Improved Point-Voxel Region Convolutional Neural Network: 3D Object Detectors for Autonomous Driving

Yujie Li, Shuo Yang, Yuchao Zheng, Huimin Lu

Abstract— **Recently, 3D object detection based on deep learning has achieved impressive performance in complex indoor and outdoor scenes. Among the methods, the two-stage detection method performs the best; however, this method still needs improved accuracy and efficiency, especially for small size objects or autonomous driving scenes. In this paper, we propose an improved 3D object detection method based on a two-stage detector called the Improved Point-Voxel Region Convolutional Neural Network (IPV-RCNN). Our proposed method contains online training for data augmentation, upsampling convolution and k-means clustering for the bounding box to achieve 3D detection tasks from raw point clouds. The evaluation results on the KITTI 3D dataset show that the IPV-RCNN achieved a 96% mAP, which is 3% more accurate than the state-of-the-art detectors.**

Keywords- 3D object detection; region proposal method; point cloud data processing

1. Introduction

Presently, with the universal applications of three-dimensional (3D) LiDAR cameras, 3D object detection based on point cloud data processing has become a fundamental technique and research focus in the robotics and autonomous driving fields. Compared to two-dimensional (2D) images from regular cameras, point cloud data include the depth and geometric space information of the object, which can not only be used to detect the category and location but also acquire the 3D space information.

In practice [1], point cloud data processing has challenges and difficulties, such as the sparsity and disorder of point clouds. The frameworks of traditional detectors include spatiotemporal clustering and classification [2, 3]. As the data volume and scene complexity increase, these detectors cannot determine the location and classification accurately. Furthermore, traditional detectors usually can process limited amounts of point cloud data. It is a crucial and challenging problem that these detectors utilize large amounts of point cloud data to detect 3D objects directly.

With the breakthrough in deep learning-based 2D detection algorithms, many 2D image-based detectors have been applied in point cloud data processing for 3D object detection. We can divide the different processing methods of point cloud data into 3 categories: view-based methods [4, 5], voxel-based methods [6] and point-based methods [7].

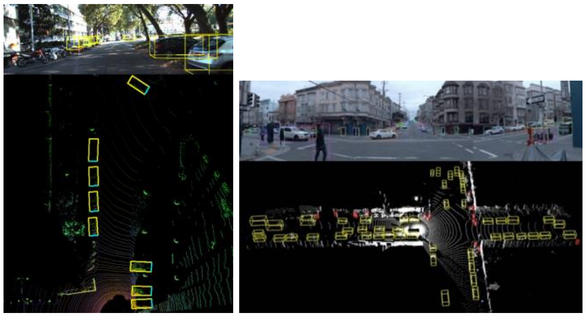
View-based methods need to convert point cloud data to 2D images for further processing. Thus, view-based detectors are not optimal because too much feature information is filtered out in the point cloud data sampling process [8].

Voxel-based methods, which can use 3D convolutional neural networks to extract objects' features, usually transform point cloud data to 3D voxels. Generally, these detectors lead to inevitable information loss and influence the localization accuracy owing to the coarse sampling for point cloud data.

Point-based methods have been used for 3D object detection in recent years. These methods directly extract the features of an object from the raw point cloud data using a learning-based method. Due to the large amount of data, these methods need to make numerous calculations. Thus, a point-based method cannot be applied using a lightweight calculation device, such as a CPU or embedded device.

Inspired by the PV-RCNN [9], a previous work proposes a unified framework that combines voxel-based methods with point-based methods and uses voxelization processing for point cloud data with a 3D convolution to extract the feature map of an object. In addition, it uses a raw point cloud sampling method to acquire the supplemental information to compensate for the information loss during voxelization and convolution. Last, it uses the region interest feature abstraction method to complete feature fusion. This method effectively fuses the convolutional feature and point feature to improve the representational ability of the model and further acquire good detection precision.

Yujie Li is with the School of Information Engineering, Yangzhou University, China, yzyjli@yzu.edu.cn
Shuo Yang is with the School of Engineering, Kyushu Institute of Technology, Japan, dlmz1shuoy@gmail.com (Corresponding Author: Shuo Yang)
Yuchao Zheng is with the School of Engineering, Kyushu Institute of Technology, Japan, zyc4718@outlook.com
Huimin Lu is with the School of Data Science and Software Engineering, Qingdao University, China, bolandi@m.ieice.org



(a) (b)

Fig 1. The problem analysis for processing a point cloud. (a)The bird's-eye view of a point cloud will affect the prediction of the z-axis distance. (b) There is a crucial problem of how to form a multiclassification detector with higher precision.

However, it only uses five 3D convolutional layers to generate the region proposals, leading to an increase in background interference. Furthermore, it can reduce the localization precision, since this method manually sets the sizes and number of anchor boxes.

In this paper, our work focuses on improving the detection precision rate and reducing the computational complexity of 3D object detection for autonomous driving. Moreover, we aim to design an efficient training method to compensate for the limited amount of data. The contributions of this paper can be summarized as follows:

1) A training method based on online data augmentation, including geometric transformation and adding noise. This method can also be extended to other 3D object detection methods.

2) A brand-new backbone with a 3D sparse convolution that adds the upsampling convolution. This backbone can obtain a larger scale feature map to acquire the fine-grained features. In addition, we increase the depth of the network to reduce the noise.

3) The use of a data-based method to set the sizes and number of anchor boxes instead of manually setting them. We use k-means clustering to calculate the sizes and number of objects from the dataset. The experiments show that this data-based method can decrease the number of invalid region proposals and increase the training efficiency.

The remainder of this paper is structured as follows. For Section 2, we introduce and analyze related work. In Section 3, we give the method analysis and problem for PV-RCNN on LiDAR data. We present the improved structure in Section 4. Then, Section 5 shows the experimental results on the KITTI 3D datasets. Finally, Section 6 gives the conclusion for the task of 3D detection.

2. RELATED WORKS

View-based methods for point cloud data. This method uses 3D-2D transformation processing to conduct object detection. This method usually projects the point cloud to the bird's-eye view (BEV) or front view. Then, it uses the 2D feature map to predict the 3D bounding box and classification for the object [10]. For example, [11] proposed a fusion method for the front view and bird's-eye view to generate the region proposals. However, it is difficult to realize an object's geometric depth using 2D images, which need extensive feature information. Thus, these methods are rarely applied to point cloud data to determine the location and classification of a 3D object.

Voxel-based methods for point cloud data. This method proposed the VoxelNet to process point cloud data for the detection task. Then, the 3D sparse convolutional neural networks can be applied to point clouds with voxelization to extract the feature maps with designed a 3D proposal generation network [12]. The research constructs multilayer 3D convolution networks and predicts the object locations and classifications using the candidate region proposals. However, this method has the problem that the voxelization loses substantial feature point information for the point cloud. Losing the key information will considerably influence the result of 3D bounding box estimation. Thus, most of these methods [13] have to refine the voxelization to improve the representational ability of the 3D convolution, but this also brings a considerable computation cost and reduces the computational complexity.

Point-based methods for point cloud data. Recently, most methods have proposed to learn the feature information from the point cloud data directly [14]. The innovation of these methods is that they use neural networks to learn the relationship feature and achieve representation learning from the point cloud data [15]. For example, the Point-RCNN, which uses the Hough voting strategy to achieve better feature grouping, only predicts the 3D bounding box from the point cloud feature. Presently, studies have designed different feature extraction networks to acquire the best performance for learning point clouds. The PointNet effectively solves the point cloud feature extraction problem, but it still suffers from information loss for the local point cloud. In addition, the PointNet has high computing costs and a slower calculation speed.

3. PV-RCNN METHOD FOR OBJECT DETECTION

For 3D object detection, an essential representational method is to combine convolutional features with point features. The PV-RCNN is a unified pipeline that uses the 3D sparse convolution to generate the region proposals, and it uses a region-of-interest layer to fuse the key point and convolution features. In this section, we summarize the characteristics of the PV-RCNN and illustrate the problems with the detection precision.

The Characteristics of the PV-RCNN

The PV-RCNN comprises three parts: First, the 3D convolution is used to generate the region proposals for the voxelization of the point cloud data. Second, a keypoint sampling model is used to extract the features from point cloud data. Third, a region-of-interest pooling module is used to fuse the keypoint features in the 3D region proposals. This method uses the fusion of 3D region proposals to predict the locations and categories of objects. The details of this method are shown as follows:

(1) ***Region proposal generation.*** According to the voxelization data of a point cloud, this method uses 5 3D convolutional layers to extract the feature maps of an object. Then, this method generates the 3D region proposals using the threshold of the last layer of feature maps. A two-stage detector calls the region proposal generation anchor box generation. An anchor-based method can set the sizes and quantity of the anchor boxes to reduce the training loss. The proposed anchor-based approach has the advantage that it has higher recall performance.

Furthermore, the bird's-eye view of the feature map helps to distinguish the foreground and background of region proposals. However, the sparsity of the voxelization and downsampling convolution still lead to the loss of numerous features for the 3D object. The 5 convolutional layers have difficulty computing the set abstraction features and introduce background noises.

(2) ***Voxel to keypoint scene encoding.*** This method fuses the features of keypoint sampling in the region proposals. First, the method uses the furthest point sampling method to sample the object's keypoints in the raw point cloud. Second, the method uses the voxel set abstraction module to encode the features of the keypoints from the different layers on the corresponding target area. Finally, according to the region proposals, the method calculates the weights for the keypoints and exports the features of the keypoints. Thus, the method can use the features of keypoints to complement the voxel loss. However, the method requires some computing time to conduct point sampling and feature extraction.

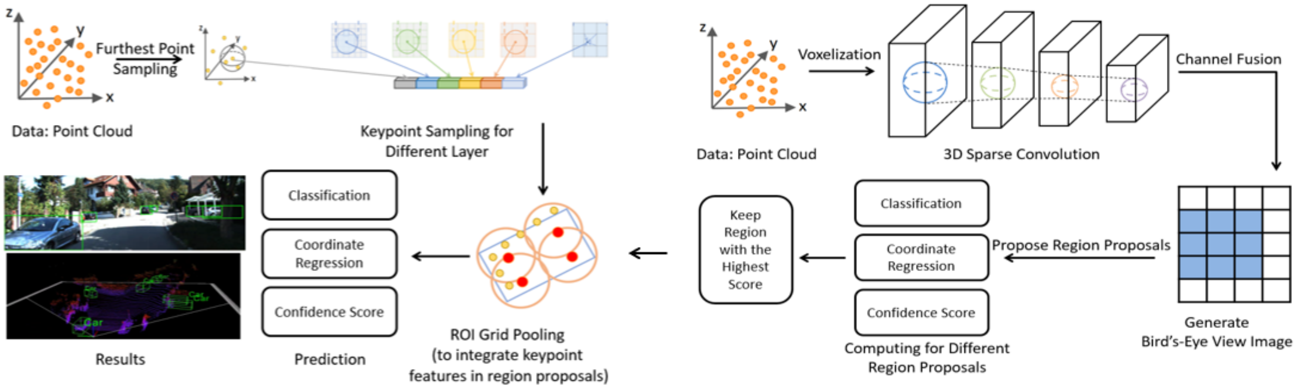


Fig 2. The pipeline analysis for the structure of the PV-RCNN. Data sampling is divided two parts. The PV-RCNN uses the voxelization and CNN to extract the feature map and generate the region proposals. In addition, it uses keypoint sampling to extract the keypoint features in the raw point cloud. In the ROI grid pooling module, the PV-RCNN fuses the keypoint features with the region proposals to further predict the object locations and classes.

(3) ***Region-of-interest pooling module.*** This module is designed to aggregate the keypoint features to the 3D region proposals with multiple receptive fields. In addition, the module is able to capture a large amount of contextual information for the object and integrate feature maps with multiple scales. Then, the module can use the output feature map to predict the 3D bounding box and categories. Moreover, the region proposals with fine-grained features contribute to the 3D bounding box regression. According to the evaluation of the KITTI 3D object detection dataset [16], this method acquires better detection precision than the traditional two-stage 3D detection methods.

At present, the PV-RCNN has excellent performance on the KITTI 3D object dataset for autonomous driving. However, the detection precision of this method still needs to be improved. The problems for this method are given as follows.

The Problems of the PV-RCNN

Because we desire to achieve a more precise 3D detector for autonomous driving, three aspects affect the detection precision of the PV-RCNN method.

(1) The PV-RCNN has a lower training efficiency and generalization ability, especially for small size objects. This method has some problems in training strategies, such as lower training efficiency and data utilization. For many robotics and autonomous driving applications, the robustness and generalization of detection methods still need to be improved. Consequently, we need to consider how to design a good training strategy for detection methods.

(2) For the detection task, most methods focus on locating the bounding box to improve the location accuracy. However, due to the classification error, the recall rate is reduced at the expense of the total detection precision. Moreover, it is difficult to use the fine-grained features from convolution to decrease the classification loss because of the downsampling backbone and the voxelization processing for point cloud data, thereby losing some fine-grained features. Thus, we need to consider how to balance of the bounding box location and classification.

(3) Most methods have lower detection accuracy for small size objects. In analyzing the backbone of the PV-RCNN, we find that the convolution network of the downsampling makes it difficult to extract great features for small size objects. Additionally, the shallower 3D convolution layer introduces more background noise information. Thus, we need to improve the representational ability of feature extraction networks.

4. IMPROVED POINT-VOXEL REGION CONVOLUTIONAL NEURAL NETWORK

In this paper, we propose the Improved Point-Voxel Region Convolutional Neural Network (IPV-RCNN), which is an anchor-based detection framework that is aimed at higher accuracy for point cloud data, for 3D object detection. In the advanced method, we propose a novel structure to improve the detection performance and enhance the training method to improve efficiency. We propose three improvement methods for solving the three subproblems in Section 3.

First, according to the analysis of the training method of data, we propose a modified training method using data augmentation to improve training efficiency and detection precision for 3D objects. Our method is an online data processing method that can achieve end-to-end training. Compared to offline data augmentation, online data augmentation will further improve the training efficiency and achieve end-to-end training.

Second, we propose the upsampling convolution based on the feature fusion of the cross-layer on the backbone of feature extraction networks. Through upsampling convolution, we can acquire more fine-grained features for small size objects. In addition, we use the region proposal network to generate higher quality bounding boxes on large size feature maps. We increase the depth of networks, which can decrease the image noise and classification loss.

Third, we use an anchor-based method [17] to predict the 3D bounding boxes and classes, and the sizes and quantities of anchor boxes play a vital role in the detectors used for object location. Most of the typical detectors use manually set the sizes and numbers of anchor boxes. In this paper, we use k-means clustering to determine the sizes and numbers of anchor boxes.

Fig 3 shows the pipeline of the proposed method. The framework of our method includes four parts: downsampling convolution, upsampling convolution, the detection head and the feature supplement model. Then, we mainly introduce the improved model in the detection framework, which includes data augmentation, upsampling convolution and the detection head.

Data Augmentation

In the training policy for the detector, data augmentation is an effective method for improving the classification and location precision. For 2D object detection tasks, data augmentation is relatively mature, and the training method innovations are growing for 2D images. For 2D object detection, the training efficiency and model quality can be improved by using color transformation, geometric transformation and noise simulation based on data augmentation. The online or offline data augmentation method has been proven to be the essential trick for training the backbone of the detection algorithms.

Compared with 2D image data augmentation, using data augmentation training for the point cloud data of 3D object detection has certain technological complexity. The reasons are as follows: First, the data format and characteristics, such as the spatial structure and color features, of the point clouds for 3D images are significantly different from those of 2D images. Second, the sparsity and disorder of point clouds influence the use the data augmentation method since they may change the locations of point clouds and cause object location errors. Third, there is no foolproof method to visualize a point cloud, particularly when visualizing the change of the z-axis of a point cloud.

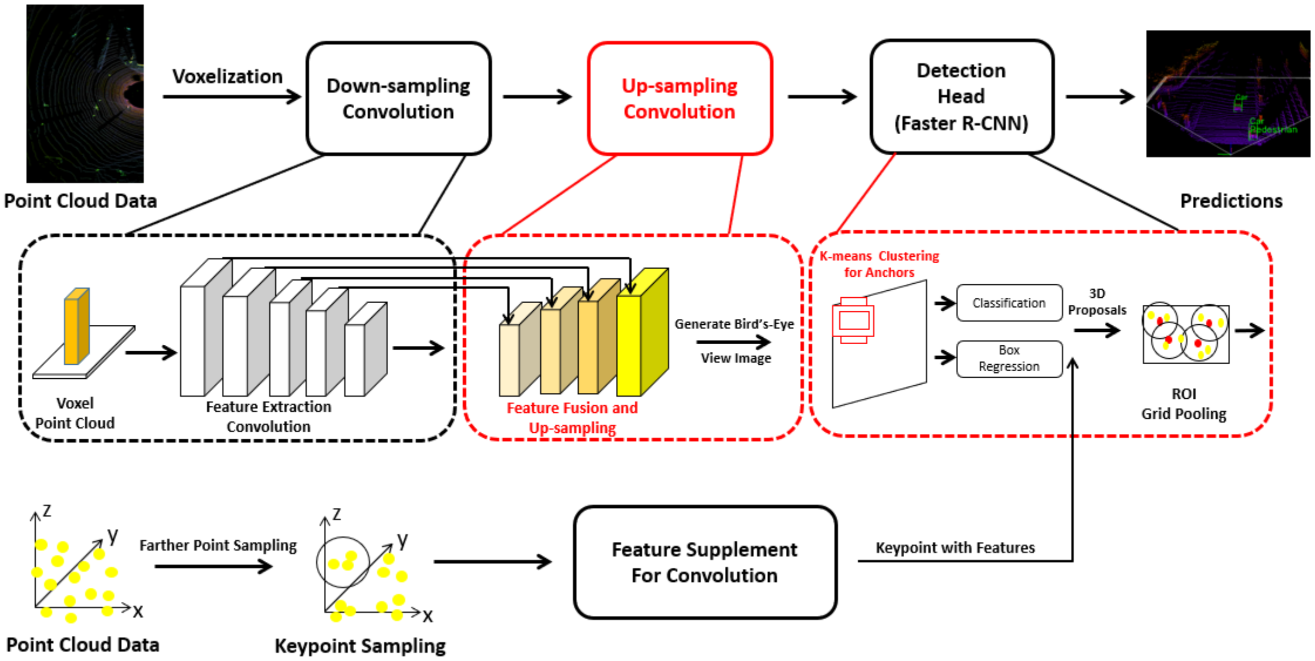
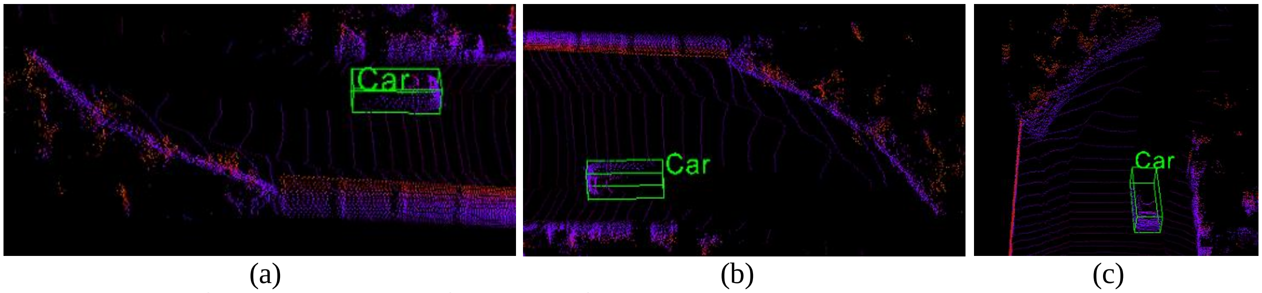


Fig 3. The framework for the improved PV-RCNN. The part in red is our improved method. The method includes data processing and an improved structure. In our work, we propose a more precise detector for 3D objects based on point clouds using data augmentation, upsampling convolution and k-means clustering for anchors.

Through the analysis of the problems, in this paper, we divide the problem into two subproblems: the augmentation method and the training method for point cloud data.

We use the geometric transformation method as the augmentation method and add noise points to achieve data augmentation. Fig 4 shows the geometric transformation method for the data augmentation. Because of the characteristics of a point cloud (4 values (*x*, *y*, *z*, and *color*)), we can add noise points to generate data. However, the noise points will affect the representational model and detection precision. Thus, our method only uses 2% noise points to generate the point cloud data.



(a) (b) (c)

Fig 4. The examples of using geometric transformation to finish data augmentation. (a) The raw point cloud data. (b) According to the mirror image method, our method can generate a certain amount of point cloud data with the controllable parameters. (c) With the rotation parameters, our method can use the x-axis and y-axis to generate image-based and target-based point cloud data.

Second, regarding the training policy of point cloud data, in order to achieve end-to-end training and decrease the memory consumption, we use online data augmentation to amplify the dataset and control the iterations to adjust the amount of point cloud data. Online data augmentation achieves data augmentation by controlling each iteration. This method costs less memory and has higher training efficiency.

Upsampling Convolution

In analyzing the backbone of the PV-RCNN, we find that the prediction objects use the latest feature map from the downsampling convolution, which influences the detection precision and representational ability of the backbone. Thus, we propose an upsampling convolution based on feature fusion to improve the representational stability of feature extraction networks. This network uses a 3×3×3 3D convolution to extract the depth features of the object and fuse the shallow layer features with 2×, 4×, and 8× upsampled sizes. Our backbone not only decreases the classification error by adding the network layer but also extracts the fine-grained features for the small size 3D objects via feature fusion. With the analysis of the FPN (Feature Pyramid Network) [18], we use the feature pyramid to construct the feature fusion method. In addition, we do not use the convolution to downsample the front layer feature map in the feature fusion process. We use the channel combination method at the same receptive field to complete the feature fusion.

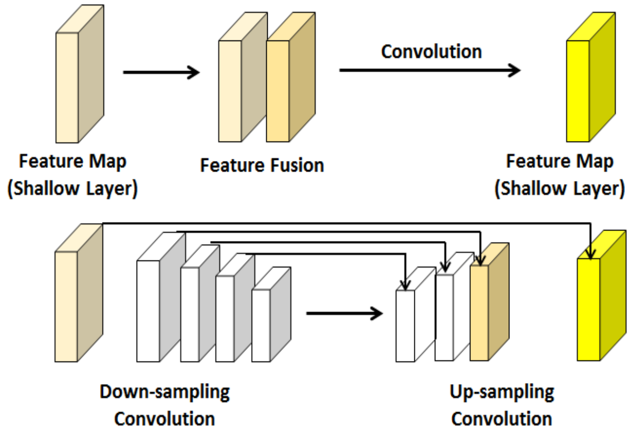


Fig 5. The examples for feature processing of the upsampling convolution. According to the FPN [17], we use the 3D sparse convolution to construct the upsampling stage to extract the fine-grained features of objects.

Fig 5 shows the structure and processing of the point cloud features. The backbone of our method consists of 9 3D sparse convolution layers and the size of the convolutional kernels is 3×3×3. In our backbone, the detector heads can acquire numerous 3D point cloud features from the fusion features of the shallow-deep layer on the up-sampling convolution. In addition, the large size feature map from the last layer can generate more region proposals on the bird's-eye view images. The large number of region proposals provides the foundation for selecting the candidate 3D bounding box.

k-means Clustering for Anchor Sizes

The PV-RCNN use the Faster RCNN as the detection head. Therefore, it manually sets a specific size and number of anchor boxes as the candidate bounding box. Generally, the method needs to compute many regressions for the anchor box to get the right bounding box of the object location. Moreover, when facing a 3D object with a massive size change, the difficulty of obtaining a better detection result for the detectors will increase.

Inspired by YOLOv3 [19] and RefineNet [20], we propose an anchor box selection method that selects the sizes and number of anchor boxes using k-means clustering. In our method, we use the ground truth bounding box to determine the sizes and number of anchor boxes. To decrease the calculation for the bounding box regression, we apply k-means clustering to the bounding box of the ground truth to control the number of anchor boxes.

The process of determining the anchor boxes is as follows: (1) The raw point cloud is transformed into bird's-eye view images. Then, we extract the bounding box of the ground truth. (2) We apply k-means clustering to the ground truth with a random k and M (includes two values: *length* × *width*) to compute the sizes and number of anchor boxes. (3) We use the clustering result as the parameters of the anchor boxes to generate the region proposals. Fig 6 shows our processing method for the sizes and number of anchor boxes.

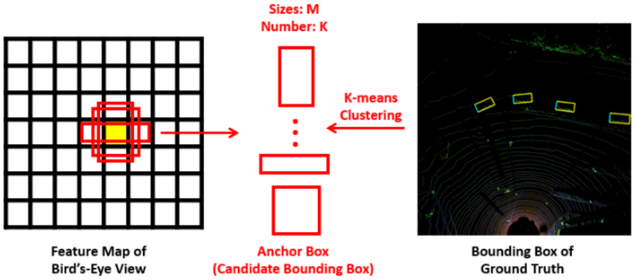


Fig 6. The examples for setting the sizes and number of anchor boxes. This method uses k-means clustering for the sizes and number of anchor boxes on the bounding box of the ground truth. According to the clustering result, we can control the number and sizes of anchor boxes to acquire the optimal detection model in the training stage.

Here, we chose 5 sizes of anchor boxes to complete the stage of generating region proposals on the KITTI 3D object dataset. Due to the differences in the dataset, we need to select different sizes of anchor boxes to acquire higher detection precision. Furthermore, the balance of the detection speed and precision can also be adjusted by controlling the number of parameters of anchor boxes.

5. EXPERIMENTS

In this paper, all experiments used the KITTI 3D object detection dataset, which include LiDAR data of point clouds and 2D images. We only used the LiDAR data to train our detection model and used the point cloud for data augmentation. This dataset includes 7481 training images and 7518 testing images. We divided the dataset into training data (3712 images) and validation data (3769 images). In our experiments, we evaluated the proposed method with the other 3D object detector (LiDAR only) on the validation dataset and test dataset.

The improved point-voxel region convolutional neural network framework was trained using an end-to-end method. We trained the detection model with a batch size of 32 and an initial learning rate of 0.01 for 100 epochs on 2 boards of 22 GB NVIDIA Tesla P40 GPUs. For data augmentation, we used half of the image data to generate the extended data in a batch. In region proposal generation, we use the 3D IoU (Intersection over Union) to divide the foreground region and background region. If the region proposals have a 3D IoU value with the ground truth labels of at least 0.7, they remained as positive samples. For prediction, we used the 160 region proposals from the backbone to detect the object position and classes. We also used the NMS (Non-Maximum Suppression) method to delete the extra 3D bounding boxes.

Table 1. The results of the accuracy comparison on the KITTI 3D test set

Method	Modality	Classes: Car (3D object)			Classes: Cyclist (3D object)		
		Easy	Moderate	Hard	Easy	Moderate	Hard
SECOND[21]	LiDAR only	86.55	77.42	68.38	66.58	523.74	49.24
PointPillars[22]	LiDAR only	88.32	79.19	70.35	68.55	54.09	50.18
Fast Point-RCNN[23]	LiDAR only	89.72	81.84	74.28	70.31	53.28	50.33
Part-A^2[24]	LiDAR only	90.05	86.48	76.53	78.33	59.35	52.24
PV-RCNN[9]	LiDAR only	93.07	90.59	81.47	80.17	66.41	60.18
IPV-RCNN (Our method)	LiDAR only	**96.28**	**92.26**	**84.39**	**82.64**	**68.27**	**61.32**
Improvement	-	+3.21	+1.67	+0.92	+2.47	+1.86	+1.14

Accuracy Comparison on the KITTI 3D Object Dataset

In our experiments, we compared the 3D detection methods, such as SECOND, PV-RCNN and others. All of these comparison methods are used LiDAR data as the input data and the detection precision reached a state-of-the-art level on the load board. To compare the detection accuracy, we selected the optimal result of the training method for every method.

Table 1 shows the comparison results for the different classes on the KITTI 3D test set. Table 2 shows the comparison results for the mean average precision (mAP) on the KITTI 3D test set. Table 3 shows the running time of the methods for single LiDAR data.

Table 2. The results of mAP on the KITTI 3D test set

Method	Modality	3D mAP
SECOND[21]	LiDAR only	79.58
PointPillars[22]	LiDAR only	82.33
Fast PointRCNN[23]	LiDAR only	85.39
Part-A^2[24]	LiDAR only	88.36
PV-RCNN[9]	LiDAR only	90.18
IPV-RCNN (Our method)	LiDAR only	**93.26**
Improvement	-	+3.08

Table 1 and Table 2 prove that our method's detection precision is superior to those of the previous state-of-the-art 3D object detection methods on the KITTI 3D object dataset. Especially for the car class, our method increases the mAP by 3.21% on the easy levels. Furthermore, our method also improves the detection accuracy by 1.67% and 0.92% on the difficulty levels of the moderate and hard difficulty levels of the car class, respectively. Regarding the mean average precision, our method improves the detection accuracy by 3.08% compared with the original PV-RCNN. The chart confirms that the improved point-voxel region convolutional neural network is superior on the KITTI 3D object dataset.

Table 3. The results of speed on the KITTI 3D test set

Method	Modality	Test time/seconds
SECOND[21]	LiDAR only	1.8
PointPillars[21]	LiDAR only	1.5
Fast PointRCNN[22]	LiDAR only	1.6
Part-A^2[23]	LiDAR only	2.3
PV-RCNN[9]	LiDAR only	2.0
IPV-RCNN (Our method)	LiDAR only	1.7

For the methods of Fast PointRCNN, SECOND and PointPillars, none of these was designed the feature compensation structure to improve the representational ability of networks. Thus, these methods have a good running rate on the detection task. For Part-A^2 and PV-RCNN, they are contributing to improve the detection precision. Compared with other methods, IPV-RCNN achieves a good balance for precision and running rate.

Evaluation of the Improved Method

We used the verification experiments to explore the effect of the three sub methods at improving the detection accuracy. The sub methods include data augmentation, upsampling convolution and k-means clustering for anchor boxes. Consequently, we complete the ablation evaluation of our improved method on the KITTI 3D object dataset (we only use the easy car class).

Table 4. The performance of the improved method

Condition	IPV-RCNN (Our method)			
Data Augmentation?	−	√	√	√
Upsampling?	-		√	√
Clustering for Anchor Boxes?	-			√
AP (Car on KITTI 3D test set)	93.07	94.72	95.74	**96.28**
Improvement	-	+1.65	+2.67	+3.21

Table 4 shows the results of the ablation evaluation that the different sub methods have different promotional effects on the detection accuracy. In the three sub methods, data augmentation and the upsampling convolution obviously improve 3D object detection by 2.67% on the easy difficulty of the car class.

6. Conclusion

In this paper, we proposed an improved method for 3D object detection based on the PV-RCNN. First, our work focuses on improving the detection precision for the three-dimensional objects, especially for the small size objects. Second, we analyze and summarize the characteristics and problems of the state-of-the-art PV-RCNN method. Third, according to the problems, we design an improved detection algorithm for three-dimensional objects by designing a data augmentation training policy, adding the upsampling of three-dimensional sparse convolution with feature fusion and using k-means clustering for the size of candidate bounding box on the region proposal network. By using data augmentation in the training stage, we can obtain a better detection model for point cloud data. By applying the upsampling convolution on the backbone, we can obtain a better representational model of the point features and improve the classification accuracy of a point cloud. By clustering anchor boxes by size, we can obtain a better candidate bounding box in the large size feature map and improve the training efficiency for our model. In addition, the upsampling convolution can fuse the feature maps from the shallow layer at different scales. Thus, our backbone can acquire a better representational ability for feature extraction. The experiments have verified the efficiency and precision of the improved method in our model. In future work, we will focus our research on improving the running rate of the detector and achieve robotics applications.

References

[1] Shi Shaoshuai, Guo Chaoxu, Li Jiang, et al. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2020).

[2] Li Peiliang, Chen Xiaozhi, Shen Shaojie. Stereo R-CNN based 3D Object Detection for Autonomous Driving. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2019).

[3] Yin Bi, Aaron Chadha, Alhabib Abbas, et al. Graph-based object classification for neuromorphic vision sensing. IEEE/CVF International Conference on Computer Vision (ICCV). 491–501. (2019).

[4] Mousavian Arsalan, Anguelov Dragomir, Flynn John, et al. 3D Bounding Box Estimation Using Deep Learning and Geometry. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). 5632-5640. (2017).

[5] Tekin Bugra, Sinha Sudipta N, Fua Pascal. Real-Time Seamless Single Shot 6D Object Pose Prediction. IEEE International

Conference on Computer Vision and Pattern Recognition (CVPR). (2017).

[6] Yin Zhou, Oncel Tuzei. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2018).

[7] Liu Zhijian, Tang Haotian, Lin Yujun, et al. Point-Voxel CNN for Efficient 3D Deep Learning. Conference and Workshop on Neural Information Processing Systems (NIPS). (2019).

[8] Maximilian Jaritz, Gu Jiayuan, Su Hao. Multi-view Pointnet for 3d scene understanding. IEEE/CVF International Conference on Computer Vision (ICCV). 0-0, 2019.

[9] Shi Shaoshuai, Guo Chaoxu, Li Jiang, et al. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2020).

[10] Martin Simon, Stefan Milz, Karl Amende, et al. Complex-YOLO: An Euler-Region-Proposal for Real-time 3D Object Detection on Point Clouds. arXiv: 1803.06199v2. (2018).

[11] Hu Houning, Cai Qizhi, Wang Dequan, et al. Joint Monocular 3D Vehicle Detection and Tracking. IEEE/CVF International Conference on Computer Vision (ICCV). (2019).

[12] Shi Shaoshuai, Wang Xiaogang, Li Hongsheng. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2019).

[13] Qi Charles R, Liu Wei, Wu Chenxia, et al. Frustum PointNets for 3D Object Detection from RGB-D Data. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2018).

[14] Qi Charles R, Su Hao, Mo Kaichun, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2017).

[15] Qi Charles R, Li Yi, Su Hao, et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Conference and Workshop on Neural Information Processing Systems (NIPS). (2017).

[16] Andreas Geiger, Philip Lenz, Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2012).

[17] Ren Shaoqing, He Kaiming, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 39(6), 1137. (2017).

[18] Lin Tsung-Yi, Piotr Dollár, Ross Girshick, et al. Feature Pyramid Networks for Object Detection. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2017).

[19] Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement. arXiv.org, (2018).

[20] Zhang Shifeng, Wen Longyin, Bian Xiao, et al. Single-Shot Refinement Neural Network for Object Detection. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). 4203-4212. (2018).

[21] Yan Yan, Mao Yuxing, Li Bo. Second: Sparsely embedded convolutional detection. Sensors, 18(10):3337. (2018).

[22] Lang Alex H, Sourabh Vora, Holger Caesar, et al. Pointpillars: Fast encoders for object detection from point clouds. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2019).

[23] Chen Yilun, Liu Shu, Shen Xiaoyong, et al. Fast Point R-cnn. IEEE/CVF International Conference on Computer Vision (ICCV). (2019).

[24] Johannes Lehner, Andreas Mitterecker, Thomas Adler, et al. Patch refinement - localized 3d object detection. Computing Research Repository (CoRR), abs/1910.04093. (2019).