ITC 1/54
Information Technology and Control
Vol. 54 / No. 1/ 2025
pp. 198-218
DOI 10.5755/j01.itc.54.1.35616

An Efficient Point Cloud Correlation Enhancement RCNN for 3D Object Detection

Received 2023/11/14 | Accepted after revision 2024/05/18

HOW TO CITE: Du, J., Huang, H., Tan, Q., Li, Y., Ding, L., Shuang, F. (2025). An Efficient Point Cloud Correlation Enhancement RCNN for 3D Object Detection. *Information Technology and Control*, 54(1), 198-218. https://doi.org/10.5755/j01.itc.54.1.35616

An Efficient Point Cloud Correlation Enhancement RCNN for 3D Object Detection

Jialong Du
Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, School of Electrical Engineering, Guangxi University, Nanning 530004, China; e-mail: 2212391008@st.gxu.edu.cn

Hanzhang Huang
China Tobacco Guangxi Industrial Co., Ltd., Nanning 530001, China; e-mail: hhz62@outlook.com

Qingji Tan
School of Mechanical Engineering, Guangxi University, Nanning 530004, China; e-mail: tanqingji@st.gxu.edu.cn

Yong Li
Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, School of Electrical Engineering, Guangxi University, Nanning 530004, China;
Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou 550025, China;
e-mail: yongli@gxu.edu.cn

Lu Ding
Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, School of Electrical Engineering, Guangxi University, Nanning 530004, China; e-mail: dinglu@gxu.edu.cn

Feng Shuang
Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, School of Electrical Engineering, Guangxi University, Nanning 530004, China; e-mail: fshuang@gxu.edu.cn

Corresponding author: yongli@gxu.edu.cn

To meet the requirement of 3D object detection task, an efficient point cloud correlation enhancement RCNN(EPCE-RCNN) is proposed. The proposed method reduces the computational complexity and time consumption of the network through a lightweight proposal generation module, and accelerates the generation of the 3D proposal box. Meanwhile, during region of interest feature coding, the relevance among different grid points is enhanced through an efficient self-attention pooling module, so that the limitation that the pooling operation is influenced by the radius of a neighborhood query sphere is addressed. In addition, the combination of an attention mechanism and a feedforward network ensures the nonlinearity of the model, so that the model can perform feature expression better. Thus, the synchronous improvement of the network detection efficiency and the detection precision is realized. On the KITTI dataset, the detection accuracy of three difficulty levels reaches 89.99%, 81.69% and 77.17% respectively. Compared with the baseline Voxel-RCNN, the detection efficiency of EPCE-RCNN is improved by 12%. To verify the generalization and application value of the proposed method, a power equipment dataset with 3D label information is constructed, the 3D label frame information of the YCB dataset is also supplemented. Experiments are carried out on these datasets. In the experimental results of the validation set, the mAP of a mug, gelatin box, single clip, wedge clip and C clip can reach 37.67%, 40.06%, 35.63%, 30.01% and 37.31% respectively. Compared with the baseline, the proposed algorithm has a significant improvement and its generalization has been fully verified.

KEYWORDS: 3-D Object Detection, Lightweight Proposal, Self-Attention, Point Cloud, Autonomous Driving.

1. Introduction

With the rapid development of intelligent driving, intelligent manufacturing and other related fields, 3D object detection with point cloud data is faced with more and more challenging requirements. In the realm of intelligent driving, autonomous vehicles need to identify the surrounding environment timely. Similarly, in intelligent production, 3D object detection technology should provide effective guidance for robots to accurately identify and grasp unknown objects. Existing 3D object detection methods with point cloud can be categorized into Point-based methods [36, 37, 18, 26, 28, 43] and Grid-based methods [42, 27, 44, 35, 51, 17, 6].

The Point-based methods take original point cloud as input directly, utilizing techniques such as multi-layer perceptron [13] and abstract aggregation to achieve high detection accuracy. However, a limitation of Point-based methods is their inability to well capture the relevance of feature space in point cloud.

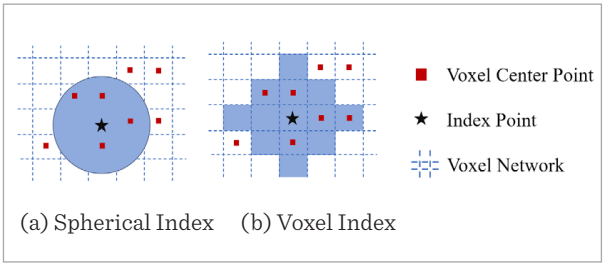
On the other hand, The Grid-based methods partition the point cloud into regular grids, which are more suitable for convolutional neural network feature extraction compared to the irregular structure of original point cloud. Nevertheless, the general Grid-based methods do not recover the 3D structure information after projecting the 3D voxel features to the 2D bird's-eye view, which will cause more information loss.

To solve this problem, Shi et al. proposed Voxel-RCNN [4]. The structure of Voxel-RCNN is consistent with the basic structure of the conventional two-stage detection network, and only has Grid-based branches.

A novel voxel query operator is used to directly extract the Region of Interest (RoI) features from voxel features, which compensates the loss of 3D structural information in the process of projecting the voxel to the bird's-eye view.

The main function of voxel query is to find K and, neighboring voxels within a certain distance range, which is similar to Ball query in PointNet++ [23]. Here, Manhattan distance is used to calculate the extent for the queried voxels: $D_m(\alpha,\beta)=\left|i_\alpha - i_\beta\right|+\left|j_\alpha - j_\beta\right|+\left|k_\alpha - k_\beta\right|$

Figure 1
Schematic diagram of the ball query and voxel query



The process of Voxel-RCNN aggregating the voxel features within the neighborhood can be expressed as: $\eta_i = \max_{k=1,2,\cdots,K}\{\psi([v_i^k - g_i\phi_i^k])\}$, where v_i^k represents the center coordinate of the non-empty voxel, g_i represents the grid point coordinate of the voxel, ϕ_i^k represents the voxel feature vector, Ψ represents the multilayer perceptron, and η_i represents the fused feature vector. In fact, Voxel-RCNN extracts the voxel features in the last two layers of 3D sparse convolution [42], then fuses the voxel features within each layer using two different Manhattan distances. Finally, the voxel features obtained from each layer through different Manhattan distances are fused as the features of RoI (Region of Interest). After extracting and fusing voxel features, the Voxel-RCNN network projects these features to a 2D bird's eye view.

However, the 2D backbone of Voxel-RCNN is stacked by traditional convolution, which has a lot of redundant computation, it directly affects the generation speed of proposal box. In addition, when RoI Pooling is performed, the encoding feature of each RoI grid point by the network is limited to the size of the query sphere, the interdependence relationship among different voxel grid points is also lacking.

To solve the above problems, this paper improves the two-dimensional backbone network and RoI pooling of Voxel-RCNN, proposes EPCE-RCNN (Efficient point cloud correlation enhancement RCNN for 3D object detection).

The main contributions of this paper are as follows:

1 A lightweight proposal generation module (LWPG) is proposed. In this module, few convolution kernels are use when an original feature graph is generated, the feature graph is complemented through cheap linear calculation, the module is built in a bottleneck connection mode, so that the calculation amount is further reduced.

2 This paper proposes an efficient Self-Attention Pooling Module (ESA). By exploiting the dependencies between grid points during ROI pooling, the problem of RoI grid point feature encoding limited by query sphere size is solved, and the key information in the data is captured by dynamically adjusting the weights. Meanwhile, a feed-forward network (FFN) structure is introduced to make the whole module nonlinear and better model representation.

3 To meet the perceptual needs of industrial robots, we reconstructed the YCB dataset with 3D information and built a power component dataset based on the common annotation format of KITTI dataset. Compared with the benchmark algorithm on these datasets, the advantages of the proposed algorithm are proved, it can reach or approach the effect of SOTA in many evaluation indexes.

2. Related Works

2.1. Object Detection Algorithm Based on Voxelization of Point Cloud

The general point-based method has high precision but large feature calculation amount, and Voxel-RCNN [4] is more suitable for feature extraction. In the object detection algorithm based on point cloud voxelization, the proposal of VoxelNet [53] breaks through the bottleneck of manually defining voxel feature information [3, 31, 1, 32]. In view of the defects of large amount of calculation and poor speed, SECOND [42] introduces sparse 3D convolution to replace the 3D convolution layer in VoxelNet, which improves the detection speed and memory utilization. Unlike regular 2D convolution, 3D sparse convolution uses a three-dimensional convolution kernel to perform convolution and pooling operations in three-dimensional space, taking into account the depth, height, and width of the data to enable the network to capture spatial information. Sparse refers to the characteristic of the data that only a small number of points or voxels in the data contain meaningful information, while the majority of the region is empty or devoid of information. 3D Sparse convolution is also widely used because of its high efficiency in processing point cloud data [27, 51]. Pointpillars [17] is an achievement produced in the industrial field, by mapping the point cloud into 2D pseudo-image and then using the 2D objects detection mode to detect 3D targets, its detection speed is far ahead of other schemes at that time, and it is widely used in the industrial field. In particular, PV-RCNN [25] simultaneously adopts the hybrid expression form of point cloud and voxel, reduces the calculation consumption through voxel and sparse convolution, and adds the information obtained from point-based operations in the proposal frame refining

process to retain more accurate position information. Similar to PV-RCNN, HVPR [22] also emphasizes the strengths and avoids the weaknesses of the two types of data. It first extracts voxel-based and point-based features respectively on the two branches, and then carries out weighting and interaction between the two. However, such hybrid methods bring greater computational overhead, which brings challenges to the timeliness of detection. Through the customized multi-scale attention module, a single-stage object detection network is realized. At the same time, some works first assign semantic information to point cloud and then voxelize and detect them, such as Pointpainting [34]. These methods significantly improve the detection accuracy of the algorithm, but the introduction of semantic segmentation brings greater computational overhead. For the real-time requirement of automatic driving task, the balance between detection accuracy and time is of great significance. To improve the efficiency of detection, sparse convolution is applied to the skeleton network of Voxel-RCNN to extract features from the original point cloud. However, there is still redundancy in the calculation of the skeleton network of Voxel-RCNN for 2D data processing, so it is necessary to design a structure to further improve its detection speed.

2.2. Attention Mechanisms

Treisman et al. [21] believe that the attention mechanism strengthens the influence of key parts of input on output by assigning weights to input data. The Attention mechanism has certain been widely used in various fields, a large number of studies have shown that the self-attention mechanism [33] is beneficial to improve the performance of networks, especially convolution modules [33, 45, 38].

To address the challenges inherent to 3D semantic segmentation, such as vast scenes and heterogeneous anisotropic distributions, Qingyong et al. [15] integrated local spatial coding with an attention pooling module to facilitate autonomous learning of key local features. In the field of 3D object perception, Li et al. introduced a graph attention module designed to assign different weights to different feature spaces for different nodes, which is achieved by evaluating the degree of relationship, and the module is applied iteratively across multiple layers to merge features and dynamically adjust node states [19]. When dealing with sparse and disordered point cloud for 3D object detection, Wu and Ogai [41] utilized a self-attention mechanism to amplify salient features while suppressing irrelevant ones. Similarly, Xie et al. [39] designed two attention modules and a feature fusion module to provide comprehensive contextual information at all layers to enhance 3D object detection.

Due to the disorder and irregularity of point cloud, the information acquisition of relevance between points is insufficient for processing point cloud data. The self-attention mechanism can be used to enhance the information acquisition, researchers found that the attention mechanism has also achieved gratifying results in the field of point cloud.

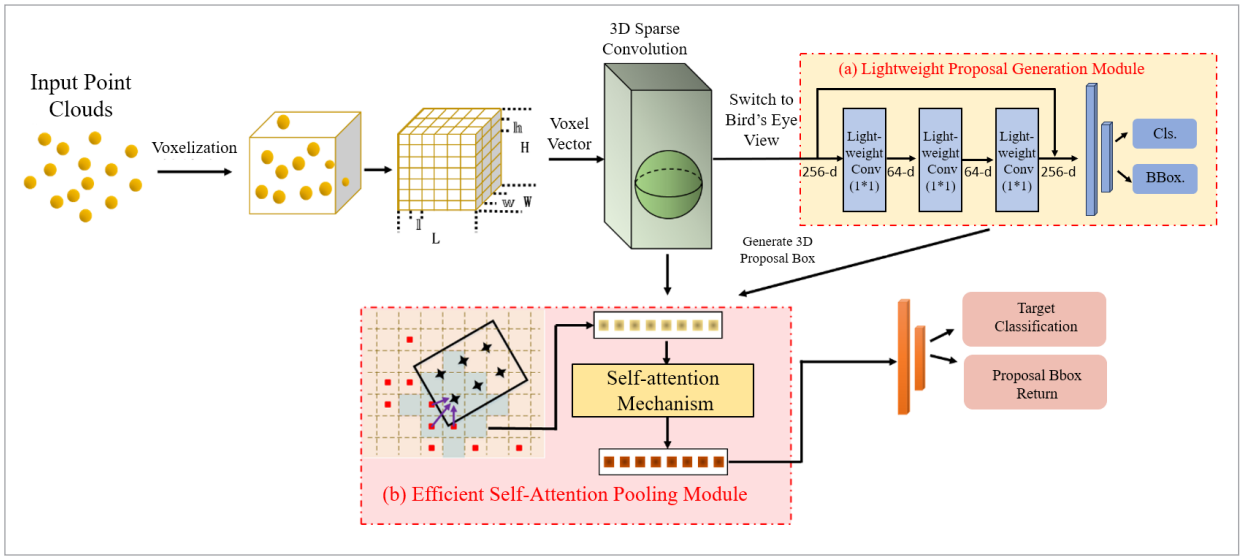
2.3. Acceleration Method of Convolution Layer

In fact, the characteristic images of many channels are highly similar when the traditional convolutional neural network is used for calculation, and more calculation redundancy will be generated if all channels are completely calculated. In view of the computational redundancy in the traditional convolution operation process, MobileNet [14] uses deep separable convolution to approximate the full convolution layer, which achieve the purpose of compressing the number of parameters and reducing the weight. Through point-by-point grouping convolution and channel rearrangement, ShuffleNet [50] greatly reduces the calculation overhead under the condition of ensuring accuracy. GhostNet [9] uses traditional convolution to obtain part of feature maps, then complements the feature maps of other channels through simple linear transformation, which effectively reduces the parameter quantity and calculation complexity without changing the dimension of output feature maps. These strategies are referred to as lightweight convolution modules.

3. EPCE-RCNN for 3D Object Detection

The structure of EPCE-RCNN is shown in Figure 2, which contains two proposed modules, i.e., (a) a lightweight proposal generation module (LWPG) and (b) an efficient self-attention pooling module (ESA). The whole network is a two-stage network based on vox-

Figure 2
EPCE-RCNN network structure. (a) The lightweight proposal generation module (LWPG), (b) The efficient self-attention pooling module (ESA)



els. After the original point cloud is divided into regular cubes in the three-dimensional coordinate system, the average values of all the points in non-empty voxels are taken as the features of the voxel. Then, the voxel features are processed by three-dimensional sparse convolution, and the three-dimensional voxel information is transformed into a two-dimensional aerial view representation. After that, the LWPG efficiently obtains the feature information of the 2D bird's-eye view through accelerated convolutional layers with bottleneck connection structure, and generates 3D proposal box based on these feature information through an anchor frame-based approach. Based on the scope of the 3D proposal box, the ESA directly extracts RoI features from the corresponding 3D voxel features. Overcoming the voxel query operator size limitations and the data complexity, the ESA focus on the connection between grid points and the key information in the features through the redesigned attention mechanism at a low computational cost, finally realizes the refinement of the proposal box based on the RoI features in the second stage of the detection head.

3.1. Voxel Grid Downsampling

Inspired by the work of [42, 53], in order to ensure the lightweight of the model, the point cloud filtering method in the data preprocessing stage will not be used, but the convolution filter will be used to filter the features extracted by the VFE (Voxelwise Feature Extractor) coding layer. Our voxel downsampling method will traverse all the original point cloud and establish the corresponding relationship between each point cloud and voxels. We prepare a hash table before traversing with reference to [53]. If the voxel related to the point does not exist, a voxel is created, and if so, a corresponding relationship is added. At the same time, the maximum number of points in each voxel needs to be specified in advance to avoid excessive computation. Finally, we will significantly reduce the amount of point cloud and obtain voxel information (voxel coordinates, the number of voxel interior points per voxel and the coordinates of the point cloud) to support voxel-level feature extraction in subsequent VFE layers. The specific settings during voxel downsampling will be mentioned in Section 4.2.

3.2. Backbone

After voxel downsampling, we build our Backbone using a design similar to the [42, 53] network.

Voxel-level feature extraction: first, the voxels are processed one by one through the VFE coding layer designed by [53], all the point cloud in the same voxel are taken as inputs, and the point cloud-level features

are obtained point by point through the fully connected network, then the aggregate features of each voxel are obtained by using MaxPooling layer [42] for aggregation features. Finally, the aggregation features are spliced with the point cloud level features. After all, the global point cloud will be transformed into the global voxel-level features of the grid format. In addition, all voxels share the same fully connected network to reduce the number of parameters.

Efficient sparse convolution filter: the voxel features extracted by VFE will be further processed by the sparse convolution structure proposed in [42]. First of all, sparse convolution structure designs an efficient rule generation algorithm through GPU rules, and constructs the input-output index matrix. After that, the sparse convolution layer aggregates the input sparse voxel features into dense features, and 3D convolution directly slides and filters on the dense features, self-learning for extracting important features and smoothing noise between neighboring features, which obtain dense output features. Then the dense output features are mapped back to the sparse output features through the input-output index matrix, thus the efficient voxel feature extraction is realized. After that, the features will flow to LWPG and ESA structures respectively, and the features in LWPG will be aggregated into 2D image features on the z-axis, the 2D detection similar to image object detection will be performed. Moreover, voxel features will be further filtered in the ESA structure based on the results of 2D detection.

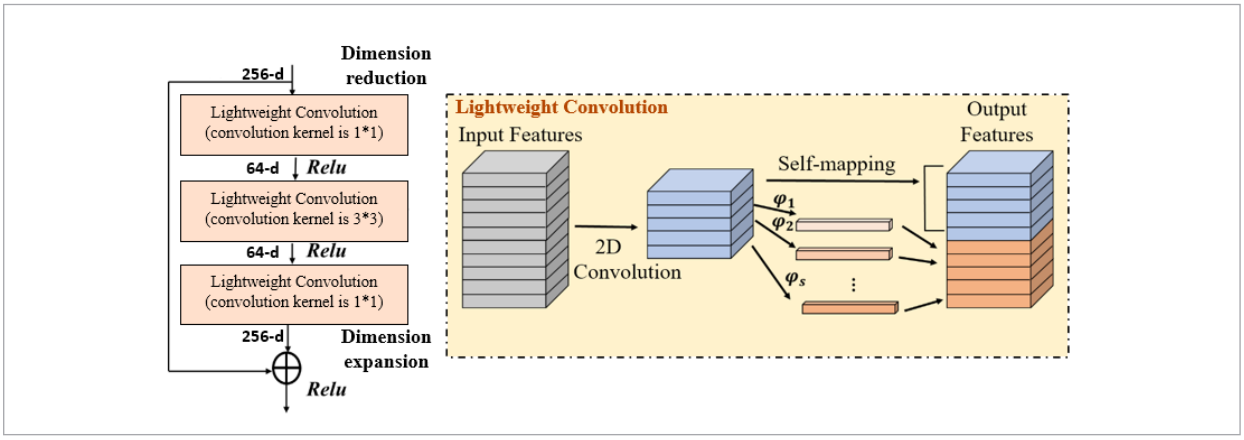
3.3. Lightweight Proposal Generation Module

The processing network of bird's-eye view features in Voxel-RCNN is stacked by traditional two-dimensional convolutional neural networks. Let the input feature be $X \in R^{c \times h \times w}$, where c is the number of channels of the input feature, h and w denote the height and width of the input feature graph, respectively. For traditional convolution layer, the operation process of generating the feature map can be expressed as follows:

$$Y = X * f + b\,, \tag{1}$$

wherein, $Y \in R^{h' \times w' \times n}$ is an output characteristic graph containing channels, $f \in R^{c \times k \times k \times n}$ represents a convolution kernel, h' and w' respectively represent the height and width of the output characteristic graph, $k \times k$ represents the size of the kernel in the convolution kernel f. $*$ is the convolution operator, $X * f$ means the output after 2D convolution operation on feature X using convolution kernel f, b is a deviation term. The amount of parameters involved in the entire convolution process is large. Inspired by GhostNet, this paper proposes a lightweight proposal generation module, which uses traditional convolution to get a part of feature graphs, then maps them by simple linear transformation to complement the rest of feature graphs. At the same time, the bottleneck structure is used to reduce the dimension of the input feature, finally the feature dimension is restored to the specified feature dimension, so as to further reduce the parameter quantity during calculation.

Figure 3
Lightweight proposal generation module



Assuming that the conventional convolution only generates m original feature maps, the output $Y' \in R^{h' \times w' \times m} (m \le n)$ can be expressed as:

$$Y' = X * f' + b \,, \tag{2}$$

where $f' \in R^{c \times k \times k \times m}$ represents a convolution kernel. To fill the number of characteristic graphs from m to n, a simple linear operation can be performed on Y':

$$y_{ij} = \varphi_{i,j}(y_i'), \quad \forall i = 1,\ldots,m, \quad j = 1,\ldots,s \,, \tag{3}$$

where, s is the total number of feature graphs generated by linear operation, y_i' represents the i-th original feature graph in Y', and $\varphi_{i,j}$ represents the j-th linear operation based on the i-th original feature graph. It can be seen from the equation that y_i' can map multiple feature graphs $\{y_{ij}\}_{j=1}^{s}$. By simple linear operation, $n = m \times s$ characteristic graphs can be obtained ($Y = [y_{11}, y_{12,\ldots,} y_{ms}]$). The linear operation is actually to perform grouping operation on the convolution channels, and the number of groups is consistent with the number of channels, so that the parameter quantity of the linear operation is: $w' \times h' \times (c / s) \times (c' / s) \times s$, where c is the number of channels of the input feature, c' is the number of channels of the output feature. The parameter quantity using the conventional convolution is G_{conv}, and the operation quantity using the lightweight convolution is G_{light}, as shown in Equation (4), where $\mu \in (0,1)$ is a scaling coefficient of the parameter quantity.

$$G_{\text{light}} = \mu G_{\text{conv}} \,. \tag{4}$$

To further reduce the computational complexity, we use two 1×1 lightweight convolutions and one 3×3 lightweight convolution (Bottleneck Structure) to build a lightweight proposal generation module. One unit with a kernel size of 1×1 is used for dimension reduction to reduce the amount of computation, the other units are used for dimension increase to restore to the specified output dimension. Taking the traditional two-dimensional convolution as an example, if the convolution kernel size is 3×3, c is the number of input channels, r is the proportion of feature dimension reduction, when the input and output dimensions are the same, the parameter quantity can be simplified from $9c^2$ to $(2r + 9r^2) \times c^2$, when the value of r is 0.25, the amount of the parameter is reduced to $1.0625c^2$. For intuitive expression, let G_{conv} be the parameter quantity using the conventional convolution, and G_{bottle} be the operation quantity using the bottleneck structure, as shown in Equation (5), where $\sigma \in (0,1)$ is the proportional coefficient of the parameter quantity.

$$G_{\text{bottle}} = \sigma G_{\text{conv}} \,. \tag{5}$$

According to the equation, the parameter quantity G_{LWM} of the lightweight proposal generation module is:

$$G_{\text{LWM}} = \sigma(\mu G_{\text{conv}}) \,. \tag{6}$$

As shown in Figure 3, our LWPG module is designed at the 2D backbone of the model, which accelerates the computation of the convolutional layers in the 2D backbone by replacing the traditional convolutional layers with lightweight convolution, and organizes the transformed convolutional layers with bottleneck connections to truncate the data through the downscaling of 1×1 convolutional kernel, after which a 3×3 convolutional kernel is used to capture the features with a relatively large sensory field, then upscaling is carried out with 1×1 convolutional kernel again, and each convolutional layer carries a nonlinear activation function to enhance the feature expression capability. The lightweight feature of LWPG can greatly reduce the computational parameters, accelerate the production of the proposal box in the first stage, and improve the detection efficiency of the algorithm while guaranteeing the quality of the proposal box.

3.4. Efficient Self-Attention Pooling Module

The 2D proposal boxes generated by the LWPG module will be used as ROI in ESA module, thus features outside the ROI will be filtered. Voxel-RCNN uses voxel query to aggregate 3D voxel features into RoI grid points to refine the proposal boxes, but the features encoded at each RoI grid point in this method are limited to the size of the sphere query. Generally speaking, self-attention mechanisms can be used to focus on the correlation between different grid points, while self-attention mechanisms are conducive to focusing on key information in complex data. In point cloud data, due to the influence of the acquisition environment and equipment, there is a lot of noise and interference, focusing on important information is conducive to enhancing the anti-interference ability of the model, thus improving the robustness of the model. Since the calculation method of the self-attention is a matrix product of $\phi(QK^{\mathrm{T}}) \in R^{N \times N}$ and value

vector matrix, the time consumption and the video memory complexity of the self-attention mechanism have a quadratic relationship with the resolution N of the input feature. Traditional self-attention mechanism is shown by Equation (7), d represents the dimension of the feature, Q, K and V are query, key and value vector matrix respectively, which are obtained from the input feature $X \in R^{N\times C}$ through linear transformation, N represents the resolution of the input feature, C represents the number of channels, and the $\mathrm{Softmax}()$ represents the softmax activation function.

$$A(X)=\mathrm{Softmax}(QK^{\mathrm{T}}/\sqrt{d})V$$
$$Q=XW_{\mathrm{Q}},K=XW_{\mathrm{K}},V=XW_{\mathrm{V}}. \quad (7)$$

To address the complexity problem of self-attention mechanism, we propose an efficient self-attention pooling module, as shown in Figure 4. Inspired by UFO-ViT [30], the Softmax operator is replaced by a simple L2 norm, and the computational complexity of the attention module is reduced by a simple associative law of matrix multiplication. Then, the structure of feedforward layer is adopted, and the activation function is introduced for nonlinear mapping [49].

To reduce the computational complexity of the self-attention mechanism, the XNorm operator is used instead of Softmax. The improved self-attention mechanism is shown in Equation (8), where γ is a learning parameter, and h represents embedding dimensions. XNorm is in fact a conventional L2 norm, and is applicable to both the spatial dimensn of $K^{\mathrm{T}}V$ and the channel dimension of Q, so XNorm is also called cross-normalization. Next, the associative law of matrix multiplication is adopted, the key and the value are multiplied first, then the key and the query are multiplied, as shown in Figure 4, the complexity of the two parts of multiplication are both $O(h\times N\times d)$, h represents the number of embedding dimension, it can be known that the process has a linear calculation amount approaching N.

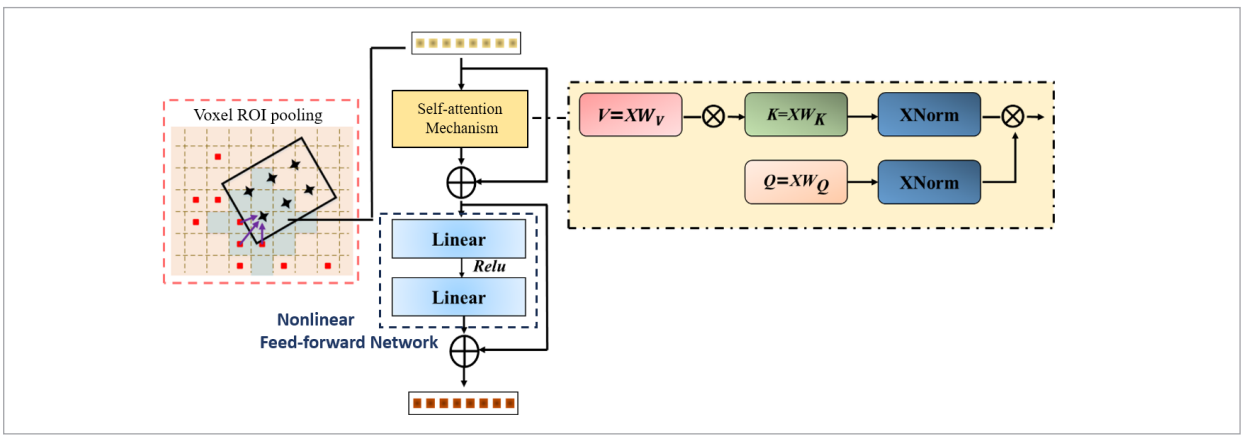
$$A(X)=\mathrm{XN}_{\mathrm{dim=filter}}(Q)(\mathrm{XN}_{\mathrm{dim=space}}(K^{T}V))$$
$$\mathrm{XN}(\mu)=\frac{\gamma\mu}{\sqrt{\sum_{i=0}^{h}||\mu||^{2}}}. \quad (8)$$

Because the matrix multiplication in the self-attention mechanism is linear, the ability to express data is limited. In this paper, a feed-forward network (FFN) is designed for nonlinear transformation of features. The FFN consists of two linear layers containing nonlinear activation functions. The first level extends the embedding dimension of the input feature from d_m to d_f, and the second level reduces the embedding dimension of the feature from d_f to d_m. FFN is expressed as Equation (9), where W_1 and W_2 are the weights of the two linear layers, b_1 and b_2 are the bias terms, and $\psi()$ is the RELU activation function:

$$\mathrm{FFN}(X)=\psi(XW_1+b_1)W_2+b_2. \quad (9)$$

In combination with Figure 4 and the equations, the output of the entire efficient self-attention pooling module can be obtained as:

Figure 4
Efficient Self-Attention Pooling Module (⊗ is the matrix multiplication operator)



$$EO = \mathrm{FFN}(A(X) + X) + A(X) + X. \quad (10)$$

In summary, as shown in Figure 4, after the aggregation operation of 3D voxels by voxel query, the features are fed into the self-attention mechanism modified using the Xnorm operator, and its processed features are output to the next structure after passing through the involved nonlinear structure. Our design allows the XNorm operator to improve the computational efficiency of the attention mechanism, and the nonlinear feedforward network enhances the feature representation of the ESA module. ESA focuses on the correlation between different mesh points through the attention mechanism, and realizes the refinement of the proposal frames by dynamically adjusting the weights to capture the more critical features in the complex point cloud data.

3.5. Loss Function

When the data is input to the detection head, EPCE-RCNN will perform detection and output a series of detection boxes with class probability distribution. After screening the detection bounding boxes with the Non-Maximum Suppression [4], we design the following loss function to measure the inconsistency between the test results and the real value.

The loss function of the RPN in the first stage of generating the proposal includes two parts, namely, classification loss and detection box regression loss, as shown in Equation (11).

$$\begin{aligned} \mathrm{L}_{\mathrm{rpn}} &= \frac{1}{N_{\mathrm{fg}}}[\sum\nolimits_i \mathrm{L}_{\mathrm{cls}}(p_i^a, c_i^*) \\ &+ \Gamma(c_i^* \geq 1)\sum\nolimits_i \mathrm{L}_{\mathrm{reg}}(\xi_i^a, t_i^*)] \end{aligned}, \quad (11)$$

where N_{fg} is the number of anchor points in the foreground, p_i^a is the output of the classification task, c_i^* is the label of the classification task, ξ_i^a is the output of the regression task, and t_i^* is the label of the regression task. $\Gamma(c_i^* \geq 1)$ means the regression loss calculated by using only the foreground anchor frame. $\mathrm{L}_{\mathrm{cls}}()$ uses Focal loss, while $\mathrm{L}_{\mathrm{reg}}()$ uses Huber loss.

Loss function of the second stage detector: This section first calculates the value of the confidence prediction, as shown in Equation (12).

$$l_i^*(\mathrm{IoU}_i) = \begin{cases} 1, & \mathrm{IoU}_i > \vartheta_{\mathrm{H}} \\ (\mathrm{IoU}_i - \vartheta_{\mathrm{L}}) / (\vartheta_{\mathrm{H}} - \vartheta_{\mathrm{L}}), & \vartheta_L \leq IoU_i < \vartheta_{\mathrm{H}} \\ 0, & \mathrm{IoU}_i < \vartheta_{\mathrm{L}} \end{cases}, \quad (12)$$

where IoU_i is the intersection ratio between the ith proposed box and its corresponding actual box, ϑ_{H} and ϑ_{L} are the upper and lower thresholds of the intersection ratio between the foreground and the background. Here, EPCE-RCNN uses a binary cross-entropy loss function to predict the confidence. The box regression uses the Huber loss function. The loss function of the detection head is shown in Equation (13), wherein N_s represents the number of proposal boxes sampled in the training phase, $\Gamma(\mathrm{IoU}_i \geq \vartheta_{\mathrm{reg}})$ means that only proposal boxes with the intersection ratio higher than the threshold are included in the calculation of the loss function.

$$\begin{aligned} \mathrm{L}_{\mathrm{detect}} &= \frac{1}{N_{\mathrm{s}}}[\sum\nolimits_i \mathrm{L}_{\mathrm{cls}}(p_i, l_i^*(\mathrm{IoU}_i)) + \\ &\Gamma(\mathrm{IoU}_i \geq \vartheta_{\mathrm{reg}})\sum\nolimits_i \mathrm{L}_{\mathrm{reg}}(\xi_i, t_i^*)] \end{aligned} \quad (11)$$

When the calculation of the loss function is completed, EPCE-RCNN uses the adam optimizer to update the model parameters in order to achieve a smaller value of each loss function, so as to correct the inconsistency between the detection results and the real values.

4. Experiment and Experimental Results

In this section, we first test the proposed EPCE-RCNN on KITTI dataset and two self-built datasets, the self-built datasets are extended YCB dataset and power component dataset. Then, we compare and analyze with existing methods. In addition, we validate the effectiveness of each part of the EPCE-RCNN through ablation experiments.

4.1. Experimental Dataets

KITTI dataset: The KITTI dataset [8] contains 6 categories of targets in multiple scenes with various degrees of occlusion and truncation. In the object detection task of this dataset, the samples are divided into three levels according to the difficulty: Simple, medium and difficult. The target in simple task is completely visible, and the maximum truncation rate is 15%. The target of medium task is partially occluded, and the maximum truncation rate is 30%. Objects in the difficult task are harder to observe, and the maximum truncation rate is 50%.

Improved YCB dataset: YCB video dataset [40] is a widely used 6d pose annotation dataset, which contains 21 categories of targets. A scene is built by 3-9 objects selected from these targets, and data acquisition is carried out by RGB-D camera. Depth image and RGB image are provided in the dataset. Since the proposed algorithm is based on point cloud, this paper converts the dataset depth image into point cloud data based on the mapping relationship between image physical coordinates and camera coordinates in advance, and independently labels the 3D information of the target, and adds the centroid point coordinates (x, y, z), length, width and height (l, w, h) and rotation angle θ information of the 3D detection frame (refers to the angle between the long side of the target in the Z-axis direction and the X-axis in the defined coordinate axis).

Self-built power component dataset: The main scene of proposed self-built dataset is indoor, 500 samples are taken in total, of which 300 are taken as training set and the rest are taken as validation set. The dataset includes three types of targets with different placement positions and placement angles: Single clip, C clip and wedge clip. The acquisition equipment is realsense-d435 i depth camera. Different from the reconstruction of YCB dataset, we directly convert the depth image into 3D point cloud data through the built-in function of the camera. In the labeling process, we also generate the centroid point coordinates (x, y, z), length, width and height (l, w, h) data and rotation angle θ information of the target 3D bounding box.

4.2. Implementation

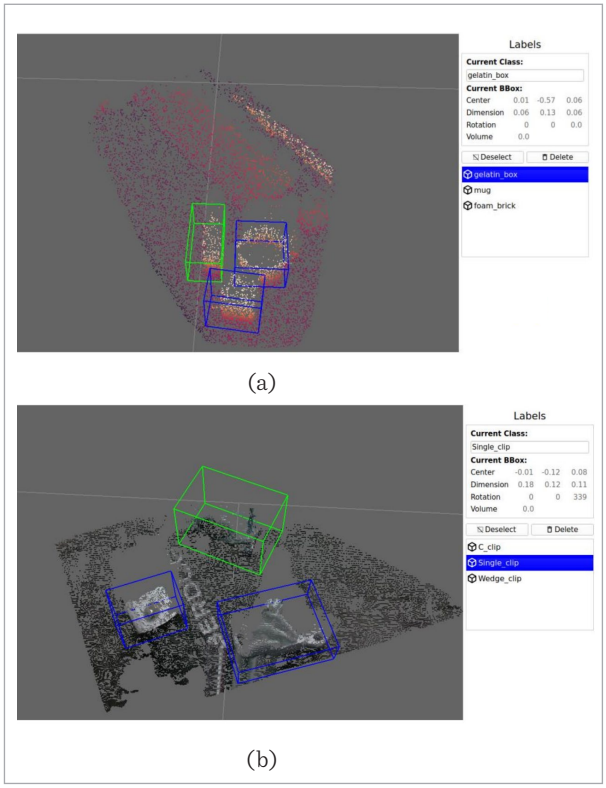
The algorithms are optimized by ADAM optimization algorithm. On three datasets, EPCE-RCNN was trained for 80 epochs on three RTX 2080 GPUs with a batch size of 6 and an initial learning rate of 0.01.

Figure 6 reflects the change of the two loss functions with the number of training periods. From the curve, we can see that the L_{rpn} and L_{detect} loss functions have reached a lower point and converge well, indicating that the training has reached an ideal state.

According to KITTI official metrics [8], the training indicators are selected as 3D detection accuracy indicator and average directional similarity [49], which are calculated based on multiple recall positions, such as represents calculation based on 11 recall locations.

For voxelization, the extent of the point cloud in the KITTI dataset is set to (0, 70.4) meters in the X axis,

Figure 5
Labeling the proposed datasets. The improved YCB dataset is (a) and the self-built power component dataset is (b)



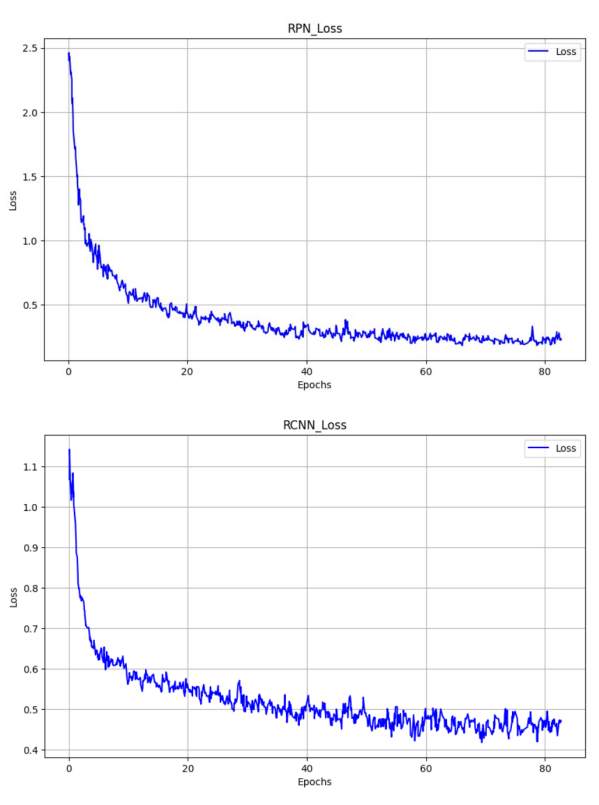
(-40, 40) meters in the Y axis, and (-3, 1) meters in the Z axis, and the size of the voxel grid is set to (0.05, 0.05, 0.1) meters. In the two self-built datasets, the point cloud extent is set to (-2, 2) meters on the X axis, (0, 2) meters on the Y axis, and (-2, 2) meters on the Z axis. The maximum number of point cloud in each voxel is 5.

EPCE-RCNN is developed and deployed based on OpenPCDet and PyTorch framework. It has good compatibility with all kinds of GPU devices that support CUDA operators, and directly supports distributed training or deployment through PyTorch. At the same time, with the help of OpenPCDet's unified dataset format conversion tools, the network also has broad compatibility for datasets with different formats.

We also adopt some data enhancement schemes as our regularization strategies, including: random sampling, random flipping, random rotation and global scale scaling, which can improve the diversity of data and improve the generalization ability of the model [48].

Figure 6

Training loss curve. The abscissa represents the epoch count, and the ordinate represents the loss value. The plot on the first row depicts the L_{cls}, while the second row represents the L_{detect}



4.3. KITTI Dataset Experimental Results

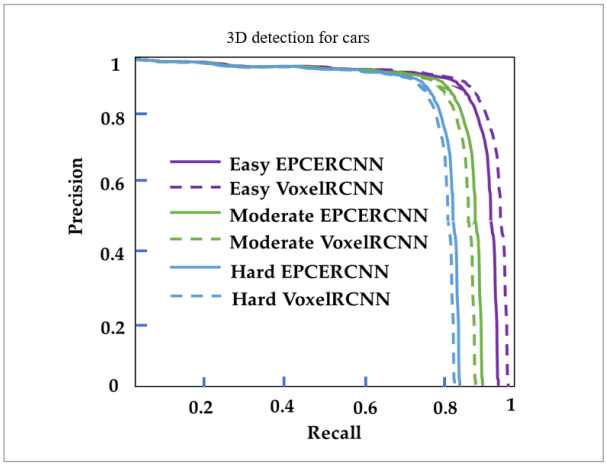
4.3.1. Analysis of Experimental Results on KITTI Online Test Set

According to the benchmark of KITTI, this section mainly talks about the 3D detection index $mAP|_{40}$ and direction estimation index $AOS|_{40}$ of EPCE-RCNN and SOTA methods on the Car class. The results are shown in Figure 7 and Table 1. Note that the benchmark network Voxel-RCNN does not give performance in the Cyclist class, but focuses primarily on the performance of the Car class. To ensure the fairness of the comparison, this section only carries out the experimental analysis on the Car class. Depend on Figure 7 and Table 1, for 3D detection performance, EPCE-RCNN outperforms the benchmark algorithm Voxel-RCNN in vehicle detection performance in all difficulty levels.

Figure 7 shows the RP curves of the 3D detection algorithm EPCE-RCNN and its benchmark algorithm Voxel-RCNN. The area enclosed by the curve and the coordinate axis is the average accuracy, and the larger the area is, the higher the accuracy of the algorithm is.

Figure 7

The RP curves of 3D detection for cars (IoU = 0.7)



For the performance of 3D detection, EPCE-RCNN has 2 evaluation indicators (out of 3 indicators) ranked in the top three among all comparison algorithms, and the hard difficulty ranks first, as shown in Table 1.

Specifically:

1. EPCE-RCNN outperforms the Grid-based method on the moderate and difficult levels. The $mAP|_{40}$ on moderate difficulty increased by at least 0.07% compared to the Grid-based method, and the $mAP|_{40}$ on hard difficulty increased by at least 0.10%.
2. EPCE-RCNN draws better performance compared to all Point-based methods on hard difficulty, and outperforms most Point-based methods on moderate difficulty.
3. Compared with the method of combining point cloud and voxel, the $mAP|_{40}$ of EPCE-RCNN at moderate and hard levels reached 81.69% and 77.17%, which were the top three levels.

For the performance of direction estimation, EPCE-RCNN has two evaluation indicators that rank among the top three among all the comparison algorithms. Looking at Table 1, the following conclusions can be drawn.

Table 1
The Performance on the KITTI online Test Server, Red, Green and Blue Represent the First, Second and Third Ranked Results Respectively (IoU=0.7)

Method	Modality	$mAP\|_{40}$			$AOS\|_{40}$		
		Easy	Moderate	Hard	Easy	Moderate	Hard
SASA [2]	Point	88.76	82.16	77.16	96.00	95.29	92.42
Faraway-Frustum [47]		87.45	79.05	76.14	-	-	-
3DSSD [43]		88.36	79.57	74.55	-	-	-
EPNet [16]		89.81	79.28	74.59	96.13	94.22	89.68
SSL-PointGNN [7]		87.78	79.36	74.15	38.55	37.21	36.53
PointRCNN [26]		86.96	75.64	70.70	95.90	91.77	86.92
BADet [24]	Grid	89.28	81.61	76.58	98.65	95.34	90.28
SVGA-Net [11]		87.33	80.47	75.91	96.02	94.45	91.54
Part-A2 [27]		87.81	78.49	73.51	95.00	91.73	88.86
TANet [20]		84.39	75.94	68.82	93.52	90.11	84.61
Harmonic- PointPillar [46]		82.26	73.96	69.21	94.23	90.78	87.42
Voxel-RCNN [4]		90.90	81.62	77.06	96.47	94.96	92.24
PointPillars [17]		82.58	74.31	68.99	77.10	58.65	51.92
SA-SSD [10]	Point-Grid	88.75	79.79	74.16	39.40	38.30	37.07
DVFENet [12]		86.20	79.18	74.58	95.33	94.44	91.55
SE-SSD [52]		91.49	82.54	77.15	96.55	95.17	90.00
PV-RCNN [25]		90.25	81.43	76.82	98.15	94.57	91.85
AFE-RCNN [29]		88.41	81.53	77.03	95.84	94.63	92.07
EPCE-RCNN (Ours)	Grid	89.99	81.69	77.17	96.83	95.13	92.32

(1) EPCE-RCNN outperforms all Grid-based methods on hard difficulty, and has at least 0.08% improvement in $AOS|_{40}$. (2) EPCE-RCNN outperformed all point-based methods on easy difficulty by at least a 0.7% improvement in $AOS|_{40}$. (3) Compared with the method combining point cloud and voxel, EPCE-RCNN outperforms the compared algorithm in the hard difficulty, the $AOS|_{40}$ is improved by at least 0.47% compared with the fused method.

As for the improvement of baseline network, EPCE-RCNN outperformed Voxel-RCNN in 5 tasks except easy difficulty of 3D detection, and improved $mAP|_{40}$ by 0.07% and 0.11% respectively in moderate and hard difficulty of 3D detection, and improved $AOS|_{40}$ by 0.36%, 0.17% and 0.08% respectively in three tasks of direction estimation.

The above experimental results show that EPCE-RCNN significantly improves the baseline algorithm's ability to detect 3D objects with a high accuracy, especially in hard difficulty. This may be due to that the efficient attention mechanism of EPCE-RCNN strengthens the correlation between points.

In the case of severe object masking, the target point cloud is sparse and incomplete, while the efficient attention mechanism strengthens the ability of the algorithm to contact context information, thus ensuring the accuracy of algorithm detection.

4.3.2. KITTI Validation Set Test Results Analysis

Most of the relevant papers on the KITTI validation set only provide the calculation of the 3D detection index $mAP|_{11}$ by using 11 recall positions in the Car class. This section compares the EPCE-RCNN and SOTA methods in Table 2 according to $mAP|_{11}$. EPCE-RCNN has two indexes in the top three.

Table 2

3D Detection Results on the Validation Set of KITTI, Red, Green and Blue Represent the First, Second and Third Ranked Results, respectively (IoU=0.7)

Method	$mAP\|_{11}$		
	Easy	Moderate	Hard
PointRCNN [26]	88.88	78.63	77.38
3DSSD [43]	89.71	79.45	78.67
BADet [24]	90.06	85.77	79.00
Part-A2 [27]	89.47	79.47	78.54
Harmonic- PointPillar [46]	87.66	77.76	73.44
TANet [20]	87.52	76.64	73.86
DVFENet [12]	89.81	79.52	78.35
SE-SSD [52]	90.21	86.25	79.22
PV-RCNN [25]	89.34	83.69	78.70
AFE-RCNN [29]	89.61	83.99	79.18
Voxel-RCNN [4]	89.41	84.52	78.93
EPCE-RCNN(Ours)	89.78	84.66	79.13

Considering the number of indexes in the top three, the proposed algorithm has more advantages. EPCE-RCNN does not perform well on all metrics, which may be related to the different partitions of the training and validation sets. Compared with the benchmark network, EPCE-RCNN outperforms the benchmark network Voxel-RCNN in $mAP|_{11}$ at all difficulty levels, increasing by 0.37%, 0.14% and 0.20%, respectively.

4.3.3. Analysis of Algorithm Efficiency and Robustness

To verify the efficiency of the proposed algorithm, the detection time of EPCE-RCNN on KITTI validation set is shown in Table 3. Compared with the reference network, the detection time of EPCE-RCNN is reduced by 12%, and the parameter quantity is reduced by 10%. On the KITTI dataset, EPCE-RCNN achieved

Table 3

Running Time of the Algorithm on The Validation Set of KITTI

Method	Runtime(s)	Parameters(M)
PointPillars [17]	0.0204	55.40
SECOND [42]	0.0243	61.03
Part-A2 [27]	0.0732	457.47
PV-RCNN [25]	0.0286	150.64
AFE-RCNN [29]	0.0345	156.13
Voxel-RCNN [4] Baseline	0.0249	86.95
EPCE-RCNN (Ours)	0.0221	78.39

a detection time of 0.0249 s, which is comparable to the single-stage detector SECOND of 0.0243 s. In addition, according to Table 3, EPCE-RCNN can improve the operation efficiency and improve the accuracy to a certain extent, which can better meet the timeliness requirements of object detection tasks.

In order to analyze the effect of the number of object classes on the efficiency and accuracy of the algorithm proposed in this section, experiments on multi-class objects are carried out on the KITTI validation set. The results are shown in Table 4. The $mAP|_{40}$ values of EPCE-RCNN were 85.33%, 85.29% and 85.31% respectively. The number of target classes has little effect on the performance of the proposed algorithm ($mAP|_{40}$). At the same time, the time consumed by EPCE-RCNN of different categories and quantities is almost the same, the difference is less than 0.001s. These results

Table 4

Experimental Results for Multiple Object Detection on the Validation Set of KITTI. IoU =0.7 for Car, IoU =0.5 for Cyclist and Pedestrian

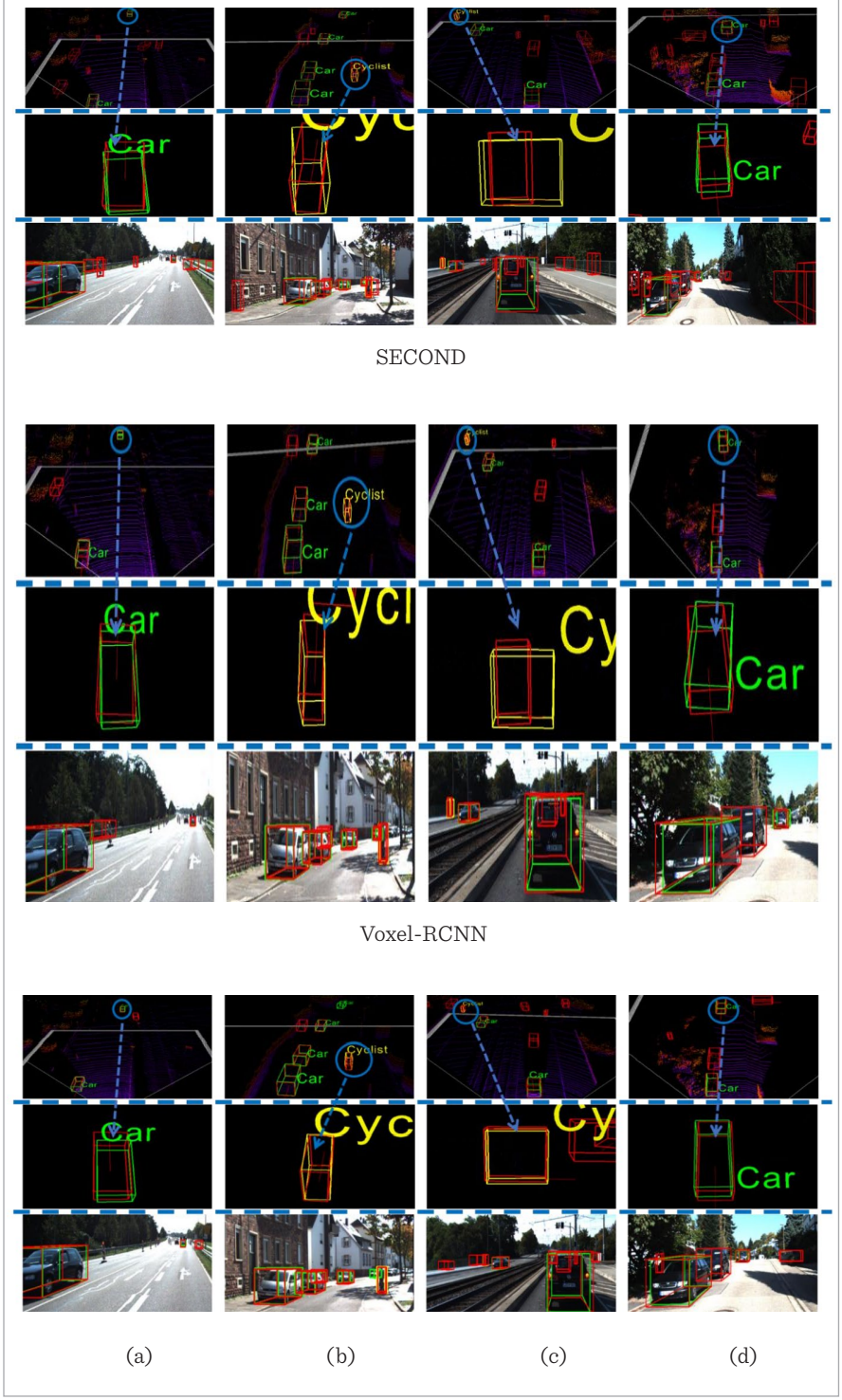
Number of categories	Moderate ($mAP|_{40}$)			Runtime (s)
1 class	Car			0.0215
	85.30			
2 classes	Car	Cyclist		0.0221
	85.27	72.09		
3 classes	Car	Cyclist	Pedestrian	0.0225
	85.27	72.07	59.11	

show that EPCE-RCNN can maintain good efficiency and robustness in multi-class object detection.

4.3.4. Qualitative Analysis

This section compares the visualizations of EPCE-RCNN, Voxel-RCNN and SECOND, as shown in Figure 8. The red boxes represent the prediction boxes, while the green and yellow boxes represent the actual labels for the Car class and Cyclist class respectively. EPCE-RCNN and Voxel-RCNN extract the RoI feature from the voxel feature, which compensates for the loss of 3D structure information in the process of voxel projection to bird's-eye view. While SECOND does not supplement the 3D information, so the coincidence ratio between the prediction boxes and the actual annotation boxes is not as good as the above two methods. In addition, EPCE-RCNN enhances the dependency relationship between grid points during feature coding of RoI grid points through an efficient self-attention mechanism, and simultaneously enhances the nonlinearity of the model through a feedforward network, so that the model can perform better feature expression. In this way, EPCE-RCNN completes the refinement of the proposal box with high quality. As can be seen from Figure 8, the predicted box of EPCE-RCNN is closest to the actual labeled box.

Figure 8
Visualization results on KITTI validation set



4.4. Ablation Experiment Based on KITTI Validation Set

Since EPCE-RCNN is proposed on the basis of Voxel-RCNN, the performance of Voxel-RCNN is taken as a baseline to compare the performance of each module. The results of the ablation experiments are shown in Table 5. Referring to the metric used in the Voxel-RCNN article, this section uses $mAP|_{40}$ (calculated from 40 recall locations) as the evaluation metric. Experimental results show that each module used in EPCE-RCNN can bring performance gains to the original model.

As shown in Table 5, compared to Voxel-RCNN's performance, Ours1 had a maximum $mAP|_{40}$ improvement of 0.53% on 6 evaluation indicators in three difficulty levels.The LWPG module is designed based on the acceleration strategy, so its accuracy improvement is limited. The effect of ESA module effectively strengthens the correlation between different grid points during RoI feature coding, and the accuracy is more significant than that of LWPG module. On each task in Table 5, the accuracy of Ours2 is higher than that of Ours1.

In order to explore the influence of different K values on the detection accuracy of EPCE-RCNN, we try to compare several K values on the KITTI validation set. The experimental results are shown in Table 6. It can be seen from Table 6 that there is no obvious relationship between the change of K value and the detection accuracy, because the point cloud data in the 3D object detection data set is not uniformly distributed, and accompanied by a lot of noise, too large K value may capture more key features, and it is also possible to collect more noise (the same reason that K value is too small), so the influence of K value change on detection accuracy is uncertain, we need to try the best value in practice. We finally choose 16 as the K value, which is consistent with the results of Voxel-RCNN, and shows better results in Table 6, which we think is more appropriate.

In order to test the performance of EPCE-RCNN under various conditions, we carried out corruption experiments on KITTI validation set with reference to the work of [5], and the experimental results are shown in Table 7. Compared to the baseline, EPCE-RCNN outperformed Voxel-RCNN in nine out of ten

Table 6

Detection Accuracy of EPCE-RCNN. IoU=0.7 for Car, IoU =0.5 for Cyclist and Pedestrian. corresponding to different K values ($mAP|_{40}$)

K Value	Car (Moderate)	Cyclist (Moderate)	Pedestrian (Moderate)
8	83.11	74.10	58.23
16	**85.27**	72.07	59.11
24	84.23	73.07	**59.28**
32	82.87	72.35	58.06
64	85.03	**74.11**	58.73

Table 5

Ablation Experiment Results (IoU=0.7 for Car, IoU =0.5 for Cyclist)

Method		Module		$mAP|_{40}$					
		LWPG	ESA	Easy		Moderate		Hard	
				Car	Cyclist	Car	Cyclist	Car	Cyclist
Voxel-RCNN Baseline				92.16	88.81	85.01	71.82	82.48	67.20
Ours	Ours1	√		92.19	88.78	85.03	72.35	82.49	67.28
	Ours2		√	92.38	90.01	85.27	73.79	82.78	68.85
	EPCE-RCNN	√	√	92.48	90.93	85.36	73.81	83.86	69.06
		LWPG	ESA	$AOS|_{40}$					
Voxel-RCNN Baseline				98.74	96.13	94.79	82.38	92.37	78.11
EPCE-RCNN		√	√	98.87	96.56	94.67	83.29	94.18	79.86

Table 7
Detection Accuracy under Extreme Conditions. IoU=0.7 for Car, IoU =0.5 for Cyclist

Method	Corruption	Car (Moderate)	Cyclist (Moderate)
EPCE- RCNN	Gaussian Noise	**84.63**	**72.38**
	Impulse Noise	**84.67**	**77.21**
	Density Decrease	**83.10**	**72.91**
	Lidar Crosstalk Noise	82.71	**73.92**
	Spatial Alignment	**48.73**	**41.40**
VOXEL -RCNN	Gaussian Noise	83.21	69.98
	Impulse Noise	83.07	73.68
	Density Decrease	83.07	69.49
	Lidar Crosstalk Noise	**82.76**	71.78
	Spatial Alignment	47.28	40.93

metrics, with the highest improvement of 1.60% in the Car class and 3.53% in the Cyclist class. These results demonstrate the robustness of EPCE-RCNN.

4.5. Analysis of Experimental Results Based on Improved YCB Dataset and Power Device Dataset

4.5.1. Quantitative Analysis

While IoU=0.7 is taken, EPCE-RCNN has excellent performance in the reconstructed YCB dataset and power component dataset. The performance on the reconstructed YCB dataset is shown in Table 8. Compared with the Voxel-RCNN, the $mAP|_{11}$ of EPCE-RCNN for mugs and gelatin boxes is improved by 0.66% and 1.04% respectively, and the detection time is reduced by 0.0173s.

The performance of the algorithm on the self-built power component dataset is shown in Table 9. Compared with the listed algorithm, the $mAP|_{11}$ of Single clip, Wedge clip and C clip is improved by 1.14%, 0.02% and 0.52% respectively compared with the Voxel-RCNN, and the detection time is reduced by 0.0148s.

Table 8
The Results of Algorithm on the Validation Set of YCB (IOU=0.7)

Method	$mAP\|_{11}$		Detection time (s)
	mug	gelatin box	
PointRCNN [26]	33.24	35.62	0.3339
SECOND [42]	30.03	31.28	0.0793
PV-RCNN [25]	36.88	38.79	0.1866
AFE-RCNN [29]	37.37	39.91	0.1971
Voxel-RCNN [4]	37.01	39.02	0.1084
EPCE-RCNN(Ours)	37.67	40.06	0.0911

Table 9
The Results of Algorithm on the Validation Set of Electric Components Dataset (IOU=0.7)

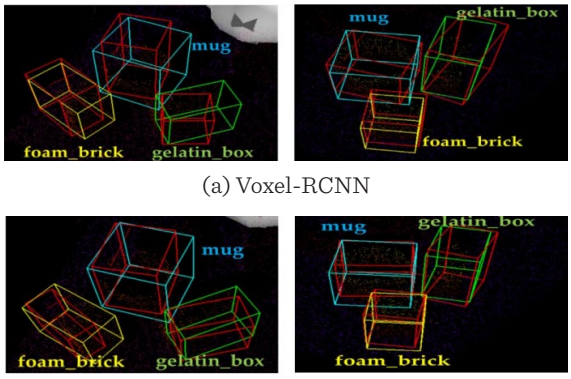
Method	$mAP\|_{11}$			Detection time (s)
	Single clip	Wedge clip	C clip	
PointRCNN [26]	31.04	26.07	33.33	0.3713
SECOND [42]	29.95	24.37	30.01	0.0817
PV-RCNN [25]	34.37	29.23	36.36	0.1466
AFE-RCNN [29]	35.86	30.05	37.68	0.1561
Voxel-RCNN [4]	34.49	29.99	36.79	0.1182
EPCE-RCNN(Ours)	35.63	30.01	37.31	0.1034

Experimental results show that EPCE-RCNN has the highest detection accuracy on both datasets, and the detection accuracy on some tasks is slightly lower than that of AFE-RCNN, while the detection time of EPCE-RCNN is much shorter than that of AFE-RCNN on these tasks, which means that EPCE-RCNN omits more complex and time-consuming feature engineering to improve accuracy to ensure lightweight design. Considering the high real-time requirements of actual tasks, EPCE-RCNN has the best robustness on workpiece positioning tasks. In a word, the improved strategy of EPCE-RCNN is effective, it can improve the accuracy while improving the lightweight. Meanwhile, since there are differences in detection scenarios, target categories, target sizes, and target distributions among the datasets, and our algorithm still achieves good results on all three datasets, it proves the good generalization ability of our model.

4.5.2. Qualitative Analysis

The visualization of the improved YCB dataset and self-built power component dataset is shown in Figure 9 and Figure 10, where the red boxes represent the prediction boxes. In Figure 9, the green, yellow, and blue boxes represent the actual labeled boxes of the gelatin box, foam brick, and mug respectively. Due to the lack of physical foam tiles, two boxes were used in the visualization experiment as an alternative. In Figure 10, the green, yellow and blue boxes represent the actual marking boxes of Single clip, C clip and Wedge clip respectively. It can be seen that in the same scene, the prediction frame of EPCE-RCNN is more consistent with the actual frame than that of the baseline algorithm Voxel-RCNN.

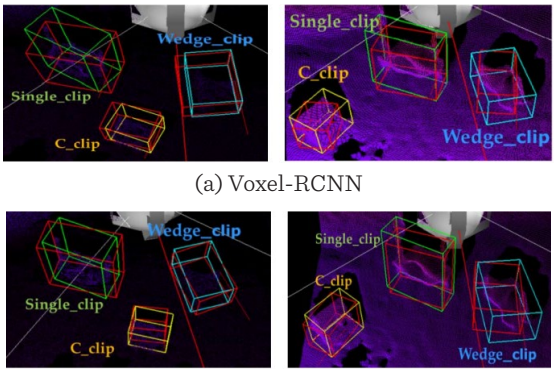
Figure 9
Visualization of the reconstructed YCB validation set



(a) Voxel-RCNN

(b) EPCE-RCNN

Figure 10
Visualization of self-built power component validation set



(a) Voxel-RCNN

(b) EPCE-RCNN

4.6. Summary of Experiments

In the KITTI online test set, 3D detection accuracy of EPCE-RCNN is better than Voxel-RCNN in medium and difficult difficulty levels, and direction detection accuracy is better than Voxel-RCNN in all three difficulty levels (Table 1). At the same time, EPCE-RCNN is better than the benchmark network Voxel-RCNN in all difficulty levels of the KITTI validation set. In the two self-constructed datasets (Table 2), Table 3-4 EPCE-RCNN 8-9 shows that EPCE-RCNN has less computing time and fewer parameters. Moreover, in qualitative analysis, Figures 8-10 all indicate that EPCE-RCNN is more suitable for the real box of the target. In addition, the excellent performance of EPCE-RCNN in high occlusion difficulty proves the robustness of the algorithm, and the excellent performance in many different targets and different task scenarios proves that the algorithm has excellent generalization ability. In the ablation experiment, we verify the effectiveness of each module, and through the corruption experiment to further prove the robustness of EPCE-RCNN under different conditions.

5. Conclusion

To improve the efficiency of 3D object detection, an efficient 3D object detection algorithm EPCE-RCNN based on point cloud correlation enhancement is proposed in this paper. The network is a two-stage network, which enhances the correlation between different grid points through an efficient self-attention pooling module during RoI feature encoding. Moreover, it solves the limitation of the influence of the radius of neighborhood query sphere during pooling operation. The attention mechanism combined with the feedforward neural network ensures the nonlinearity of the model, which makes the model better to express the features. By the lightweight proposal box generation module, the computational complexity and time consumption of the whole three-dimensional object detection network are reduced, then the timeliness requirement in the detection task is better met. In addition, this paper builds the power component dataset, and supplements the 3D annotation information of the public dataset. Experiments are carried out on a variety of scene datasets. The experimental results show that EPCE-RCNN has some advantages in location accura-

cy and detection speed, which makes the algorithm can be applied to time-sensitive object detection tasks with a certain accuracy. Meanwhile, the good performance on multiple datasets also shows that the algorithm has good generalization and has the potential to be widely applied to various scenes and maintain robustness.

Although the research of 3D object detection algorithm based on point cloud has been completed in this paper, there are still some shortcomings in the research work, which need to be further optimized and perfected. The summary is as follows:

1 The proposed algorithm uses the anchor frame method to generate the detection frame. When the object tilts to a certain angle, the anchor frame method cannot coordinate this angle well, which leads to a large error. Future work will make full use of the rotation invariance of point cloud to improve the detection box generation mode.

2 When making the dataset, the scene in the camera coordinate system needs to be manually aligned to the specified plane, and there are too many human errors in this process, which makes the annotation work of the dataset have natural errors. In the future, it is necessary to optimize the annotation work and set a unified alignment coordinate system.

3 The scene of self-built dataset is single, and its performance in complex environment has certain limitations. In addition, compared with mature data collection equipment, it is difficult to ensure the original data quality only by using depth camera for data collection in this paper. In the follow-up work, the production mode of dataset will be improved, and the combination of laser radar and camera will be used for data collection in various working scenes. At the same time, the point cloud data acquired by the depth camera in our work actually contains the 3D coordinate information and RGB color information of each point, but the proposed algorithm does not use the RGB information. Future work will try to analyze and utilize the valuable information in a multi-modal fusion way.

Acknowledgement

This work was supported by the Natural Science Foundation of Guangxi under Grant No.2022GXNSFBA035661, the Open Foundation of the Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education under Grant No. GZUAMT2021KF [04], the Key Laboratory of AI and Information Processing (HechiUniversity) Education Depart-ment of Guangxi Zhuang Autonomous Region under 2022GXZDSY006, and Bagui Scholar Program of Guangxi.

References

1. Bariya, P., Nishino, K. Scale-Hierarchical 3D Object Recognition in Cluttered Scenes. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; IEEE, 2010, 1657-1664. https://doi.org/10.1109/CVPR.2010.5539774
2. Chen, C., Chen, Z., Zhang, J., Tao, D. SASA: Semantics-Augmented Set Abstraction for Point-Based 3D Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36, 221-229. https://doi.org/10.1609/aaai.v36i1.19897
3. Chua, C. S., Jarvis, R. Point Signatures: A New Representation for 3D Object Recognition. International Journal of Computer Vision, 1997, 25, 63. https://doi.org/10.1023/A:1007981719186
4. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H. Voxel R-CNN: Towards High Performance Voxel-Based 3D Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35, 1201-1209. https://doi.org/10.1609/aaai.v35i2.16207
5. Dong, Y., Kang, C., Zhang, J., Zhu, Z., Wang, Y., Yang, X., Su, H., Wei, X., Zhu, J. Benchmarking Robustness of 3D Object Detection to Common Corruptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023, 1022-1032. https://doi.org/10.1109/CVPR52729.2023.00105
6. Du, L., Ye, X., Tan, X., Feng, J., Xu, Z., Ding, E., Wen, S. Associate-3Ddet: Perceptual-to-Conceptual Association for 3D Point Cloud Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 13329-13338. https://doi.org/10.1109/CVPR42600.2020.01334
7. Erçelik, E., Yurtsever, E., Liu, M., Yang, Z., Zhang, H., Topçam, P., Listl, M., Çaylı, Y. K., Knoll, A. 3D Object Detection with a Self-Supervised LiDAR Scene Flow

Backbone. In Proceedings of the Computer Vision-ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part X; Springer, 2022, 247-265. https://doi.org/10.1007/978-3-031-20080-9_15

8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R. Vision Meets Robotics: The KITTI Dataset. The International Journal of Robotics Research, 2013, 32, 1231-1237. https://doi.org/10.1177/0278364913491297
9. Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C. GhostNet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 1580-1589. https://doi.org/10.1109/CVPR42600.2020.00165
10. He, C., Zeng, H., Huang, J., Hua, X.-S., Zhang, L. Structure-Aware Single-Stage 3D Object Detection from Point Cloud. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, 11870-11879. https://doi.org/10.1109/CVPR42600.2020.01189
11. He, Q., Wang, Z., Zeng, H., Zeng, Y., Liu, Y. SVGA-Net: Sparse Voxel-Graph Attention Network for 3D Object Detection from Point Clouds. In Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36, 870-878. https://doi.org/10.1609/aaai.v36i1.19969
12. He, Y., Xia, G., Luo, Y., Su, L., Zhang, Z., Li, W., Wang, P. DVFENet: Dual-Branch Voxel Feature Extraction Network for 3D Object Detection. Neurocomputing, 2021, 459, 201-211. https://doi.org/10.1016/j.neucom.2021.06.046
13. Hornik, K. Approximation Capabilities of Multilayer Feedforward Networks. Neural Networks, 1991, 4, 251-257. https://doi.org/10.1016/0893-6080(91)90009-T
14. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861, 2017.
15. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 11108-11117. https://doi.org/10.1109/CVPR42600.2020.01112
16. Huang, T., Liu, Z., Chen, X., Bai, X. EPNet: Enhancing Point Features with Image Semantics for 3D Object Detection. In Proceedings of the Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV; Springer, 2020, 35-52. https://doi.org/10.1007/978-3-030-58555-6_3
17. Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O. PointPillars: Fast Encoders for Object Detection from Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 12697-12705. https://doi.org/10.1109/CVPR.2019.01298
18. Li, X., Guivant, J., Khan, S. Real-Time 3D Object Proposal Generation and Classification Using Limited Processing Resources. Robotics and Autonomous Systems, 2020, 130, 103557. https://doi.org/10.1016/j.robot.2020.103557
19. Li, Z., Zhang, J., Li, G., Liu, Y., Li, S. Graph Attention Neural Networks for Point Cloud Recognition. In 2019 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2019, 387-392. https://doi.org/10.1109/ICME.2019.00074
20. Liu, Z., Zhao, X., Huang, T., Hu, R., Zhou, Y., Bai, X. TANet: Robust 3D Object Detection from Point Clouds with Triple Attention. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34, 11677-11684. https://doi.org/10.1609/aaai.v34i07.6837
21. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K. Recurrent Models of Visual Attention. Advances in Neural Information Processing Systems, 2014, 27.
22. Noh, J., Lee, S., Ham, B. HVPR: Hybrid Voxel-Point Representation for Single-Stage 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, 14605-14614. https://doi.org/10.1109/CVPR46437.2021.01437
23. Qi, C. R., Yi, L., Su, H., Guibas, L. J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Advances in Neural Information Processing Systems, 2017, 30.
24. Qian, R., Lai, X., Li, X. BADet: Boundary-Aware 3D Object Detection from Point Clouds. Pattern Recognition, 2022, 125, 108524. https://doi.org/10.1016/j.patcog.2022.108524
25. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 10529-10538. https://doi.org/10.1109/CVPR42600.2020.01054
26. Shi, S., Wang, X., Li, H. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer

Vision and Pattern Recognition, 2019, 770-779. https://doi.org/10.1109/CVPR.2019.00086

27. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H. From Points to Parts: 3D Object Detection from Point Cloud with Part-Aware and Part-Aggregation Network. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43, 2647-2664. https://doi.org/10.1109/TPAMI.2020.2977026
28. Shi, W., Rajkumar, R. Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 1711-1719. https://doi.org/10.1109/CVPR42600.2020.00178
29. Shuang, F., Huang, H., Li, Y., Qu, R., Li, P. AFE-RCNN: Adaptive Feature Enhancement RCNN for 3D Object Detection. Remote Sensing, 2022, 14, 1176. https://doi.org/10.3390/rs14051176
30. Song, J. UFO-ViT: High Performance Linear Vision Transformer without Softmax. arXiv preprint arXiv:2109.14382, 2021.
31. Stein, F., Medioni, G. Structural Indexing: Efficient 3D Object Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14, 125-145. https://doi.org/10.1109/34.121785
32. Tuzel, O., Liu, M.-Y., Taguchi, Y., Raghunathan, A. Learning to Rank 3D Features. In Proceedings of the Computer Vision-ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I; Springer, 2014, 520-535. https://doi.org/10.1007/978-3-319-10590-1_34
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. Attention Is All You Need. Advances in Neural Information Processing Systems, 2017, 30.
34. Vora, S., Lang, A. H., Helou, B., Beijbom, O. PointPainting: Sequential Fusion for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 4604-4612. https://doi.org/10.1109/CVPR42600.2020.00466
35. Wang, L., Fan, X., Chen, J., Cheng, J., Tan, J., Ma, X. 3D Object Detection Based on Sparse Convolution Neural Network and Feature Fusion for Autonomous Driving in Smart Cities. Sustainable Cities and Society, 2020, 54, 102002. https://doi.org/10.1016/j.scs.2019.102002
36. Wang, L., Zhao, D., Wu, T., Fu, H., Wang, Z., Xiao, L., Xu, X., Dai, B. Drosophila-Inspired 3D Moving Object Detection Based on Point Clouds. Information Sciences, 2020, 534, 154-171. https://doi.org/10.1016/j.ins.2020.05.006
37. Wang, Q., Chen, J., Deng, J., Zhang, X. 3D-CenterNet: 3D Object Detection Network for Point Clouds with Center Estimation Priority. Pattern Recognition, 2021, 115, 107884. https://doi.org/10.1016/j.patcog.2021.107884
38. Wang, X., Girshick, R., Gupta, A., He, K. Non-Local Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 7794-7803. https://doi.org/10.1109/CVPR.2018.00813
39. Wu, Y., Ogai, H. Realtime Single-Shot Refinement Neural Network for 3D Object Detection from LiDAR Point Cloud. In 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), IEEE, 2020, 332-337. https://doi.org/10.23919/SICE48898.2020.9240279
40. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. arXiv preprint arXiv:1711.00199, 2017. https://doi.org/10.15607/RSS.2018.XIV.019
41. Xie, Q., Lai, Y.-K., Wu, J., Wang, Z., Zhang, Y., Xu, K., Wang, J. MLCVNet: Multi-Level Context VoteNet for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 10447-10456. https://doi.org/10.1109/CVPR42600.2020.01046
42. Yan, Y., Mao, Y., Li, B. SECOND: Sparsely Embedded Convolutional Detection. Sensors, 2018, 18, 3337. https://doi.org/10.3390/s18103337
43. Yang, Z., Sun, Y., Liu, S., Jia, J. 3DSSD: Point-Based 3D Single Stage Object Detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 11040-11048. https://doi.org/10.1109/CVPR42600.2020.01105
44. Ye, Y., Chen, H., Zhang, C., Hao, X., Zhang, Z. SARPNet: Shape Attention Regional Proposal Network for LiDAR-Based 3D Object Detection. Neurocomputing, 2020, 379, 53-63. https://doi.org/10.1016/j.neucom.2019.09.086
45. Zhang, G., Davoodnia, V., Sepas-Moghaddam, A., Zhang, Y., Etemad, A. Classification of Hand Movements from EEG Using a Deep Attention-Based LSTM Network. IEEE Sensors Journal, 2019, 20, 3113-3122. https://doi.org/10.1109/JSEN.2019.2956998
46. Zhang, H., Mekala, M., Nain, Z., Park, J. H., Jung, H.-Y. 3D Harmonic Loss: Towards Task-Consistent and Time-Friendly 3D Object Detection on Edge for Intelligent Transportation System. arXiv preprint arXiv:2211.03407, 2022. https://doi.org/10.1109/TVT.2023.3291650

47. Zhang, H., Yang, D., Yurtsever, E., Redmill, K. A., Özgüner, Ü. Faraway-Frustum: Dealing with LiDAR Sparsity for 3D Object Detection Using Fusion. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, 2646-2652. https://doi.org/10.1109/ITSC48978.2021.9564990

48. Zhang, J., Chen, L., Ouyang, B., Liu, B., Zhu, J., Chen, Y., Meng, Y., Wu, D. PointCutMix: Regularization Strategy for Point Cloud Classification. Neurocomputing, 2022, 505, 58-67. https://doi.org/10.1016/j.neucom.2022.07.049

49. Zhang, Q., Yang, Y.-B. ReST: An Efficient Transformer for Visual Recognition. Advances in Neural Information Processing Systems, 2021, 34, 15475-15485.

50. Zhang, X., Zhou, X., Lin, M., Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 6848-6856. https://doi.org/10.1109/CVPR.2018.00716

51. Zheng, W., Tang, W., Chen, S., Jiang, L., Fu, C.-W. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector from Point Cloud. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35, 3555-3562. https://doi.org/10.1609/aaai.v35i4.16470

52. Zheng, W., Tang, W., Jiang, L., Fu, C.-W. SE-SSD: Self-Ensembling Single-Stage Object Detector from Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, 14494-14503. https://doi.org/10.1109/CVPR46437.2021.01426

53. Zhou, Y., Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud-Based 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 4490-4499. https://doi.org/10.1109/CVPR.2018.00472