Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6228797, 9 pages
https://doi.org/10.1155/2022/6228797

Research Article

3D Object Detection from Point Cloud Based on Deep Learning

Ning Hao

School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China

Correspondence should be addressed to Ning Hao; haoning@shu.edu.cn

Received 2 May 2022; Revised 14 May 2022; Accepted 25 May 2022; Published 16 June 2022

Academic Editor: Kalidoss Rajakani

In order to study the modern 3D object detection algorithm based on deep learning, this paper studies the point-based 3D object detection algorithm, that is, a 3D object detection algorithm that uses multilayer perceptron to extract point features. This paper proposes a method based on point RCNN. A three-stage 3D object detection algorithm improves the accuracy of the algorithm by fusing image information. The algorithm in this paper integrates the information and image information of the three stages well, which improves the information utilization of the whole algorithm. Compared with the traditional 3D target detection algorithm, the structure of the algorithm in this paper is more compact, which effectively improves the utilization of information.

1. Introduction

Nowadays, with the rapid development of artificial intelligence and deep learning, many problems that are difficult to overcome by traditional machine learning can be successfully solved by using deep learning methods, such as image classification and detection [1], machine translation in natural language processing [2] and reading comprehension [3], and reinforcement learning [4]. At the same time, with the rapid development of high-performance computing devices, operating speed is no longer the main factor restricting the development of deep neural networks. These hardware and software factors have contributed to the vigorous development of deep learning.

Image analysis and detection is a major research hotspot in the computer field. Target detection plays an important role in traditional research. For example, the well-known open source image analysis library OpenCV already supports algorithms such as face detection tracking and vehicle detection tracking. Object detection technology is widely used in the military field. [5] applied target detection to bullet defect detection, and [6, 7] used target detection techniques to detect CT ships and radar targets. In the civilian field that is closely related to people's lives, object detection also has many applications in the civilian field. [8] applied object detection to helmet-wearing detection, and [9] used object detection to refine real-time pedestrian detection in orchards. Object detection is also widely used in the medical field. [10] used object detection to realize automatic diagnosis of diabetic fundus lesions. However, traditional methods need to manually design features to detect objects, which is time-consuming and labor-intensive. In the era of rapid development of deep learning, it is of great significance to apply deep learning to object detection.

Object detection technology also plays an important role in the field of autonomous driving. Autonomous driving has always been a research hotspot in the automotive field. Current autonomous driving systems can be divided into two types: driverless and ADAS (advanced driver assistance systems). This form focuses on fully driverless car driving to save the driver's labor cost, and the other focuses on assisting the driver, reducing the driver's stress while driving, and improving the safety of the vehicle. Both use various sensors installed on the vehicle to collect data and combine the map data for system calculation, so as to realize the planning of the driving route and control the vehicle to reach the predetermined position [11]. Breakthroughs in the field of artificial intelligence such as machine vision and deep learning are of great significance to the development of autonomous driving. A series of excellent artificial intelligence companies such as Mobileye have emerged to serve autonomous driving.

Self-driving cars typically use LiDAR (Light Detection and Ranging) and several cameras installed in different locations of the vehicle to collect perception data and then analyze and localize the collected visual data to locate objects such as lanes, vehicles, and pedestrians. Recently, with the rapid development of deep learning and artificial intelligence technology, the ability of computers to analyze image data has been significantly improved compared with traditional methods, and many companies have begun to develop artificial intelligence-specific chips. In intelligent driving systems, mainstream sensor solutions include LiDAR, cameras, and millimeter-wave radar (RADAR). The advantage of LiDAR lies in 3D modeling, wide detection range, and high detection accuracy. Therefore, deep learning target detection systems using LiDAR detectors are a hot research direction. However, there are still some problems in the current deep learning-based LiDAR detection systems. First, the data output by the LiDAR sensor is a sparse point cloud, not as dense as the image output by the camera, so traditional deep learning target detection methods cannot be used, directly used for LiDAR perception. The second is that the point cloud output by the LiDAR is three-dimensional data. Due to the time-consuming operation of traditional 3D convolution, current point cloud-based object detection methods usually run slowly. Therefore, it is of great significance to study the fast target detection of LiDAR sensors based on point cloud output. The improved point cloud target detection method with sparse convolution acceleration proposed in this paper has theoretical and practical significance and has certain research and application value. Model experiments are carried out, and it is proved that it can achieve high detection accuracy, reaching 79.51% (moderate). The accuracy of the model is 0.66% higher than the baseline, demonstrating the effectiveness of the third-level point cloud classifier and image classifier.

2. Related Work

We observe that point cloud-based object detection is closely related to point cloud-based 3D instance segmentation algorithms. For point clouds in 3D scenes, the 3D instance segmentation algorithm needs to give each point cloud a class label and individual instance labels and needs to distinguish different instances of the same class. Several approaches to 3D instance segmentation are based on 3D detection bounding boxes and an additional mask branch to predict masks for objects inside the boxes.

Reference [12] proposes a 2D-driven 3D object detection method, where they use hand-crafted features (hitograms based on point coordinates) to regress the position and pose of a 3D bounding box. Thanks to the advent of PointNet [13], a new possibility is provided to directly process native point clouds, which can directly learn features and identify objects in point clouds. [14] et al. used the latest deep point cloud 3D feature learning network PointNet [15] and proposed a more flexible and effective solution F-PointNet [16], by implementing 3D instance segmentation and finally obtaining 3D bounding boxes. It is estimated that experiments show that 3D segmentation in the point cloud can make the 3D localization clearer and more accurate. [17] By dividing the space into a series of small cubic grids, the point cloud of the cubic grid also uses the feature extraction method of PointNet and finally fills it into a space, using an end-to-end trainable deep neural network, intermediate features. The extraction stage adopts the form of 3D convolution and then uses 2D convolution by compressing the height dimension of the point cloud.

PointRCNN [18] proposes a new way of generating 3D proposal candidate frames from point clouds. By semantically segmenting the point cloud, a bin-based localization method is used to determine the object's location in the predicted foreground point, center point, while STD [19] proposes a method of taking each foreground point as the center of the object, using a spherical anchor to generate 3D region candidate proposals, and finally, by aggregating sparse point clouds into a more compact representation to predict the frame, further fine-tuning. [20] proposed a multitask learning framework to learn feature embeddings and orientation information of instance centers to better cluster points into instances.

PointRCNN is a paper on 3D target detection in CVPR2019. The article uses two-stage approach, using PointNet++ as the backbone network, to first complete the segmentation task to determine the label of each 3D point. For each point divided into foreground, a box is generated using FEATURE. Then, the box is roi crop for the optimization of the box. The previous methods are to do the box on the basis of the detected object; the method mentioned in this paper is to predict the box for all points and then remove the box predicted by the background points, then the box generated by the foreground points left basically contains the detection target, and then filter and optimize from these boxes to get the final prediction box.

3. The Proposed Model

The algorithm in this paper is based on the improvement of the PointRCNN network [21] and improves the detection accuracy of the algorithm by fusing image information. PointRCNN is a point-based 3D object detection algorithm, which itself has two stages: in the first stage, PointRCNN extracts the semantic feature of each point through a network and then uses this feature to separate the foreground points and extract the preselected frame. In the second stage, by fusing the semantic features and classification confidence of the first stage, PointRCNN further refines the candidate frame proposed in the first stage and then uses nonmaximum suppression (NMS, nonmaximum suppression) to screen the candidate frame to obtain the final result [22]. The network framework of PointRCNN is shown in Figure 1.

After researching the network, this paper finds that when the network extracts foreground points, some background points have a similar appearance to the foreground points in the form of point cloud, which is prone to misjudgment or omission. Therefore, this paper proposes a way to fuse image information that improves this situation. Mainly by adding a point cloud classifier and an image classifier. The framework of this paper is shown in Figure 2, where each stage is trained separately. In the next few chapters, this paper will first introduce the PointRCNN network; then, the method of fusing image information in this paper will be explained; then, the point cloud classifier and image

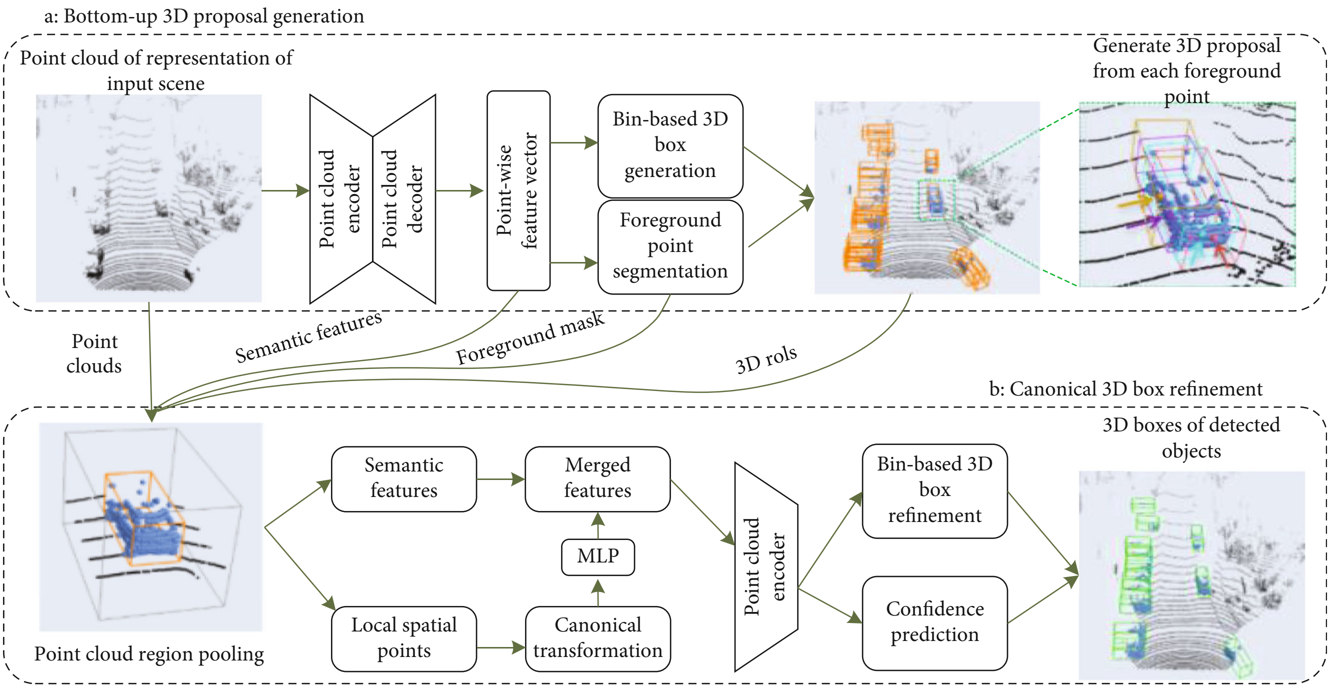

a: Bottom-up 3D proposal generation
Point cloud of representation of input scene
Point cloud encoder
Point cloud decoder
Point-wise feature vector
Bin-based 3D box generation
Foreground point segmentation
Generate 3D proposal from each foreground point
Point clouds
Semantic features
Foreground mask
3D rols
b: Canonical 3D box refinement
Semantic features
Merged features
MLP
Local spatial points
Canonical transformation
Point cloud encoder
Bin-based 3D box refinement
Confidence prediction
3D boxes of detected objects
Point cloud region pooling

Figure 1: Framework of PointRCNN.

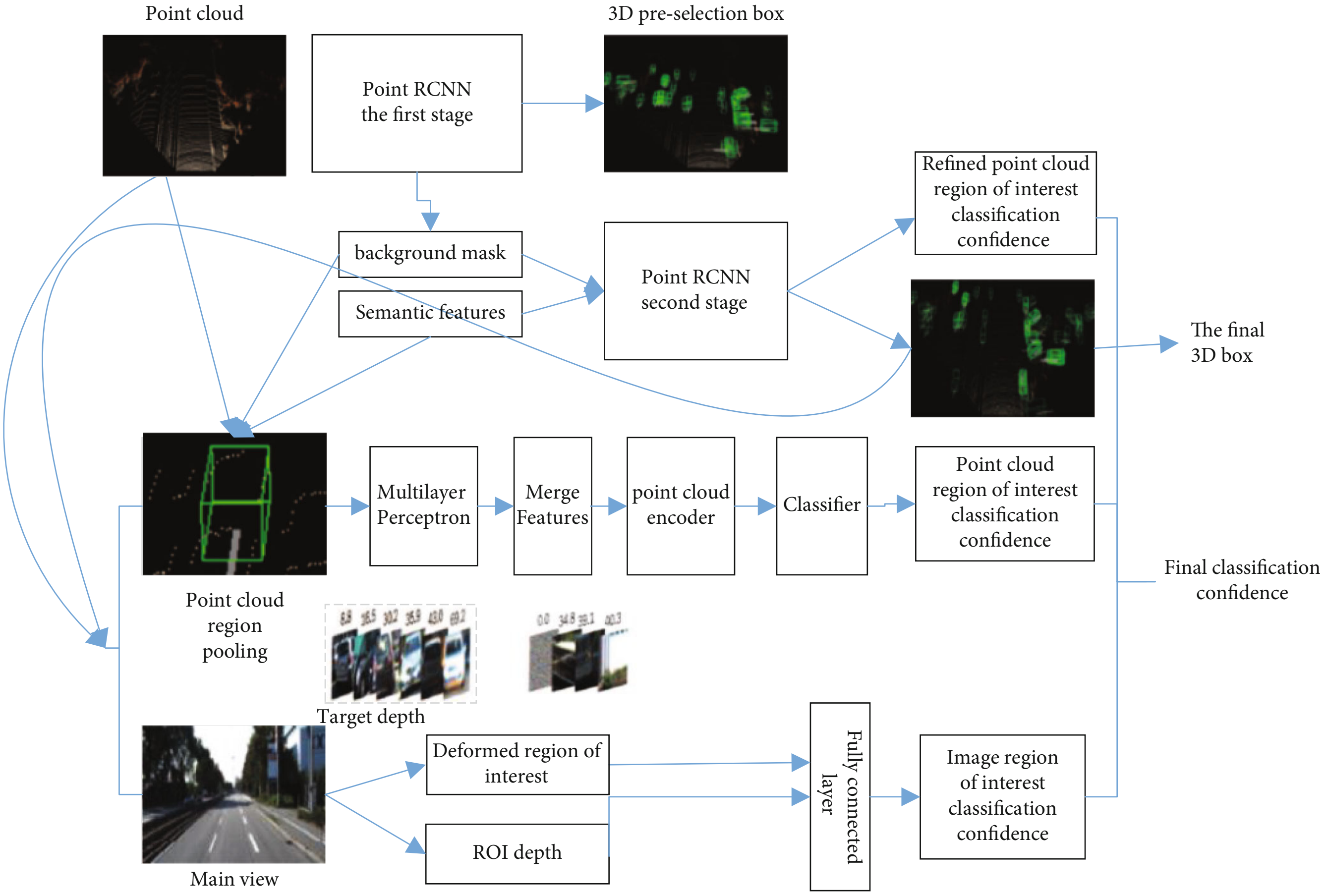

Point cloud
Point RCNN the first stage
3D pre-selection box
background mask
Semantic features
Point RCNN second stage
Refined point cloud region of interest classification confidence
The final 3D box
Multilayer Perceptron
Merge Features
point cloud encoder
Classifier
Point cloud region of interest classification confidence
Final classification confidence
Point cloud region pooling
Target depth
Main view
Deformed region of interest
ROI depth
Fully connected layer
Image region of interest classification confidence

Figure 2: Framework of 3D object detection combined with image.

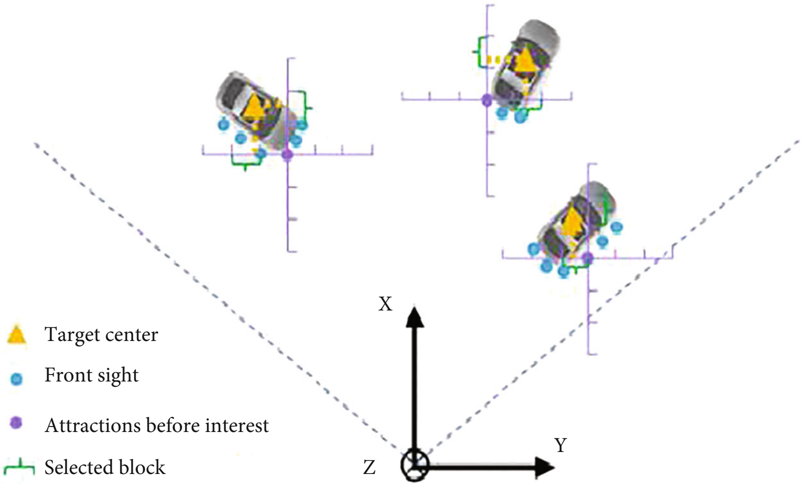


Figure 3: Figure of bin-based regressing method.

classifier of this paper, as well as the corresponding model structure will be introduced; finally, the experimental verification will be carried out and the speed and accuracy of the above algorithm.

3.1. PointRCNN Network. First, the network model needs to extract features from the point cloud data. In this paper, the PointNet network is chosen for feature extraction because it is fast and performs well in the semantic segmentation task. Next, foreground points and 3D preselected frames need to be extracted. The precursors themselves contain rich information that can be used to locate the position, size, and orientation of the target. The network model obtains the foreground points by learning semantic information [23]. Since this semantic information also helps to locate 3D bounding boxes, predicting both forespots and 3D preselected boxes can be mutually beneficial. For large outdoor site attraction clouds, the number of background points is much larger than the number of preattractions. Therefore, to solve the problem of positive and negative sample imbalance, focal is chosen for the foreground point prediction task in this paper. Loss is used as the classification loss, and its expression is shown in Equation (2).

$$L_{\text{focal}}(p_t) = -\alpha_t (1 - p_t)^\gamma \log (p_t), \tag{1}$$

$$p_t = \begin{cases} p, p_g = 1, \\ 1 - p, p_g = 0. \end{cases} \tag{2}$$

In the preselected box regression task, PointRCNN needs to predict a preselected box for each foreground point. The preselected box is in the form of $(x, y, z, w, l, h, \theta)$, P stands for semantic value, (x, y, z) is the center coordinate of the preselected box, (w, l, h) is the size of the preselector, and θ is the orientation of the preselected box in the top view. To limit the range of generated preselection boxes, PointRCNN uses a patch-based regression loss.

In order to predict the center coordinates of an object, PointRCNN divides the surrounding area under the top view of each foreground point into small blocks. Then, a one-dimensional search distance S is set, and according to this distance and a fixed parameter δ, it is determined which small block different targets are located in the top view. A small block categorical cross-entropy loss is also used here, which can improve the accuracy and robustness of localization. The schematic diagram of the block-based regression method is shown in Figure 3.

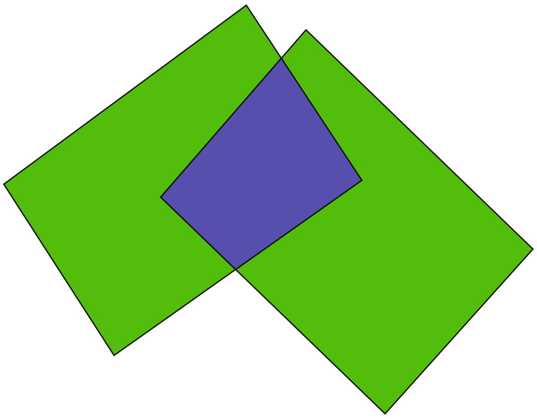


Figure 4: Image of IOU.

For the regression loss of (x, y), it consists of two parts, one is the classification loss function of which patch the target center is located in, and the other is the regression loss of (x, y) relative to this patch. For the regression of the z value, since the fluctuation of the target in the z direction is not large, the regression loss function is directly used [24]. Therefore, the regression target can be obtained by Equation (4).

$$\text{bin}_x^{(p)} = \left\lfloor \frac{x^p - x^{(p)} + S}{\delta} \right\rfloor, \text{bin}_y^{(p)} = \left\lfloor \frac{y^p - y^{(p)} + S}{\delta} \right\rfloor, \tag{3}$$

$$\underset{u\in\{x,z\}}{\text{res}_u^{(p)}} = \frac{1}{C}\left(u^p - u^{(p)} + S - \left(\text{bin}_u^{(p)} \cdot \delta + \frac{\delta}{2}\right)\right), \tag{4}$$

$$\text{res}_y^{(p)} = y^p - y^{(p)}. \tag{5}$$

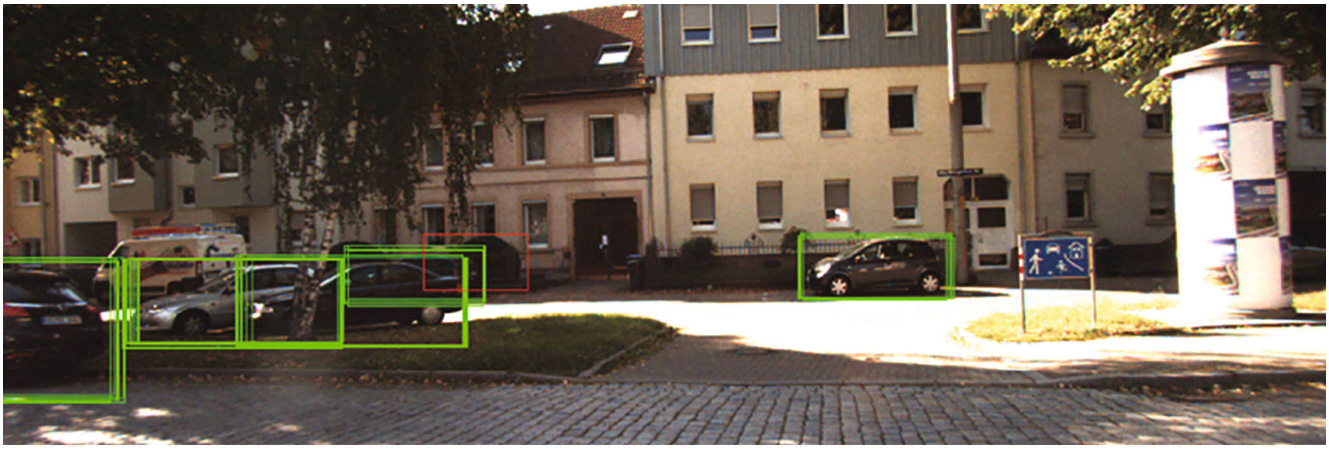


Figure 5: Result figure of proposals with high confidence projected to image.

$(x^{(p)}, y^{(p)}, z^{(p)})$ is the coordinate of the foreground point of interest, (x^p, y^p, z^p) is the center coordinate of the target corresponding to the point, $(\mathrm{bin}_x^{(p)}, \mathrm{bin}_y^{(p)})$ is the coordinate of the patch to which the label is mapped and is the distance of the point relative to the corresponding patch, and C is used for normalization parameter. For the parameter (w, l, h), the smooth $L1$ loss function is directly used for regression, and the expression of the smooth $L1$ function is shown in Equation (6). For θ, first divide 2π into n parts, and then, as above, predict which part of $\mathrm{bin}_\theta^{(p)}$ it belongs to and the angle value $\mathrm{res}_\theta^{(p)}$ relative to this part.

$$\mathrm{Smooth}_{L1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1, \\ |x| - 0.5, & \text{otherwise.} \end{cases} \tag{6}$$

Therefore, the overall loss function is shown in Equations (6)–(7).

$$L_{\mathrm{bin}}^{(p)} = \sum_{u \in \{x,z,\theta\}} \left(F_{\mathrm{cls}}\left(\widehat{\mathrm{bin}}_u^{(p)}, \mathrm{bin}_u^{(p)} \right) + F_{\mathrm{reg}}\left(\widehat{\mathrm{res}}_u^{(p)}, \mathrm{res}_u^{(p)} \right) \right), \tag{7}$$

$$\mathrm{L}_{\mathrm{res}}^{(p)} = \sum_{v \in \{y,h,w,l\}} F_{\mathrm{reg}}\left(\widehat{\mathrm{res}}_v^{(p)}, \mathrm{bin}_v^{(p)} \right), \tag{8}$$

$$L_{\mathrm{reg}} = \frac{1}{N_{\mathrm{pos}}} \sum_{p \in \mathrm{pos}} \left(L_{\mathrm{bin}}^{(p)} + L_{\mathrm{res}}^{(p)} \right). \tag{9}$$

Among them, N_{pos} is the number of foreground points, $\widehat{\mathrm{bin}}_u^{(p)}$ and $\widehat{\mathrm{res}}_u^{(p)}$ are the predicted small block to which the point belongs and the corresponding deviation value, F_{cls} is the cross entropy loss, and F_{reg} is the smooth $L1$ loss. In order to reduce redundant preselection boxes, in the training phase, this paper uses nonmaximum suppression algorithm to filter all preselection boxes. The 512 preselected boxes with the highest confidence are handed over to the second stage for refining. For the two preselected boxes b_1 and b_2, the calculation formula of IOU is shown in Equation (10), and its schematic diagram is shown in Figure 4. IOU is the area of the blue part divided by the area of the green part. In the prediction stage, a smaller number of preselected boxes are kept.

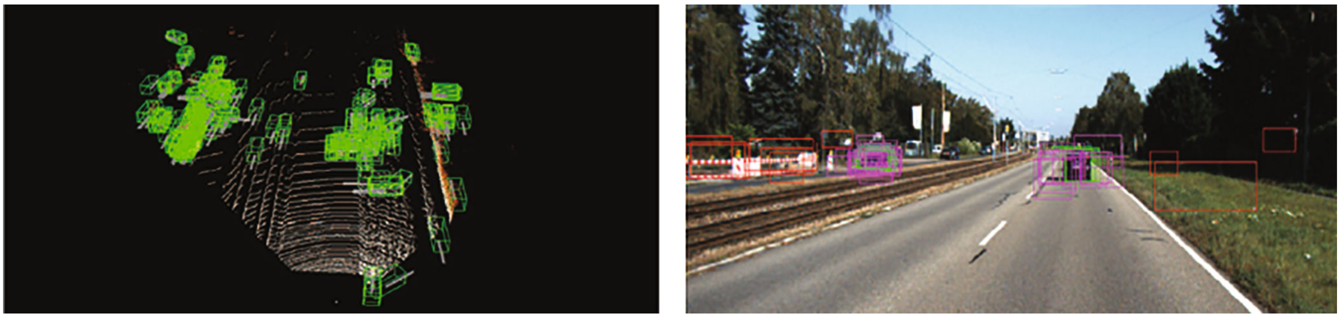
$$\mathrm{IOU} = \frac{b_1 \cap b_2}{b_1 \cup b_2}. \tag{10}$$

After obtaining the preselected boxes in the first step, these preselected boxes need to be further refined to obtain more accurate bounding boxes. For any preselected box $b_i = (x_i, y_i, z_i, h_i, w_i, l_i, \theta_i)$, PointRCNN slightly expands it to obtain nearby semantic information, namely, $b_i^e = (x_i, y_i, z_i, h_i + \eta, w_i + \eta, l_i + \eta, \theta_i)$.

For any point $p = (x^{(p)}, y^{(p)}, z^{(p)})$, see if it lies within b_i^e. If true, keep the point. The feature of the point in the preselected box consists of the following parts: its three-dimensional coordinate $(x^{(p)}, y^{(p)}, z^{(p)})$, its laser point intensity R, the mask value $m^{(p)}$ obtained in the first stage, and the semantic information $f^{(p)}$.

The above features are global features. In order to further refine the preselection box, the point in position b_i^e and the corresponding target point are converted into a standard coordinate system. This coordinate system must meet the following conditions: (1) the origin is located in the center of the prediction box. (2) The X-axis and the Y-axis should be parallel to the ground, and the X-axis should point to the direction of the prediction frame; (3) The Z-axis should be consistent with the original coordinate axis. Through rotation and translation, all internal points are transferred to this coordinate system to extract good local features.

Although this coordinate system can obtain local features well, it inevitably loses depth information. In order to preserve this information, a feature $d^{(p)} = \sqrt{(x^{(p)})^2 + (y^{(p)})^2 + (z^{(p)})^2}$ is defined. Finally, connect $[r^{(p)}, d^{(p)}, m^{(p)}]$ as the final local feature, and use a multilayer perceptron for feature extraction to obtain a feature with the same dimension as the first stage $f^{(p)}$. Then, concatenate this feature with $f^{(p)}$ to get a new feature. This feature is input into a network that extracts point cloud features to obtain the final refined 3D box and the confidence of the corresponding box. The box will



(a) 3D proposals

(b) ROIs of image

Figure 6: 3D proposals and ROIs of image.

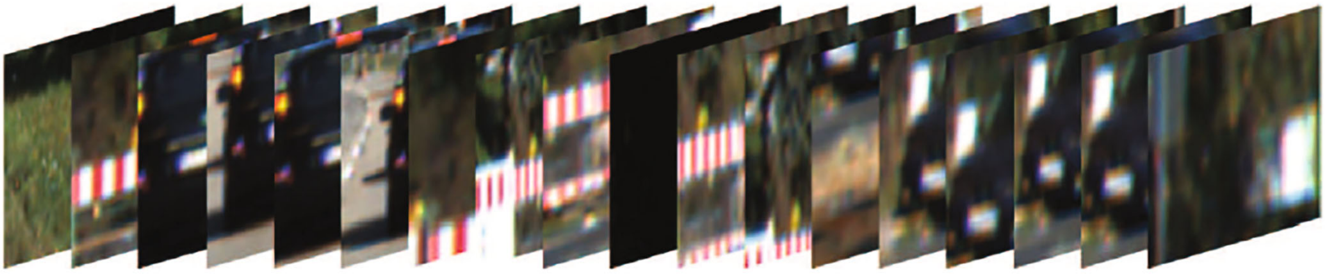


Figure 7: ROIs of resized image.

Table 1: The accuracy of the algorithm on the KITTI validation set.

Algorithm	Simple (%)	Medium (%)	Difficulty (%)	Speed (ms)
PointRCNN	89.20	78.85	77.91	157
Algorithms for fusing image information	89.76	79.51	78.58	392

Table 2: The accuracy of the algorithm on the KITTI validation set for 3D objects.

Algorithm	Simple (%)	Medium (%)	Difficulty (%)
L	89.20	78.85	77.91
$L+P$	89.61	79.25	78.38
$L+P'$	89.50	79.10	78.20
$L+I$	89.61	79.31	78.25
$L+P+I$	89.76	79.51	78.58

Table 3: Effects of algorithms on video memory, speed, and accuracy 1.

Algorithm	Display memory usage (MB)	Speed (ms)	Accuracy (%) (medium)
L	1997	154	78.85
$L+P$	2005	241	79.25
$L+I$	4597	308	79.31
$L+P+I$	4612	392	79.51

participate in the final backpropagation only if the IOU of the label box, and the box is greater than a threshold.

In terms of loss function, a loss function similar to that of the first stage is used here, but when the search distance S is selected, a smaller search distance is selected. And, since all preselection boxes and annotations will be transformed into the standard coordinate system, that is, the preselection boxes $b_i = (x_i, y_i, z_i, h_i, w_i, l_i, \theta_i)$ and the annotation boxes $b_i^{gt} = (x_i^{gt}, y_i^{gt}, z_i^{gt}, h_i^{gt}, w_i^{gt}, l_i^{gt}, \theta_i^{gt})$ will be transformed into Equations (11).

$$\tilde{b}_i = (0, 0, 0, h_i, w_i, l_i, 0), \tag{11}$$

$$\tilde{b}_i^{gt} = \left(x_i^{gt} - x_i, y_i^{gt} - y_i, z_i^{gt} - z_i, h_i^{gt}, w_i^{gt}, l_i^{gt}, \theta_i^{gt} - \theta_i\right). \tag{12}$$

Therefore, the loss function of the second stage is shown in

$$L_{\text{refine}} = \frac{1}{\|B\|} \sum_{i \in B} F_{\text{cls}}(\text{prob}_i, \text{label}_i) + \frac{1}{\|B\|} \sum_{i \in B_{\text{pos}}} \left(\tilde{L}_{\text{bin}}^{(i)} + \tilde{L}_{\text{res}}^{(i)}\right). \tag{13}$$

Among them, B is the preselected box obtained in the first stage, B_{pos} is the number of positive samples predicted, prob_i is the confidence level of the prediction, label_i is the corresponding label, and the remaining parameters are similar to formula (7). Then, after sorting the confidence of the final prediction box result, the maximum value is

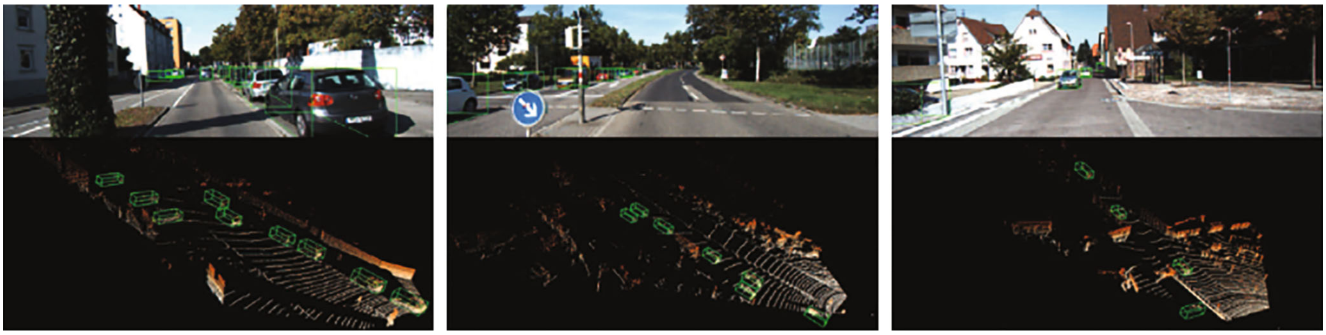


Figure 8: Result of 3D object detection combined with image.

suppressed, and the overlapping 3D boxes are filtered out to obtain the final result.

4. Algorithms for Fusing Image Information

After studying PointRCNN in this paper, it is found that in the form of point clouds, some foreground points have similar appearance characteristics to background points, so it is difficult to distinguish. However, it can be easily distinguished from the image corresponding to the point cloud, so a third stage is added [25]. In the third stage, an algorithm that uses image information to enhance the confidence of the prediction frame is proposed to improve the accuracy of the algorithm. As shown in Figure 5, the green bounding box is the preselection box that the second stage and the third stage agree with high confidence, the red is the preselection box that the third stage thinks is high confidence, and the second stage is low confidence frame. It can be seen that the algorithm in this paper can effectively identify distant foreground points after adding image information and improve the confidence of these preselected boxes.

The algorithm in this paper is a three-stage 3D target detection algorithm. After the second stage of PointRCNN, in order to improve the confidence of the second-stage preselected box, this paper adds a point cloud classifier and an image classifier in the third stage. By combining these two classifiers, the confidence of foreground boxes with similar appearance to the background is improved. This improves the recognition accuracy of the entire algorithm. After obtaining the preselected frame b_i from the first stage, for any point $p = (x^{(p)}, y^{(p)}, z^{(p)})$ in the point cloud, it is also checked whether it is located in b_i. If true, keep the point. The features of the points in the preselection box in the third stage are similar to those in the second stage and are composed of the following parts: its three-dimensional coordinate $(x^{(p)}, y^{(p)}, z^{(p)})$, its laser point intensity r, the mask value $m^{(p)}$ obtained in the first stage, and the semantics obtained in the second stage information $f^{(p)}$. These point clouds are then directly fed into a multilayer perceptron to extract features and obtain the final classification confidence. In order to obtain better results here, a higher intersection ratio threshold than the second stage is set, that is, when the intersection ratio between the preselected box and the annotation on the bird's eye view is greater than the threshold, it will be trained. In addition, this paper does not further regress the size of the preselected box, because the second stage of the algorithm has obtained a sufficiently accurate preselected box, and no further refinement is required. In order to improve the classification confidence, this paper uses the projection matrix in the annotation file to project the 3D prediction box into the corresponding images, as shown in Figure 6, and then convolves these images. In order to facilitate convolution and obtain outputs of the same size, this paper scales the corresponding images to the same resolution, as shown in Figure 7, and then uses the VGG-16 network for feature extraction. At the same time, the depth of the corresponding image frame is calculated and connected with the image features to obtain new features, and finally, a fully connected layer is used to obtain the classification confidence at the image level. The final classification confidence can be obtained by Equation (14).

$$\mathrm{score}(x, I) = \mathrm{sigmoid}(\mathrm{pred}(x) + \mathrm{cls}(x, I)), \tag{14}$$

$$\mathrm{cls}(x, I) = cls_{\mathrm{3D}}(\mathrm{reg}(x)) + cls_{\mathrm{img}}(\mathrm{proj}(\mathrm{reg}(x), I), \mathrm{depth}(\mathrm{reg}(x))). \tag{15}$$

Among them, score is the final classification confidence, x refers to the three-dimensional prediction frame, I is the corresponding image prediction frame, pred is the classifier of the second stage, cls is the classifier of the third stage, $\mathrm{cls}_{\mathrm{3D}}$ is the point of the third stage cloud classifier, $\mathrm{cls}_{\mathrm{img}}$ is the image classifier of the third stage, prog is the projection operation, reg is the prediction box regressor of the second stage, and depth is the operation of calculating the depth.

5. Experimental Results and Analysis

The 3D detection results of the model in this paper on the KITTI validation set are compared with the detection results of the traditional 3D target detection algorithm. In this paper, according to the requirements of the KITTI official website, if the 3D intersection ratio between the detection frame and the label is higher than 0.7, the detection frame is considered to be successfully detected; otherwise, the detection is considered to fail. The detection results obtained by the model on the validation set according to the above principles are shown in Table 1. It can be seen that the detection rate of the algorithm in this paper is higher than that of the original 3D target detection algorithm, but the algorithm speed is higher than that of the original 3D target detection algorithm and has declined. This is because the algorithm in this paper is based on points, and the three stages require repeated projection, mask operation, coordinate transformation, etc., which waste a lot of time. Experiments show that the detection accuracy of the algorithm is very high, but it also cannot meet the real-time requirements of automatic driving. The number of training and validation sets selected is 100.

In order to verify the effectiveness of each algorithm in this paper, the model is then subjected to stripping experiments, and the results are shown in Table 2, where L refers to the original 3D target detection algorithm, I is the image classifier in the third stage, P refers to adding a point cloud classifier using the training method in this paper, and P refers to directly assigning the parameters of the second stage to the point cloud the way of the classifier. The results show that both the new point cloud classifier and the image classifier can effectively improve the detection accuracy of the algorithm.

In order to analyze the impact of each classifier on the speed of the algorithm and the memory consumption, a comparative experiment is carried out in this paper, and the results are shown in Table 3. It can be found that with the increase of the classifier, the memory and algorithm consumption time of the model increase. Among them, the point cloud classifier only occupies 8 MB of video memory, but the algorithm consumption time increases by 87 ms, indicating that the connection between each stage is time-consuming, while the image classifier consumes 2600 MB of memory, and the time also increases by 154 ms, indicating that image classifiers are time-consuming. It can be seen that the algorithm cannot meet the real-time requirements. The final detection results of this model are shown in Figures 8.

6. Conclusion

To address the limitations of the overall framework of traditional 3D object detection algorithms, this paper investigates a point-based 3D object detection algorithm. In this paper, PointRCNN, a pure deep learning 3D object detection algorithm, is investigated. A third-level point cloud classifier and an image classifier are proposed to enhance the classification confidence. First, the network structure of PointRCNN is introduced and how it returns the target and refines it; then, the proposed point cloud classifier and image classifier are described in detail, including the network design, loss function; finally, model experiments are conducted to demonstrate that it can achieve a high detection accuracy of 79.51% (medium). The accuracy of the model is 0.66% higher than the baseline, which proves the effectiveness of the third level point cloud classifier and image classifier.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declared that there are no conflicts of interest regarding this work.

References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, no. 2, 2012.

[2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster rcnn: towards real-time object detection with region proposal networks,” *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[3] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[4] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, “Augmented reality technologies, systems and applications,” *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 341–377, 2011.

[5] Z. Zhang, “Microsoft Kinect sensor and its effect,” *IEEE Multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[6] H. Cheng, *Autonomous Intelligent Vehicles: Theory, Algorithms, and Implementation*, Springer Science & Business Media, 2011.

[7] D. Maturana and S. Scherer, *Voxnet: A 3D Convolutional Neural Network for Real-Time Object Recognition. Intelligent Robots and Systems*, IEEE Compute Society Press, Washington DC, 2015.

[8] D. Z. Wang and I. Posner, “Voting for Voting in Online Point Cloud Object Detection,” *In Proceedings of the Robotics: Science and Systems, Rome, Italy*, vol. 1, 2015.

[9] Z. Wu, S. Song, A. Khosla et al., “3D shape nets: a deep representation for volumetric shape modeling,” in *IEEE Conference on Computer Vision & Pattern Recognition*, IEEE Computer Society, 2015.

[10] R. Klokov and V. Lempitsky, “Escape from cells: deep kd-networks for the recognition of 3d point cloud models,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 863–872, 2017.

[11] P. S. Wang, Y. Liu, Y. X. Guo, C. Y. Sun, and X. Tong, “O-CNN,” *ACM series on computing methodologies*, vol. 36, no. 4, pp. 1–11, 2017.

[12] M. Donald, “Geometric modeling using octree encoding,” *Computer Graphics and Image Processing*, vol. 19, no. 1, p. 85, 1982.

[13] B. Shi, S. Bai, Z. Zhou, and X. Bai, “DeepPano: deep panoramic representation for 3-D shape recognition,” *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[14] A. Sinha, J. Bai, and K. Ramani, *Deep Learning 3D Shape Surfaces Using Geometry Images*, Springer International Publishing, Berlin, 2016.

[15] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, *Multi-View Convolutional Neural Networks for 3D Shape Recognition*, 2015.

[16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, *Point Net: Deep Learning on Point Sets for 3D Classification and Segmentation*, 2016.

[17] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[18] R. Ayman, M. F. Domingues, and J. Rodriguez, "Mobile caching-enabled small-cells for delay-tolerant e-Health apps," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 103–108, IEEE, 2017.

[19] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, *Frustum Pointnets for 3d Object Detection from rgb-d Data. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Salt Lake City, USA, 2018.

[20] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[21] S. Yasir, A. S. Malik, D. Sidibe, M. T. Simsim, N. Saad, and F. Meriaudeau, *Compressed VFH Descriptor for 3D Object Classification. 3DTV-Conference: The True Vision-Capture Transmission and Display of 3D Video (3DTV-CON) 2014*, pp. 1–4, 2014.

[22] W. Shang, K. Sohn, D. Almeida, and H. Lee, *Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units*, PMLR, 2016.

[23] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, IEEE Conference on Computer Vision & Pattern Recognition, IEEE Computer Society, 2016.

[24] Y. Zhou and O. Tuzel, "Voxelnet: end-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, IEEE Press, Salt Lake City, USA, 2018.

[25] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/R SJ International Conference on Intelligent Robots and Systems (IR OS)*, pp. 1–8, IEEE Press, Madrid, Spain, 2018.