Journal of Computers Vol. 32 No. 1, 2021, pp. 242-255
doi:10.3966/199115992021023201020

Method of 3D Vehicle Object Detection Based on Improved VoxelNet

Yi-Fan Zhao, Shao-Bo Wu*, Shi-Peng Dong

School of Information & Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China
yfzhao_email@163.com, wushaobo@bistu.edu.cn, 2609575722@qq.com

Received 20 December 2020; Revised 20 January 2021; Accepted 27 January 2021

Abstract. VoxelNet is a classical end-to-end target detection architecture, only the point cloud is used as the input to generate high precision 3D bounding box. However, the network performs the same operation for all points in the scene, without taking into account the close-density and far-sparse characteristics of the LiDAR, which makes different points may have different importance for target detection. And it's not negligible that a large amount of calculation in 3D convolution makes the inference speed slow. In consideration of the above deficiencies, this paper proposes AttentionVoxelNet model by introducing attention mechanism and sparse convolution. This method learns the weights of points by attention network and fuses them with features in the stage of voxel feature extraction so that paying more attention to the features of the points with high importance. Sparse convolution is also used to replace the 3D convolution to improve detection efficiency. Finally, through the vehicle detection experiment on the KITTI dataset, the average precision of detection and inference speed are compared with those of the classical algorithm under three difficulty modes. The results show that relatively accurate bounding boxes are generated in the point cloud and bird's eye view scenes. Then contrasted with VoxelNet, the average precision has been greatly improved in the simple, medium, and difficult modes, increasing by 5.98, 12.12, and 8.47 percentage points respectively, it also saves more than half of the time. Experiments illustrate that the proposed method is effective in introducing attention mechanism and sparse convolution, and it's meaningful compared with the current 3D vehicle detection structure.

Keywords: 3D vehicle detection, point cloud, voxel-based, attention mechanism

1 Introduction

As one of the significant modules of automatic driving, environmental perception realizes the recognition of road scene information by vehicle sensors and provides support for decision-making control. Vehicle detection plays an extremely prominent role in the recognition process. The essence of vehicle detection is to extract the region of interest from the real scene, which not only needs to identify the vehicle target but also determines its location and size to generates the corresponding bounding box, providing the necessary vehicle data information for the intelligent transportation system.

Within the scope of deep learning, it has achieved a series of significant research and results in 3D target detection based on 2D images. However, influenced by many elements such as distance, the intensity of illumination, camera resolution, weather effect, etc., there are still many problems in the expression of 3D information. Laser radar is gradually in an irreplaceable position in road scene understanding because of its strong anti-interference ability and the characteristics of high precision. The point cloud data generated by LiDAR have the following characteristics. (1) The point cloud are 3D and discrete point sets, which is a set of depth points without adjacency, and there is no fixed topological structure between each point, so the storage is relatively convenient and difficult to be affected by

* Corresponding Author

external factors. (2) Point cloud data can provide physical information such as intensity and color, as well as more geometric information, which makes the 3D detection pays attention not only to the location of the target in the image plane but also to its posture and other information. (3) LiDAR detection accuracy can reach centimeter-level, and the maximum detection radius can reach 100m, in the field of vehicle detection, it is possible to continuously track and obtain the trajectory of the vehicle in the detection area. Therefore, vehicle target detection based on point cloud has its unique advantages over image-based methods and has become an important way to obtain traffic information data. The research on more intelligent and robust object detection algorithm based on LiDAR is also one of the frontier directions of deep learning at present.

Object detection algorithm in the field of deep learning is based on mass training data. Feature extraction network can automatically learn the features of each layer, thus fully expressing the high-level semantic information in the scene. So it greatly improves the detection accuracy and speed and has strong applicability in a variety of complex tasks. Similar to 2D, 3D detection method based on deep learning is mainly divided into the following two kinds according to the principle: candidate region-based method (two-stage) and regression-based method (one-stage). The two-stage approach firstly extracts candidate regions by selective search or Region Proposal Network (RPN), then classifies and predicts candidate regions by regression. This kind of algorithm has high detection accuracy but slow speed, representative algorithms are MV3D [1], F-Pointnet [2], PointRCNN [3], VoteNet [4], etc. One-stage approach carries on the regression in the scene, locates and recognizes the object directly, which greatly improves the detection speed, such as 3D FCN [5], YOLO3D [6], PIXOR [7], VoxelNet [8]. This kind of method can predict the bounding box by sending the original data to the network only once. The end-to-end characteristics make it very suitable for the mobile terminal, which is also the development trend of vehicle detection algorithm. VoxelNet is a typical architecture for object detection by inputting point cloud directly. Point cloud feature coding layer is introduced to encode the data in 3D voxel into a unified feature dimension, and the 3D point cloud data is no longer transformed into Bird's Eye View (BEV) for manual feature representation. In addition, RPN mechanism is introduced to propose an end-to-end point cloud object detection network. The state-of-the-art performance of the 3D obstacle detection task on the KITTI dataset was obtained when it is proposed. The subsequent SECOND [9] based on VoxelNet design the sparse convolution middle layer, which has the improvement in speed.

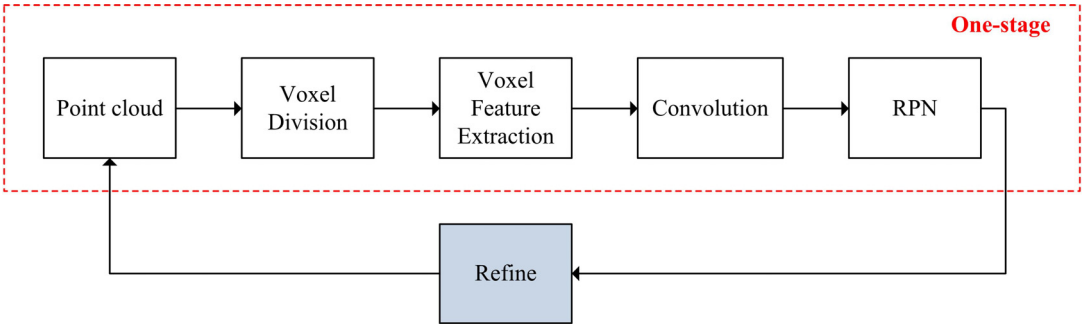


Fig. 1. One-stage and two-stage structures with voxel-based as examples. The whole part of the figure represents the two-stage, and the part contained in the red box represents one-stage

Although excellent experimental performance has been achieved for target detection of point cloud-based, problems remain. The current network treats all input point clouds equally, but considering the inhomogeneous density of points, the importance of points in the scene is different. Meanwhile, the traditional module of 3D convolutional network contains large calculations, which show a slow detection speed. Aiming at these problems above, in order to improve the detection accuracy and speed under the application background of vehicle detection, this paper proposes AttentionVoxelNet, a 3D detection network that introduces the attention network into the classical VoxelNet. The main research contents are as follows:

(1) The attention mechanism is combining with the feature learning stage to obtain the corresponding weights while acquiring the features, which makes the meaningful points or features prominently.

(2) We design the sparse convolution substitution for 3D convolution to increase the speed of network.

(3) We conduct the vehicle detection experiment on the KITTI dataset. The training and testing results

show that compared with VoxelNet, the average accuracy of 3D detection in simple, medium, and difficult modes is increased by 5.98, 12.12, and 8.47 percentage points respectively. Meanwhile, the reasoning speed is 2.5 times that of VoxelNet. The experiments also indicate that the proposed approach has a certain practical significance with reference to the existing 3D object detection methods.

2 Related Work

2.1 3D Object Detection

With the rapid rise of autonomous driving and computer vision, 3D target detection has been more deeply studied. It no longer outputs a 2D bounding box for the object but outputs a 3D bounding box for accurately determining the location of the target in 3D space. In general, 3D target detection is summarized from three aspects: 2D image, BEV, and direct processing of point cloud.

Image-based 3D Object Detection. In terms of the principle of photography geometry, it is impossible to accurately restore the 3D position of the object by relying solely on one image. Even if the relative position information can be obtained, there is still a lack of real size information. Therefore, this kind of algorithm uses monocular or stereo camera images to restore the pose information of the object, or through multiple cameras to form a stereo vision system. Proposed in the initial stage, Mono3D [10] first generates the 2D bounding box, the target category data, and the instance segmentation data, then through the methods of manually designing features to generate the feature map. Multi-Level3D [11] utilizes multi-level fusion of monocular images to predict 3D bounding box information. Subsequently, with the application of machine learning and deep learning, Mousavian proposed Deep3Dbox [12], an algorithm for 3D target detection (mainly for vehicles) and 3D position estimation, which uses the deep neural network to regress the relatively stable 3D target features, then by the convolution network to obtain the 2D bounding box information as the geometric constraint of the 3D pose of the object. It is necessary to know the position of the vehicle in the scene, the angle of the vehicle relative to the camera, and the size of the vehicle, then the projection formula is used to calculate the center point vector. GS3D [13] is based on guidance and surface, which is completed by Buyu Li et al., Chinese University of Hong Kong. Firstly, the 2D detection box and observation angle are predicted, then the rough bounding box of the target is generated based on the prior scene, and reproject it to the image plane to calculate the surface features. Finally, through the sub-network classification learning from the reprojection features, obtain the refined 3D bounding box. Although some optimization has been achieved, for the lack of depth information, the above methods can only produce rough detection effects.

Bird's Eye View-based 3D Object Detection. The point cloud is projected onto the aerial view or the front view, and the classification feature and bounding box regression feature of the point cloud are learned by convolution neural network. MV3D [1] is the first method to use BEV, which proposes a method to convert point cloud into multi-channel 2D aerial images, the density and intensity of the BEV point cloud are converted into two 2D feature images. ComplexYOLO [14] uses YOLO [15] as the target detection framework, encodes the angle into a 2D vector, so as to obtain higher execution speed and better angle regression performance. PIXOR [7] improved BEV coding and designed a high-speed area recommendation network. HDNET [16] inherited the design of PIXOR and carried out pixel-wise prediction. The main improvement is that High-Definition Map is added to the input, which makes the neural network have the guidance of geometric and semantic perception in the learning of BEV. BirdNet [17] projects the input point cloud to a 3-channel BEV map, and the channel is height, intensity, and density in turn. Then Faster R-CNN [18] is used for 2D object oriented detection on BEV. Finally, combine the detection results and Ground estimation, offline for 3D object oriented detection. But the above BEV-based method will lose much information, resulting in the Z-axis information can not be fully taken into account, making these methods in 3D target detection performance will decline.

Point Cloud-based 3D Object Detection. The Point cloud data directly as the input of the network model, mainly divided into point-based and voxel-based two categories. In terms of feature extraction of independent points, PointNet [19] is a 3D deep learning model directly acting on disordered point cloud. Applying symmetric function to the processing of 3D point cloud, and the high-level features are obtained by max pooling. It takes the point cloud as the input of network and presents the architecture for the tasks of classification and segmentation. Pointnet + + [20] algorithm optimizes local features while focusing on global features to obtain better results. PointNet series performs well in point cloud

classification and object instance segmentation, but it is not applicable for 3D detection. To improve the work efficiency of PointNet/PointNet + + and solve the problem of target spatial location, Qi et al. proposed Frustum PointNets [2], introduced 2D detection based on RGB images, and extracted the cone in the space through the 2D detection results, then utilized PointNet/PointNet++ to segment the target instance. Finally, a non-modal boundary frame pre-test model was used to predict the 3D bounding box, so that PointNet / PointNet + + could carry out the 3D target detection task. In 3D FCN [5], point cloud data is discretized into binary grids and then 3D convolution is used. Further, PointCNN [21] uses the K-nearest neighbor method to obtain the adjacent points of the midpoint to learn the spatial information in the point cloud. These methods directly take point cloud data as input and then apply one-dimensional convolution or one-dimensional convolution based on a certain neighbor method. However, they cannot be directly used to process large-scale point cloud. PointRCNN [3] is a two-stage detection framework. Stage-1 sub-network divides the point cloud of the whole scene into foreground and background points and generates a small number of high-quality 3D proposals directly from the point cloud in a bottom-up manner. Stage-2 sub-network converts the pooled points of each proposal into standard coordinates to better learn the local spatial features. But the drawback of this method cannot be ignored is time-consuming.

For the voxelization method, point cloud with irregular distribution is transformed into grid representation of regular distribution. Voxel is a unit that divides the space of point cloud regularly, which includes multiple points in the subspace. This method retains the spatial position information of points. In the initial stage of the voxel-based method, the voxel feature extraction is the Multi-Layer Perception extraction, and the corresponding backbone network is the 3D convolutional neural network that is easy to implement. However, the point cloud data is often sparsely distributed after being transformed into voxel representation, many voxels do not contain 3D points, but still occupy the memory, which brings a lot of unnecessary computational overhead and reduces efficiency. FPNN [22] proposed some methods to deal with sparse problems but still operate on sparse voxels. Vote3D [23] converts the point cloud into voxel with feature vectors afterward uses a voting-based algorithm to avoid the huge amount of computation required for 3D convolution. Vote3Deep [24] further takes advantage of the sparsity of point cloud, which proposes a new voting mechanism with feature-centric to achieve a new convolution, thereby improving the calculation speed. However, due to the use of manual design features, the above methods are not suitable for the complex environment faced by autonomous driving. VoxelNet [8] combines the sparse point cloud structure in 3D target detection has made breakthrough progress and proposes an end-to-end efficient scheme. It's a general 3D object detection model, which can integrate feature extraction and boundary estimation into one-step trainable deep network. After random sampling and normalization of the points, VoxelNet extracts local features from each non-empty voxel using several Voxel Feature Encoding layers. Then the feature is further abstracted by 3D Convolutional Middle Layers, the target classification, detection, and location regression are completed by RPN finally.

In order to solve the problem of high calculation consumption of 3D convolution, SECOND [9] designs an effective sparse convolution model based on VoxelNet, which greatly improves the detection speed, but still does not take into account the difference in the importance of points. [25] proposed a new convolution dedicated to sparse 3D data, which does not change the activation point position of sparse data, and further improves the calculation speed at the expense of accuracy. In [26], this new sparse 3D convolution is used in 3D semantic segmentation task. At present, the summary indicates there are few methods to use 3D sparse convolution in point cloud target detection.

2.2 Attention Mechanism

Attention mechanism originates from the research on human vision. In cognitive science, due to the bottleneck of information processing, human beings will selectively focus on a part of all information, invest more attention resources from this focus area to obtain more details of the target that need to be concerned, and suppress other useless information. Generally speaking, the attention mechanism in various advanced fields is essentially similar to the visual attention mechanism of human beings and the core goal is also to extract more critical factors for the current task. With the extensive research on deep learning, attention mechanism has been studied in multiple disciplines such as natural language

processing [27] and image recognition [28]. In these fields, the attention mechanism can also be regarded as a methodology.

Attention mechanism has been widely used is derived from [29] that proposed by Google Mind team in 2014, the attention mechanism is applied to the image classification task on the circular neural network and achieved remarkable results. On this basis, [30] used similar attention mechanisms in machine translation tasks, and carried out translation and alignment simultaneously which also achieved great results. This is the first recognized application of attention mechanisms in the field of natural language processing. Taking this as an opportunity, attention mechanisms are introduced into the field of deep learning. In the process of image recognition, the attention mechanism is to calculate the weighted sum of important features in a graph, and repeatedly emphasize the importance of the feature, so as to better learn the features and improve the performance of the model. In the 2017 ImageNet classification competition, Hu Jie et al proposed SENet [31] won the competition. This network uses a channel-based attention architecture, which focuses on the relationship between different channels, then obtains the importance of the features about different channels through modeling, and assigns weights through input data according to different tasks, which is simple and effective. Subsequently, CBAM [32] combines the spatial and channel attention mechanism modules, and the spatial attention mechanism pays attention to the different regions in the feature map. It has achieved better results compared with SENet, which only focuses on channel. Also in the application of autonomous driving, attention mechanism has made great progress in the latest research work. Donghoon Chang [33] designed a more optimized multi-lane detection scheme. The accuracy of lane segmentation is 99.87 % at an average speed of about 55 FPS on the corresponding dataset. To sum up, it is of great significance to introduce the attention mechanism into the point cloud-based deep learning network.

The structure of attention mechanism is shown in Fig. 2. Given input x, the feature $T_i(x)$ and the weights between features $W_i(x)$ is obtained by network learning, then through the attention mechanism, the output is the feature $F_i(x)$ with weight information, as given in Eq. 1.

$$F_i(x) = T_i(x) \cdot W_i(x) \tag{1}$$

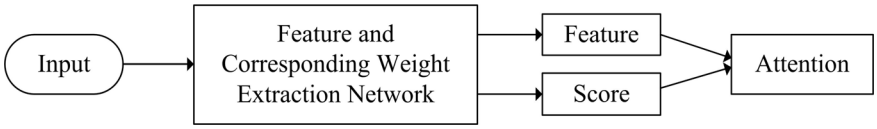


Fig. 2. Attention mechanism

3 Approach

In this chapter, we design the architecture of the improved method shown in Fig. 3, which is composed of three phases: Feature Extraction Network Introducing Attention Mechanism, Sparse Convolution, and RPN. It first generates voxel features integrating importance information of points, followed by abstracts and aggregates voxel features to add more content to shape description through Sparse Convolution Layer, the RPN in the traditional method is used for vehicle detection and location regression finally. We mainly introduce from the above three parts.

3.1 Feature Extraction Network Introducing Attention Mechanism

The structure of the classical voxel-based feature extraction network includes the following steps: voxelization, random sampling, and voxel feature learning of non-empty voxel. The approach proposed in this paper, AttentionVoxelnet, combines the attention network with the above architecture.

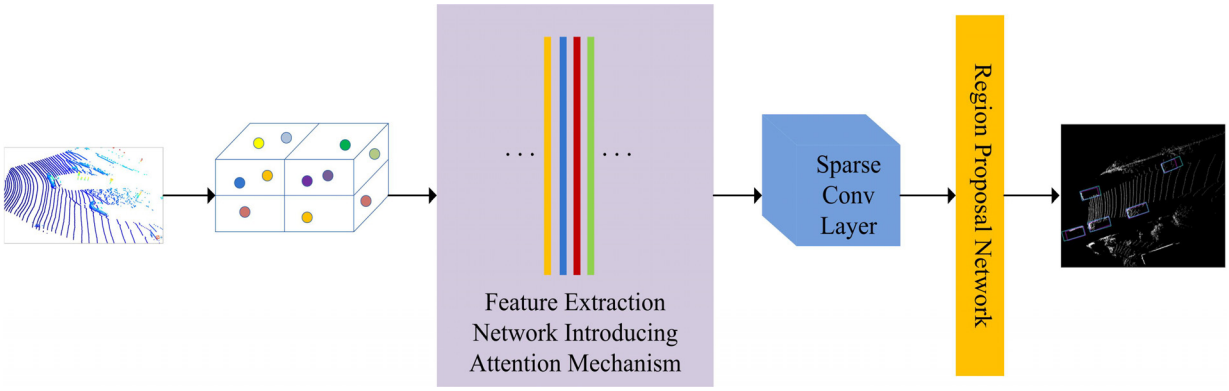


Fig. 3. Network architecture

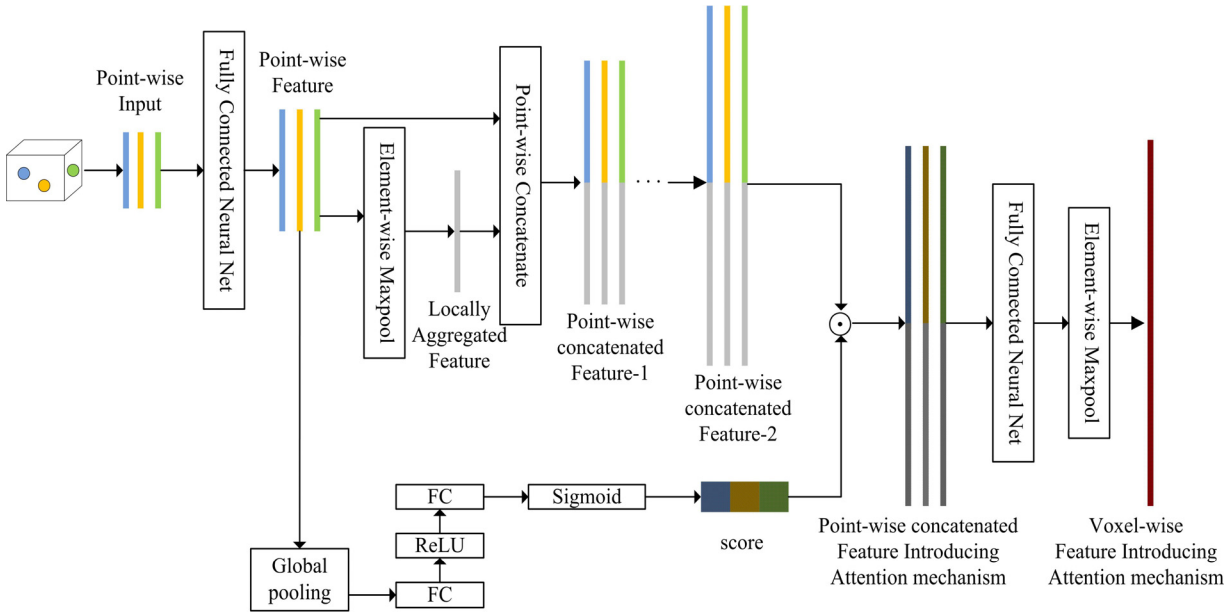


Fig. 4. Voxel feature extraction layer

Point-wise Feature Extraction. This article is still based on classic Voxel Feature Encoding module in VoxelNet [8]. Firstly, for the input of the point cloud scene, assume its size along the Z, Y, X axes is D, H, W respectively, and the corresponding size of each voxel cube is (v_D, v_H, v_W), then obtain the voxel grid size is $D' = D/v_D$, $H' = H/v_H$, $W' = W/v_W$ (we assume D, H, W is the multiple of v_D, v_H, v_W). After voxelization grouping, since the point cloud has density variability, each voxel will contain a variable number of points. Due to the large amount of point cloud data generated by LiDAR, it will increase more burden to process all. Moreover, the spatial density distribution of point cloud is highly variable, so it's necessary to preprocess the points in voxels. The maximum number of points in each non-empty voxel for car detection is set to $T = 35$, when the number of points is greater than T, random sampling is carried out so that only 35 points are retained in the voxel, otherwise, the points in the voxel remains unchanged.

For the point set $t <= T$ in a voxel, it first to calculate the mean value (v_x, v_y, v_z) of all points in the Voxel and take it as the centroid, then the combining the coordinates of the points and the reflection intensity r_i of laser beam, the feature of the points in a voxel is extended as follows:

$$V_{input} = \{p_i = [x_i, y_i, z_i, r_i, x_i - v_x, y_i - v_y, z_i - v_z]^T\}_{i=1...t} \quad \textbf{(2)}$$

The above-mentioned points containing 7-dimensional features will point-wise extend to 16-dimensional representation through the fully connected network. Followed by taking the max pooling,

which is operated between elements to obtain the locally aggregated feature, then we cascade it with the point-wise feature to generate 32-dimensional feature, which is named point-wise concatenate feature-1. Repeat the above steps, extend the feature dimension to 128, and get the point-wise concatenate feature-2 as input to the attention network.

Attention Network. The attention module mainly learns the feature weight of input points. Firstly, the feature is compressed through global pooling, next reduces the dimension by the first fully connected layer. Then returns to the original feature dimension through another fully connected layer after activation by the Rectified Linear Unit (ReLU) function. Finally, the sigmoid function is utilized to transform it into a normalized weight of 0 to 1. The closer to 0, the lower the importance is, and corresponding the closer to 1 means that it has higher importance.

After the above steps to generate the weight of each point, multiply it with point-wise concatenated feature-2, calculate the feature introducing attention mechanism, and then send the results to the max pooling layer to obtain the voxel-wise feature. So that it's easier to retain the features of meaningful points when operating the max pooling, thus the features of the points with high importance are highlighted, and with low importance are suppressed in this process.

3.2 Sparse Convolution

Due to the high sparsity and irregularity of the data generated by LiDAR, although the point clouds are voxelized by the above method, there are still a large number of voxel grids. Using the traditional 3D convolution to process these point cloud data will face two dilemmas. (1) Distorted features can be more likely to be extracted while extracting dense features from sparse data. (2) It will consume huge time and memory, which is not conducive to the real-time performance of operation. So sparse convolution is used instead of 3D convolution in AttentionVoxelNet model. In fact, sparse convolution means that to do convolution input only for voxels with points, and if the voxel is empty, it will not participate in the convolution calculation directly.

The sparse convolution used in this paper is shown in Fig. 5. Firstly, the spatial index of non-empty voxel in the original space is recorded, and its features are arranged into a column of maps. The convolution operation is also completed by calculating the index, that is, the final result is only obtained by index calculation in two dimensions. And ultimately the final-feature-map is returned to voxel expression through the final spatial index. At the same time, in order to further use the two-dimensional RPN network, we directly compress the H layer to the feature.

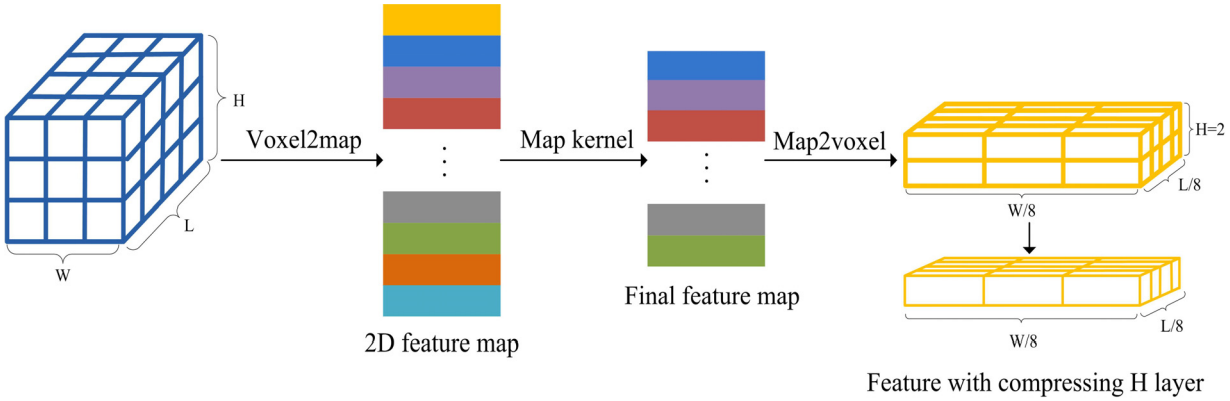


Fig. 5. Schematic diagram of sparse convolution

The receptive field is enlarged and the geometric space representation is learned to further abstract the features through the sparse convolution. Meanwhile, the information on the z-axis is merged into the channel information, and then the 4-dimensional feature tensor is transformed into the 3D feature tensor. Therefore, in the stage of vehicle detection and location regression, we can use the classical RPN in 2D.

3.3 Region Proposal Network

In 2D target detection, multi-scale RPN network is usually used to process multi-scale targets. These methods have well results in image-based target detection. However, due to the high complexity of point

cloud features, although we have transformed it into 3D feature tensor, it is still not suitable for complex RPN. Therefore, in the case of fixed size of the detection target, this paper designs a simplified RPN, that the structure is shown in Fig. 6, which only contains the several layers of convolution neural network. The input of the network is the 3D feature tensor generated by sparse convolution, that is, the 2D feature map. After the convolution neural networks, three 1×1 convolutions are directly used to obtain the classification output and the regression of bounding box. It is worth mentioning that the output of the RPN network also contains the orientation classifier so that the angle-related information can be introduced in the subsequent process of the loss function design.

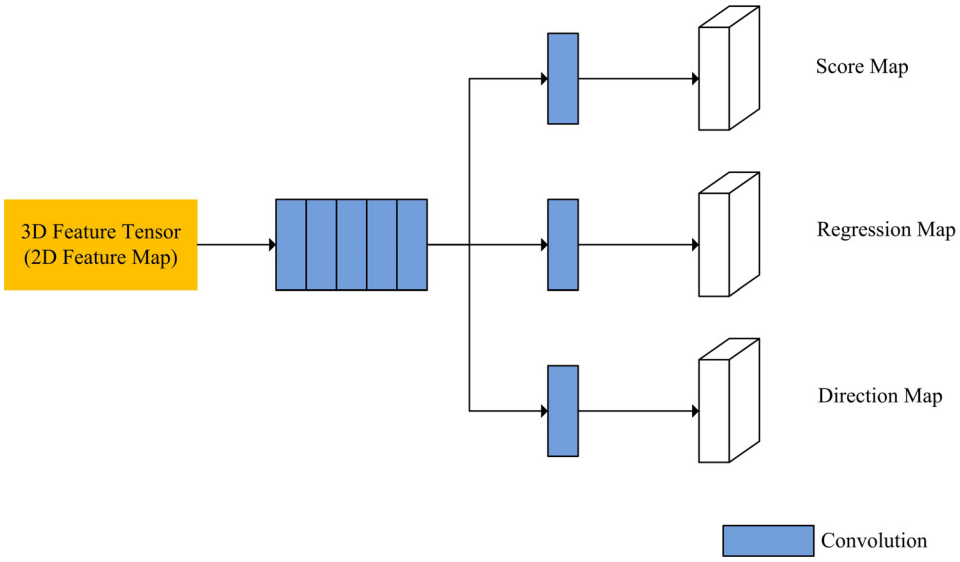


Fig. 6. The structure of Region Proposal Network

3.4 Loss Function

Loss function is an indispensable module in the design of deep learning network, which is mainly used to measure the inconsistency between the predicted value and the real value of the model. It is a non-negative real value function. The smaller the loss function is, the better the robustness of the network is.

In this paper, the module of loss function is designed as Eq. 3. L_{pro} is the module of region score loss function (classifier), which is consistent with the classifier module in Faster-RCNN. L_{reg} is the module of region location about the target (regression).

$$Loss = L_{pro} + L_{reg} \tag{3}$$

The 3D bounding box of the ground truth is parameterized as $(x_i^g, y_i^g, z_i^g, w_i^g, l_i^g, h_i^g, \theta_i^g)$, where x_i^g, y_i^g, z_i^g is the center position coordinate, w_i^g, l_i^g, h_i^g represent the width, length, height of the ground truth respectively, and θ_i^g is the angle of rotation around the z-axis. Similarly, the generated 3D bounding box is set to $(x_i^a, y_i^a, z_i^a, w_i^a, l_i^a, h_i^a, \theta_i^a)$. L_{reg} is defined as follows:

$$L_{reg} = \sum_{i=0}^{n}\sum_{j=0}^{s} R_{ij}[(x_i^a - x_i^g)^2 + (y_i^a - y_i^g)^2 + (z_i^a - z_i^g)^2 + (w_i^a - w_i^g)^2 + (l_i^a - l_i^g)^2 + (h_i^a - h_i^g)^2 + (\theta_i^a - \theta_i^g)^2] \tag{4}$$

Where n is the number of voxels about point cloud samples, s represents the number of candidate target bounding boxes, R_{ij} uses the $SmoothL1$ function [18]. In particular, based on the orientation classifier output in the above RPN, we also introduce the angle-related content into the loss function to make the information considered more complete.

4 Experiments

4.1 Experimental Details

Dataset and Implementation. This method is implemented using TensorFlow framework, training and testing on KITTI target detection benchmark. KITTI is a public dataset for autonomous driving, providing 7481 training scenarios and 7518 test scenarios. Each scene contains an RGB front view, registered point cloud data, and tags. According to the method proposed in reference [19], 7481 training data are divided into 3712 training sample point clouds and 3769 evaluation sample point clouds. Experiments were carried out in simple, medium, and difficult scenes, then compared with the traditional classical algorithm after test 3D Average Precision (3D AP), Bird's Eye View Average Precision (BEV AP) respectively. The classification of difficulty level is mainly based on the diversity, different size, perspective, and occlusion of the detection object, so that the model can be evaluated more comprehensively. Easy represents the completely visible target, moderate represents the partially occluded target, and hard corresponds to the target with severe occlusion or long distance.

Table 1. Experimental configuration

Item	CPU	GPU	System	CUDA
Content	Intel Xeon E5-2678 v3	NVIDIA GTX 1080 Ti	Ubuntu16.04	CUDA10.0

Parameter Setting. During the experiment process, the ranges of D, H, and W are set as [-3.0, 1.0], [-40.0, 40.0], [0, 70.4] (in meters), the points within this range are retained. We choose the size of each voxel as v_D =0.4, v_H =0.2, v_W =0.2 meters. The size of the anchor is fixed and its width, length, and height are 1.60, 3.90, and 1.56 meters respectively. The center of z is -1.0 meters. The training epoch is set to 160, and the batch is set to 2. According to the official standard of KITTI, for the vehicle image, only the detection results overlap more than 70 % of the 3D boundary with the label to be considered correct. Therefore, the IoU threshold of BEV AP and 3D AP is set to 0.7.

4.2 Experimental Results

The experiment is divided into two parts: training and test. We will evaluate our method from the perspectives of training effect and test results.

According to the above settings, in the case of the same training of 160 epoch, the training curve of VoxelNet and our method is shown in Fig. 7. It can be seen that in the simple mode, the data converges faster, while in the difficult mode, the convergence rate is relatively slow. Furthermore, the accuracy and convergence speed in the BEV situations are faster than those of the corresponding 3D status. And for various scenarios, with all its stability as the measurement standard, VoxelNet tends to converge after about 109 epochs, while the AttentionVoxelNet proposed in this paper has reached a relatively stable state when training close to 89 epochs. Especially in the difficult mode, 3D and BEV training still show well convergence performance. Obviously, our method has better convergence in training and more satisfactory results in three kinds of difficulty corresponding to BEV and point cloud scene.

We conduct verification experiments based on the training model. Extraction results of visual candidate region are shown in Fig. 8 as follows: KITTI 2D image, 3D detection visualization results, and BEV visualization results. As shown in part b, the point clouds representing vehicles in the scene are directly marked with 3D bounding box, which indicates that the detection result is effective and accurate, then we use BEV to compare for a more intuitive display of experiment results. The BEV map is generated by projecting the data of 3D point cloud space into 2D space, and the detection result of the target is obtained in the BEV map. The detection result in this form is a 2D detection box, which is illustrated in part c of Fig. 8. The red bounding box in the figure represents ground truth of the vehicle object in the BEV map, and the blue represents the detection result. It can be seen that the results of the BEV of point clouds are one-to-one correspondence with the results of the 3D detection box. The predicted bounding boxes also have high coincidence with ground truth. Meanwhile, in the process of visualization detection, with the improvement of the difficulty level, the number of vehicles in the scene

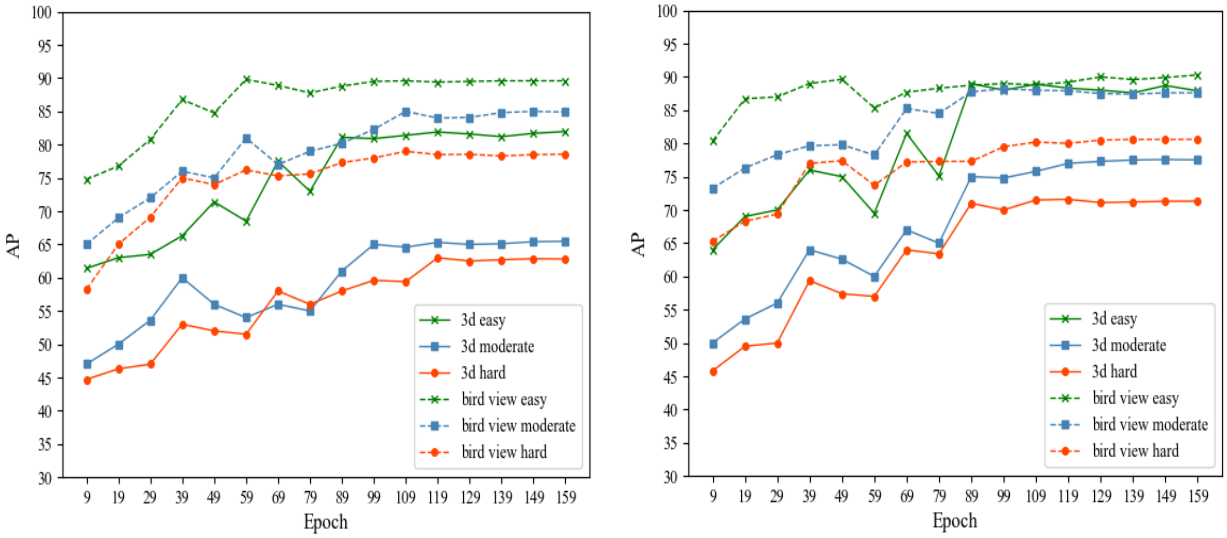
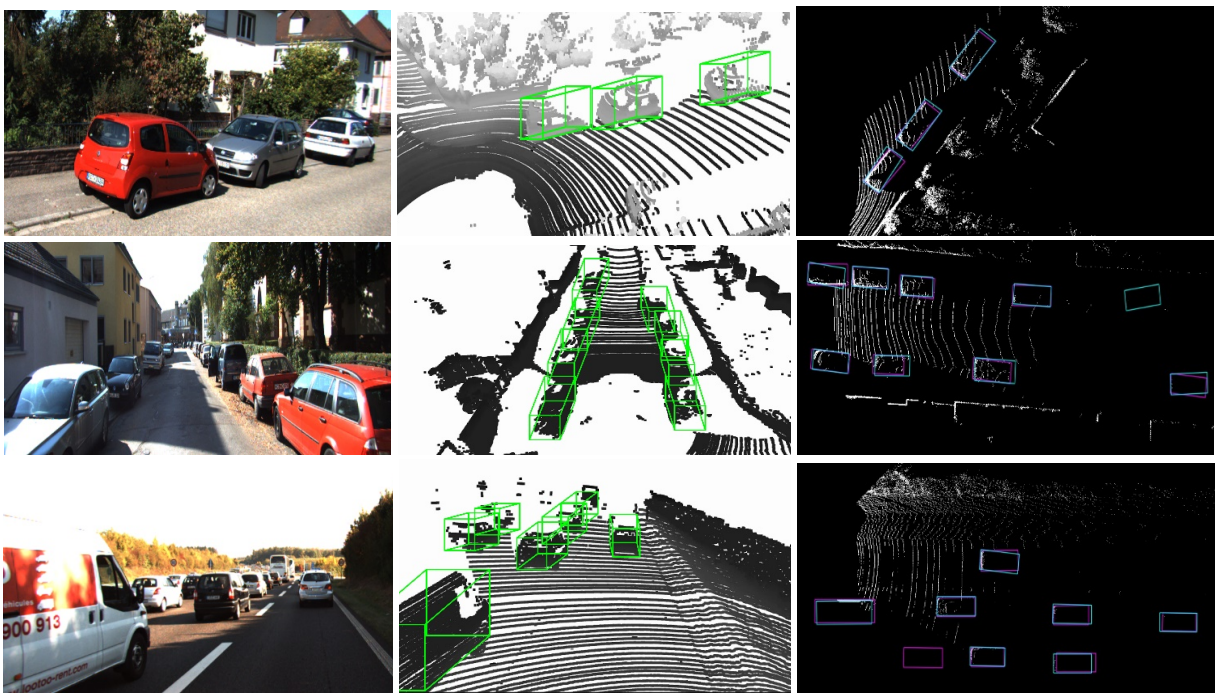


Fig. 7. Comparison of training curves



(a) 2D images of KITTI (b) corresponding 3D detection results on point cloud (c) detection results on BEV

Fig. 8. Visualization of 3D Vehicle detection on the KITTI validation set

increases and the occlusion becomes more and more serious. The analysis of the detection results shows that the miss ratio is very low in the range of LiDAR scene, and the prediction of the angle of vehicle orientation is more accurate.

It is analyzed with the classical algorithm from two aspects: BEV AP and 3D AP. The BEV detection results of vehicle are shown in Table 2. The data indicate that BEV AP of simple, moderate, and difficult level reached 90.28, 87.63, and 80.62 respectively. Compared with classical MV3D, F-PointNet, and VoxelNet, accuracy has been improved to some extent. Meanwhile, the reasoning speed is 0.09s, which is 60% less time than VoxelNet, and 4 times that of MV3D. Although there is a slight gap with SECOND in time, it is still superior in the current detection algorithms.

Table 2. Comparison of BEV vehicle detection AP for different models

Method	Modality	Easy	Moderate	Hard	mAP	Time[s]
MV3D	LiDAR+ Camera	86.55	78.10	76.67	80.44	0.36
F-PointNet	LiDAR+ Camera	88.16	84.02	76.44	82.87	0.17
SECOND	LiDAR	89.96	87.07	79.66	85.56	0.05
VoxelNet	LiDAR	89.60	84.81	78.57	84.33	0.23
Ours (AttentionVoxelNet)	LiDAR	90.28	87.63	80.62	86.18	0.09

The corresponding 3D vehicle detection results are shown in Table 3. BEV only detects position accuracy on the plane, it is more challenging that 3D bounding box needs to detect accurate pose information in 3D space. The analysis results signify that contrasted with VoxelNet, the 3D vehicle detection AP of the simple, medium, and difficult levels have been improved by 5.98, 12.12, and 8.47 percentage points respectively, mAP reached 78.95. It is worth mentioning that in medium and difficult modes with serious occlusion and a large number of vehicles, the 3D AP increases more, indicating that the network introducing attention mechanism can still extract feature information from remote sparse point clouds.

Table 3. Comparison of 3D vehicle detection AP for different models

Method	Modality	Easy	Moderate	Hard	mAP	Time[s]
MV3D	LiDAR + Camera	71.29	62.68	56.56	63.51	0.36
F-PointNet	LiDAR + Camera	83.76	70.92	63.65	72.78	0.17
AVOD-FPN	LiDAR + Camera	84.41	74.44	68.65	75.83	0.1
SECOND	LiDAR	87.43	76.48	69.10	77.67	0.05
VoxelNet	LiDAR	81.97	65.46	62.85	70.09	0.23
Ours (AttentionVoxelNet)	LiDAR	87.95	77.58	71.32	78.95	0.09

To sum up, the experimental results illustrate that the accuracy and inference speed of the vehicle detection algorithm based on attention mechanism has been significantly improved. In particular, the detection effect is still well with the scene becomes more complex, which means our approach is more optimized in medium and difficult modes. From the perspective of principle, for long-distance or high-occlusion vehicle targets, the point cloud that can represent their characteristics is relatively limited. After introducing the attention mechanism, to a certain extent, giving a higher weight to the key point cloud can highlight its weak feature representation. Thus the detection accuracy of complex targets is improved.

5 Summary

This paper designs a 3D target detection model AttentionVoxelNet by improving the classical voxelization network VoxelNet. It first utilizes the attention mechanism in the process of voxel features generation, making the feature combined with the weight learned by the attention network, so that the points with high importance can be highlighted and play a greater role in vehicle detection after the max pooling operation. Then the sparse convolution is used to replace the 3D convolution to improve the detection speed and further abstract the features.

Experiments of vehicle detection are carried out on KITTI dataset, the satisfactory visualization and data results show that the design effectively improves the accuracy and efficiency of detection. The average precision of this method is 87.95 %, 77.58 %, and 71.32 % in simple, medium, and difficult modes respectively, and the calculation speed is 0.09 s. Compared with VoxelNet, mAP for 3D vehicle detection increases by 8.86 percent points and saves about 60%t of the time. Especially, our framework shows better performance especially in the difficult mode of long distance and high occlusion. And this model also has certain advantages over other typical algorithms. The above results indicate that it has been significantly optimized by introducing attention mechanism and sparse convolution into object detection network. Meanwhile, the network does not change the original properties and structure of the point cloud, it is universal in theory and can be applied to any framework with point cloud as input.

Although AttentionVoxelNet has achieved satisfactory results in vehicle detection, there are still

limitations remain. (1) This paper inputs the complete road scene into the attention mechanism-based network. Despite the different importance of points is considered, the voxel division of the same size and the voxel feature coding of the same scale is still used for all point clouds. In the follow-up research, the point cloud can be divided into voxels of different sizes and feature extraction of different scales according to the difference in sparsity. (2) Our framework is based on point cloud refers to the advantages of LiDAR, but the sparse characteristics of point cloud can not be ignored, so there are some limitations in the positioning and regression of small targets. Therefore, the RGB image and our method can be fused with multiple modules in the next stage, which to fully combine dense texture information of the RGB image and complete depth information of LiDAR, further enhance the robustness of the target feature extraction. (3) This paper only tries to add attention model to the classic VoxelNet, we can also try to introduce attention mechanism to some more advanced algorithms in the future, and study the effect of different attention mechanisms on the target detection results.

Acknowledgements

Thanks for promoting the connotation development of colleges and universities - the graduate science and technology innovation project (number: 5112011036) provides support.

References

[1] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, Multi-view 3D object detection network for autonomous driving, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[2] C.R. Qi, W. Liu, C. Wu, H. Su, L.J. Guibas, Frustum PointNets for 3D Object Detection from RGB-D Data, arXiv: 1711.08488, 2017.

[3] S. Shi, X.Wang, H. Li, PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud, in: Proc. 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[4] C.R. Qi, O. Litany, K. He, & L. Guibas, Deep Hough Voting for 3D Object Detection in Point Clouds, in: Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[5] B. Li, 3D fully convolutional network for vehicle detection in point cloud, in: Proc. 2017 International Conference on Intelligent Robots and Systems (IROS), 2017.

[6] W. Ali, S. Abdelkarim, M. Zahran, YOLO3D: End-to-end real-time 3D Oriented Object Bounding Box Detection from LiDAR Point Cloud, in: Proc. 2018 European Conference on Computer Vision (ECCV), 2018.

[7] B. Yang, W. Luo, R. Urtasun, PIXOR: Real-Time 3D Object Detection From Point Clouds, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] Y. Zhou, O. Tuzel, VoxelNet: End-to-end learning for point cloud based 3D object detection, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[9] Y. Yan, Y. Mao, and B. Li, SECOND: Sparsely Embedded Convolutional Detection, Sensors, 2018.

[10] X. Chen, K. Kundu, Z. Zhang, H. Ma, Fidler S, Urtasun R, Monocular 3D object detection for autonomous driving, in: Proc. 2016 IEEE Conference on Computer Vision & Pattern Recognition (CVPR), 2016.

[11] B. Xu, and Z. Chen, Multi-level Fusion Based 3D Object Detection from Monocular Images, in: Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2018.

[12] A. Mousavian, D. Anguelov, J. Flynn, J. Košecká, 3D bounding box estimation using deep learning and geometry, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[13] B. Li, W. Ouyang, L. Sheng, GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving, in: Proc. 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[14] M. Simon, S. Milz, K. Amende, H.M. Gross, Complex-YOLO: Real-time 3D Object Detection on Point Clouds, arXiv: 1803.06199, 2018.

[15] J. Redmon, S. Divvala, R. Girshick, A. Farhad, You only look once: Unified, real-time object detection, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[16] B. Yang, M. Liang, R. Urtasun, HDNET: Exploiting HD Maps for 3D Object Detection, in: Proc. 2nd Conference on Robot Learning (CoRL), 2018.

[17] J. Beltran, C. Guindel, F.M. Moreno, BirdNet: a 3D Object Detection Framework from LiDAR information, in: Proc. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018.

[18] S. Ren, K. He, R. Girshick, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv: 1506.01497, 2018.

[19] C.R. Qi, H. Su, K. Mo, Guibas L.J, Pointnet: Deep learning on point sets for 3D classification and segmentation, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[20] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Proc. Advances in Neural Information Processing Systems, December (2017) 5105-5114.

[21] Y. Li, R. Bu, M. Sun, PointCNN: Convolution on X-transformed points, in: Proc. 2018 Conference and Workshop on Neural Information Processing Systems (NIPS), 2018.

[22] Y. Li, S. Pirk, H. Su, C.R. Qi, and L.J. Guibas, FPNN: Field probing neural networks for 3d data, in: Proc. 2016 Conference and Workshop on Neural Information Processing Systems (NIPS), 2016.

[23] D.Z. Wang and I. Posner, Voting for voting in online point cloud object detection, in: Proc. 2015 Robotics: Science and Systems, 2015.

[24] M. Engelcke, D. Rao, D.Z. Wang, C.H. Tong, I. Posner, Vote3deep: Fast object detection in 3D point clouds using efficient convolutional neural networks, in: Proc. 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017.

[25] B. Graham, L.V.D. Maaten, Submanifold Sparse Convolutional Networks, arXiv: 1706.01307, 2017.

[26] B. Graham, M. Engelcke, L.V.D. Maaten, 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[27] M.T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, Computer Science, 2015.

[28] L.C. Chen, Y. Yang, J. Wang, Attention to scale:Scale-aware semantic image segmentation, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[29] V. Mnih, N. Heess, A. Graves, Recurrent Models of Visual Attention, arXiv: 1406.6247, 2014.

[30] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, arXiv: 1409.0473, 2014.

[31] J. Hu, L. Shen, S. Albanie, Squeeze-and-Excitation Networks, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[32] S. Woo, J. Park, J.Y. Lee, I.S. Kweon, CBAM: Convolutional Block Attention Module, in: Proc. 2018 European Conference on Computer Vision (ECCV), 2018.

[33] D. Chang, V. Chirakkal, S. Goswami, Multi-lane Detection Using Instance Segmentation and Attentive Voting, in: Proc. The 19th International Conference on Control, Automation and Systems (ICCAS), 2019.