scanner. The following field measurements were obtained after extensive surveying of the pine site (coniferous site: R=0.96, broadleaf site: R=0.70, RMSE=1.63 m; root mean square error/RMSE=0.59 m). But, they were unable to achieve high-performance standards for their system despite their experimental methods. Using camera and Lidar fusion, the study [16] proposed a multi-adaptive technique for completing depth and converting the two-dimensional sparse depth map (Lidar) into a dense depth map. The KITTI array was utilized to align the two data plane sensors. Although they demonstrated the value and approach of multi-sensor fusion, the results lacked precision. The paper [17] used a Gaussian process in implementing its proposed algorithm to segment objects in time based on 3D point drag. They also applied two types of Gaussian operations models, to work on the limit of fragmentation that contributes to the separation of the body into several parts. Although their results showed an 11.4 % increase in tracking accuracy over segmentation accuracy, they did not address the use of deep learning for point columns and therefore their work does not match recent developments. The research [18] is concerned with forecasting 3D lengths and modeling road centerlines using a Lidar point cloud and flat road centerline data. By using a 3D vector model based on Linear Reference Systems (LRS) techniques to describe and forecast the centerlines of a road in three dimensions. Although this study indicated that the proposed 3D technique employing Lidar data was effective in collecting 3D road lengths, their plan was sluggish and out of date. The researchers in [19] discussed the utility of 3D object tracking in panoramic video and Lidar for radioactive source object attribution and improved source detection by implementing this analysis pipeline on a specially developed system consisting of a 2”4” 16-inch static NaI (Tl) detector with 64-beam Lidar and four monocular cameras. Although their results demonstrated the potential to track pedestrians and cars at the same time, their experiments were unable to provide satisfactory results in some situations, such as fog.

All this allows asserting that it is expedient to conduct a study mitigating the particular difficulties in processing point clouds with deep neural networks by using deep learning algorithms on point clouds.

3. The aim and objectives of the study

The aim of the study is to detect objects from point clouds of a 3D dataset by Point-Pillars neural architecture. This will make it possible to detect a 3D object employing 2D convolutional neural network (CNN) layers.

To achieve this aim, the following objectives are accomplished:

– to produce a point cloud with full-vision including the semantic label, cuboid label, and point clouds information, as well as the ground truth box labels and the cropping point cloud;

– to use data augmentation techniques to increase the detection accuracy while assisting in avoiding over-fitting problems during training;

– to train the point-pillars object detector using a deep learning network to regress and detect 3D bounding boxes.

4. Method and materials

4. 1. Research object and hypothesis

There are three main stages in Point-Pillars architecture:

1) producing a sparse pseudo-image from a point cloud using a feature encoder;

2) using a 2D convolution backbone to process the pseudo-image into high-level representations;

3) using detection heads to regress and detect 3D bounding boxes.

The Point-Pillars networks require two inputs; pillar features as a P-by-N-by-K and pillar indices as a P-by-2 matrix, where P represents the network pillars number, N denotes the number of points/pillar, while K represents the dimension feature as shown in Fig. 1.

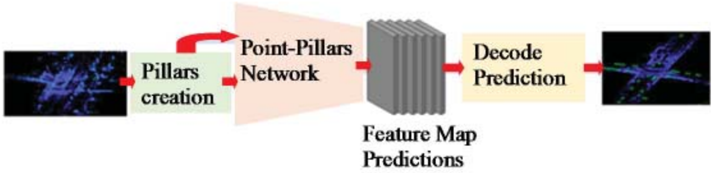


Fig. 1. The mechanism of Point-Pillars architecture

The network starts by running the feature encoder to simplify the Point-Net and includes a number of serially connected convolution layers, ReLu, and batch-normalize layers ending with a max-pool layer. Using the pillar indices, a scatter layer at the conclusion transforms the collected features into a 2D space.

After that, the architecture contains a 2D backbone CNN, which includes decoder-encoder blocks. Every encoder includes ReLu, batch-norm, and convolution layers for extracting features at various spatial resolutions. Every decoder contains ReLu, batch-norm, and transpose convolution layers. The layers of the Point-Pillars network are shown in Fig. 2.

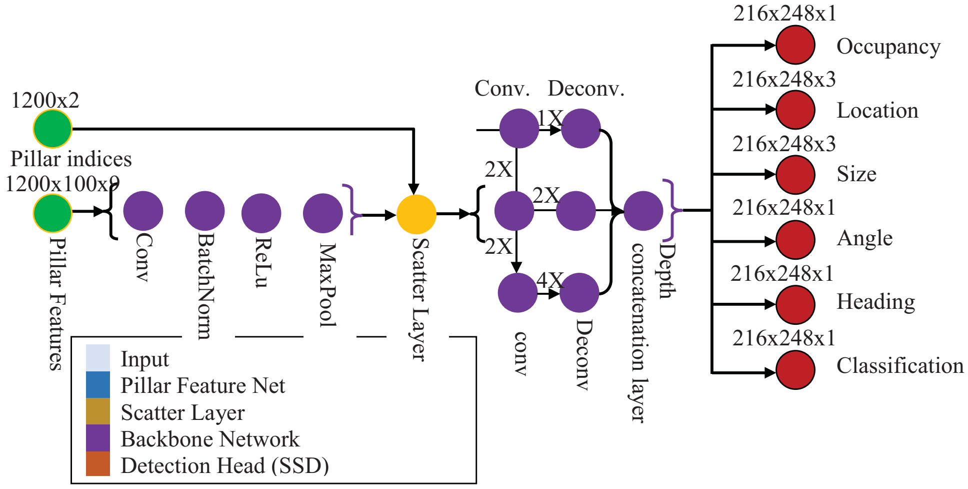


Fig. 2. The layers of the Point-Pillars network

The architecture after that concatenates the output characteristics at each decoder end to pass these characteristics using six heads detection combined by sigmoid and convolutional layers to calculate class, heading, angle, size, location, and occupancy.

This work trains Point-Pillars architecture to detect an object in a 3D environment point cloud. Data from the Lidar-based point cloud are collected using various Lidar sensors including Ouster, Pandar, and Velodyne® sensors. This can capture 3D location information regarding an object of a specific scene. Although this is helpful for many applications for augmented reality and self-directed driving, training point cloud-based data of strong detections is a challenging task due to the sensor noise, object occlusions, and sparsity of data per object. Deep CNN algorithms have been considered to solve such difficulties by training the representations of the robust features straightly using the data of point clouds. One deep CNN method to detect a 3D object is by considering Point-Pillars [13]. According to a similar network to Point-Net, the Point-Pillars architecture can extract dense, which are strong features from point clouds sparsely identified as pillars. Then the method employs a 2D deep CNN architecture through a customized detection of SSD object architecture to predict class predictions, orientations, and joint 3D bounding boxes.

The dataset is downloaded from Panda-Set [20] of 5.2 GB size, which prepared 2560 point cloud data. All the point clouds cover 360° viewing and are assigned 64-by-1856 dimensions. Each point cloud is accumulated with PCD format, while its associating ground truth information is stored with (file.mat). This file includes three classes and 3D bound pack data, which are pedestrian, truck, and car.

4. 2. Methodology steps

In this work, we pick the point clouds of the full-vision of the Panda-Set dataset and convert them into point clouds with front-view by means of standard factors that determine the network size for the pass input [21]. We choose a smaller range point cloud for the *XYZ*-axis to distinguish objects nearer to the source. This will decrease the whole network learning time. Table 1 shows the considered parameters with their symbols and values.

Fig. 3 shows the adopted steps followed to apply the proposed object detection from point clouds of a 3D dataset by Point-Pillars architecture.

The data preprocessing stage includes:

1) calculating the pseudo-image dimensions;

2) defining the parameters of the point cloud;

3) cropping the front vision point cloud from the input full-vision;

4) selecting the pack classes located inside the ROI area.

The creation of training data-store objects contains:

1) splitting the dataset into test or evaluation (30 %) and training (70 %) datasets for the network;

2) saving the training data as PCD folders;

3) creating a data-store file for loading the PCD data;

4) creating a box class data store to load the 3D bound box classes;

5) combining the 3D bounding box classes with point clouds to a single training data store.

Table 1

Parameters with their symbols and values

Parameter	Symbol and value
Down-sampling factor	dsFactor=2.0
Resolution along *Y*-axis	yStep=0.16
Resolution along *X*-axis	xStep=0.16
Maximum *Z*-axis value	zMax=5.0
Maximum *Y*-axis value	yMax=39.68
Maximum *X*-axis value	xMax=69.12
Minimum *Z*-axis value	zMin=–5.0
Minimum *Y*-axis value	yMin=–39.68
Minimum *X*-axis value	xMin=0.0

We apply the function (*sample_Lidar_Data*) for sampling the 3D bound packs of the training dataset and their associated point clouds and apply the function (*pc_Bbox_*

Oversample) for arbitrarily adding a constant amount of truck and car label objects for all point clouds. Additionally, we use the function (transform) for applying the custom data and ground truth augmentations to the training dataset. The application of the above augmentation techniques for the training data for all point clouds includes:

– random translation for the *x*-, *y*-, and *z*-axis with (0.2, 0.22, and 0.1) meters respectively;

– *z*-axis random rotation in the range [–pi/4, pi/4];

– random scaling by 5 percent;

– random flipping along the *x*-axis.

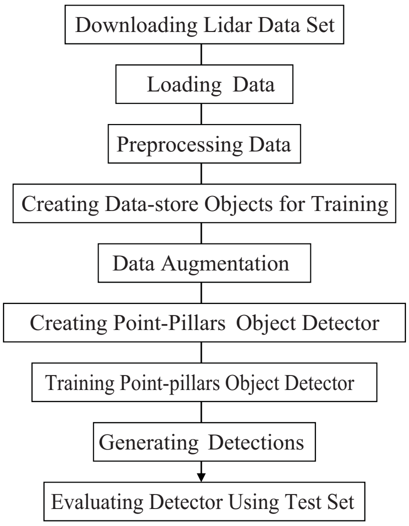

Downloading Lidar Data Set
Loading Data
Preprocessing Data
Creating Data-store Objects for Training
Data Augmentation
Creating Point-Pillars Object Detector
Training Point-pillars Object Detector
Generating Detections
Evaluating Detector Using Test Set

Fig. 3. The methodology steps of the proposed object detection from point clouds of a 3D dataset by Point-Pillars architecture

The training of the Point-Pillars object detector is conducted using a deep learning network with the training options listed in Table 2.

We apply a function called (*train_Point_Pillars_Object_Detector*) for training the Point-Pillars object detector. For creating detection objects, we use the trained architecture for the test data by:

1) reading a test data point cloud;

2) running the detection network on the tested point cloud data to obtain the estimated confidence scores and bound packs.

Table 2

The training options for the deep learning network

Description	Value
Max. Epochs	60
Min. Batch Size	3
Gradient Decay Factor	0.9
Squared Gradient Decay Factor	0.999
Learning Rate Schedule	piecewise
Initial Learning Rate	0.0002
Learning Rate Drop Period	15
Learning Rate Drop Factor	0.8
Execution Environment	Execution Environment
Dispatch In Background	Dispatch In Background
Batch Normalization Statistics	moving
Reset Input Normalization	false
Checkpoint Path	Temp dir

5. Results of the object detection from point clouds of a 3D dataset

5. 1. The point cloud full-vision

The point cloud full-vision is shown in Fig. 4, while Fig. 5 shows the ground truth box labels and the cropping point cloud according to a MATLAB-based helper function.

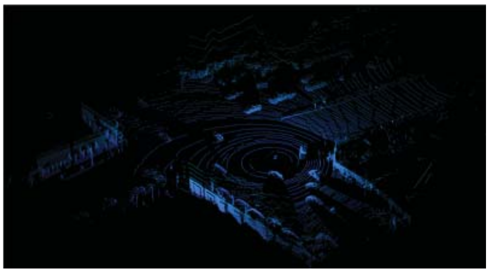


Fig. 4. Point cloud full-vision

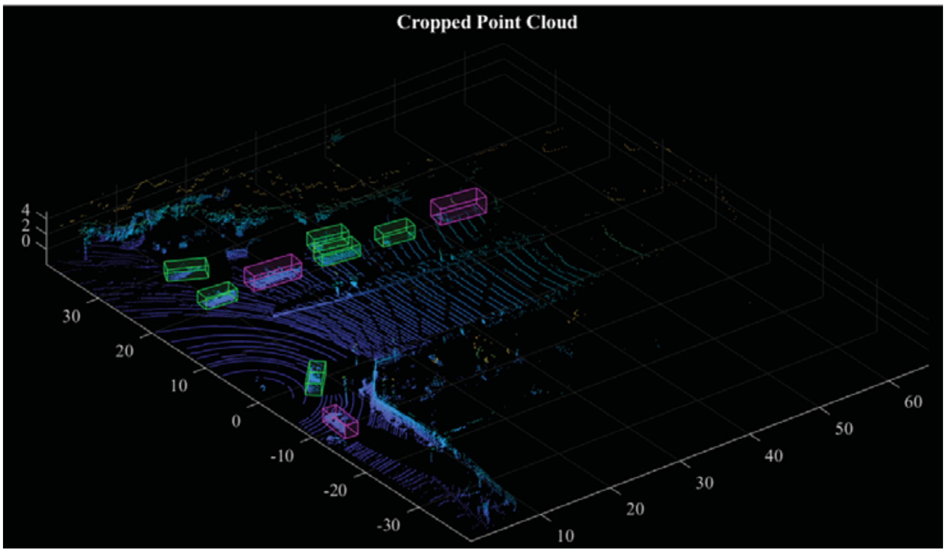


Fig. 5. Ground truth box classes and the cropping point cloud

The downloaded file includes folders with semantic Labels, Cuboids, and Lidar that have the semantic label, cuboid label, and point clouds information, respectively.

5. 2. Applying data augmentation technique

Fig. 6 reads and displays a prior augmentation point cloud.

This technique increases the detection accuracy while assisting in avoiding over-fitting problems during training. Fig. 7 shows the ground truth augmented boxes with the augmented point clouds.

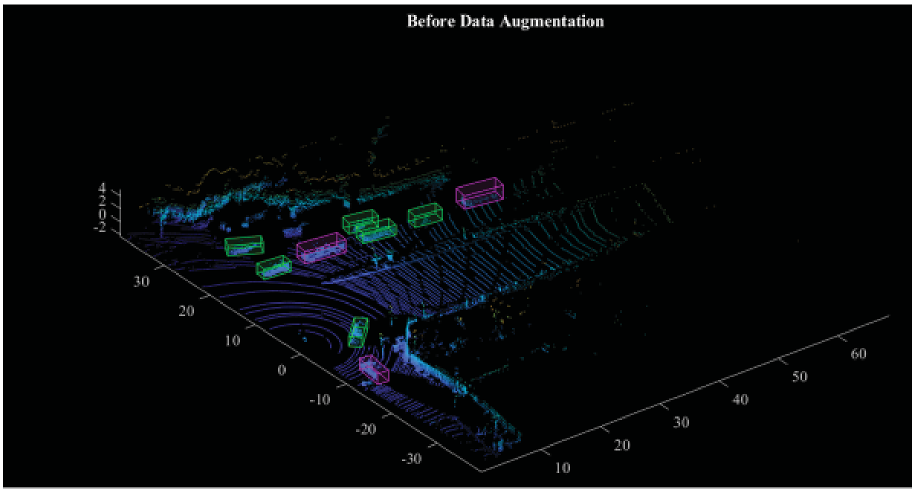


Fig. 6. A prior augmentation point cloud

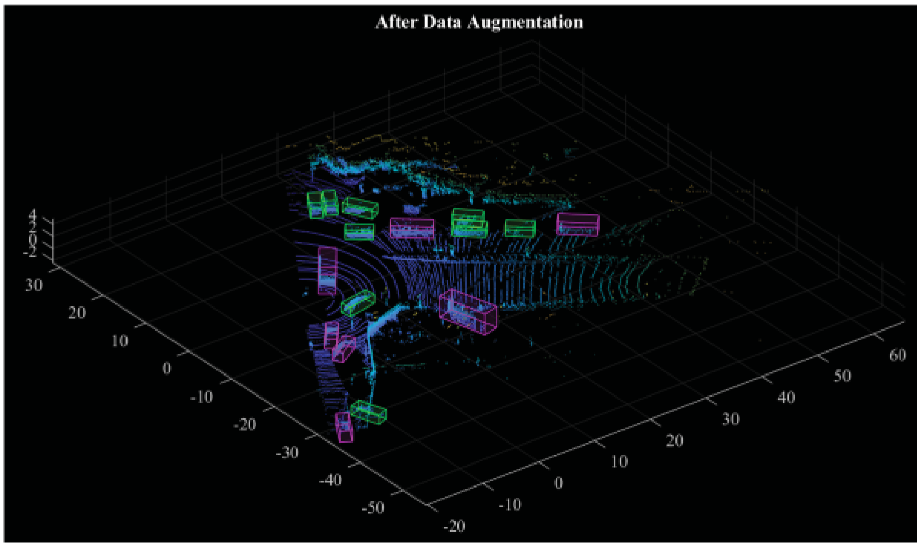


Fig. 7. The ground truth augmented boxes with the augmented point cloud

The application of the function of the Point-Pillars object detector necessitates specifying the number of inputs to parameterize the Point-Pillars architecture:

– amount of points/pillar (N=100);

– prominent pillars number is (P=12.000);

– voxel size;

– point cloud range;

– anchor boxes;

– class names.

5. 3. Training Point-Pillars object detector

The regress and detect 3D bounding boxes and point cloud, on 0.5 assurance threshold of the detections, is shown in Fig. 8.

The evaluation of the trained object point cloud detection applying the test dataset is implemented on a large set to measure the performance. This is done by generating rotating rectangles in the cuboid classes and setting the threshold quantities by 0.5 for the number of PositiveIoUThreshold and 0.25 for the confidence threshold. Then, we convert the bound packs to rotating rectangles format and compute the assessment metrics including the average orientation similarity (AOS) and average precision (AP), which is a common metric in assessing the accuracy of object detections. Furthermore, AOS was used to evaluate the performance of mutual object detection to estimate its 3D orientation and it has been defined in [1] by:

$$AOS=\frac{1}{11}\sum_{r\in\{0,0.1,\ldots,1\}}\max s(\tilde{r}),$$

$$\tilde{r}:\tilde{r}\geq r, \qquad (1)$$

where s∈[0,1], r=TP/(TP+FN) represents the recall of the object Pascal detection, which indicates that it is correct detection if the detecting 2D bounding box overlaps with 50 % minimum with ground truth bound boxes. TP is the true positive, TN is negative, and FN is the false negative. AP can be defined by:

$$\mathrm{AP}=\frac{1}{11}\times$$

$$\times\begin{pmatrix}\mathrm{AP}_r(0)+\\+\mathrm{AP}_r(0.1)+\ldots+\\+\mathrm{AP}_r(1.0)\end{pmatrix}. \qquad (2)$$

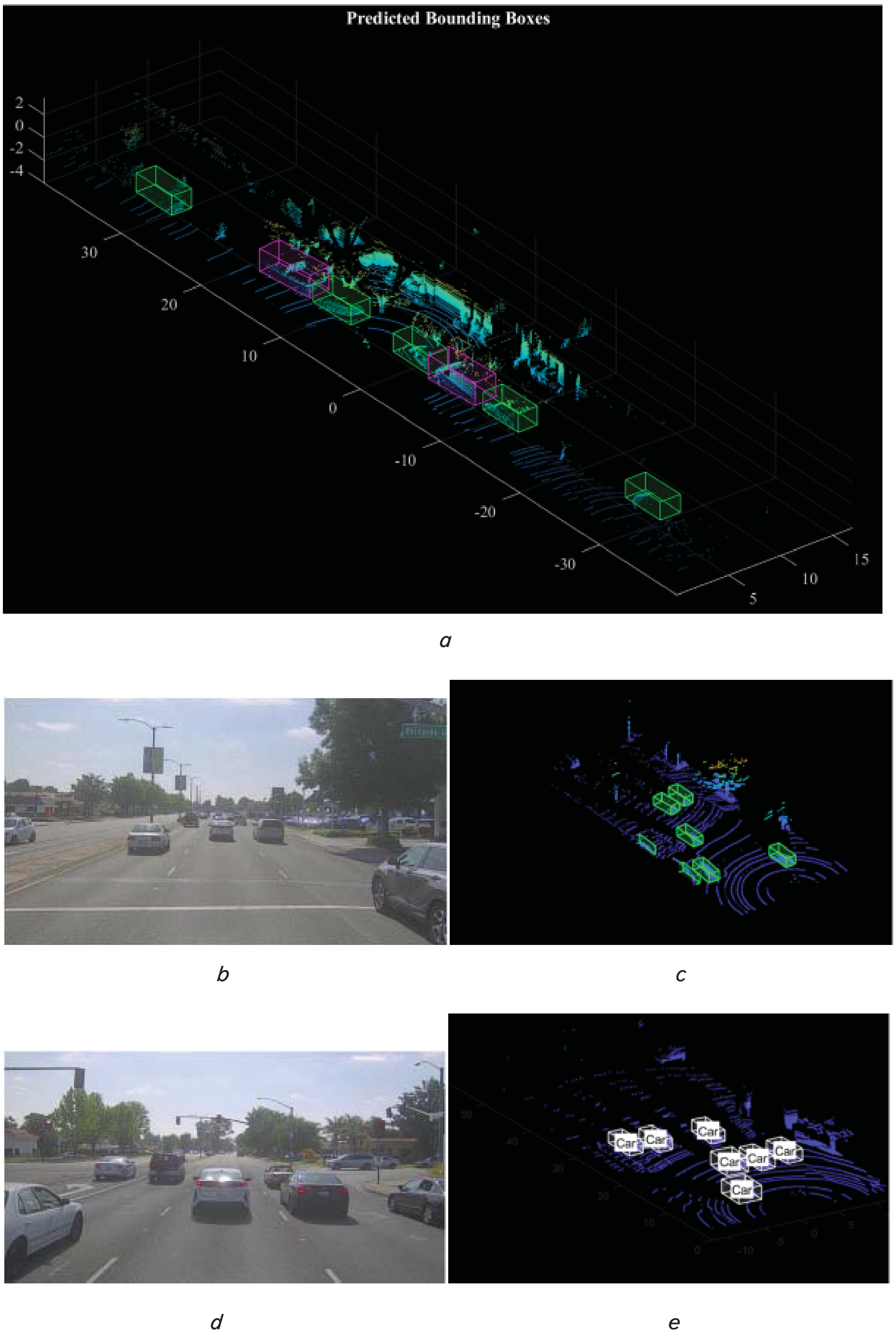
We can assume the last terms are zero when AP becomes very small. i. e., until the recall rate hits 100 %, we may not necessarily make predictions. Therefore, by comparing with two related studies, the AOS and AP metrics for the detection are listed in Table 3.

Table 3

The average orientation similarity (AOS) and average precision (AP) metrics for the car and truck objects

Reference	Dataset	Object type	AOS	AP
Proposed	Panda-Set	Truck	0.60989	0.61157
		Car	0.74377	0.75569
[22]	ScanNet[5] and SUN RGB-D [22]	Furniture pieces	0.4628	0.4701
[21]	KITTI [23]	Truck	0.528	0.544
		Car	0.7212	0.7256

The results showed that the average orientation similarity (AOS) and average precision (AP) for trucks and autos, respectively, were 0.60989 and 0.61157.



a

b

c

d

e

Fig. 8. The predictions on the point cloud: *a* – bounding boxes prediction; *b* – first image of the front camera; *c* – car detections on the point cloud; *d* – 399th image from front view camera feed; *e* – lidar feed with the track

6. Discussion of the object detection from point clouds of a 3D dataset

This work utilizes an augmentation for ground truth data as well as additional augmentations of global data methods to include further diversity in the data training and associating packs. Data augmentation techniques increase detection accuracy while assisting in avoiding over-fitting problems during training. The application of the data augmentation technique increases the detection accuracy while assisting in avoiding over-fitting problems during training as shown in the results of Fig. 6. After data augmentation, the ground truth augmented boxes with the augmented point cloud are clear in Fig. 7. The training of point-pillars object detector results in regression and detection of 3D bounding boxes and point cloud with (0.5) thresholding assurance as shown in Fig. 9. Such study can contribute to advanced driver assistance systems, perception and robot navigation, scene classification, surveillance, stereo vision, and depth estimation.

To evaluate how well the proposed approach performs on 3D point pillar data, multiple benchmark models, including the dataset (ScanNet and SUN RGB-D) by the study [24], and KITTI dataset by [21] are compared in this study. The comparison showed that the developed approach performs substantially better than other models, achieving average orientation similarity (AOS) and average precision (AP) for tracking the car and truck objects than other algorithms as shown in Table 3.

The limitation of this work is that the simulations have been applied on 2.560 point cloud data, covering 360° viewing, and are assigned 64-by-1.856 dimensions with PCD format. Future work may expand the applications of this approach to include various types of datasets.

7. Conclusions

This study developed an approach to detect objects from point clouds of a 3D dataset by Point-Pillars neural architecture, which makes it possible to detect a 3D object employing 2D convolutional neural network (CNN) layers:

1. Point cloud with full-vision including the semantic label, cuboid label, and point clouds information has been produced with the ground truth box labels and the cropping point cloud.

2. Data augmentation has been used to increase the detection accuracy while assisting in avoiding over-fitting problems during training to parameterize the point-pillars

architecture with N=100 points/pillar and P=12.000 prominent pillars.

3. The presented work trains the point-pillars object detector using a deep learning network to regress and detect 3D bounding boxes. The obtained results demonstrated that the average orientation similarity (AOS) and average precision (AP) were 0.60989, 0.61157 for trucks, and 0.74377, 0.75569 for cars.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

The manuscript has associated data in a data repository.

Acknowledgments

The authors would like to express their deepest gratitude to the Northern Technical University, Mosul-Iraq for their support to complete this research.

References

1. Geiger, A., Lenz, P., Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. 2012 IEEE Conference on Computer Vision and Pattern Recognition. doi: https://doi.org/10.1109/cvpr.2012.6248074
2. Fuseya, Y., Kariyado, T., Ogata, M. (2009). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite Andreas. Journal of the Physical Society of Japan.
3. Song, X., Wang, P., Zhou, D., Zhu, R., Guan, C., Dai, Y. et al. (2019). ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2019.00560
4. Liu, H., Guo, Y., Ma, Y., Lei, Y., Wen, G. (2021). Semantic Context Encoding for Accurate 3D Point Cloud Segmentation. IEEE Transactions on Multimedia, 23, 2045–2055. doi: https://doi.org/10.1109/tmm.2020.3007331
5. Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., Niessner, M. (2017). ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2017.261
6. Rossi, D., Veglia, P., Sammarco, M., Larroca, F. (2012). ModelNet-TE: An emulation tool for the study of P2P and traffic engineering interaction dynamics. Peer-to-Peer Networking and Applications, 6 (2), 194–212. doi: https://doi.org/10.1007/s12083-012-0134-x
7. Sabry, A. H., Nordin, F. H., Sabry, A. H., Abidin Ab Kadir, M. Z. (2020). Fault Detection and Diagnosis of Industrial Robot Based on Power Consumption Modeling. IEEE Transactions on Industrial Electronics, 67 (9), 7929–7940. doi: https://doi.org/10.1109/tie.2019.2931511
8. Nazir, D., Afzal, M. Z., Pagani, A., Liwicki, M., Stricker, D. (2021). Contrastive Learning for 3D Point Clouds Classification and Shape Completion. Sensors, 21 (21), 7392. doi: https://doi.org/10.3390/s21217392
9. Hamza, E., Aziez, S., Hummadi, F., Sabry, A. (2022). Classifying wireless signal modulation sorting using convolutional neural network. Eastern-European Journal of Enterprise Technologies, 6 (9 (120)), 70–79. doi: https://doi.org/10.15587/1729-4061.2022.266801
10. Fernandes, D., Silva, A., Névoa, R., Simões, C., Gonzalez, D., Guevara, M. et al. (2021). Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. Information Fusion, 68, 161–191. doi: https://doi.org/10.1016/j.inffus.2020.11.002
11. Zhou, Z., Gong, J. (2018). Automated residential building detection from airborne LiDAR data with deep neural networks. Advanced Engineering Informatics, 36, 229–241. doi: https://doi.org/10.1016/j.aei.2018.04.002
12. Qing, L., Yang, K., Tan, W., Li, J. (2020). Automated Detection of Manhole Covers in MLS Point Clouds Using a Deep Learning Approach. IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium. doi: https://doi.org/10.1109/igarss39084.2020.9324137
13. Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O. (2019). PointPillars: Fast Encoders for Object Detection From Point Clouds. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2019.01298
14. Tu, J., Wang, P., Liu, F. (2021). PP-RCNN: Point-Pillars Feature Set Abstraction for 3D Real-time Object Detection. 2021 International Joint Conference on Neural Networks (IJCNN). doi: https://doi.org/10.1109/ijcnn52387.2021.9534098
15. Hu, T., Sun, X., Su, Y., Guan, H., Sun, Q., Kelly, M., Guo, Q. (2020). Development and Performance Evaluation of a Very Low-Cost UAV-Lidar System for Forestry Applications. Remote Sensing, 13 (1), 77. doi: https://doi.org/10.3390/rs13010077
16. Guan, L., Chen, Y., Wang, G., Lei, X. (2020). Real-Time Vehicle Detection Framework Based on the Fusion of LiDAR and Camera. Electronics, 9 (3), 451. doi: https://doi.org/10.3390/electronics9030451

17. Shin, M.-O., Oh, G.-M., Kim, S.-W., Seo, S.-W. (2017). Real-Time and Accurate Segmentation of 3-D Point Clouds Based on Gaussian Process Regression. IEEE Transactions on Intelligent Transportation Systems, 18 (12), 3363–3377. doi: https://doi.org/10.1109/tits.2017.2685523
18. Cai, H., Rasdorf, W. (2008). Modeling Road Centerlines and Predicting Lengths in 3-D Using LIDAR Point Cloud and Planimetric Road Centerline Data. Computer-Aided Civil and Infrastructure Engineering, 23 (3), 157–173. doi: https://doi.org/10.1111/j.1467-8667.2008.00518.x
19. Marshall, M. R., Hellfeld, D., Joshi, T. H. Y., Salathe, M., Bandstra, M. S., Bilton, K. J. et al. (2021). 3-D Object Tracking in Panoramic Video and LiDAR for Radiological Source–Object Attribution and Improved Source Detection. IEEE Transactions on Nuclear Science, 68 (2), 189–202. doi: https://doi.org/10.1109/tns.2020.3047646
20. Xiao, P., Shao, Z., Hao, S., Zhang, Z., Chai, X., Jiao, J. et al. (2021). PandaSet: Advanced Sensor Suite Dataset for Autonomous Driving. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). doi: https://doi.org/10.1109/itsc48978.2021.9565009
21. Shi, W., Rajkumar, R. (2020). Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr42600.2020.00178
22. Song, S., Lichtenberg, S. P., Xiao, J. (2015). SUN RGB-D: A RGB-D scene understanding benchmark suite. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2015.7298655
23. Geiger, A., Lenz, P., Stiller, C., Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. The International Journal of Robotics Research, 32 (11), 1231–1237. doi: https://doi.org/10.1177/0278364913491297
24. Qi, C. R., Litany, O., He, K., Guibas, L. (2019). Deep Hough Voting for 3D Object Detection in Point Clouds. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). doi: https://doi.org/10.1109/iccv.2019.00937