Research on 3D Target Detection Algorithm Based on PointFusion Algorithm Improvement

Jun Wang, Shuai Jiang, Linglang Zeng, Ruiran Zhang
School of Mechanical and Electrical Engineering, Jiangxi University of Science and Technology, Ganzhou, China

***Abstract*—With the continuous development of automatic driving technology, the requirements for the accuracy of 3D target detection in complex traffic scenes are getting higher and higher. To solve the problems of low recognition rate, long detection time, and poor robustness of traditional detection methods, this paper proposes a new method based on PointFusion model improvement. The method utilizes the PointFusion network architecture to input 3D point cloud data and RGB image data into the PointNet++ and ResNeXt neural network structures, respectively, and adopts a dense fusion method to predict the spatial offsets of each input point to each vertex in the 3D selection box point by point, to output the 3D prediction box of the target. Experimental results on the KITTI dataset show that compared with the PointFusion network model, the improved PointFusion-based model proposed in this paper improves the 3D target detection accuracy in three different difficulty modes (easy, medium, and hard) and performs best in the medium difficulty mode. These findings highlight the potential of the method proposed in this paper to be applied in the field of autonomous driving, providing a reliable basis for navigating self-driving cars in complex environments.**

***Keywords*—Neural network; target detection; autonomous driving; PointFusion; deep learning**

I. INTRODUCTION

With the rapid development of computer vision and deep learning technology, driverless vehicles are moving towards the practical stage. However, in intricate road conditions and uncertain traffic scenarios, the safety of automated driving technology is becoming more and more important. How to recognize obstacles efficiently and accurately has become an important challenge in the field of autonomous driving.

Target detection plays an important role in autonomous driving [1, 2]. A large number of methods have been proposed to solve the obstacle recognition problem in autonomous driving. In the field of 3D target detection, depending on the modality of the sensor data used in 3D detection networks, they can be broadly categorized into detection methods based on image, point cloud, and bimodal information fusion of image and point cloud. The SMOKE network model proposed by Liu et al. [3] is an image-based 3D target detection that utilizes feature point estimation and 3D spatial variable regression to determine the spatial location of the target, which has the advantage of a simple data preprocessing stage that improves the detection speed, while the network model solves the effect of noise introduced due to redundancy of 2D detection networks. The 3D-SSD network model proposed by Luo et al. [4] is a one-stage network for target detection based on depth information, and the prediction is realized with multi-scale mapping, the method has a good improvement in small target detection, as well as excellent performance in depth estimation against images. The Mono3D (Monocular 3D) model proposed by CHEN et al. is based on the improvement of the 3DOP model [5], which generates 3D candidate frames, then scores them with 2D image features and classifies and regresses the candidates with high scores, where the 2D image features are generated based on the information of semantic segmentation, instance segmentation, and location a priori. The practice has shown that the accuracy of the 3DOP model is insufficient relative to the estimation of depth, and some scholars have found that more accurate depth information can be obtained by utilizing parallax estimation. In 2019, LI et al. proposed a Stereo R-CNN model [6], which, relative to the 3D-SSD and the 3DOP model, utilizes the parallax estimation method to obtain more accurate depth information. In point cloud-based 3D target detection methods, many researchers rasterize the point cloud and convert it into voxel form representation to easily handle irregular point cloud data. SIMON et al. [7] proposed Complex-YOLO, a point cloud-based 3D real-time target detection network based on YOLO, where the pose of an object is estimated by adding an imaginary and a real number to the regression network in a specific Euler-Region-Proposal Network (E-RPN). The voxel size setting is a difficult problem to be solved in point cloud voxelization. In 2020, M.Y et al. [8] proposed a hybrid voxel network (HVNet) for mixing point clouds with voxels, which solves this problem by fusing voxel features of different scales at the point-level of the point cloud and projecting them into multiple pseudo-image feature maps. Deng et al. proposed a two-stage voxel-based framework Voxel R-CNN network [9], which first generates region proposals based on a bird's-eye view, and then extracts region-of-interest features directly from voxel features using a designed voxel RoI pool. However, the process of transforming the point cloud into either a voxel or bird's eye view projection map form causes some loss of point cloud data. To minimize or avoid this loss, some scholars have investigated the direct processing of the original point cloud. Qi et al. proposed the PointNet network [10], which ensures that the order of the points in the point cloud remains unchanged during the processing, and the structural connection between the points is preserved completely. Lehner et al. introduced a two-stage model, which consists of two VoxelNet [11] based networks, the Region Proposal Network (RPN) and the Local Refinement Network (LRN), for accurate detection and localization of 3D targets from point cloud data. Although different modal data have obvious effects when used individually in some specific scenarios [12], however, a sensor can only acquire a single modal data in the environment, and a large number of

experiments have proved that there are obvious inherent deficiencies in the environment sensing tasks accomplished by relying on only single modal data. The fusion of LiDAR point cloud data and camera image data with complementary relationships to improve the detection of targets in autonomous driving environment sensing tasks has been the focus of researchers [13]. Xu et al. proposed a PointFusion network structure [14], which is one of the typical pre-fusion structures. PointFusion first generates 2D selection frames on the image with a 2D detector projects the point cloud to the image plane, and selects the appropriate target region in the point cloud using the 2D selection frames region, and the selected 2D image data and the 3D point cloud data are used for feature extraction with ResNet [15] and PointNet networks for feature extraction to predict the location of the target in space. The advantage of this method is that the point cloud is directly input as raw data so that the information is preserved. However, the network is limited to dense point cloud data and is poor for sparse point clouds. Inspired by fusion methods such as PointFusion, Sindagi et al. proposed the MVX-Net network architecture for hybrid fusion based on earlier fusion [16], which utilized the Voxel Net network structure introduced at the time to combine two modal data, RGB images, and point clouds. Wang et al. proposed the F-ConvNet network [17], which, unlike the method of fusing voxelized point clouds with images, consists of a set of view cones proposed to be generated from 2D checkboxes, which are used to group the local point clouds, and the view cone features are formed through feature extraction. The VeloFCN network proposed by Li et al. [18] draws on the experience of 2D image detection by projecting a 3D point cloud to a front view similar to a camera image, and the data obtained from this processing is converted to a 2D image form, and the image is detected with a 2D target detector, but the point cloud in this method has multiple points that overlap in the process of projecting to the front view, resulting in loss of information.

In recent years, with the continuous development of computer vision and deep learning, the detection of targets using deep learning techniques has become a popular research direction. Saranya. K.C et al. [19] proposed YOLO v3 to detect pedestrians, using this method reduces the computational resources and speeds up the computation speed based on guaranteeing the detection accuracy, but there will still be misdetection and omission problems for the targets occluding each other, overlapping and so on. Ren S [20] proposed to utilize neural networks instead of selective search and proposed the concept of anchor frames, the highlight of this method is the integration of subsequent steps such as feature extraction in the same network, which leads to an improvement in the overall performance.

However, the current target detection algorithms are still unable to meet the practical needs, and many problems still need to be solved and improved. On the one hand, most of the autonomous driving scenarios are outdoor open scenarios, containing a large number of static and dynamic targets, and the traffic situation is complex; on the other hand, most of the sensors used for target detection in autonomous driving vehicles are more than three types, and the currently designed target detection algorithms have a single task on the network, and the fused sensor data types are fewer, which will deplete the limited arithmetic power of the vehicle control unit when carrying out multiple tasks at the same time. When performing multiple tasks at the same time, it consumes the limited arithmetic power of the vehicle control unit. Therefore, how to recognize obstacles efficiently and accurately is still of great significance in the field of autonomous driving.

The rest of the paper is organized as follows: Section II describes the theoretical approach and optimization process of the proposed improved model, including the relevant parameter settings of the PointNet++ network and ResNeXt network. Section III examines the accuracy of this paper's model on the KITTI dataset, followed by a comprehensive analysis and discussion of the experimental results. Section IV summarizes the main contributions of this paper's model and proposes future research directions.

II. Proposed Method

A. Pointfusion Network Model Optimization

In this paper, we improve the PointFusion architecture based on the PointFusion architecture, optimize the point cloud feature extraction module and the image feature extraction module, introduce the better performing PointNet++ and ResNeXt neural network structures instead of the PointNet module and the ResNet module, and use only the dense fusion structure to predict the 3D selection box point by point from each input point to the 8 corners (i.e., each vertex) of the spatial offsets, and in this way outputs the 3D prediction frame of the target. The structure is shown in Fig. 1.

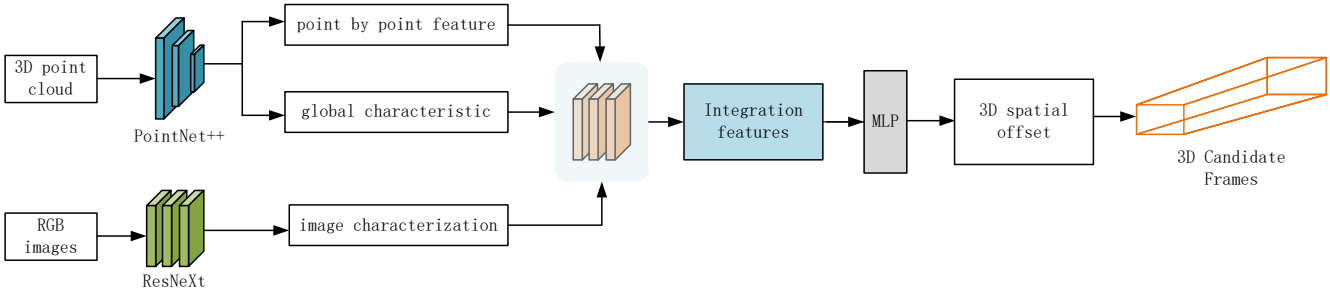


Fig. 1. Improved converged architecture based on PointFusion.

The PointFusion network architecture is a pre-fusion two-stage network structure. The network architecture performs target detection on the image data with a 2D target detector and enhances the point cloud information with the detected 2D target image information. Then the processed RGB image and point cloud image are used as input information, and the corresponding image and point cloud feature extraction network are used to extract features from the input data. Finally, the points in the point cloud of the target area are used as localization anchor points in space and predicted to obtain the 3D candidate frame of the target.

PointFusion uses heterogeneous network architecture to process the input. 3D point cloud data and RGB image data are fed into different branches for feature extraction. A variant model of the PointNet network architecture is used to process the raw point cloud data directly, avoiding the lossy input preprocessing caused by converting the point cloud data into Range maps or voxel forms. However, the PointNet network itself has a poor ability to process sparse point cloud data, which leads to weak detection of sparse point cloud targets in PointFusion itself. In the architecture of PointFusion, on the one hand, the data enhancement of point cloud data is performed with a 2D target detection method before data input, which makes the input point cloud information a cropped dense point cloud, and its network architecture is also trained in the dense region of point cloud, which has a weak adaptive ability to sparse point cloud; on the other hand, when utilizing the point cloud as a spatial localization point, the dense point cloud can be very On the other hand, when using point clouds as spatial localization points, dense point clouds can predict the spatial location of the target, while sparse point clouds cannot predict the spatial location of the target very well using this method.

B. PointNet++ Network

PointNet++ is a deep optimization improvement based on PointNet.PointNet++ is a neural network structure consisting of a combination of multiple individual layers, which applies the PointNet network in feature extraction of the input point set. The PointNet++ network, by calculating the spatial distances between points in space, can follow the change in size between two neighboring layers to extract local features. Moreover, each ensemble sampling layer can be adapted to sample point cloud regions with different densities and can automatically combine feature information at different scales.

PointNet++ employs hierarchical point set feature learning with a hierarchy consisting of multiple ensemble sampling layers. At each ensemble sampling layer, the point cloud sets within a region are first grouped and sampled, then undergo feature extraction and move to the next ensemble sampling layer. In the new ensemble sampling layer, each set of features from the previous layer is combined into a new point cloud element, and the previous operation is repeated until all the features are extracted. The ensemble sampling layer is the core layer and consists of the sampling layer, the grouping layer, and the point network layer. The sampling layer uses the farthest point sampling (FPS) method for sampling, which randomly takes a set of input points as the center of the region, then calculates the distances between other points and that point, and determines the adjacent points based on the distances between the points. The grouping layer constructs the neighborhood of each point cloud based on the distances between spatial points. The PointNet layer focuses on feature extraction of point clouds in the domain, using a simplified version of the PointNet network to extract features from different groups of point clouds in the grouping layer. The PointNet++ structure is shown in Fig. 2.

The input to the ensemble sampling layer is N × (d + C), where d represents the coordinate dimension of the point cloud, C represents the feature vector dimension of each point in the point cloud, and N represents the number of points in each point cloud ensemble. The output of this layer is $N'\times$（d+C'）, where the coordinate dimensions of the points are unchanged, C'represents the eigenvector dimension of each point in the point cloud collection in the input to the next ensemble sampling layer, and N' represents the number of points in each point cloud collection in the input to the next ensemble sampling layer.

A set of points$\{x_1, x_2, \ldots, x_n\}$ is input to the sampling layer, and a set of points $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ is found using constant iterative sampling to determine that x_{i_j} is the maximum distance from the set $\{x_{i_1}, x_{i_2}, \ldots, x_{i_{j-1}}\}$. This sampling ensures that all points in the point cloud collection are utilized.

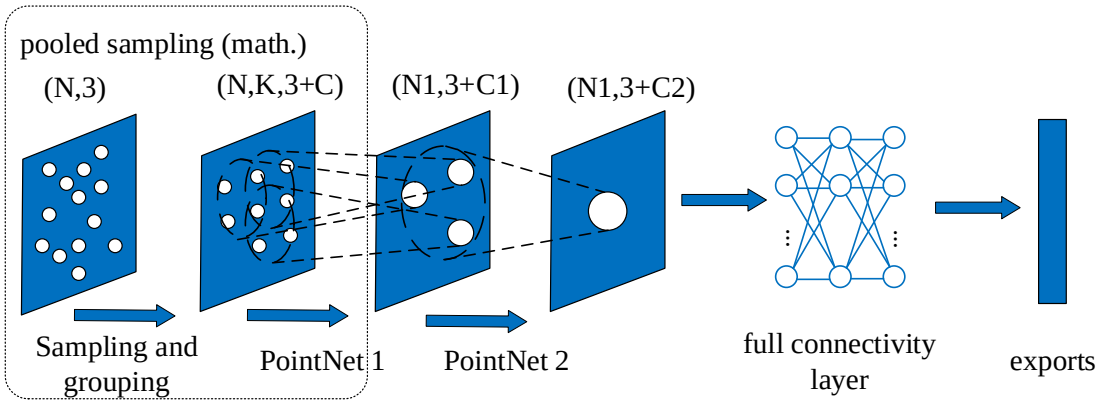


Fig. 2. PointNet++ network structure.

The grouping layer inputs are a point cloud collection and a center of mass collection. The point cloud collection contains the coordinate dimension of the point cloud, the eigenvector dimension of each point, and the number of points in the point cloud; the center of mass collection contains a set of center of mass coordinates. The output of this layer is N'×K×(d+C), with N' denoting the number of neighbors in the set, and K being the center-of-mass points selected in the neighborhood with variable size.

In the PointNet layer, the input is a localized region of N' points of size N'×K×（d+C）, and each localized region in the output is abstracted by its center of mass and local features encoding the neighborhood of the center of mass, with a data size of N'×K×（d+C'）. The PointNet layer is to extract the features of the point cloud data within each neighborhood partitioned by the previous layer, represented by a feature vector of uniform size. The input to this layer is N'×K×（d+C ） and the output is N'×K×（d+C'）.

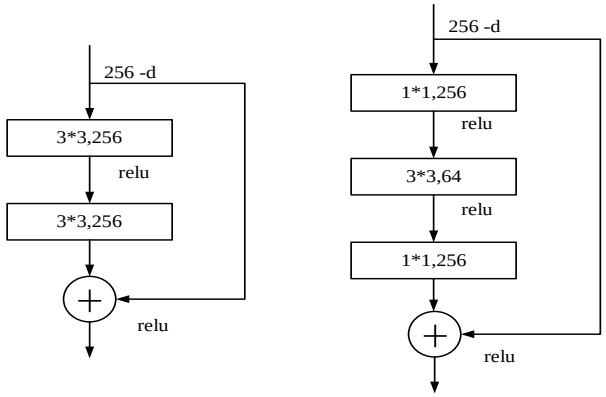
The PointNet++ network structure specifically addresses the problem that when the point cloud is inhomogeneous, the features learned in the dense region may not be suitable for the sparse region, and proposes a multiscale grouping (MSG), where the point cloud data with different densities are grouped into multiple local neighborhoods with different sizes, and these neighborhoods are first feature extracted, and then later the local neighborhood features are extracted by a point cloud feature extraction network to obtain the global features.

C. ResNeXt Network

In the image feature extraction branch, the ResNeXt neural network structure is used, which is developed based on ResNet and combines the experience of classic network structures such as VGG, ResNet, and Inception. It is composed of stacking multiple residual blocks of similar structure and in each block, three convolutional layers are used to realize various transformations from "dimensionality reduction-transformation-upgrading".

The ResNeXt network has been used as the core of many advanced networks for its performance in speed and accuracy in many image vision tasks. Although Resnet can be subdivided into 18, 34, 50, and 101 layers, the main body of the network structure is the same. The starting input of the Resnet network structure is a convolutional layer with 7*7 convolutional kernels, followed by a maximum pooling downsampling layer with 3*3 convolutional kernels, followed by a residual structure with multiple blocks stacked in layers conv2~conv5, and the last layer is an average pooling downsampling layer with 3*3 pooling kernels. The first input of the Resnet network structure is a convolutional layer with 7*7 convolutional kernels, followed by a maximally pooled downsampling layer with 3*3 convolutional kernels, followed by conv2~conv5 layers which are residual structures stacked with multiple blocks, and the last layer is an average pooled downsampling and fully-connected layer, and the result is outputted by softmax. The structure of the ResNeXt network is shown in Fig. 3.

Fig. 3(a) shows the residual module of ResNet-18/34, and Fig. 3(b) shows the residual module of ResNet-50/101/152. Each module consists of a main branch and a shortcut branch, which are output after the convolution operation of the main branch on one hand, and the shortcut branch is directly the input, and then the results of the two branches are summed up and processed by the activation function, so the size and depth of the feature matrices of the shortcut branch and the output of the main branch should be the same. The size of the kernel of convolution and the input/output of the residual module of ResNet-18/34 are the same at each layer, and the size of the kernel of convolution and the input/output of ResNet-50/101/152 is the same, while the size of the convolution kernel of the convolutional layers in the residual module of ResNet-50/101/152 is not consistent. In Fig. 3(b) 1*1 convolution kernel is used for downscaling and upscaling, and it can be seen from the figure that the input of 256 channels in the first layer is output down to 64 channels, while the 64 channels input in the second layer is increased to 256 channels by the output of the third layer. In Fig. 3(b), this structure serves to save more number of convolution kernels. Taking the input of 256 channels as an example, after calculation, there are 1179648 convolutional kernels in Fig. 3(a) and 69632 convolutional kernels in Fig. 3(b), compared to the reduction of 111,016 convolutional kernels, which saves resources and improves the efficiency at the same time.



(a) The residual module of ResNet-18/34 (b) The residual module of ResNet-50/101/152

Fig. 3. PointNet++ network structure.

The improved network uses a dense fusion network structure. The network uses the input spatial points as spatial anchors and predicts the spatial positional offsets of each spatial point to each vertex of the neighboring preselected boxes. An unsupervised approach is utilized to predict a score for each spatial point, and the point with a high prediction score receives a high confidence level. The high confidence point is used as the final prediction point. The unsupervised scoring loss function is:

$$L_{score} = \frac{1}{N}\sum\left(L^{i}_{offset} \cdot c_i - w \cdot log(c_i)\right) + L_{stn} \quad (1)$$

where, w is the weight coefficient, c_i is the confidence level, and L^{i}_{offset} is the spatial angular offset loss at the first spatial point. L_{stn} is the spatial transformation regularization loss. The loss function of the dense fusion network is:

$$L_{dense} = \frac{1}{N}\sum_i smothL\left(x^{i^*}_{offset,} x^{i}_{offset}\right) + L_{score} + L_{stn} \quad (2)$$

where, N is the number of input points, $x_{offset}^{i^*}$ is the offset between the true checkbox vertex position and the ith spatial point, and x_{offset}^{i} is the spatial position offset between the predicted checkbox vertex position and the ith spatial point.

III. Experimental Results

A. Experimental Parameter setting

1) PointNet++ parameter settings: This experiment uses the MSG network (Multi-Scale Network) of the PointNet++ network to set up three ensemble sampling layers, and some of the parameters of each ensemble sampling layer are shown in Table I.

TABLE I. Parameter Settings for each Ensemble Sampling Layer

Ensemble sampling layer	Number of input points	Number of sampling points	Sampling radius	characteristic channel
SA1	1024	512	[0.1,0.2,0.4]	-
SA2	512	128	[0.2,0.4,0.8]	320
SA3	128	-	-	640+3

During model training, batch is set to 24 and epoch is set to 200. The MSG model has three layers SA1, SA2, and SA3 used for point cloud data feature extraction.

1) Ensemble Sampling Layer: The MSG network model has three layers, sa1, sa2, and sa3, which are used for point cloud data feature extraction. The output of this layer mainly has seven-dimensional features. The first-dimensional feature is the number of sampled points; the second-dimensional feature is the radius size, a total of three radius parameters are set, [0.1,0.2,0.4] for the first layer, [0.2,0.4,0.8] for the second layer; the third-dimensional feature is the number of points in the group corresponding to the radius; the fourth-dimensional feature is the number of features or the channel size of the input points; and the fifth to the seventh-dimensional features are corresponding to the three radiuses respectively. The fifth to seventh-dimensional features are the number of feature dimensions corresponding to each of the three radii.

2) ResNeXt parameter settings: The ResNeXt101 network is used in this experiment, where the batch is set to 32, num_workers is set to 4, and epoch is set to 100. The optimizer is ADAM, and the learning rate is 0.001. The parameters of conv2~conv5 of the ResNeXt residual structure are set to {3, 4, 23, 3}.

B. Analysis and Discussion of Results

To test the model precision on the KITTI dataset, to better compare with other target detection methods, the experiment uses the evaluation method provided by the official KITTI dataset, sets the thresholds for cars, cyclists, and pedestrians at 0.7 respectively, and divides the test set among the KITTI dataset into three modes: simple, medium and difficult. The accuracy and recall in the three modes are calculated to obtain the average precision as the evaluation performance metric. According to the method of PointFusion, Fast-Rcnn is used as a 2D target detector with ResNeXt-101 to extract the input image feature information and average over the feature map positions. For each input 2D frame, the image is cropped and resized to 224 × 224, and up to 400 3D point clouds are randomly sampled as input for training and evaluation.

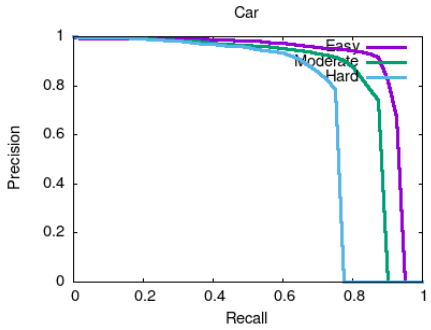
The P-R curves obtained from the experiment are shown in Fig. 4, and the final experimental AP result statistics are shown in Table II.

Table II results show the improved fusion network model detection results. The evaluation results for 3D detection in both evaluation criteria show that the detection accuracy for the automobile class is improved to 79.97% in the simple model; the detection accuracy for pedestrians is still below fifty percent. Some of the visualized test results are shown in Fig. 5.

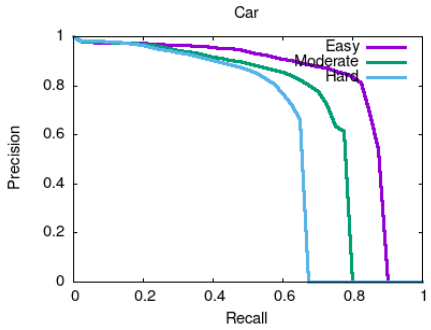
TABLE II. Model Test Results

objectives	AP_{BEV}(%)			AP_{3D}(%)		
	Easy	**Medium**	**Difficult**	**Easy**	**Medium**	**Difficult**
vehicle	89.10	83.14	71.41	79.97	69.13	58.57
pedestrians	56.52	47.80	43.63	48.65	40.11	35.99
cyclist	71.92	57.99	51.55	65.51	51.33	45.05

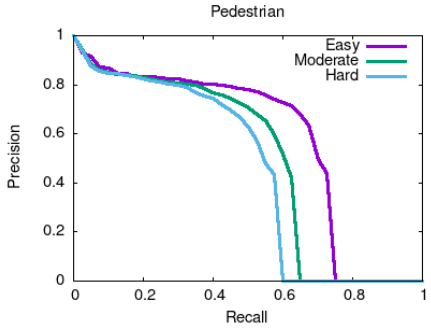
The PointFusion model before and after improvement is compared with other state-of-the-art models in Table III. The experimental results show that the improved PointFusion-based model proposed in this paper performs well for the target detection task in general and improves the performance compared to the original model. Among them, the highest improvement is 6.13% in the medium mode, 2.05%, and 5.3% in the easy and difficult modes, respectively. In the medium and difficult modes, the proportion of accuracy improvement before the improvement relative to that after the improvement is larger, indicating that the generalization ability of the improved model to the original model has been improved. The method proposed in this paper reaches or approaches the results of the current state-of-the-art methods in all three difficulty modes, but there is an obvious gap in accuracy compared to AOVD and F-PointNets in the difficulty level. The comparison reveals that both the original model and the improved model based on this paper have poor performance in the difficulty modes relative to the above two models. This indicates that the detection of large occluded targets and small targets needs to be improved.



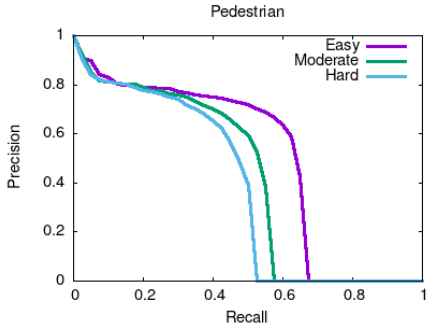
(a) PR curve for Bev_Car



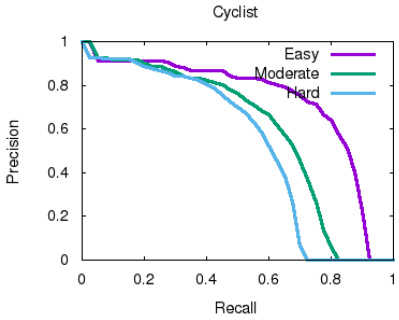
(b) PR curve for 3D_Car



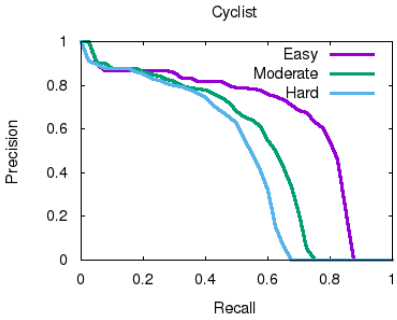
(d) PR curve for 3D_Pedestrian



(c) PR curve for Bev_Pedestrian

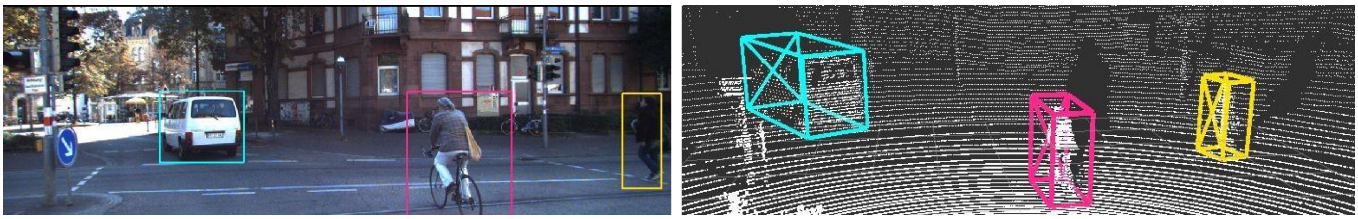


(e) PR curve for Bev_Cyclist

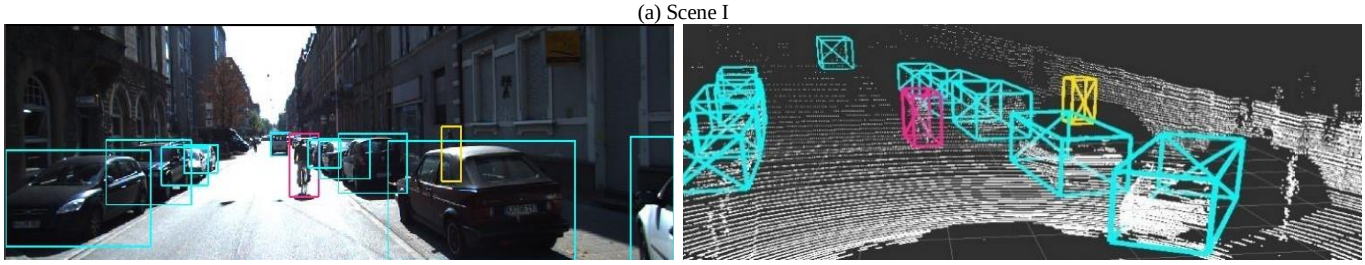


(f) PR curve for 3D_Cyclist

Fig. 4. Improved P-R curves of Disp R-CNN on bird's eye view (Bev) and 3D detection tasks.



(a) Scene I



(b) Scene II

Fig. 5. Visualization of some of the test results.

TABLE III. MODEL TEST RESULTS

Network model	APBEV(%)			AP3D(%)		
	Easy	Medium	Difficult	Easy	Medium	Difficult
F-PointNets	91.17	84.67	74.77	82.19	69.79	60.59
AOVD	89.75	84.95	78.32	76.39	66.47	60.23
MV3D	86.62	78.93	69.80	74.97	63.63	54.00
PointFusion	-	-	-	77.92	63.00	53.27
paper model	89.10	83.14	71.41	79.97	69.13	58.57

IV. CONCLUSION

In this paper, we propose to optimize the PointFusion architecture based on the improved PointFusion fusion algorithm by replacing the PointNet module and ResNet module in the point cloud feature extraction module and image feature extraction module with PointNet++ and ResNeXt network structures, respectively, to achieve more accurate 3D target detection. The key difference with the existing methods is that we consider the problems of poor generalization ability of the PointNet network and poor feature extraction in sparse regions of the point cloud. We feature extract image and point cloud information from the input data and use a dense fusion approach to obtain the final prediction, which effectively reduces the number of hyperparameters in the image feature extraction module and improves the computing speed without affecting the accuracy of the final prediction. Finally, we used experiments on the KITTI dataset to demonstrate the effectiveness of the method, and compared with the original prediction model, the model in this paper has the highest improvement among the medium modes, reaching 6.13%. Most of the current neural networks for autonomous driving perception tasks are designed for a single task, which generates a large amount of arithmetic power consumption for the in-vehicle main controller that works on multiple tasks at the same time. How to integrate the networks of different tasks into one main network, optimizing the arithmetic power, and improve the cooperative work of each task will be a future research direction.

REFERENCES

[1] Balasubramaniam and S. Pasricha, "Object detection in autonomous vehicles: Status and open challenges," arXiv preprint arXiv:2201.07706, 2022.

[2] N. M. A. A. Dazlee, S. A. Khalil, S. Abdul-Rahman, and S. Mutalib, "Object detection for autonomous vehicles with sensor-based technology using yolo," International Journal of Intelligent Systems and Applications in Engineering, vol. 10, no. 1, pp. 129-134, 2022.

[3] Liu Z, Wu Z, Tóth R. Smoke: Single-stage monocular 3d object detection via keypoint estimation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 996-997.

[4] Luo Q ,Ma H ,Tang L , et al. 3D-SSD: Learning hierarchical features from RGB-D images for amodal 3D object detection[J]. Neurocomputing,2020,378.

[5] XIAOZHI CHEN, KAUSTAV KUNDU, ZIYU ZHANG, et al. Monocular 3D Object Detection for Autonomous Driving[C]. //29th IEEE Conference on Computer Vision and Pattern Recognition: 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 26 June – 1 July 2016, Las Vegas, Nevada.:Institute of Electrical and Electronics Engineers, 2016:2147-2156.

[6] PEILIANG LI, XIAOZHI CHEN, SHAOJIE SHEN. Stereo R-CNN based 3D Object Detection for Autonomous Driving[C]. //2019 IEEE/CVP Conference on Computer Vision and Pattern Recognition: CVPR 2019, Long Beach, California, USA, 15-20 June 2019, [v.11].:Institute of Electrical and Electronics Engineers, 2019:7636-7644.

[7] MARTIN SIMON, STEFAN MILZ, KARL AMENDE, et al. Complex-YOLO: An Euler-Region-Proposal for Real-Time 3D Object Detection on Point Clouds[C]. //Computer Vision - ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part I.:Springer, 2019:197-209.

[8] M. Y ,S. X ,T. C . HVNet: Hybrid Voxel Network for LiDAR Based 3D Object Detection[J]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,2020.

[9] Deng J, Shi S, Li P, et al. Voxel r-cnn: Towards high performance voxel-based 3d object detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(2): 1201-1209.

[10] Qi C R, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segment-ation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 652-660.

[11] Lehner J, Mitterecker A, Adler T, et al. Patch Refinement--Localized 3D Object Detection[J]. arXiv preprint arXiv:1910.04093, 2019.

[12] LI, YING, MA, LINGFEI, ZHONG, ZILONG, et al. Deep Learning for LiDAR Point Clouds in Autonomous Driving: A Review[J]. IEEE transactions on neural networks and learning systems,2021,32(8):3412-3432. DOI:10.1109/TNNLS.2020.3015992.

[13] CUI, YAODONG, CHEN, REN, CHU, WENBO, et al. Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review[J]. 2022,23(2):722-739. DOI:10.1109/TITS.2020.3023541.

[14] Xu D, Anguelov D, Jain A. Pointfusion: Deep sensor fusion for 3d bounding box estimation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 244-253.

[15] KAIMING HE, XIANGYU ZHANG, SHAOQING REN, et al. Deep Residual Learning for Image Recognition[C]. //29th IEEE Conference on Computer Vision and Pattern Recognition: 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 26 June – 1 July 2016, Las Vegas, Nevada.:Institute of Electrical and Electronics Engineers, 2016:770-778.

[16] Sindagi V A, Zhou Y, Tuzel O. Mvx-net: Multimodal voxelnet for 3d object detection[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 7276-7282.

[17] Wang Z, Jia K. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 1742-1749.

[18] Li B, Zhang T, Xia T. Vehicle detection from 3d lidar using fully convolutional network[J]. arXiv preprint arXiv:1608.07916, 2016.

[19] C K S ,Thangavelu A . Vulnerable Road User Detection using YOLO v3[J]. International Journal of Advanced Computer Science and Applications (IJACSA),2019,10(12).

[20] REN, SHAOQING, HE, KAIMING, GIRSHICK, ROSS, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2017,39(6):1137-1149.DOI:10.1109/TPAMI.2016.2577031.