EURASIP Journal on Advances
in Signal Processing

RESEARCH Open Access

EFMF-pillars: 3D object detection based on enhanced features and multi-scale fusion

Wenbiao Zhang[1], Gang Chen[2,3]*, Hongyan Wang[1], Lina Yang[2] and Tao Sun[1]

*Correspondence:
gchenchine@foxmail.com

[1] College of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China
[2] College of Information Science and Engineering, Jiaxing University, Jiaxing 314001, China
[3] Jiaxing Soy Intelligent Co. Ltd., Jiaxing, China

Abstract

As unmanned vehicle technology advances rapidly, obstacle recognition and target detection are crucial links, which directly affect the driving safety and efficiency of unmanned vehicles. In response to the inaccurate localization of small targets such as pedestrians in current object detection tasks and the problem of losing local features in the PointPillars, this paper proposes a three-dimensional object detection method based on improved PointPillars. Firstly, addressing the issue of lost spatial and local information in the PointPillars, the feature encoding part of the PointPillars is improved, and a new pillar feature enhancement extraction module, CSM-Module, is proposed. Channel encoding and spatial encoding are introduced in the new pillar feature enhancement extraction module, fully considering the spatial information and local detailed geometric information of each pillar, thereby enhancing the feature representation capability of each pillar. Secondly, based on the fusion of CSPDarknet and SENet, a new backbone network CSE-Net is designed in this paper, enabling the extraction of rich contextual semantic information and multi-scale global features, thereby enhancing the feature extraction capability. Our method achieves higher detection accuracy when validated on the KITTI dataset. Compared to the original network, the improved algorithm's average detection accuracy is increased by 3.42%, it shows that the method is reasonable and valuable.

Keywords: 3D object detection, PointPillars, CSPDarknet, SENet

1 Introduction

The rapid advancements in unmanned systems and robotic perception, 3D object detection has also made long-term progress. 3D object detection technology stands as a pivotal technique within the realm of computer vision, focusing on identifying and locating objects in a three-dimensional scene, to enable autonomous perception and decision-making in various domains such as obstacle avoidance for intelligent robots [1, 2], unmanned vehicle navigation, and others.

Compared to images, point cloud contain spatial information of the scene and are less affected by lighting conditions, with a wider perception range, thus possessing a natural advantage in three-dimensional perception tasks. The data collected by LiDAR can provide rich 3D geometry and scale information of the road environment, which can be used to detect the geometry, distance information, azimuth and traveling speed of vehicles, pedestrians and other road targets in the driving direction [3]. To effectively accomplish point cloud detection tasks, many methods have been proposed successively. In 2017, Qi et al. [4] proposed Pointnet, which can directly learn point cloud features. VoxelNet [5] applied PointNet to object detection and achieved strong performance, greatly advancing the pace of end-to-end learning. In 2018, SECOND [6] dropped 3D convolution, which is good in terms of performance and speed. In 2019, PointPillars [7] deployed new encoders and achieved better results, it voxelizes the scene to extract features and projects it into BEV for object detection using 2D convolution. However, in the feature extraction process, due to the characteristics of voxelization and convolution operations, a large amount of local spatial information of the point cloud is inevitably lost, resulting in insufficiently high detection accuracy for vehicles and pedestrians.

In order to optimize the detection results of pedestrians and other targets, and to solve the problem of information loss in the pillars, the PointPillars model is improved in this paper. The main work of this paper is as follows:

1. To solve the issue of losing spatial and local information in the PointPillars, we propose a new pillar feature extraction module called CSM-Module. In this module, channel encoding and spatial encoding are introduced into the encoding network, fully considering the spatial information and detailed local geometric information of each pillar. This enhancement significantly improves the feature representation capability of each pillar.
2. By modifying the backbone after integrating CSPDarknet and SENet, the network is able to capture rich contextual semantic information and multilevel features more effectively, resulting in stronger expressive and context-aware capabilities.
3. The proposed method has been evaluated using KITTI and the results show that the method has good detection accuracy and running speed.

2 Related work

2.1 Enhanced features and multi-scale fusion

Multi-scale Fusion and Feature Enhancement are commonly used techniques in computer vision and machine learning. Multi-scale Fusion refers to the simultaneous consideration of information at different scales when processing data. This approach helps the algorithm to capture objects of different sizes and features at multiple levels, thus improving the accuracy and robustness of recognition. Method [8] aligned feature maps according to the scale through ROI, and features at different scales can be fused effectively. However, ROI is utilized to process and store feature maps at multiple viewpoints and resolutions, which can significantly increase the amount of computation. M3DETR [9] used the multi-scale Transformer to process point clouds to establish relationships between different points and enhance the understanding of complex scenes. However, when processing high-resolution data, the Transformer's self-attention mechanism is extremely computationally complex, significantly increasing computation time for each layer of features and affecting real-time performance. Method [10] used selective kernels so that the most appropriate convolutional kernel size can be dynamically selected based on the characteristics of the input data. However, since the selective kernel contains multiple convolutional kernel branches of different sizes, the number of parameters of the model increases significantly, especially in larger networks or deep network structures. Voxel-FPN [11] considered that point clouds are not uniformly dense in 3D space, and a single-sized voxel grid may not adequately represent all the information in the scene, so voxelized point clouds from multiple scales to capture more local structural information. However, multi-scale voxelization is sensitive to the density and distribution of the point cloud data and does not work well with sparse or unevenly distributed data.

Point cloud feature enhancement techniques are used to process 3D point cloud data through a series of algorithms and deep learning methods to extract and enhance its geometric and semantic features. These techniques include denoising, down-sampling, feature extraction, alignment, data enhancement, etc., and are widely used in the fields of autonomous driving, robot navigation, 3D modeling, and virtual reality to enhance the recognition, classification, and segmentation capabilities of the model, and thus improve the robustness and accuracy of the system [16, 17]. CL3D [12] computed parallax maps from stereo images to generate pseudo-lidar points, and fused the generated pseudo-lidar points with the original lidar points to enhance the point cloud features. However, this relies heavily on the pseudo-lidar point cloud, which may affect the quality of the final fused features if there are errors in the depth estimation or coordinate transformation process. EPNet [13] proposed to use the LI-Fusion layer to establish correlations between point cloud data and camera images and adaptively estimate image semantic features. But the semantic information that depends on the camera image can be affected by lighting variations, occlusion, or other factors. SIENet [14] designed the spatial information enhancement module for predicting the shape of the front sights within a candidate box. However, if the initial prediction of the candidate box is not accurate enough, it may affect the subsequent shape prediction and structural information extraction. Small-track [15] augmented the response of the target region by the GEM module, which utilizes a graph neural network to update the relationship between nodes and edges, and implements pixel-level and irregular weight modulation to refine the classification response. However, the robustness of the GEM module may be affected if the input image contains noise or interference information.

Our approach enhances features by extracting weighted spatial features and channel features for each pillar. Compared to other methods, CSM-Module improves the model's representation of features by greatly reducing computational overhead through simple average and maximum pooling operations, adaptively adjusting the weights, and enabling the model to more accurately focus on discriminative features. In the backbone network, splitting and merging the feature graphs through cross-level connections reduce the duplicate computations in the network, which reduces the computation of the model with less number of parameters and faster inference speed, which is more satisfying for real-time; at the same time, it increases the depth and complexity of the network, and obtains the features at different levels; with the addition of the SENet, the model can focus on the useful features, and inhibit the features that have a smaller contribution to the task small or even irrelevant features.

2.2 Object detection methods

According to the different ways of feature extraction, the relevant algorithms can be categorized into three main groups: point-based, voxel-based and point-and-voxel-based methods.

2.2.1 Point-based methods

Point clouds are irregular data, and converting them into regular 3D voxels or 2D views can make the data unnecessarily verbose and mask the natural invariance of the original data. For this reason, a series of approaches have emerged in recent years, including PointNet [4] and PointNet++[18], have been proposed in recent years to directly obtain spatial geometric features from the original points, and then classify and locate interested targets according to the extracted features. PointRCNN [19] first selected some possible targets from the data, and then classifies and optimizes the targets to get the final result. STD [20] got the anchor first and then produced the bounding box, which is more efficient. 3D-SSD [21] adopted a feature distance-based farthest point sampling method, then to captured the semantic message and excluded irrelevant points. By introducing point-level semantic information, SASA [22] avoided the semantic enhancement module from selecting too many background points. BtcDet [23] found that the performance could be improved by completing the occlusion missing, by predicting the shape share of RoI, BtcDet integrated it into the point cloud features and then carried out target detection. SPG [24] first generated semantic point sets, and combined semantic point sets with original point, finally used a detector to obtain detection results. LiDAR-RCNN [25] addressed the dimensional ambiguity problem by solving it and proposing an effective solution. PointFormer [26] effectively applied the Transformer model to the point cloud by refining the point cloud with a self-attention map.

2.2.2 Voxel-based methods

Converting point cloud data into voxels is a common method to deal with irregular point cloud data. 3DFCN [27] extended the detection technology based on 2D full convolutional network to the 3D spatial domain for 3D target detection. Vote3Deep [28] algorithm used a central-point symmetric voting mechanism, to deal with sparse input point clouds. VoxelNet [5] divided the space into small squares and extracted relevant information from each square with data, but it consumes huge resources. SECOND [6] used sparsely embedded convolution to achieve efficient object detection. Voxel R-CNN [29] aggregated 3D information on 3D features and used coarse-grained voxels to complete high-precision detection. PointPillars [7] sliced the space into pillars one by one, and did not deal with empty pillars, which improved its operation efficiency, but the local feature information is lost and the feature extraction capability is weak. HVNet [30] taken into account that different voxel proportion divisions will affect the detection accuracy and calculation time. TANet [31] used stacked attention modules to process each voxel separately, enhanced the key information of the target while suppressing unstable points. VoTr [32] used the self-attention mechanism to extract and aggregate features for each point, applicable to complex scenarios. PDV [33] worked effectively by finding the key points in space and then matching the relevant features. Focals-Conv

[34] achieved better performance by introducing the concept of learnable sparsity. TED [35] proposed novel TeBEV pooling and TiVoxel pooling modules, which are designed to help efficiently learn isotropic features from point clouds. SST [36] improved the detection accuracy by effectively combining the Transformer architecture and sparse convolution.

2.2.3 Point-and-voxel-based

PV-RCNN [37] considered not only the characteristics of the original points, but also the characteristics of the voxels, and performed the detection task based on them. Deformable PV-RCNN [38] paid more attention to the features of scattered points, so it performed better for distant objects in the scene. SA-SSD [39] added auxiliary network to the main network, fused point features under different resolutions. CIA-SSD [40] calibrated the two tasks of classification and location in single-step target detection. SE-SSD [41] is trained with flexible soft objectives and deterministic hard objectives without additional calculations into the reasoning. HVPR [42] extracted the features of voxels and points, respectively. For each voxel feature, the point features are aggregated according to its similarity, and a mixed representation of voxels and points is obtained. Fast Point R-CNN [43] voxelized the point cloud, and integrated the coordinate information and the corresponding features after convolution of each point to achieve the effect of preserving the context information and the accurate coordinate position. PV-RCNN+ +[44] proposed new sampling and aggregation strategies to achieve high efficiency and high accuracy detection performance. PVT-SSD [45] captured features by sampling voxels and points around a reference point and utilizing their positions and features through a Transformer block. SAT-GCN [46] proposed a new module, SAT-GCN, to enhance weak semantic information for 3D target detection.

Through the analysis of different algorithms, it is observed that methods utilizing pillars consume fewer computational resources and exhibit faster runtime compared to various other algorithms. However, these pillar-based methods suffer from the issue of critical feature loss, which diminishes the effectiveness of detection tasks due to the loss of crucial local information. This paper proposes a new approach based on PointPillars is proposed to effectively enhance the network's feature extraction capability and reduce feature loss. This approach aims to preserve deep-level information, and it achieves excellent results in detection tasks.

3 Method

In this section, we will elaborate on the model proposed in this paper. Our model is built upon the PointPillars framework, using it as a baseline and modifying its architecture to propose the model presented in this paper. When PointPillars maps the point into a BEV, it suppresses the spatial information of points within pillars, local detail features are lost during the max-pooling process. In consideration of this problem, a new pillar coding module (CSM-Module) is proposed in this paper. In the CSM-Module, it can better capture channel and spatial information in the input feature map, thus focusing more on extracting spatial information and local features from the point cloud [49]. Additionally, to address the insufficient feature extraction capability of the original Backbone network, we integrate CSPDarknet [47] and SENet

[48] to modify the backbone network, named CSE-Net. This modification allows the network to partition and connect features at different levels and adaptively adjust the importance of each channel, thereby enhancing the focus on critical features. This helps the network better capture features at different scales and contexts, improving the extraction capability of important information in the point cloud and enhancing its expressive power and context awareness. Our model is constructed from three components: the pillar feature net, the Backbone, and the detection head. The architecture of this paper is shown in Fig. 1.

3.1 Pillar feature net

PointPillars for extracting the features of each pillar is maximum pooling to represent the features of the pillar. However, this results in the loss of detailed spatial features that is extremely important for pillar-based detection, especially for smaller targets such as pedestrians. So, we propose a new pillar coding module named CSM-Module, which takes into account the local spatial information and channel information of each pillar, and integrates local features with global features. The input to this module is the original point containing information include point coordinates and reflection intensity. P indicates the number of pillars, N is the number of points in pillars, and C indicates the feature dimension. In the point cloud coding module, the information of column center and range is first used to enhance the original point, and then the enhanced point features are mapped to high-dimensional features by MLP (multilayer perceptron). In the maximum pooling coding module, perform maximum pooling operations on point features in each pillar to obtain the maximum pooling feature. In the channel coding unit, the channel features are obtained by coding the points in each pillar. In the spatial coding module, the spatial features are obtained by weighted summation of the point cloud features. Then the pillar feature can be obtained by averaging the maximum pooling feature, channel feature and space feature. Finally, all pillar features are combined and stacked according to the position of the original pillar to form a pseudo-image. Image range is (C, H, W). As observed in Fig. 2, CSM-Module has four units: (1) channel coding unit; (2) Spatial coding unit; (3) Maximum pooling coding unit; 4) Feature fusion.

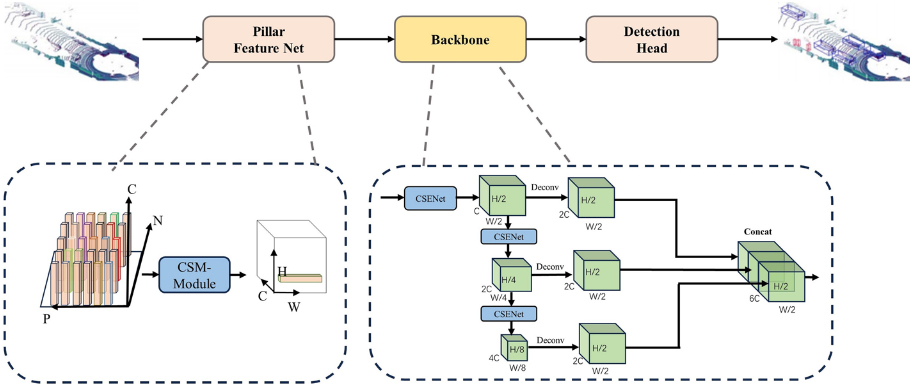


Fig. 1 Network structure of the improved PointPillars

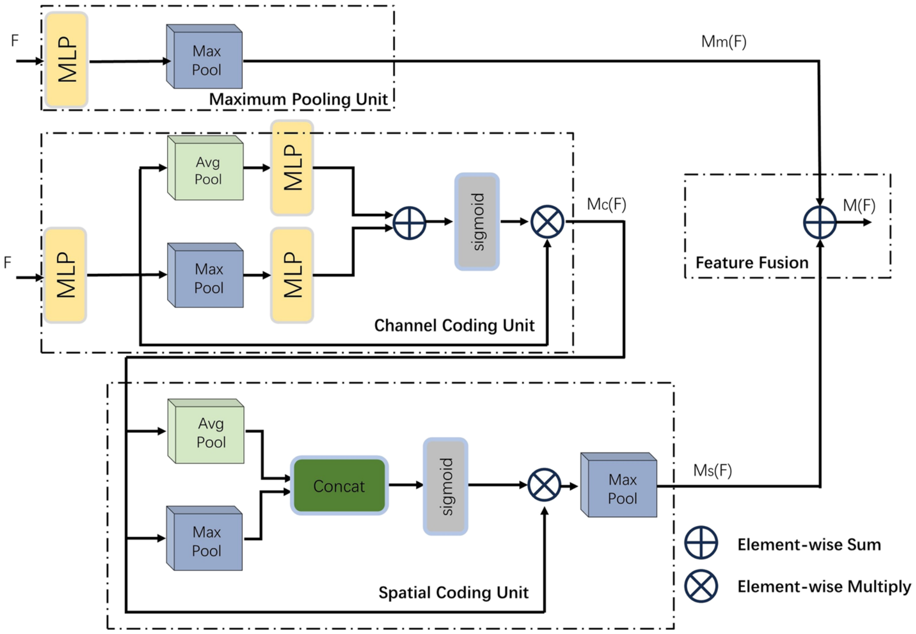


Fig. 2 Network structure of CSM-Module

3.1.1 Channel coding unit

The channel coding unit first performs two pooling operations. Then, the weights of each channel are learned, through the full connection layer and activation function, then the learned weights are normalized, and the learned weights are applied to the original input feature map, and the final output feature map is generated by weighting each channel. The formula is as follows:

$$\mathrm{Mc}\,(F) = \sigma\left(\mathrm{MLP}\left(\mathrm{Avgpool}\,(F)\right) + \mathrm{MLP}(\mathrm{Maxpool}\,(F))\right) \tag{1}$$

where F is the input feature, MLP is the multi-layer perceptron, Avgpool is the average pooling, Maxpool is the maximum pooling, σ is the sigmoid function, and $M_c(F)$ is the obtained channel feature.

The channel coding unit helps the network focus more on channels that are more important to the task by capturing information between channels and then using that information to assign different levels of importance to each channel.

3.1.2 Spatial coding unit

The spatial coding unit adaptively adjusts the attention to each location by learning the weight on the spatial location. First, the input feature map is pooled, the results are mapped to an attention weight map through the convolution layer. Finally, the learned spatial attention weights are applied to the input feature map by weighting each spatial position. Through global information pooling and adaptive spatial weight learning, this series of operations emphasizes the attention to different locations. The formula is as follows:

$$Ms\left(F\right)=\sigma\left(f^{3\times 3}\left(\left[\text{Avgpool}(M_c\left(F\right));\ \text{Maxpool}(M_c\left(F\right))\right]\right)\right) \tag{2}$$

where $f^{3\times 3}$ represents the convolution layer and $Ms(F)$ is the resulting spatial feature.

The spatial coding unit captures spatial information and then assigns different coefficients to each location in the feature map, and the network can focus on the more important areas of the feature map.

3.1.3 Maximum pooling unit

The maximum pooling unit uses maximum pooling to get the feature of the pillars, that is, the point with the most significant feature among all the points is used to represent the pillar. The formula is as follows:

$$Mm\left(F\right)=\text{Maxpool}\left(F\right) \tag{3}$$

$Mm(F)$ is the eigenvector after maximum pooling of each pillar. With the maximum pooling, the model gets the best features in each pillar.

3.1.4 Feature fusion

The final features are obtained by averaging the learned channel features, spatial features and maximum features. The formula is as follows:

$$M\left(F\right)=\frac{Mc\left(F\right)+Ms\left(F\right)}{2} \tag{4}$$

$M(F)$ is the final pillar feature. The feature dimension is C, including the global perception information and local perception information in the pillar. After feature fusion, the algorithm not only gets the most obvious features in each pillar, but also makes it easier to notice important channels and more important areas of the feature map.

3.2 Backbone

CSPDarknet is currently a feature extractor with strong feature extraction capabilities. SENet (Squeeze-and-Excitation Network) is an attentional mechanism to enhance the network's focus on important features and suppress noise. In this paper, the fusion of CSPDarknet and SENet is used as the backbone, and it is named CSENet to obtain richer feature information and improve detection accuracy.

3.2.1 CSPDarknet

CSPDarknet is based on Darknet. The input is split into two parts and then merged through a cross-stage hierarchy structure. Initially, the input is divided into two groups along the channel dimension. One remains unchanged to retain the original features, while the other undergoes processing through Dark block basic unit and 1×1 convolutions. Finally, the feature maps from both groups are concatenated using the concatenate operation to keep the channel numbers unchanged before and after fusion, followed by processing through a 1×1 convolution. Experimental results demonstrate that this segmentation and merging strategy aids in better gradient propagation, enhancing network performance. Additionally, the concatenation reuse of different feature layers in

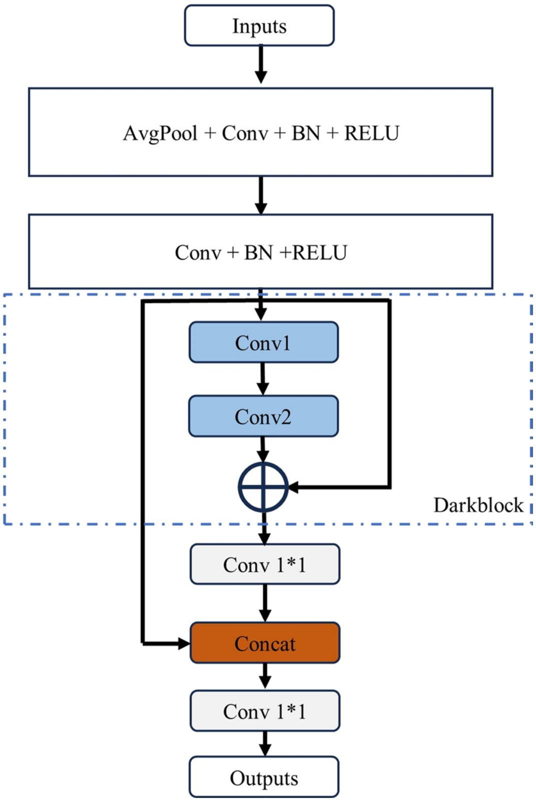


Fig. 3 Network structure of CSPDarknet

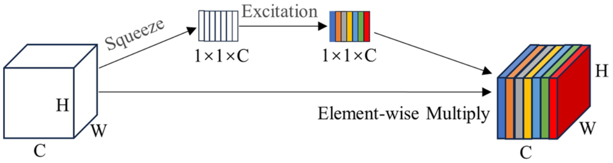


Fig. 4 Network structure of SENet

the CSPDarknet network improves the model's generalization to targets. The network structure is illustrated in Fig. 3.

3.2.2 SENet

The SENet [48] (Squeeze-and-Excitation Net) module introduced a channel attention mechanism designed to adaptively adjust the importance between features by learning weights. SENet is designed to improve the sensitivity of the key features and enhance the representation by adaptively highlighting the channels that are important to the task. Its structure is shown in Fig. 4.

3.2.3 Improved backbone

The backbone network of PointPillars uses a simple convolutional that not extract sufficient features. And some important local features and context information are ignored, it has a great impact on the recognition task.

In this paper, CSPDarknet and SENet are integrated to rebuild the two-dimensional convolutional down sampling module in the backbone network, which enables the network feature extraction stage to extract rich contextual information and global features. The network adaptively adjusts the weight of each channel. Thus, the feature extraction capability of the algorithm is enhanced. The backbone designed in this paper is shown in Fig. 5.

3.3 Detecting head

The detection header adopts SSD [51] for target detection. SSD is a typical single-step detection algorithm, and the idea of anchor is introduced into the network, which can adapt to multi-scale target detection tasks and is more in line with the characteristics of large-scale transformation of point cloud data.

3.4 Loss function

We refer to SECOND [6] to set up the loss function, use $\left(x, y, z, w, l, h, \theta\right)$ to represent the target box, where x, y, z is the center point of the target box, w, l, h is the width, length, and height of the target box, and θ the direction of the target. The linear regression residuals are

$$\begin{cases} \Delta x = \dfrac{x^{gt} - x^a}{d^a}, \Delta y = \dfrac{y^{gt} - y^a}{d^a}, \Delta z = \dfrac{z^{gt} - z^a}{d^a}, \\ \Delta w = \log \dfrac{w^{gt}}{w^a}, \Delta l = \log \dfrac{l^{gt}}{l^a}, \Delta h = \dfrac{h^{gt}}{h^a}, \\ \Delta \theta = \sin\left(\theta^{gt} - \theta^a\right), d^a = \sqrt{(w^a)^2 + (l^a)^2}, \end{cases} \tag{5}$$

This paper uses the Smooth$L1$ function for training and L_{dir} to learn the direction of the target box, so the regression loss is

$$L_{\mathrm{loc}} = \sum_{b \in (x,y,z,w,l,h,\theta)} \mathrm{Smooth}L1(\Delta b) \tag{6}$$

Classified losses are

$$L_{\mathrm{cls}} = -\alpha_a \left(1 - p^a\right)^{\gamma} \log p^a \tag{7}$$

Among them, p^a represents the probability size of the prediction box, $\alpha = 0.25,\gamma = 2$. The total loss is

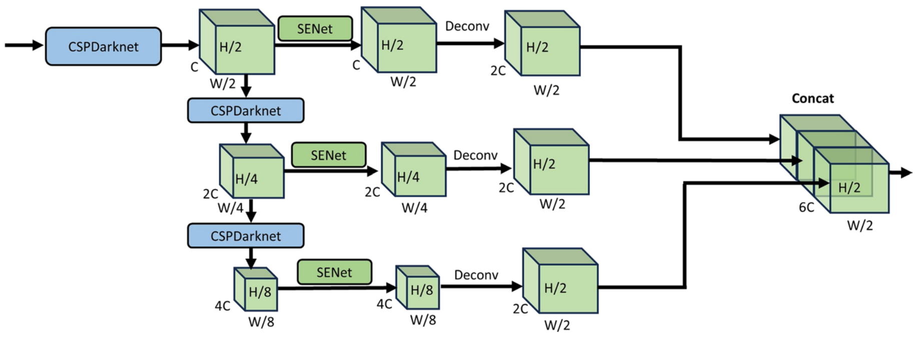


Fig. 5 Network structure of backbone

$$L = \frac{1}{N_{\text{pos}}}(\beta_{\text{loc}}L_{\text{loc}} + \beta_{\text{cls}}L_{\text{cls}} + \beta_{\text{dir}}L_{\text{dir}}) \tag{8}$$

N_{pos} is the number of candidate boxes.

4 Experiment and analysis

4.1 Dataset

The experiments in this paper were conducted on the large-scale public dataset KITTI. The KITTI dataset [50] is a comprehensive dataset used for research in autonomous driving and computer vision. It comprises 7481 training samples and 7518 test samples, and the data set included cars, pedestrians and cyclists.

4.2 Experimental setup

The hardware environment used in this paper is: NVIDIA GeForce RTX 3060 12G GPU; Intel i5 12th CPU 16G; 512G hard disk; The software environment is Ubuntu 18.04 LTS, Python3.9, Cuda11.1, and Pytorch1.8.1. In this paper, the batch size is 4, weight value was set to 0.01, momentum value to 0.9, learning rate value to 0.003, and epoch is 200.

Data augmentation improves the network's generalization ability and detection performance by increasing data samples. Firstly, create a 3D bounding box lookup table based on real targets and point clouds within the bounding boxes for all targets. For each sample, add 15 cars, 15 pedestrians, and 15 cyclists to the current data to participate in network training. Next, add the bounding boxes of the real targets one by one, and perform rotation and translation operations on each bounding box. The coordinates of the boxes are transformed according to the normal distribution of $N(0, 0.25)$. The model achieves the increase in training samples through the above operations.

4.3 Analysis of experimental results

4.3.1 Comparison studies

All experiments were carried out on KITTI dataset, and the test results showed the detection accuracy of Car, Pedestrian and Cyclist categories. We use the detection accuracy as the evaluation index and 3D-IOU and BEV-IOU to judge whether the algorithm detects the target.

The proposed algorithm is tested against other 3D target detection related algorithms on KITTI dataset. The comparison algorithms include VoxelNet [5], SECOND [6], PointPillars [7], PointRCNN [19], PFF3D [52], PillarNet [53] and EOTL [54]. Tables 1 and 2 show the detection accuracy of the proposed algorithm with other algorithms in KITTI dataset, the bolded part of the table shows the best results of all the algorithms in different evaluation metrics.

As shown in Table 1, in 3D mode, the proposed algorithm is superior to the baseline algorithm. In the three kinds of difficulty of the car, it has increased by 4.23%, 1.60% and 5.54%, respectively. The pedestrian was increased by 1.71%, 3.48% and 4.63%, respectively, and the cyclist was increased by 5.77%, 0.55% and 3.24%, respectively. And our method also achieves better detection accuracy compared with other algorithms.

Table 1 Comparison of 3D AP of different methods

Methods	Car			Pedestrian			Cyclist		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
VoxelNet	81.97	65.46	62.85	39.48	33.69	31.50	67.17	47.65	45.11
SECOND	83.13	73.66	66.20	52.05	47.07	43.40	53.85	46.90	56.69
PointPillars	83.75	75.68	70.03	53.55	47.65	42.57	78.82	65.51	59.67
PointRCNN	88.26	76.27	71.64	54.50	48.97	42.89	82.05	65.79	60.07
PFF3D	88.45	77.16	75.26	55.32	49.78	46.92	75.87	62.37	59.68
PillarNet	87.69	**77.81**	74.60	51.58	46.80	44.39	80.75	62.50	58.83
EOTL	**88.75**	76.73	75.31	**57.07**	50.07	45.44	**84.61**	65.55	58.61
Ours	87.98	77.28	**75.57**	55.26	**51.13**	**47.20**	84.59	**66.06**	**62.91**

Table 2 Comparison of BEV AP of different methods

Methods	Car			Pedestrian			Cyclist		
	Easy	Mod	Hard	Easy	Mod	Hard	Easy	Mod	Hard
VoxelNet	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55
SECOND	88.07	79.37	77.95	55.10	46.27	44.76	73.67	56.04	48.78
PointPillars	89.17	85.58	82.06	59.12	53.45	49.28	80.79	66.01	62.02
PointRCNN	91.14	87.62	**86.05**	62.82	**58.18**	51.46	84.51	69.11	64.15
PFF3D	89.59	87.17	85.42	64.86	57.23	51.47	85.95	69.26	63.44
PillarNet	89.70	86.96	84.60	58.91	53.48	50.90	83.42	67.01	62.86
EOTL	**91.63**	88.19	81.46	**66.24**	58.11	**52.09**	85.25	**72.31**	62.76
Ours	90.65	**88.31**	85.79	64.05	57.69	51.68	**89.54**	71.15	**65.72**

Table 3 Inference speed and param comparison among different methods

Methods	Publish year	Param(M)	FPS	Reasoning speed (1/FPS) (s)
VoxelNet	2018	8.35	18	0.0557
SECOND	2018	5.33	27	0.0372
PointPillars	2019	4.83	35	0.0286
PointRCNN	2019	4.04	11	0.0916
PFF3D	2021	8.67	18	0.0562
PillarNet	2022	10.99	16	0.0625
EOTL	2023	7.59	19	0.0513
Ours		**1.89**	**37**	**0.0272**

From Table 2, in BEV mode, in the three kinds of difficulty of car has increased by 1.48%, 2.73% and 3.73%, respectively. The pedestrian was increased by 4.93%, 4.24% and 2.40%, respectively, and the cyclist was increased by 8.75%, 5.14% and 3.70%, respectively.

As shown in Table 3, the bolded parts of the table are the fastest inference and the least number of parameters of all the algorithms, in terms of running speed, our method can process 37 frames per second of the point cloud. Our method achieves the best results among all the compared algorithms in terms of running speed and number of parameters while maintaining the accuracy. The 64-line LiDAR of the KITTI data acquisition device operates at 10 Hz, the inference speed of our algorithm is 0.0272 s, which ensures real-time performance.

4.3.2 Ablation studies

In this paper, the effects of the two modules on the baseline network are verified by ablation experiments. CSM-Module is a pillar feature enhancement module that introduces channel coding and spatial coding, and CSENet is a backbone network module that integrates CSPDarknet and SENet.

Table 4 shows the role of each module. The results were evaluated with mean Average Precision (mAP) across all detection difficulty levels for the three types, the baseline is PointPillars, where CCU means channel coding unit and SCU is spatial coding unit.

With the addition of channel coding unit in the pillar feature enhancement module, the mAP increases from 76.48% to 77.65% for Car, 47.92% to 49.28% for Pedestrian, and 68.00% to 70.08% for Cyclist, which suggests that the channel coding unit helps the network focus on the more important channels by capturing the information between the channels, and then utilizing this information to assign each channel a different level of importance, thus helping the network to focus more on the channels that are more important to the task. And after adding the spatial coding unit, the mAP increases to 78.67% for Car, 50.96% for Pedestrian, and 70.84% for Cyclist, which indicates that the spatial coding unit captures spatial information and then assigns different coefficients to each location in the feature map so that the network can focus on more important areas in the feature map. The experiments demonstrate that the CSM-Module proposed in this paper effectively combines features extracted from max-pooling, spatial encoding, and channel encoding, thereby preserving richer fine-grained information and effectively improving detection accuracy.

After adding CSPDarknet in backbone, the mAP of Car increased from 76.48% to 78.65%, the mAP of Pedestrian increased from 47.92% to 48.38%, and the mAP of Cyclist increased from 68.00% to 70.39%, which suggests that the cross-phase operation of CSPDarknet helps to reduce the redundant gradient information and enhance the gradient flow and learn features at different levels, thus improving the accuracy and efficiency of the network. In addition, after adding SENet into the network, the mAP of Car increases to 79.89%, the mAP of Pedestrian increases to 50.58%, and the mAP of Cyclists increases

Table 4 3D mAP detection results of different modules

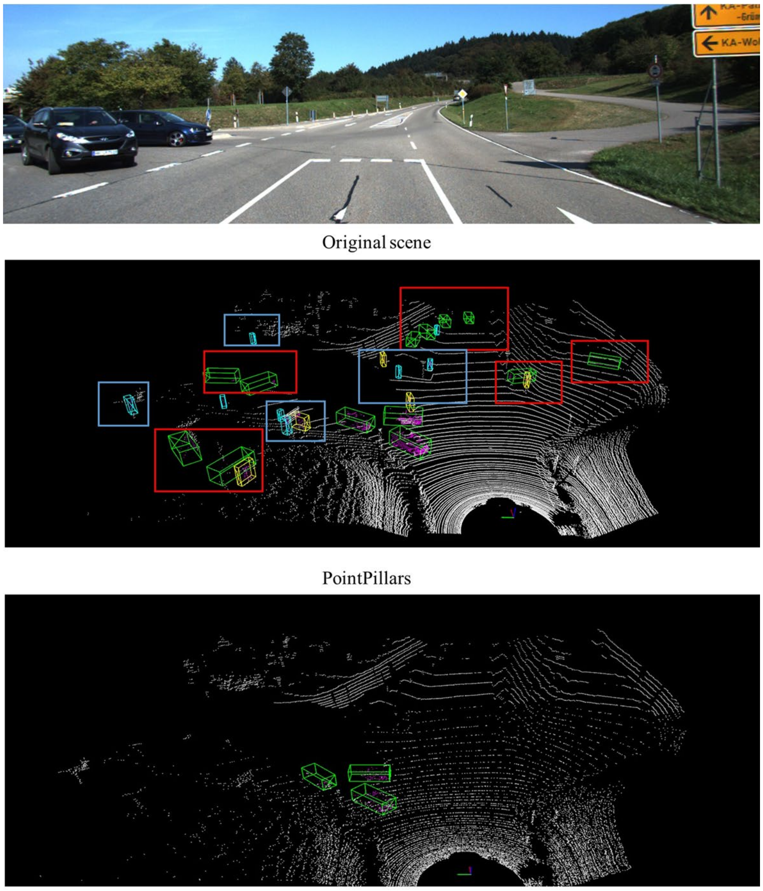
CSM-Module		CSENet		Car	Pedestrian	Cyclist
CCU	SCU	CSPDarknet	SENet			
×	×	×	×	76.48	47.92	68.00
√	×	×	×	77.65	49.28	70.08
×	√	×	×	78.17	48.97	70.68
√	√	×	×	78.67	50.96	70.84
×	×	√	×	78.65	48.38	70.39
×	×	×	√	78.96	49.75	70.41
×	×	√	√	79.89	50.58	70.62
√	√	√	√	80.28	51.20	71.19

to 70.62%, which indicates that SENet can acquire global contextual information and weight the original feature maps to enhance important features and suppress irrelevant features. The experiments demonstrate that adding CSENet to the backbone network enhances the extraction capability of important information in the point cloud, resulting in stronger expressive power and context awareness.

When adding both the CSM-Module and the CSENet to the network, the mAP for Car increased by 3.80%, for Pedestrian increased by 3.28%, and for Cyclist increased by 3.19%. This indicates that with the addition of channel coding and spatial coding, the algorithm fully considers the spatial information and detailed local geometric information of each pillar, which greatly enriches the features of each pillar. The improved backbone is able to capture rich contextual semantic information and multilevel features more effectively, with stronger expressive and context-aware capabilities.

4.4 Visual result analysis

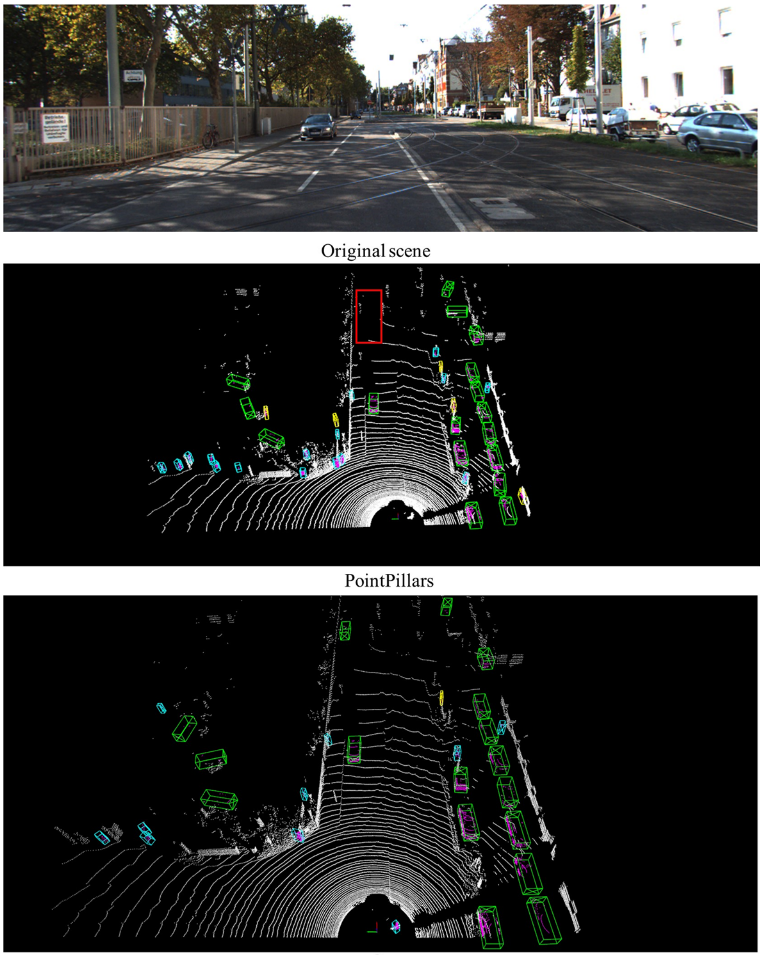
The comparison of detection effects of the proposed algorithm and PointPillars in different scenarios of KITTI dataset is shown in Figs. 6 and 7. The top half of each figure is divided into the corresponding camera image under the real scene, and the bottom


Original scene

PointPillars

Ours

Fig. 6 Comparison of detection effects in scene one



Original scene

PointPillars

Ours

Fig. 7 Comparison of detection effects in scene two

half is the 3D object detection effect diagram of PointPillars and the algorithm in this paper under the point cloud scene.

From Fig. 6, PointPillars exhibits numerous false positives in detecting small objects in complex scenes. Misclassifications of pedestrians and cyclists are highlighted with blue bounding boxes, while false detections of cars are indicated by the red border. From the blue bounding boxes: The PointPillars algorithm erroneously detects objects such as roadside railings, tree branches, and traffic signs as pedestrians or cyclists. Roadside railings, tree branches, and some small streetlights are easily misidentified as pedestrians. Continuous and closely spaced railings are often misclassified as cyclists. From the red bounding boxes: There are also partial false detections in the detection of distant cars. The PointPillars mistakenly identifies square-shaped objects as cars in distant object detection. Our algorithm effectively mitigates the issue of excessive false detections of small objects encountered in the PointPillars algorithm, resulting in improved visualization performance.

In Fig. 7, cars that were missed by the detection are indicated by red bounding boxes. Due to occlusion from car in the front, the PointPillars successfully identifies only the foremost car, resulting in missed detections for car behind. Our algorithm successfully detects the car behind, thereby avoiding missed detections caused by occlusion issues.

5 Conclusions

This paper proposes a method for 3D object detection, aiming to address the challenges of poor performance for small objects such as pedestrians and feature loss in pillar-based algorithms. Firstly, the CSM-Module is introduced to enhance the representation capability of each pillar feature, thereby the algorithm has more powerful capabilities. Secondly, the 2D convolution down sampling module in the backbone network is improved based on CSPDarknet and SENet to enhance the algorithm's feature extraction capability. Experimental results on the KITTI dataset show a significant improvement in detection accuracy for all categories compared to the baseline. In the experiments, our method significantly reduces detection errors and missed detections. In addition, the method can detect 37 frames per second during the inference process, which achieves faster detection speed compared to other methods while maintaining accuracy.

Abbreviations

CNN	Convolution neural network
IOU	Intersection of union
RELU	Rectified linear unit
BN	Batch normalization
SENet	Squeeze and excitation network
CSPDarknet	Cross stage partial Darknet
FPS	Frames per second
AP	Average precision
mAP	Mean average precision
avgpool	Average pooling
maxpool	Maximum pooling
BEV	Bird's eye view

Author contributions

WZ, GC and HW conceived and designed the study. WZ, GC initiated the project, WZ proposed the method, implemented the algorithms, and wrote the paper. WZ, GC conducted experiments, analyzed the data and wrote the paper. LY, TS reviewed and edited the manuscript. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The datasets generated or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare no competing interests.

Received: 29 March 2024 Accepted: 19 September 2024
Published online: 05 October 2024

References

1. Y. Wang, et al., Pseudo-LiDAR from visual depth estimation: bridging the gap in 3D object detection for autonomous driving, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Long Beach, 2019). pp. 8437–8445
2. K, Minemura, H, Liau, A. Monrroy, et al., LMNet: Real-time multiclass object detection on CPU using 3D LiDAR, In: Proceedings of the 2018 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS). (IEEE, Singapore, 2018). PP. 28–34
3. B. Brown, The social life of autonomous cars. Computer **50**(2), 92–96 (2017)

4. R. Charles, H. Su, M. Kaichun, L. Guibas, PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Honolulu, 2017), pp. 77–85
5. Y. Zhou, O. Tuzel, VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Salt Lake City, 2018), pp. 4490–4499
6. Y. Yan, Y. Mao, B. Li, SECOND: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
7. A. Lang, et al., PointPillars: Fast Encoders for Object Detection From Point Clouds, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Long Beach, 2019), pp. 12689–12697
8. R. Guo, D. Li, Y. Han, Deep multi-scale and multi-modal fusion for 3D object detection. Pattern Recogn. Lett. **151**, 236–242 (2021)
9. T. Guan, et al., M3DETR: Multi-representation, Multi-scale, Mutual-relation 3D Object Detection with Transformers, In: Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). (IEEE, Waikoloa, 2022), pp. 2293–2303
10. X. Gao, G. Zhang, Y. Xiong, Multi-scale multi-modal fusion for object detection in autonomous driving based on selective kernel. Measurement **194**, 111001 (2022)
11. H. Kuang et al., Voxel-FPN: multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds. Sensors **20**(3), 704 (2020)
12. C. Lin et al., CL3D: camera-LiDAR 3D object detection with point feature enhancement and point-guided fusion. IEEE Trans. Intell. Transp. Syst. **23**(10), 18040–18050 (2022)
13. T. Huang, et al., Epnet: Enhancing point features with image semantics for 3d object detection, in Proceedings of the European conference on computer vision (ECCV). (Springer, Glasgow, 2018), pp. 35–52
14. Z. Li et al., Spatial information enhancement network for 3D object detection from point cloud. Pattern Recogn. **128**, 108684 (2022)
15. Y. Xue, et al. SmallTrack: Wavelet pooling and graph enhanced classification for UAV small object tracking. IEEE Transactions on Geoscience and Remote Sensing (2023)
16. S. Wen, T. Wang, S. Tao, Hybrid CNN-LSTM architecture for LiDAR point clouds semantic segmentation. IEEE Robot. Automation Lett. **7**(3), 5811–5818 (2022)
17. X. Liu, et al, A multi-sensor fusion with automatic vision-LiDAR calibration based on Factor graph joint optimization for SLAM. IEEE Trans Instrument Measure (2023)
18. Qi, L. Yi, et al., Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv Neural Inform Process Syst 30 (2017)
19. S. Shi, X. Wang, H. Li, PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Long Beach, 2019), pp. 770–779
20. Z. Yang, Y. Sun, S. Liu, X. Shen, J. Jia, STD: Sparse-to-Dense 3D Object Detector for Point Cloud, In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). (IEEE, Seoul, 2019), pp. 1951–1960
21. Z. Yang, Y. Sun, S. Liu, J. Jia, 3DSSD: Point-Based 3D Single Stage Object Detector, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Seattle, 2020), pp. 11037–11045
22. Chen, Chen, et al., Sasa: Semantics-augmented set abstraction for point-based 3d object detection, In: Proceedings of the 36th AAAI Conference on Artificial Intelligence. (AAAI, Vancouver, 2022), pp. 221–229
23. Q. G. Xu, Y. Q. Zhong, U. Neumann, Behind the curtain: Learning occluded shapes for 3D object detection, In: Proceedings of the 36th AAAI Conference on Artificial Intelligence. (AAAI, Vancouver, 2022), pp. 2893–2901
24. Q. Xu, Y. Zhou, W. Wang, C. Qi, D. Anguelov, SPG: Unsupervised Domain Adaptation for 3D Object Detection via Semantic Point Generation, In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). (IEEE, Montreal, 2021), pp. 15426–15436
25. Z. Li, F. Wang, N. Wang, LiDAR R-CNN: An Efficient and Universal 3D Object Detector, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Nashville, 2021), pp. 7542–7551
26. X. Pan, Z. Xia, S. Song, L. Li, G. Huang, 3D Object Detection with Pointformer, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Nashville, 2021), pp. 7459–7468
27. B. Li, 3D fully convolutional network for vehicle detection in point cloud, In: Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). (IEEE, Vancouver, 2017), pp. 1513–1518
28. M. Engelcke, D. Rao, et al., Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks, In: Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA). (IEEE, Singapore, 2017), pp. 1355–1361
29. J. Deng, S. Shi, et al., Voxel R-CNN: Towards high performance voxel-based 3D object detection, In: Proceedings of the 35th AAAI Conference on Artificial Intelligence. (AAAI, British Columbia, 2021), pp. 1201–1209
30. M. Ye, S. Xu and T. Cao, HVNet: Hybrid Voxel Network for LiDAR Based 3D Object Detection, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Seattle, 2020), pp. 1628–1637
31. Z. Li, X. Zhao, et al., TANet: Robust 3D object detection from point clouds with triple attention, In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, (AAAI, New York, 2020). pp. 11677–11684
32. J. Mao, et al., Voxel Transformer for 3D Object Detection, In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). (IEEE, Montreal, 2021), pp. 3144–3153
33. J. K. Hu, T. Kuai, S. Waslander, Point Density-Aware Voxels for LiDAR 3D Object Detection, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, New Orleans, 2022), pp. 8459–8468
34. Y. Chen, Y. Li, X. Zhang, J. Sun, J. Jia, Focal Sparse Convolutional Networks for 3D Object Detection, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, New Orleans, 2022), pp. 5418–5427
35. H. Wu, et al., Transformation-equivariant 3d object detection for autonomous driving, In: Proceedings of the AAAI Conference on Artificial Intelligence. (AAAI, Washington, 2023), pp. 2795–2802
36. L. Fan, et al., Embracing Single Stride 3D Object Detector with Sparse Transformer, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (IEEE, New Orleans, 2022), pp. 8448–8458

37. S. Shi, et al., PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Seattle, 2020), pp. 10526–10535
38. P. Bhattacharyya, K. Czarnecki, Deformable PV-RCNN: Improving 3D object detection with learned deformations (2020). arXiv: 2008.08766
39. C. He, H. Zeng, J. Huang, X. Hua, L. Zhang, Structure Aware Single-Stage 3D Object Detection From Point Cloud, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Seattle, 2020), pp. 11870–11879
40. W. Zheng, W. L. Tang, et al., CIA-SSD: Confident IoU-aware single-stage object detector from point cloud, In: Proceedings of the AAAI Conference on Artificial Intelligence. (AAAI, British Columbia, 2021), pp. 3555–3562
41. W. Zheng, W. Tang, L. Jiang, C. Fu, SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (IEEE, Nashville, 2021). pp. 14489–14498
42. J. Noh, S. Lee, B. Ham, HVPR: Hybrid Voxel-Point Representation for Single-stage 3D Object Detection, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Nashville, 2021), pp. 14600–14609
43. Y. Chen, S. Liu, X. Shen, J. Jia, Fast Point R-CNN, In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). (IEEE, Seoul, 2019), pp. 9774–9783
44. S. Shi et al., PV-RCNN++: point-voxel feature set abstraction with local vector representation for 3D object detection. Int. J. Comput. Vision **131**(2), 531–551 (2023)
45. H. Yang, et al., PVT-SSD: Single-Stage 3D Object Detector with Point-Voxel Transformer, In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (IEEE, Vancouver, 2023), pp. 13476–13487
46. L. Wang et al., SAT-GCN: Self-attention graph convolutional network-based 3D object detection for autonomous driving. Knowl.-Based Syst. **259**, 110080 (2023)
47. A. Bochkovskiy, C. Y. Wang, et al., Yolov4: Optimal speed and accuracy of object detection (2020), arXiv:2004.10934
48. J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. **42**(8), 2011–2023 (2020)
49. S. Woo, J. Park, et al., CBAM: Convolutional block attention module, In: Proceedings of the European conference on computer vision (ECCV). (Springer, Munich, 2018), pp. 3–19
50. Geiger, Andreas, et al., Vision meets robotics: The kitti dataset. The International Journal of Robotics Research 32(11), 1231–1237 (2013)
51. W. Liu, et al., SSD: Single shot multibox detector, In: Proceedings of the European conference on computer vision (ECCV). (Springer, Amsterdam, 2016), pp. 21–37
52. L.H. Wen, K.H. Jo, Fast and accurate 3D object detection for lidar-camera-based autonomous vehicles using one shared voxel-based backbone. IEEE Access **9**, 22080–22089 (2021)
53. G. Shi, R. Li, C. Ma, Pillarnet: Real-time and high-performance pillar-based 3d object detection, In: Proceedings of European Conference on Computer Vision (ECCV). (Springer, Switzerland, 2022), pp. 35–52
54. R. Yang, et al., Efficient online transfer learning for road participants detection in autonomous driving. IEEE Sensors Journal (2023)

Publisher's Note