IEICE TRANS. INF. & SYST., VOL.E107–D, NO.9 SEPTEMBER 2024

PAPER

Reinforced Voxel-RCNN: An Efficient 3D Object Detection Method Based on Feature Aggregation*

Jia-ji JIANG[†a)], **Hai-bin WAN**[†b)], **Hong-min SUN**[††c)], **Tuan-fa QIN**[†d)], ***Nonmembers***, ***and*** **Zheng-qiang WANG**[†††e)], ***Member***

SUMMARY In this paper, the Towards High Performance Voxel-based 3D Object Detection (Voxel-RCNN) three-dimensional (3D) point cloud object detection model is used as the benchmark network. Aiming at the problems existing in the current mainstream 3D point cloud voxelization methods, such as the backbone and the lack of feature expression ability under the bird's-eye view (BEV), a high-performance voxel-based 3D object detection network (Reinforced Voxel-RCNN) is proposed. Firstly, a 3D feature extraction module based on the integration of inverted residual convolutional network and weight normalization is designed on the 3D backbone. This module can not only well retain more point cloud feature information, enhance the information interaction between convolutional layers, but also improve the feature extraction ability of the backbone network. Secondly, a spatial feature-semantic fusion module based on spatial and channel attention is proposed from a BEV perspective. The mixed use of channel features and semantic features further improves the network's ability to express point cloud features. In the comparison of experimental results on the public dataset KITTI, the experimental results of this paper are better than many voxel-based methods. Compared with the baseline network, the 3D average accuracy and BEV average accuracy on the three categories of Car, Cyclist, and Pedestrians are improved. Among them, in the 3D average accuracy, the improvement rate of Car category is 0.23%, Cyclist is 0.78%, and Pedestrians is 2.08%. In the context of BEV average accuracy, enhancements are observed: 0.32% for the Car category, 0.99% for Cyclist, and 2.38% for Pedestrians. The findings demonstrate that the algorithm enhancement introduced in this study effectively enhances the accuracy of target category detection.

key words: *3D object detection, inverted residual sparse convolution, spatial semantic feature fusion, weight normalization*

1. Introduction

With the rapid development of deep learning and image processing layout, and the improvement of computer hardware level, the field of target detection continues to receive high attention from all walks of life since 2012 [1], [2]. As one of the main downstream tasks of computer vision, target detection is mainly divided into two-dimensiona l(2D) target detection and three-dimensional (3D) target detection. Due to the lack of depth message in 2D target detection, it can not provide accurate 3D spatial information of the environment. Meanwhile, the 3D target detection can be more intuitively close to the real-world scene, including the position, angle and distance of the target object and other status information. The research of 3D target detection has received continuous attention. 3D object detection plays an irreplaceable role in the fields of autopilot, robotics, and virtual reality, augmented reality at present [3]–[5].

The current 3D object detection methods are mainly divided into three types. Mainly based on the different data set used, they are divided into monocular 3D object detection [6]–[8], point cloud 3D object detection [9]–[12], and multimodal 3D object detection [13], [14]. As the cost of LiDAR decreases, 3D point cloud object detection becomes the mainstream method. The current mainstream 3D point cloud detection methods are mainly based on two types, point-based and voxel-based.

The point-based methods [15]–[17] directly input the raw data collected by the lidar into the network model without preprocessing. These methods can keep the original information of the real scene well and have the highest accuracy, but they spend a long time to train and detect. Because points are needed to represent the results of the abstract search of the nearest neighbor set.

The grid-based voxelization methods [18]–[20] are to divide the original point cloud information into girds of a fixed size, and perform feature extraction through a 3D convolutional neural network. These methods abandon the complex set abstractions in the point-based methods, and can greatly speed up detection. Due to the occurrence of empty voxels or insufficient points within the voxel grid in the process of point cloud voxelization, conventional feature extraction networks are unable to fully capture the original feature information. Therefore, the detection accuracy and predicted position is slightly lower than that of point-based methods. Meanwhile, because the field of automatic driving requires real-time detection, the method based on gird voxelization has become a research hotspot.

Consider issues such as insufficient point cloud feature extraction capabilities in voxel-based 3D target detection methods. To further enhance the performance of grid-

Manuscript received September 20, 2023.
Manuscript revised January 10, 2024.
Manuscript publicized April 24, 2024.

[†]School of Computer, Electronics and Information and Guangxi Key Laboratory of Multimedia Communications and Network Technology, Guangxi University, China.
[††]Guangxi Vocational & Technical Institute of Industry, Nanning 530001, Guangxi, China.
[†††]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunication, Chongqing, 400065 P.R.China.

*This work was supported in part by the NSF of China # 61961004 and # 62261003.

a) E-mail: 1434193694@qq.com
b) E-mail: hbwan@gxu.edu.cn (Corresponding author)
c) E-mail: sunhm2009@qq.com (Corresponding author)
d) E-mail: tfqin@gxu.edu.cn
e) E-mail: wangzq@cqupt.edu.cn

DOI: 10.1587/transinf.2023EDP7200

based voxelization approaches, we used Voxel-RCNN [21] as the benchmark network and proposed an inverted residual sparse convolutional network, which improved the feature extraction ability of the 3D backbone. In the 2D convolution layer part, an attention-based spatial semantic feature fusion network was proposed, which could fuse spatial features and abstracted semantic features to further improve the feature integration ability of the model. In conclusion, a high-performance point cloud 3D object detection network based on a combination of submanifold inverse residual convolution and feature aggregation was proposed (Reinforced Voxel-RCNN). Specifically, the contributions of this paper can be summarized as follows:

- A submanifold inverted residual network feature fusion network was designed to replace the regular 3D convolution backbone in the baseline network. Submanifold 3D convolution [22] was used to extract backbone features, which effectively solved the issue of losing original feature information when employing excessively deep stacks of 3D sparse convolutional layers.
- A spatial semantic feature fusion network based on convolutional attention was proposed, which could enrich the feature extraction of the network from the perspective of BEV and further improved network performance.

2. Related Work

2.1 Point-Based 3D Point Cloud Object Detection Methods

These methods take the original point cloud as input, use PointNet [9] or PointNet++ [10] as a point cloud information extraction tool, and use iterative sampling and grouping to take the points as representative features. 3d object proposal generation and detection from point cloud (PointRCNN) was proposed by Shaoshuai Shi, *et al.* [15], which proposed a 3D region proposal network to obtain regional features. A method of voting through images (ImVoteNet) was proposed by Charles R. Qi, *et al.* [16], it was proposed to fuse 2D votes in the image and 3D votes in the irregular point cloud to obtain more feature information and achieves good detection results. The authors in [17] fused D-FPS and F-FPS to build a one-stage 3D detector (3DSSD), this method achieved a good balance between accuracy and efficiency. Although PointNet [9] or PointNet++ [10] can provide a flexible receptive field for point cloud feature extraction, the neighborhood search of points in three-dimensional space spends a lot of time.

2.2 Voxel-Based 3D Point Cloud Object Detection Methods

Techniques employing grid voxelization partition the point cloud into a grid of constant dimensions, subsequently employing 2D/3D Convolutional Neural Networks (CNNs) for the purposes of information extraction and detection. An End-to-End Learning for Point Cloud Based 3D Object Detection (VoxelNet) was proposed by Yin Zhou, *et al.* [18], which divided points into 3D voxels, and used tiny PointNet [9] to select a representative feature from each voxel. In 2019, Pointpillars was proposed by Alex H. Lang, *et al.* [19], this method divided the point cloud into pillar from the top view, and used 2D CNN for feature extraction, which achieved a certain balance between speed and accuracy. Shaoshuai Shi, *et al* proposed PV-RCNN [20], which used multi-scale voxel feature aggregation as keypoints through the design of voxel set abstraction. PV-RCNN greatly improved the detection accuracy of 3D object detection at that time. The voxel-based methods abandon the complicated point cloud neighborhood search, which bring certain efficiency improvements and reasoning speed. However, in the process of point cloud voxelization, information loss will inevitably occur. In this paper, we mainly optimize the 3D backbone and 2D convolutional network parts to enhance the feature expression ability of the model.

3. Network Design

The network model in this paper is designed with reference to Voxel-RCNN [21]. From the input to output of point cloud data, it is mainly divided into the following modules:

1) Point cloud data voxel encoding module.
2) 3D submanifold inverted residual convolutional network feature fusion module.
3) Attention-based spatial semantic feature convolution module.
4) Region proposal network.
5) Voxel ROI Pooling.
6) Multi-tasking detection head.

3.1 Voxelization Encoding Module

The objective of voxel coding is to partition point cloud data into uniformly volumetric grids along the three axis: X, Y, and Z. Set the length, width and height of each voxel to V_l, V_w, V_h. Correlation scale corresponding to the point cloud are L, W, H. Then there are $L' = L/V_l$, $W' = W/V_w$, $H'=H/V_h$. Therefore, a non-empty voxel can be expressed as:

$$V = \left\{P_i = [x_i, y_i, z_i, r_i] \in R^4\right\}_i^N, \tag{1}$$

where P_i is the feature output of the point, including the coordinate values x_i, y_i, z_i of the X-axis, Y-axis, and Z-axis and the reflection intensity r_i. i represents i-th point, and N represents the greatest amount of points in each voxel. We adopt the method of benchmark network, and compute the average value of the points in the voxel after gridding as the feature of the voxel, and the compution process is shown as:

$$V_k = \sum_i^T P_i/T, \tag{2}$$

where V_k is the average value of points in the k-th voxel, with

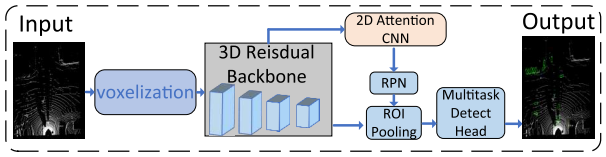


Fig. 1 Network structure

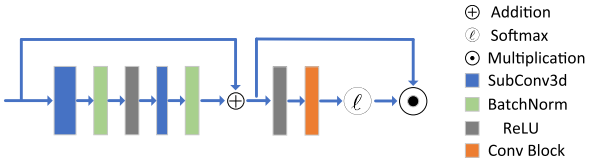


Fig. 2 The basic unit of the module

V_k as the feature of the voxel. P_i is the feature vector of the i-th point in the voxel grid, the meaning is the same as Eq. (1), and T represents the maximum number of sampling points in each voxel, here set T to 5. V_k is the output of the voxelization of the network, followed by 3D sparse convolution to extract the features of each voxel. This work is in the second and third parts of the network structure in Fig. 1.

3.2 3D Inverted Residual Network Feature Fusion Module

This paragraph provides an introductory overview of the designed 3D Inverted Residual Network Feature Fusion Module (3DFFM). In conventional 3D convolution, as outlined in [23], the necessity to traverse the entire spatial area during the convolution process, coupled with the sparsity of point cloud data, leads to substantial computational costs. To address this, two convolution methods have emerged as the mainstream direction in extracting features for 3D point cloud detection in state-of-the-art networks, these are sparse 3D convolution and submanifold sparse 3D convolution. However, with increasing depth of convolution layers, sparse 3D convolution loses some level information and dilutes the feature communication between wach convolutional layer.

In the Voxel-RCNN baseline network, the 3D backbone section employs merely four layers of 3D sparse convolutional layers for feature extraction. Notably, only adjacent convolutional layers facilitate information exchange, inevitably leading to a significant information gap between the first and last convolutional layers. This disparity in feature information extraction is a contributing factor to reduced accuracy in detection outcomes. To solve this issue, a 3D convolution fusion network based on inverted residual is designed. Inspired by Deep Residual Learning for Image Recognition (ResNet), which was proposed by Kaiming He,*et al*. [24], ResNet used the design of residual networks for the first time, and inspired by MobileNetV2 [25], which was proposed by Mark Sandler, *et al*. Our fusion network leverages the inverted residual submanifold sparse convolution method, and enhances the extracted feature information in the latter portion of the network. Specifically, the structure we propose is expressed as shown in Fig. 2.

Table 1 3D backbone network parameters

Components	kernel	channel	Input size	Output size
Conv1	3*3*3	16	1600*1408*41	1600*1408*41
Conv2	3*3*3	32	1600*1408*41	800*704*21
Conv3	3*3*3	64	800*704*21	400*352*11
Conv4	3*3*3	128	400*352*11	200*176*5

The design based on the residual network can deepen the network structure, which has a positive effect on the extraction of sparse features in point clouds, and effectively solves the problem of gradient disappearance of gradient descent caused by the deepening of network layers. Drawing inspiration from the concept of channel expansion in MobileNetV2 [25], the initial convolutional layer doubles the number of channels. This channel expansion is designed to preserve the maximum amount of original feature information. The design of the inverted residual mitigates information loss resulting from the deployment of the Rectified Linear Unit (ReLU) activation function within the convolutional process. Experimental results demonstrate that the strategy of augmenting the number of channels positively contributes to enhancing detection accuracy, and the specific channel number design is given in Sect. 3.3 and Table 1. The comprehensive design is elucidated as follows:

$$\begin{cases} f(x) = w_2 \{\sigma w_1(x)\}, \\ y = f(x) + x, \end{cases} \tag{3}$$

where w_1, w_2 denote the convolution parameter weight of convolution kernel with a size of 3*3*3, which are used for the expansion or compression of the feature channel. In this context, σ denotes the ReLU activation function, and x symbolizes the input feature. The resultant feature following the operation of the inverted residual network is denoted as y. After the inverted residual convolutional network, a feature fusion network based on the weight normalization function is designed to derive the normalized weights (the structure of the second half of Fig. 2). Normalized weights are acquired to extract underlying information, thereby enhancing the model's capacity for feature expression. The acquired weights are subjected to a pointwise multiplication with the eigenvalues generated by the convolution operation. The findings substantiate the superior performance achieved by this method, and the complete design can be formally articulated as follows:

$$F = y * \{\ell c \sigma(y)\}, \tag{4}$$

where y represents the features produced by the inverted residual network, ℓ denotes the Soft Maximum Activation Function (SoftMax), and c corresponds to the Conv Block illustrated in Fig. 2, encompassing a comprehensive convolutional process. It contains three parts, submanifold convolution, batch normalization and ReLU activation function.

Since the submanifold sparse convolution restricts non-empty voxels, there is a disadvantage that the expression of some original features is not clear enough. So in the Conv

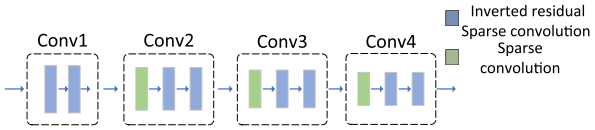


Fig. 3 3D feature extraction network

Table 2 Spatial semantic feature convolution parameters

Components	kernel	Input size	Output size
Spatial Feature CNN	3*3	200*176	200*176
	3*3	200*176	200*176
	3*3	200*176	200*176
Semantic Feature CNN	3*3	200*176	100*88
	3*3	100*88	100*88
	3*3	100*88	100*88

Block module, using the submanifold sparse convolution with a convolution kernel size of 3*3*3 to compress the feature map output by the inverted residual sparse convolution into a channel number of 1, and adaptively extract its channel information, followed by batch normalization and ReLU activation. Then we use the SoftMax function to normalize the feature and obtain the weight (the Softmax function generates values within the range of 0 to 1, facilitating gradient propagation in the network and aiding in model training). Finally, we multiply the weight with the feature map output by the inverted residual sparse convolution in an element-by-element manner, and the result is used as the final output.

3.3 3D Backbone Design

In the network of our method, the 3D backbone feature extraction network is designed immediately after voxelization encoding. The 3D backbone which designed to extract feature is mainly composed of four down-sampling convolution modules, where the down-sampling operation is mainly concentrated in the sparse convolution layer. The convolution module is structured with two components: sparse 3D convolution and a novel sparse convolution founded on inverse residual and feature aggregation. Sparse 3D convolution is characterized by height computational efficiency, whereas the inverted residual sparse convolution excels in capturing pivotal features, the composition relationship is shown in Fig. 3.

Table 1 shows the down-sampling parameter information of each layer. The output features are acquired through the reduction in size of the input features, achieved primarily through two prevalent down-sampling techniques: convolution and pooling. In this study, convolution downsampling is used, and feature map size downsampling can be achieved by changing the convolution step size. Table 1 details the specific convolution step values for the four convolutional layers, which are set to 1, 2, 2, and 2, respectively. The feature sizes are displayed according to the Y*X*Z axis, and the usage is consistent with Table 2 below. Based on the

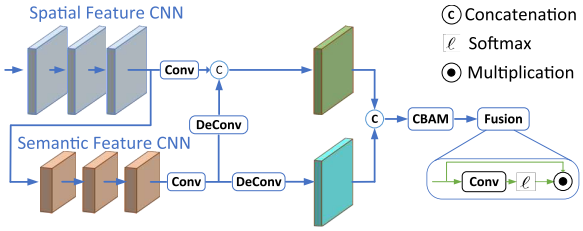


Fig. 4 Spatial semantic feature convolution module structure

input feature size, the down-sampling multiples are set to 1, 2, 4, and 8 times, the convolution kernel size is set to 3*3*3, and the number of channels is designed to be 16, 32, 64, 128 respectively. The output feature is converted into a feature map with a size of 200*176*2 after a layer of 3D sparse convolution with a convolution kernel size of 3*1*1. This adjustment can increase the receptive field in the height direction and speed up the training speed of the network.

3.4 Spatial Semantic Feature Convolution Module Based on Convolutional Attention

In this section, our primary focus is to provide a comprehensive introduction to the specially designed 2D convolutional feature extraction network.

After downsampling the 3D convolutional network to obtain the feature data of the point cloud, the next step of feature extraction needs to be performed in the BEV perspective. For the calculation method of the BEV perspective, we refer to Voxel-RCNN. By compressing along the Z-axis, the output feature map size is altered from 200*176*2 to 200*176. This adjustment is guided by real-world scenarios where objects tend not to overlap along the Z-axis. Furthermore, the tensor derived from the 3D convolutional network segment is inherently sparse, necessitating a conversion from sparse to dense data before the commencement of 2D convolution feature extraction. Although the method of compressing the feature map on the Z-axis improves efficiency, in the baseline, only two branches of down sampling and transposed convolution are used to extract spatial and semantic information, and the number of stacked layers is too deep, which will cause a certain degree of information loss. Therefore, it is necessary to redesign the 2D convolutional network part.

To address this issue, with reference to the spatial semantic feature aggregation network in CIA-SSD, which was proposed by Wu Zheng, *et al*. [26], and the design ideas of attention modules in Convolutional Block Attention Module (CBAM), which was proposed by Sanghyun Woo, *et al*. [27]. We propose a 2D spatial semantic feature extraction network which is based on channel and spatial attention. The use of spatial feature convolution groups and semantic feature convolution groups in the network can achieve robuster feature extraction. Figure 4 shows the module structrue.

Different from the benchmark network, we only stack three layers of convolutional layers in the semantic and spatial feature convolution groups, which reduces the processing complexity of the network while retaining certain spatial and

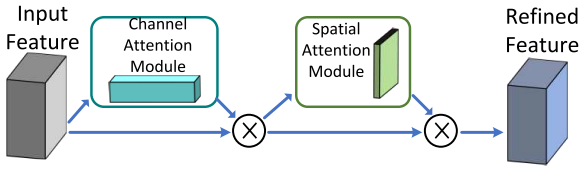


Fig. 5 Convolutional block attention module

semantic information. The amount of output channels of the spatial feature convolution part is 64. The convolution step is set to 1, and the feature size is guaranteed to remain unchanged. The number of output channels of the semantic feature convolution part is 128, the convolution step is set to 2, and the feature size is reduced to original half. The specific convolution parameters are shown in Table 2.

The configuration of the spatial and semantic feature convolution group enables the acquisition of more extensive high-level semantic features and deep spatial feature information. After obtaining the spatial and semantic features, the semantic features are transposed and convolved. Transposed convolution, also known as deconvolution, is an upsampling operation that maps an image from a small resolution to a large resolution. The purpose of upsampling is to make the resolution size of semantic features consistent with the spatial features.

In our approach, the convolution kernel size of the deconvolution is set to 3*3, the convolution stride is set to 2, the ReLU activation function is used to activate, and then stacked with the spatial features in the channel direction. The stacking operation is the operation of juxtaposing two feature maps with the same resolution and size on the channel. The semantic features after transposed convolution and the spliced spatial features are stacked again in the channel direction. As a result of decreasing the number of semantic feature channels and augmenting the dimensions of semantic features during the transposed convolution process, a portion of the feature information may become attenuated. To find solutions to problems, after the second stacking of semantic features and spatial features, a CBAM attention module is added to strengthen the stacked spatial semantic features in terms of channels and spatial directions. The module structure is depicted shown in Fig. 5.

CBAM contains channel attention and spatial attention. The role of the channel attention module is to keep the channel dimension unchanged, compresses the spatial dimension, and focus on the classification information of the target, while the spatial attention module compresses the channel dimension based on the unchanged spatial dimension, this part focuses on target's location information. For the input feature $F \in R^{C*H*W}$, passing through the first channel attention module and the subsequent spatial attention module, the corresponding weights are $M_C \in R^{C*1*1}$, $M_S \in R^{1*H*W}$. Assuming the input features is F, the output features are obtained as:

$$F' = M_C(F) \otimes F, \tag{5}$$

$$F'' = \mathrm{M}_S(F') \otimes F', \tag{6}$$

where $\otimes$ is dot product between elements, channel attention weight M_C and spatial attention weight M_S can be expressed as:

$$M_C(F) = \sigma\{MLP(AvgPool(F))\} + \sigma\{MLP(MaxPool(F))\}, \tag{7}$$

$$M_S = \sigma\left\{f^{7\times7}([AvgPool(F), MaxPool(F)])\right\} \tag{8}$$

where $AvgPool$, $MaxPool$ are average pooling and maximum pooling operations respectively. MLP represents a multi-layer perceptron, here are two layers, and $f^{7\times7}$ is 7*7 convolution operation. After the attention module, the fused spatial and semantic feature information has been strengthened, and the last step is to perform the fusion operation. Assuming that the output feature after the CBAM attention is $F \in R^{C*H*W}$, the feature after the fusion operation can be defined as:

$$F' = \left(\ell f^{3X3}(F)\right) \otimes F, \tag{9}$$

where f^{3X3} is 3*3 convolution operation, and ℓ is SoftMax function. The design of feature fusion is mainly to use the SoftMax function to regularize the merged spatial semantic features, and used the result of regularization as the weight of each feature after merging. Finally, we perform dot multiplication on the corresponding weight and feature value. Such a structure can establish the dependency relationship between spatial features and semantic features, and the dot product result is output from the BEV perspective as the final spatial and semantic fusion feature.

3.5 Design of Loss Function

3.5.1 Design of Focal Loss Function

To solve the issue of unbalanced anchor categories generated on the point cloud feature map, the FocalLoss [28] is referenced, and the function is designed as follows:

$$focalloss(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t), \tag{10}$$

$$p_t = \begin{cases} p & if\ y = 1, \\ 1-p & otherwise, \end{cases} \tag{11}$$

where y is the label value of the sample, and p is the model predicts the probability that a certain sample is a positive sample. $y = 1$ indicates a positive sample, α_t, γ are hyperparameters, set 0.25 and 0.2, respectively.

3.5.2 RPN Loss Function

For point cloud detection, the setting of anchors generally contains eight vector dimensions, the first seven-dimensional vector represents the position information, length, width and height of the box. The last vector is the target category information. For different categories, different IoU thresholds are required to divide positive and negative sample information. The RPN loss function is as follows:

Table 3 The comparison results of this network in 3D evaluation indicators and mainstream networks in the KITTI dataset.

Method	Car			Cyclist			Pedestrians		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
pointpillars [19]	82.56	75.84	72.13	77.83	60.74	57.30	51.56	45.53	40.90
pillarnet [30]	85.79	76.28	72.49	77.64	59.23	55.57	48.18	42.86	39.43
F-PointNet [31]	83.76	70.92	63.65	77.15	56.49	53.37	N/A	N/A	N/A
pointrcnn [15]	88.65	78.32	77.64	**87.12**	72.46	66.92	61.82	53.63	46.63
TANet [32]	87.52	76.64	73.86	84.53	61.64	57.44	N/A	N/A	N/A
VoxelNet [18]	81.97	65.46	62.85	67.17	47.65	45.11	57.86	53.42	48.87
Part-A^2 [33]	89.32	78.87	78.21	83.64	71.57	68.94	61.08	53.30	49.30
SECOND [34]	88.15	77.63	76.12	78.41	62.91	59.92	53.62	48.63	45.15
Ours	87.97	77.46	76.51	78.74	63.31	59.57	55.23	50.06	45.69
	-0.18	-0.17	+0.39	+0.33	+0.40	-0.35	+1.61	+1.43	+0.54
Voxel-RCNN [21]	**89.60**	79.35	78.52	85.63	71.24	68.06	61.87	54.09	49.26
Ours	89.57	**79.93**	**78.66**	86.43	**72.53**	**68.32**	**62.21**	**57.11**	**52.16**
	-0.03	+0.58	+0.14	+0.80	+1.29	+0.26	+0.34	+3.02	+2.90

$$L_{RPN} = \frac{1}{N_f}\left\{\sum_i L_{cls}\left({c_i}^a, {p_i}^*\right)\right\} + \frac{1}{N_f}\left\{\mathbb{I}\left({p_i}^* \geq 1\right)\sum_i L_{reg}\left({\delta_i}^a, {t_i}^*\right)\right\}, \quad (12)$$

where N_f is the number of foreground anchor boxes, and L_{cls}, L_{reg} are classification and regression loss, respectively. ${c_i}^a$, ${p_i}^*$ represent the prediction results of classification and regression. ${\delta_i}^a$, ${t_i}^*$ indicate classification labels and regression labels respectively. $\mathbb{I}\left({p_i}^* \geq 1\right)$ indicates that the regression loss is only by calculated on the positive samples.

3.5.3 The Loss Function of Detection Head

The design of the detection head loss function refers to the benchmark, which can be expressed as:

$$L_{head} = \frac{1}{N_s}\left\{\sum_i L_{cls}\left(p_i, {l_i}^*\right)\right\} + \frac{1}{N_s}\left\{\mathbb{I}\left(IoU_i \geq \theta_{reg}\right)\sum_i L_{reg}\left(\delta_i, {t_i}^*\right)\right\}, \quad (13)$$

where IoU_i indicates the IoU ratio between the i-th suggestion box and the real box, $\mathbb{I}\left(IoU_i \geq \theta_{reg}\right)$ indicates that the region candidate box is only involved in the calculation of regression loss when $IoU_i \geq \theta_{reg}$, and N_s is the number of region candidate boxes.

4. Experimental Results and Analysis

4.1 Experimental Settings

The hardware configurations for this study were housed on the local host, comprised of a 64-bit Linux system (Ubuntu 18.04), an Nvidia RTX2070s graphics card, and 8GB of video memory. The experimental environment consisted of pytorch1.8.1, python3.9.13, and CUDA10.2 configurations. The experiment utilized point clouds within range $[0, 70.4]$m along the X-axis, $[-40, 40]$m along the Y-axis, and $[-3, 1]$m along the Z-axis, with an input voxel size of $(0.05, 0.05, 0.1)$m. In our study, the network optimizer employed Adam, with an initial learning rate of 0.01 and an optimized momentum parameter of 0.9. The experiment was conducted on a single card with an 80-epoch training and a batch size of 2.

4.2 Selection and Evaluation in Index of Data Sets

The experiment utilized the KITTI [29] dataset by employing comprehensive assembly equipment to collect data samples of vehicles in an actual traffic scenario. The dataset consisted of 7481 training samples and 7518 testing samples, partitioned into a training set comprising 3712 samples and a validation set comprising 3769 samples. The network of our method mainly evaluates the three categories of Car, Pedestrians, and Cyclist in dataset, and the anchor box setting of these three categories are Car $[3.9, 1.6, 1.56]$m, Pedestrians $[0.8, 0.6, 1.73]$m, Cyclist $[1.76, 0.6, 1.73]$m. The performance was evaluated using the standard KITTI metrics, with the average accuracy (AP) used to measure the 3D and BEV indicators. The evaluation was conducted across three difficulty levels: easy, moderate, and hard.

4.3 Network Comparison Experiment

This section compares the experimental results of the network model with the results of the mainstream 3D object detection network, and discusses the effect of network improvement, using the average accuracy of 11 recall positions. Our results are delineated in Table 3 and Table 4, with the best results highlighted in bold black font.

Given the similarity in the architectural design of the 3D backbone network and the utilization of 2D convolutional layers, both the Voxel-RCNN and SECOND algorithms adopt a common framework. Specifically, they employed a four-layer convolutional downsampling structure in

Table 4 The comparison results of the detection indicators of this network in the BEV and the mainstream network in the KITTI dataset.

Method	Car			Cyclist			Pedestrians		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
pointpillars [19]	89.58	86.90	83.52	81.93	66.66	61.51	56.47	51.10	46.76
pillarnet [30]	89.20	86.13	83.35	83.35	65.86	61.89	55.53	49.97	46.71
F-PointNet [31]	88.16	84.02	76.44	81.82	60.03	56.32	N/A	N/A	N/A
pointrcnn [15]	89.86	87.26	86.64	87.54	74.35	71.82	**67.23**	58.81	52.59
TANet [32]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
VoxelNet [18]	89.60	84.81	78.57	74.41	52.18	50.49	65.95	**61.05**	**56.98**
Part-A^2 [33]	90.25	87.62	86.79	86.80	74.43	71.60	62.84	57.18	52.53
SECOND [34]	89.95	87.09	84.93	80.86	67.67	63.52	57.57	53.52	49.55
Ours	90.17	87.44	85.23	80.57	68.25	64.05	59.23	55.08	50.81
	+0.22	+0.35	+0.30	-0.29	+0.58	+0.53	+1.66	+1.56	+1.26
Voxel-RCNN [21]	90.41	88.15	87.45	**90.42**	73.35	70.51	63.80	57.49	52.58
Ours	**90.45**	**88.62**	**87.90**	90.25	**75.08**	**71.94**	66.09	60.17	54.75
	+0.04	+0.47	+0.45	-0.17	+1.73	+1.43	+2.29	+2.68	+2.17

Table 5 3D detection average precision of network ablation experiments in KITTI validation set

Methods	3D Backbone	2D CNN	Car			Cyclist			Pedestrians		
			Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
(a)			89.60	79.35	78.52	85.63	71.24	68.06	61.87	54.09	49.26
(b)	$Five\_Layers$		89.46	79.23	78.37	85.46	70.96	68.01	61.90	53.87	49.07
(c)	$3DRes$		89.42	79.27	78.46	85.76	71.26	67.17	61.63	53.64	50.39
(d)	$3DFFM$		89.55	79.48	78.68	85.55	71.89	67.22	61.36	56.32	51.18
(e)		$SSFA$	89.44	79.21	78.43	85.21	71.92	67.86	62.58	54.60	50.46
(f)		$CBAM$	89.34	79.27	78.50	85.33	71.62	68.27	62.23	53.85	49.68
(g)		$SSFE$	89.75	79.40	78.42	86.64	73.18	70.28	62.75	54.16	50.20
(h)	$3DFFM$	$SSFE$	89.57	79.93	78.66	86.43	72.53	68.32	62.21	57.11	52.16

the 3D network, and incorporated five 2D convolutional layers within the 2D network. Consequently, we conducted a supplementary set of comparative experiments with the SECOND algorithm to investigate the efficacy of the network improvements. From Table 3 and Table 4, in comparison with SECOND algorithm, both the 3D detection accuracy and the accuracy of the BEV perspective have been improved to a certain extent on the three target categories of Car, Cyclist, and Pedestrians. Verified that our improved network architecture enhances the expressive capability of feature information. It is noticeable that our method outperforms the baseline network (Voxel-RCNN) as well as previous methods. Under the 3D detection accuracy index and the bird's-eye view index, three difficulty levels have a certain degree of accuracy improvement compared with the baseline. On large target objects, the moderate and hard levels of vehicle category under 3D detection increased by 0.58% and 0.14%, respectively. The improvement effect is obvious on the small target category. For instance, the 3D detection metrics for pedestrian categories exhibit improvements of 0.34%, 3.02%, and 2.90%, respectively. Simultaneously, BEV detection demonstrates enhancements of 2.29%, 2.68%, and 2.17%, respectively.

4.4 Network Ablation Experiment

In this section, Voxel-RCNN is used as a benchmark to conduct ablation research on the improved model. 11 recall positions are used for average precision calculation. Table 5, Table 6 and Table 7 give the results of the ablation experiment.

There are eight different combinations of experimental settings in Table 5 and Table 6. The design of the 3D inverted residual network feature fusion module is called 3DFFM, and the attention-based spatial and semantic feature convolution module is called SSFE. 3DRes represents the conventional residual convolution method. SSFA and CBAM represent use cases in SSFE without using CBAM, as well as methods using only CBAM. The experiment with sequence number (a) adopts the same method as the baseline network in 3D and 2D networks respectively, each set of ablation experiments investigates the impact of the corresponding module structure on detection accuracy.

Table 7 explores the impact of various improved parts of the network on model inference time. It can be demonstrated from the experimental results that both the 3DFFM and SSFE methods can improve the detection accuracy, and the SSFE method is used to further shorten the model's inference time to 43 ms. Compared with the baseline, the final improved

Table 6 BEV detection average precision of network ablation experiments in KITTI validation set.

Methods	3D Backbone	2D CNN	Car			Cyclist			Pedestrians		
			Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
(a)			90.41	88.15	87.45	90.42	73.35	70.51	63.80	57.49	52.58
(b)	$Five\_Layers$		90.27	88.04	87.13	90.17	72.96	70.57	63.78	57.26	52.42
(c)	$3DRes$		90.33	88.29	87.65	88.32	73.31	70.28	63.76	58.53	53.30
(d)	$3DFFM$		90.28	88.49	87.76	89.86	74.21	71.11	63.37	58.98	53.77
(e)		$SSFA$	90.35	88.12	87.40	90.96	73.91	70.83	64.61	58.47	53.55
(f)		$CBAM$	90.40	88.32	87.73	91.37	73.93	71.37	64.06	57.98	53.53
(g)		$SSFE$	90.38	88.27	87.63	91.33	77.52	72.77	64.74	57.72	53.71
(h)	$3DFFM$	$SSFE$	90.45	88.62	87.90	90.25	75.08	71.94	66.09	60.17	54.75

Table 7 The performance of inference time on KITTI validation set. The results are evaluated with the average precision for car class.

Methods	3DFFM	SSFE	Moderate $AP_{3D}(\%)$	Time(ms)
(a)			79.35	48
(b)	✓		79.48	51
(c)		✓	79.40	43
(d)	✓	✓	79.93	46

network (method d) shortened the inference time by 2 ms.

4.4.1 Investigate the Influence of Varying the Number of Convolutional Layers within the 3D Backbone on the Accuracy of Network-Based Object Detection

In ablation experiment (b), the 3D backbone incorporates a five-layer structure of sparse convolutional layers, increasing an layer compared to the baseline network.

The experimental results demonstrate a decrease in both 3D detection accuracy and BEV detection accuracy to a certain extent. This substantiates the assertion that indiscriminately increasing the depth of convolutional layers is not an optimal solution. Consequently, it emphasizes the necessity of an adaptive convolutional layer structure design.

4.4.2 Experimental Results of 3D Inverted Residual Network Feature Fusion Module

The difference between experiment (c) and experiment (d) is only that in the 3D backbone network, the residual connection method is different. Experiment (c) is conventional residual connection, while experiment (d) is an inverted residual connection method. The results of experiment (c) show that the traditional residual connection method has a certain accuracy improvement in the evaluation scales such as moderate and hard of the category of pedestrians, but it has declined in the detection of other categories. It shows that the residual convolution method can obtain more feature information of small target categories, but deeper residual convolution will also lead to the dilution of feature information of some categories during the convolution process, such as Car and Cyclist. In the inverted residual convolutional connection, more original feature is preserved by expanding the number of output channels of the feature map. Compared with experiment (c), the detection accuracy has been significantly improved in both AP_{3D} and BEV indicators.

The experimental model of 3DFFM has a serial number of (d). The analysis in Table 5 and Table 6 shows that the three types of 3D detection achieve accuracy rate improvements of 0.13%, 0.65%, and 2.23%, respectively. In the context of medium difficulty average metrics, the accuracy of BEV displays increments of 0.34%, 0.86%, and 1.49%, respectively. Attributable to the architectural incorporation of the inverted residual convolution module and the application of the SoftMax weight function, which facilitates the network to extract deeper feature information and express it. Notably, this effect was particularly impressive for small target categories, including Cyclist, Pedestrians, and significantly contributed to the superiority of the proposed module.

4.4.3 Experimental Results of Attention-Based Spatial Semantic Feature Convolution Module

In this section, ablation experiments are categorized into three distinct groups, denoted as (e), (f), and (g). These experiments individually investigate the influence of distinct structural designs within 2D CNN networks on the accuracy of detection.

Compared with the baseline, the utilization of SSFA has led to an enhancement in the detection accuracy of the two smaller target categories, namely Cyclists and Pedestrians, but it has slightly decreased in the Car category. It shows that the spatial and semantic information convolution group in SSFA can improve the feature extraction ability of small target objects, but for large target categories, this method will cause certain feature loss. In experiment (f), the CBAM method is used to retain more characteristic information of large target objects while maintaining the accuracy of small target categories. SSFE is a combination of SSFA and CBAM, and the experiment is labeled as Experiment (g). Under the hard evaluation level, the accuracy of 3D detection decreases by 0.10% for car but increases by 2.22% and 0.94% for Cyclist and Pedestrians, respectively. This observation underscores the substantial enhancement of feature extraction capabilities for small object categories achieved by the SSFE module, while simultaneously maintaining the overall accuracy of large object categories with negligible alterations. The precision of BEV detection im-

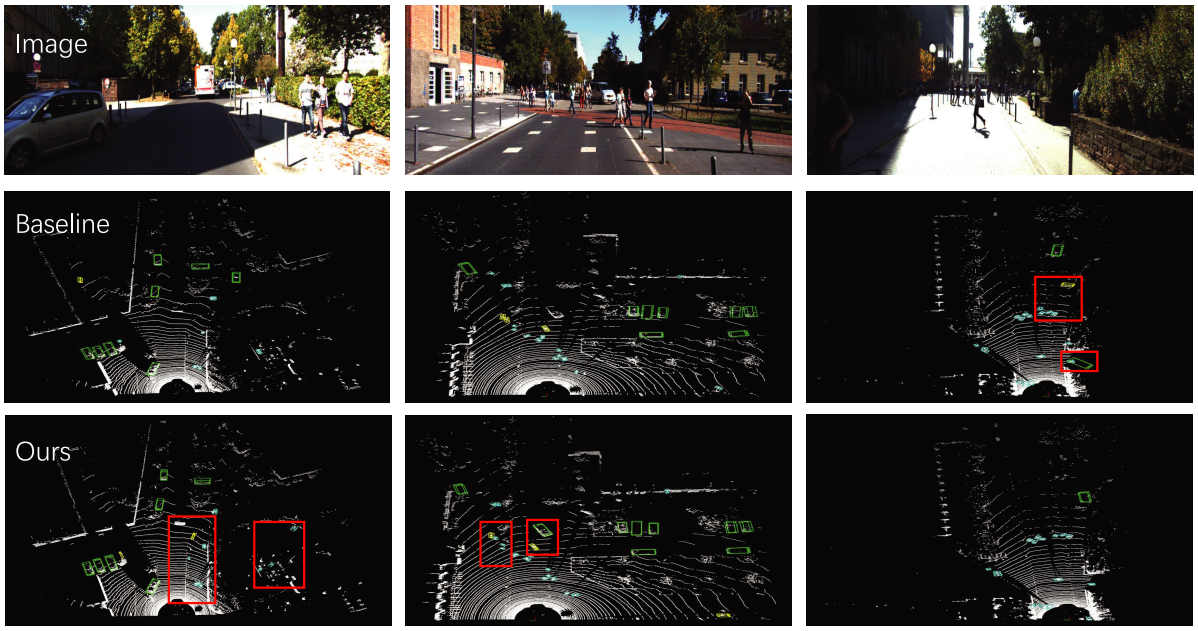


Fig. 6 The visualization of our 3D detection results on the KITTI validation set, there are three columns in total, divided into A, B, C.

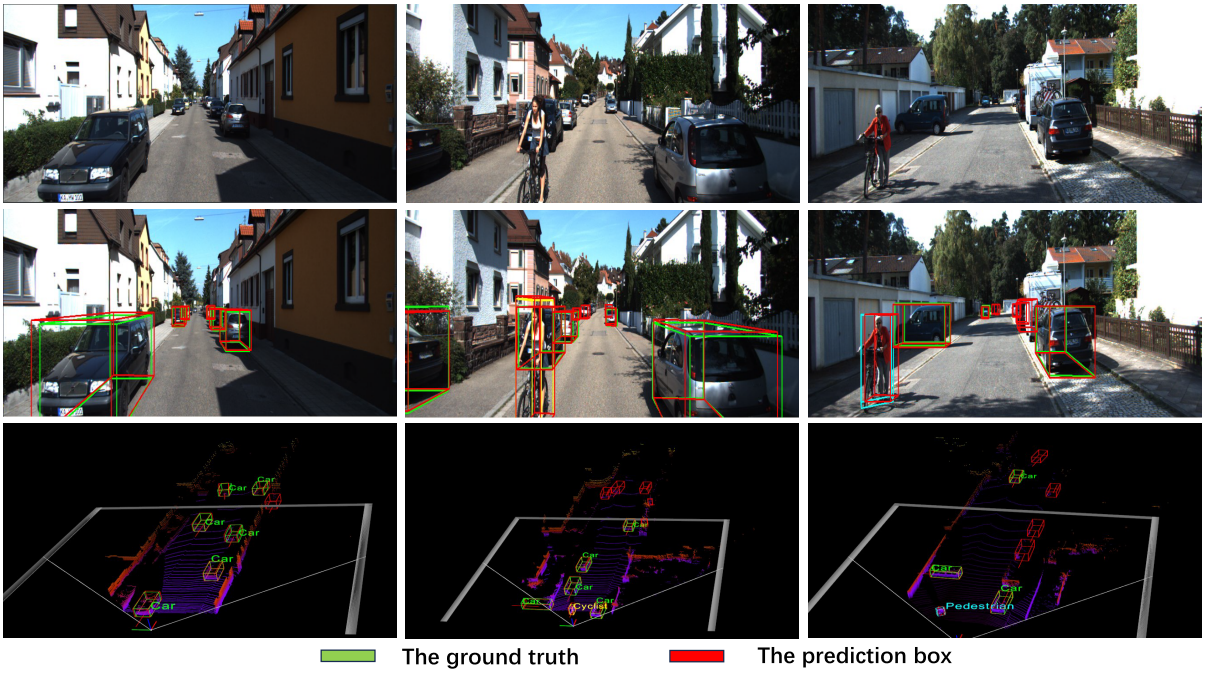


Fig. 7 The visualization on KITTI validation

proved by 0.18%, 2.26%, and 1.13%, correspondingly, in the hard evaluation tier. In conclusion, these results demonstrate that the use of channel stacking between the semantic and spatial convolution layer and CBAM structure in the SSFE module can focus on clearer channel and spatial information.

4.5 Visualization of Network Results

The enhanced network incorporates six distinct scenario groups for visualization purposes, with the resulting visualizations presented in Fig. 6 and Fig. 7. Specifically, Fig. 6 encompasses three distinct scenarios: A, B, and C.

In Scenario A, a notable contrast emerges between the enhanced network and the baseline, highlighted within the red box. The improved network effectively maintains accuracy in the Car category detection while also exhibiting enhanced recognition of Pedestrians information, resulting in an overall improvement in Pedestrian category detection

accuracy. Scenario B demonstrates the resolution of Car detection issues observed in the Reinforced Voxel-RCNN visualization results, along with improved representation of target object features, a challenge successfully addressed by our method. In Scenario C, an issue identified in the baseline where Pedestrians were incorrectly categorized as Cyclist has been diligently rectified in Reinforced Voxel-RCNN.

In Fig. 7, the green box represents the ground truth and the red box represents the network predicted value, which contains the 3D prediction value and the ground truth in the point cloud and image scenes. It can be seen from the three sets of scenes in Fig. 7 that the predicted values of the improved algorithm in this paper are in good agreement with the real values. High-precision prediction can also be made for targets without real labels in the scene, and the target object category can be correctly predicted. In conclusion, the visualization results from the aforementioned scenarios effectively prove the rationality and effectiveness of the network improvement.

5. Conclusion

In this paper, The 3D convolutional backbone extraction network in the benchmark network and the feature information extraction and expression capabilities from the BEV detection perspective are not strong, and the hierarchical information fades with the deepening of the convolutional layers, resulting in low detection accuracy and object misdetection.

Our approach (Reinforced Voxel-RCNN) proposes the design of a 3D inverse residual network feature fusion network and an attention-based spatial semantic feature convolutional network. The design of the inverted residual convolutional network preserves a higher quantity of 3D sparse features, effectively mitigating issues associated with information loss stemming from excessively deep convolutional network layers, and the introduction of the attention-based spatial semantic feature convolution network enhances the network's capability to amalgamate deeper channel and semantic information in BEV detection. The test results on the public dataset KITTI demonstrate that our method can further improve the detection accuracy when compared with the forward state-of-the-art 3D point cloud object detection methods. In addition, the proposed network improvement could be transplanted to other algorithms with the same partial structural design, this hypothesis was verified in the comparative experiments in Table 3 and Table 4, and also achieved positive results.

Future work includes the optimization of multi-task detection heads within the network, with a dedicated focus on reducing anchor box matching time and enhancing real-time detection efficiency. This undertaking represents a challenging research endeavor aimed at elevating the network's overall detection performance.

References

[1] Y. Zhao, Y. Rao, S. Dong, and J. Zhang, "A review of deep learning target detection methods," Image Graph, vol.25, no.4, pp.629–654, 2020.
[2] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object detection," Computer Vision: A Reference Guide, pp.1–9, 2020.
[3] C. Tao, J. Cao, C. Wang, Z. Zhang, and Z. Gao, "Pseudo-mono for monocular 3d object detection in autonomous driving," IEEE Trans. Circuits Syst. Video Technol., vol.33, no.8, pp.3962–3975, 2023.
[4] S.Y. Alaba and J.E. Ball, "Deep learning-based image 3d object detection for autonomous driving," IEEE Sensors J., vol.23, no.4, pp.3378–3394, 2023.
[5] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3d lidar and camera data for object detection in autonomous vehicle applications," IEEE Sensors J., vol.20, no.9, pp.4901–4913, 2020.
[6] Y. Hu, S. Fang, W. Xie, and S. Chen, "Aerial monocular 3d object detection," IEEE Robotics and Automation Letters, vol.8, no.4, pp.1959–1966, 2023.
[7] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang, "Monocular 3d object detection: An extrinsic parameter free approach," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.7556–7566, 2021.
[8] X. Wu, D. Ma, X. Qu, X. Jiang, and D. Zeng, "Depth dynamic center difference convolutions for monocular 3d object detection," Neurocomputing, vol.520, pp.73–81, 2023.
[9] R.Q. Charles, H. Su, M. Kaichun, and L.J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.652–660, 2017.
[10] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Advances in Neural Information Processing Systems, vol.30, 2017.
[11] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Std: Sparse-to-dense 3d object detector for point cloud," Proc. IEEE/CVF International Conference on Computer Vision, pp.1951–1960, 2019.
[12] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.1711–1719, 2020.
[13] S. Qi, X. Ning, G. Yang, L. Zhang, P. Long, W. Cai, and W. Li, "Review of multi-view 3d object recognition methods based on deep learning," Displays, vol.69, p.102053, 2021.
[14] V.A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," 2019 International Conference on Robotics and Automation (ICRA), pp.7276–7282, IEEE, 2019.
[15] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.770–779, 2019.
[16] C.R. Qi, X. Chen, O. Litany, and L.J. Guibas, "Imvotenet: Boosting 3d object detection in point clouds with image votes," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.4404–4413, 2020.
[17] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.11040–11048, 2020.
[18] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4490–4499, 2018.
[19] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.12697–12705, 2019.
[20] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10529–10538, 2020.
[21] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," Proc. AAAI Conference on Artificial Intelligence, vol. 35, no.2, pp.1201–1209, 2021.

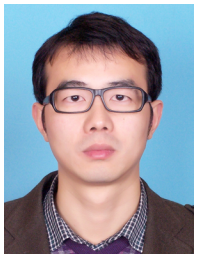
[22] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.9224–9232, 2018.
[23] C. Yan and E. Salman, "Mono3d: Open source cell library for monolithic 3-d integrated circuits," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.65, no.3, pp.1075–1085, 2017.
[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.770–778, 2016.
[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.4510–4520, 2018.
[26] W. Zheng, W. Tang, S. Chen, L. Jiang, and C.-W. Fu, "Cia-ssd: Confident iou-aware single-stage object detector from point cloud," Proc. AAAI Conference on Artificial Intelligence, vol.35, no.4, pp.3555–3562, 2021.
[27] S. Woo, J. Park, J.-Y. Lee, and I.S. Kweon, "Cbam: Convolutional block attention module," Proc. European Conference on Computer Vision (ECCV), pp.3–19, 2018.
[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," Proc. IEEE International Conference on Computer Vision, pp.2980–2988, 2017.
[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," The International Journal of Robotics Research, vol.32, no.11, pp.1231–1237, 2013.
[30] G. Shi, R. Li, and C. Ma, "Pillarnet: Real-time and high-performance pillar-based 3d object detection," Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, Oct. 23–27, 2022, Proceedings, Part X, vol.13670, pp.35–52, Springer, 2022.
[31] C.R. Qi, W. Liu, C. Wu, H. Su, and L.J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.918–927, June 2018.
[32] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "Tanet: Robust 3d object detection from point clouds with triple attention," Proc. AAAI Conference on Artificial Intelligence, vol.34, no.7, pp.11677–11684, 2020.
[33] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," IEEE Trans. Pattern Anal. Mach. Intell., vol.43, no.8, pp.2647–2664, 2020.
[34] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," Sensors, vol.18, no.10, p.3337, 2018.

Jia-ji Jiang received the B.E. degree from Guangxi University, Guangxi, China, in 2021. He is currently studying for a master degree in school of computer, electronics and information at Guangxi University. His research interests include 3D object detection and computer vision.

Hai-bin Wan received the PhD degree in Electronic Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013. He is currently with the College of Computer Science and Electronic Information, Guangxi University, China. His research interests include application research of deep learning and embedded technology in pattern recognition, information system.

Hong-min Sun received the M.S. degree in Computer application technology from Guangxi University, nanning, China, in 2009. Since December 2019, he served as the associate professor at the Guangxi Institute of Industry and Technology, China. His research interests include in big data, computer vision.

Tuan-fa Qin received the PhD degree from Nanjing University, Nanjing, China, in 1997. Since 1991, he has been with the School of Computer, Electronic and Information, Guangxi University, Nanning, China, where he became an associate professor in 1997 and a professor in 2000. He is also the laboratory director of Guangxi Key Laboratory of Multimedia Communications and Network Technology. He has authored more than 200 academic papers and participated in writing 2 monographs. He has obtained 12 authorized Chinese invention patents in the field of communication, and 7 utility model patents, and more than 20 copyrights of computer software registration. He has sponsored over 5 projects of National Natural Science Foundation of China, over 2 National Innovation Fund Projects for small and medium-sized enterprises, and over more than 10 provincial and ministerial projects. His general research interests include wireless body area network and wireless multimedia communication.



Zheng-qiang Wang received his B.S. degree from Southeast University, Nanjing, China, in 2005, and the M.S. degree from Xuzhou Normal University, Xuzhou, China, in 2008, both in applied mathematics, and Ph.D. degree in the Department of Electronic Engineering, Shanghai Jiao Tong University, China, in 2015. Since December 2017, Dr. Wang served as the associate professor at Chongqing University of Posts and Telecommunications. His research interests include 5G, cognitive radio networks, game theory and network optimization.