

DOI:10.19651/j.cnki.emt.2210335

基于激光雷达点云的车辆目标检测算法改进研究

王庆林¹ 李 辉¹ 谢礼志¹ 谢剑斌² 彭石林¹

(1.长沙理工大学物理与电子科学学院 长沙 410114; 2.湖南中科助英智能科技研究院有限公司 长沙 410205)

摘要: 本文提出了一种基于 PointRCNN 的改进目标检测算法。该方法针对原始 PointRCNN 对远距离处的车辆检测效果较差的问题进行了优化,并提高了算法目标检测的平均精度值。改进算法第 1 阶段先将激光雷达点云进行伪图像处理,降维至二维,然后利用 Point-Focus 结构对其进行处理并还原至三维点云。再将其送入 PointNet++ 主干网络中进行特征提取,得到点的分类与回归结果并进行第 1 阶段的 3D 框生成。第 2 阶段对 3D 框进行优化选择,引入 Point-CSPNet 结构进一步提升网络学习能力和鲁棒性。本文合理借鉴了 YOLO 系列算法中的 Focus、CSPNet 结构,充分提取了原始点云中的有效信息,有效整合了网络运算过程中的特征及梯度变化,提高网络的检测准确率。本文的改进算法在 KITTI 数据集的 3D 场景下平均精度值从 81.10% 提升至 81.74%; BEV 场景下平均精度值从 86.87% 提升至 88.20%,可视化效果中远距离处的车辆目标检测效果也得到了一定程度的优化,对无人驾驶技术进一步优化和完善具有一定的积极意义。

关键词: 点云数据;目标检测;PointRCNN;KITTI 数据集

中图分类号: TP391.4 **文献标识码:** A **国家标准学科分类代码:** 510.4

Research on improving vehicle target detection algorithm based on lidar point cloud

Wang Qinglin¹ Li Hui¹ Xie Lizhi¹ Xie Jianbin² Peng Shilin¹

(1. School of Physical and Electronic Sciences, Changsha University of Science and Technology, Changsha 410114, China;

2. Hunan ZK HI Intelligent Technology Research Institute Co., Ltd., Changsha 410205, China)

Abstract: This paper presents a target detection algorithm based on PointRCNN. This method is aimed at vehicle targets. Aiming at the problem that the original PointRCNN is poor in vehicle detection at a distance, the method is optimized and the average accuracy of target detection is improved. In the first stage, the lidar point cloud is processed by pseudo-image structure and dimensionality reduction to 2D, and then processed by Point-Focus structure and restored to 3D point cloud. Then it will be sent into the backbone of PointNet++ for feature extraction, classification and regression. In the second stage, 3D frame is optimized and selected, and Point-CSPNet structure is introduced to further improve network learning ability and robustness. In this paper, the Focus and CSPNet structures of YOLO series algorithms are used for reference. The effective information in the original point cloud is fully extracted and the feature, gradient changes in the network operation are effectively integrated to improve the detection accuracy of the network. The average accuracy of the improved algorithm is improved from 81.10% to 81.74% in 3D scenes of KITTI dataset; and it is improved from 86.87% to 88.20% in BEV scenes of KITTI dataset, and the detection effect of vehicle targets in the far distance of visual effect has also been optimized to a certain extent, which has certain positive significance for further optimization and improvement of unmanned driving technology.

Keywords: point cloud data; target detection; PointRCNN; KITTI dataset

0 引 言

随着无人驾驶技术的兴起,研究者们越来越关注与之相关的具体应用技术研究。实现无人智能驾驶,就离不开

车载条件下的环境感知和目标检测。点云数据即分布在三维空间中的离散点集^[1],一般由激光雷达、毫米波雷达、ToF(time of flight)相机等三维点云采集设备配备在车辆的合适位置来获取,用以进一步实现车辆的智能感知。在

收稿日期:2022-06-16

硬件完备的基础上,进一步优化点云目标检测算法的精确度和鲁棒性就具有十分重大的意义。目标检测是对视野中待检测物体的位置和大小进行判定框选,并以突出的形式让人眼能够快速识别,二维目标检测已经得到了巨大的发展,应用到了很多实际工程中^[2]。在二维目标检测发展的启发下,三维点云中的目标检测也发展的越来越迅速。

一般来说,点云数据中的目标检测有两种大的研究思路,其一是直接对原始点云进行处理,其二是将点云进行体素化处理^[3]。基于原始点云数据表达的目标检测方法就是将原始点云直接作为输入,从而获取相应的点云特征完成目标检测,当前典型的算法有 PointRCNN^[4]、STD^[5]等。基于体素的点云目标检测则是先对点云划分网格,将最终处于同一网格当中的点进行体素化处理,当前的主要典型算法有 VoxelNet^[6]、SECOND^[7]、PointPillars^[8]等。

在研究者们大力研究点云结合图片和点云体素化两种目标检测的解决思路时,PointRCNN 这一算法的提出进一步丰富了点云目标检测的研究广度,它直接对原始点云进行处理极大的保留了点云的原始特征,提高了目标检测的精度。进一步对 PointRCNN 进行优化研究也就具有十分重要的意义。

由于 PointRCNN 在 KITTI 数据集上的优质表现,且是第一个直接对点云数据进行处理,得到了广大研究者的深入研究。徐兵^[9]在 PointRCNN 的基础上,引入了特征权重提取模块,设计了多通道的自适应特征权重学习策略。Yan 等^[10]在 RPN 检测网络中添加了 Hough voting 机制,优化了 RPN 网络的目标检测效果。梁小芳^[11]在其论文中实现了 PointRCNN 融合去地面算法,提高了目标检测精度;Li 等^[12]采用多分支主干网络来处理点云密度不均匀的问题,优化了原始检测算法。

1 相关工作

1.1 PointRCNN

PointRCNN 是第 1 个直接对原始点云进行处理的两阶段点云目标检测算法,并采用 Python 语言结合 Pytorch 框架进行实现。PointRCNN 的设计思路参考了二维 RCNN(Region-CNN)网络,具有很强的 RCNN 特点。其中 RCNN 的主要流程分为候选区域选择、特征提取和分类与边界回归 3 个大的部分^[13]。将这一思路应用到三维点云目标检测中极大的促进了点云目标检测的发展。

PointRCNN 的第 1 阶段是通过语义分割操作得到前景点,并将前景点进行回归操作以减少检测框搜索范围;第 2 阶段则是采用正则化(canonical coordinate)和非极大值抑制(non-maximum suppression, NMS)等检测框优选方法进行 3D 框确定。这一阶段中,原始 PointRCNN 中提出了一种基于区间 bin 的 3D 框优选方法,有效的减少了 3D 框数量,如图 1 所示为 PointRCNN 的算法网络结构。

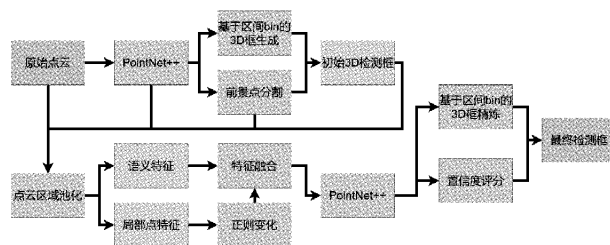


图 1 PointRCNN 算法网络结构

1.2 Focus 与 CSPNet 结构

Focus 与 CSPNet 结构已经应用于二维目标检测领域的 YOLO 算法中^[14-15]。Focus 结构处于其输入端和主干网络之间,用于对输入图片进行切片操作。经过这样的操作也就可以得到 4 张互补且没有信息丢失的图片,与此同时,输入图片的有效信息就可以集中到通道空间并将通道空间扩展到原来的 4 倍。在原始的 YOLO 算法中,一张 $640 \times 640 \times 3$ 的图像输入 Focus 结构后,会先切片成 $320 \times 320 \times 12$ 的特征图,这一特征图再经过卷积操作得到 $320 \times 320 \times 32$ 的特征图。

CSPNet 结构则是针对密集卷积网络进行的优化工作,在提高了网络的准确度的同时降低了网络运算复杂度^[16]。这一结构可以理解成一种可广泛移植的思想方法,应用于其他的网络结构中。具体来说,在单阶段的 DenseNet 结构中,每个阶段都会包括一个密集模块和过度模块。而融合 CSPNet 结构则有两种具体思路,即在输入过度模块之前进行特征融合还是之后。经过这一结构,可以在一定程度上减少梯度信息从而减小计算量并提高检测精度。

2 算法设计

针对点云数据目标检测,参考 Focus 和 CSPNet 结构的思想方法,本文在 PointRCNN 的基础上融合了 Point-Focus 和 Point-CSPNet 进一步提高点云目标检测的鲁棒性和精确度。

本文的改进算法网络为两阶段网络,其网络结构如图 2 所示,第 1 阶段的主要目的即生成 3D 边界框提案,这样可以有效避免在整个 3D 空间中使用大量的锚选框。第 2 阶段则是对 3D 候选框进行优选以得到最优的目标检测框。

2.1 改进 PointRCNN 算法的流程

原始点云数据首先进行伪图像处理,得到伪图像后再经过 Point-Focus 结构进行点云精简。经过这一操作后再将伪图像升维至三维,并输入到 PointNet++ 主干网络^[17]中进行特征提取,即得到点云数据的特征向量,对其进行进一步的分类与回归,进而得到点云数据的前景点、背景点分类结果和每一个前景点上的 bounding box 回归框。

为进一步提高检测框的准确度,再将第一阶段所得特征向量与前景点掩码、点距离采集雷达的距离、反射强度等

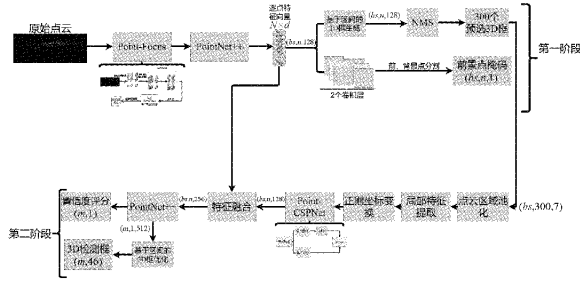


图 2 改进 PointRCNN 整体网络结构

参数进行 concat 并作为新的局部特征送入到 Point-CSPNet, 所得结果再与前述第一步得到的特征向量进行融合并送入到 PointNet++ 以得到最后 3D 检测框。

如图 3 所示的算法网络流程, 本文首次将二维 Focus、CSPNet 结构引入三维点云而提出 Point-Focus 和 Point-CSPNet, 在一定程度上提升了原始 PointRCNN 的检测精度和鲁棒性。

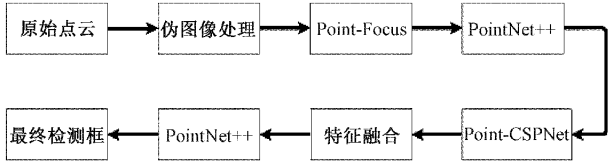


图 3 改进 PointRCNN 的算法流程

2.2 Point-Focus 结构

本文受 YOLO 算法中的 Focus 结构启发, 创造性提出将 Focus 结构的思想应用到点云目标检测当中去, 提出 Point-Focus 结构。原始点云数据传入 Point-Focus 结构后会进行一定程度的精简, 在有效减少数据量的同时保留重要的点云信息特征。

如图 4 所示, 这一结构先对原始点云数据进行鸟瞰视角的伪图像处理, 即将原始点云进行网格化操作并对每一网格编码。编码后将特征取回其原始的位置, 就可以创建大小为 (M, N, O) 的伪图像。得到伪图像之后, 在俯瞰视角的伪图像上每隔一个柱状框取一个框, 这样就可以得到 4 个互补而又类似的伪图像, 但却没有丢失任何信息, 经过这一操作, 通道空间变化为原来的 4 倍。经过 Focus 结构后的伪图像再还原成三维点云, 送入 PointNet++ 中进行点云的特征提取操作, 得到每个点的特征向量。Focus 结

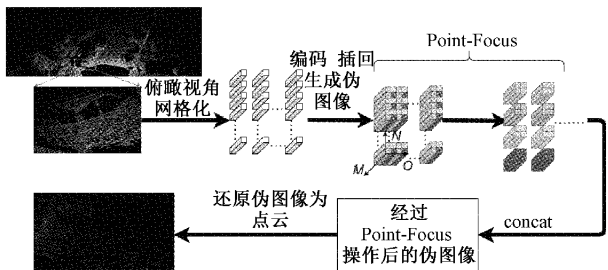


图 4 Point-Focus 结构

构的计算量和参数量要比普通卷积多 4 倍, 但是在进行下采样时没有信息丢失, 这也是 Focus 结构较邻近下采样的优势之处^[18]。

Point-Focus 结构伪代码如下:

算法 1: Point-Focus

Result: D_{out} : the point cloud data that across the Point-Focus structure

1 **Initialization:**

2 P : input raw point cloud data

3 N_{iter} : the number of iteration

4 P_k : meshed point cloud from BEV perspective

5 P_f : the result after splicing focus structure

6 Main Loop:

7 or $i=1; N_{iter}$ do

8 $model = PointFocus(P)$

9 for $k=1; |P|$ do

10 $P_k \leftarrow grid(P)$;

11 $L = FocusConv \leftarrow P_k$;

12 $P_f \leftarrow concat(L)$;

13 $restore \leftarrow P_f$, // restore P_f to three dimensional

14 return P_f to point cloud;

15 **end**

16 **end**

2.3 Point-CSPNet 结构

本文首次提出利用 Point-CSPNet 在点云目标检测中进行特征融合。具体来说就是将上一层得到的特征分为两部分, 其中一部分直接输入多层感知机 (multilayer perception, MLP) 中得到 128 维的局部特征, 另一部分不做处理。最后再将这两部分进行特征拼接以方便输入下一结构, 即进一步与第一阶段中 PointNet++ 输出结构后的特征 ($batchsize, n, 128$) 进行融合, 得到特征 ($batchsize, n, 256$)。针对 CNN 网络, CSPNet 结构可以有效的增强 CNN 网络的学习能力, 且同时能够在一定程度上提升网络的准确率。如图 5 所示, Point-CSPNet 的主要方法是将上一层网络的所有输出特征连接起来, 并将其作为下一层的输入, 使得势最大化; 同时也将梯度信息进行结合, 让高势特征和稀疏连接提升整个网络的学习能力。

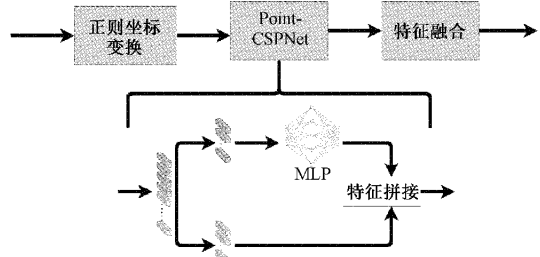


图 5 Point-CSPNet 结构

Point-CSPNet 伪代码如下:

算法 2: Point-CSPNet

Result: f : the output features after Point-CSPNet

```

1 Initialization:
2  $f_v$ : the output features after canonical
3  $k_1, k_2$ : the input characteristics are divided into
two parts
4 Main Loop:
5 for  $i=1; N_{iter}$  do
6    $model = PointCSPNet(f_v)$ 
7   for  $k$  in  $f_v$  do
8      $k_1, k_2 \leftarrow divide(f_v)$ ;
9      $k_3 \leftarrow MLP(k_1, k_2)$ ;
10     $f \leftarrow concat(k_2, k_3)$ ;
11  return  $f$ ;
12 end
13 end

```

2.4 损失函数

在点云前景点和背景点分割阶段,前景点偏少,背景点偏多,在数量上具有不平衡性,因此通过 focal loss 降低大量简单负样本在训练中所占的权重,式(1)表示其损失函数:

$$\delta_{focal} = -\alpha_t (1 - i_t)^\gamma \log(i_t)$$

$$i = \begin{cases} i, & \text{前景点} \\ 1 - i, & \text{其他} \end{cases} \quad (1)$$

式中: α 是平衡因子, γ 是可调节因子, i 表示样本的复杂程度,设定 $\alpha = 0.25, \gamma = 2$ 。

基于区间 bin 的精选化操作对初始回归结果在坐标轴上分组,当把数据进行分组处理后,数据会变得更加稳定,从而减少目标初始候选框的数量,得到候选框。式(2)表示在基于 bin 的过程中的损失函数:

$$\delta_{bin}^{(p)} = \sum_{u \in (x, z, \theta)} (T_{cls}(\widehat{bin}_u^{(p)}, bin_u^{(p)}) + T_{reg}(\widehat{res}_u^{(p)}, res_u^{(p)}))$$

$$\delta_{res}^{(p)} = \sum_{v \in (y, h, w, l)} T_{reg}(\widehat{res}_v^{(p)}, res_v^{(p)})$$

$$\delta_{reg} = \frac{1}{N_{pos}} \sum_{p \in pos} \delta_{bin}^{(p)} + \delta_{res}^{(p)} \quad (2)$$

其中, pos 是前景点集合, N_{pos} 是前景点的数量, $\widehat{bin}_u^{(p)}$ 是 bin 过程后得到的预测框, $\widehat{res}_v^{(p)}$ 是 $\widehat{bin}_u^{(p)}$ 与前景点预测框的差值, $bin_u^{(p)}$ 和 $res_v^{(p)}$ 是真值目标, T_{cls} 为交叉熵分类损失, T_{reg} 为平滑 L1 损失, u 为参数 x, z, θ 的集合, v 为参数 y, h, w, l 的集合。

式(3)则表示在 RCNN 阶段的损失函数:

$$\sigma = \frac{1}{\|m\|} \sum_{i \in m} \varphi_{cls}(prob_i, label_i) +$$

$$\frac{1}{\|m_{pos}\|} \sum_{i \in m_{pos}} (\widetilde{\sigma}_{bin}^{(i)} + \widetilde{\sigma}_{res}^{(i)}) \quad (3)$$

$$\widetilde{\sigma}_{bin}^{(i)} = \sum_{u \in (x, z, \theta)} \varphi_{cls}(b_i, b_i^{gt})$$

$$\widetilde{\sigma}_{res}^{(i)} = \sum_{v \in (y, h, w, l)} \varphi_{reg}(b_i, b_i^{gt})$$

其中, b_i 是目标初始候选框, m 是目标初始候选框的集合, m_{pos} 则是用来存储回归后得到的第一候选框, $prob_i$ 是 b_i 的置信度, $label_i$ 是第一候选框和真值框的吻合程度, φ_{cls} 是监督预测置信度的交叉熵损失, φ_{reg} 为平滑 L1 损失, b_i^{gt} 为真值框。

3 实验与结果分析

本文实验基于 OpenPCDet 框架进行实现^[19]。OpenPCDet 是为了方便统一 KITTI、nuScene 等主要点云数据集和各种不同点云目标检测算法的框架。利用这一框架可以方便的运用多种数据集和算法进行实验。

本章说明了在 KITTI 数据集上进行的实验与验证^[20], 分析了实验的结果并与其他典型算法进行了检测精度值的对比。同时也测试了所得模型在 ApolloScape^[21]、BLVD^[22] 数据集上的可视化表现效果。

本文所采用的硬件环境为: NVIDIA RTX 3080Ti 12 G 显存 GPU、Intel i7 11 代 CPU、16 G 内存、500 G 以上硬盘; 软件环境为: Ubuntu 16.04 LTS、Python 3.6.5、Cuda 11.2、Pytorch 1.10。

3.1 KITTI 数据集

KITTI 数据集是自动驾驶领域的权威数据集^[23], 本文的模型训练和评估测试也主要在 KITTI 数据集上进行。KITTI 数据集由德国卡尔斯鲁厄理工学院和丰田美国技术研究院联合创办的, 是目前世界上最大的自动驾驶领域计算机视觉算法评测数据集。图 6 所示为 KITTI 数据集中某一场景的二维图像和点云对应的情况。



图 6 KITTI 数据集中的图像点云对照示例

3.2 其他测试数据集

ApolloScape 是百度公司阿波罗自动驾驶项目的一部分, 包含激光雷达点云、多视图 RGB 图像, 其中每个像素和

每个 3D 点都进行了语义标记,此外,还提供了每张图像的采集地点。图 7 所示为 ApolloScape 数据集中的某一个点云场景。

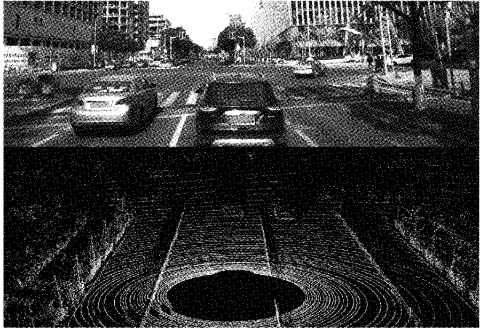


图 7 ApolloScape 数据集中的图像点云对照示例

BLVD 数据集是西安交通大学人工智能与机器人研究所研究人员为自动驾驶构建的大规模点云数据集。图 8 所示为 BLVD 数据集中的某一场景。

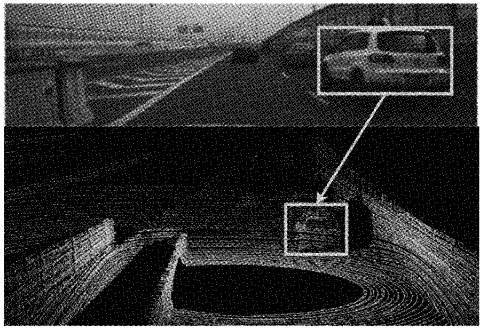


图 8 BLVD 数据集中的图像点云对照示例

本文主要利用这两个数据集进行网络模型的可视化测试,用以测试改进后模型在 KITTI 以外的数据集上的泛化能力。

3.3 检测结果可视化

表 1 列出了改进算法在 KITTI 数据集的可视化检测效果。表 2 列出了改进算法和当前主流算法在不同数据集上的检测结果,通过对比,可以清晰看出不同算法在不同场景下对于车辆目标的检测效果。图 9 则展示了改进前后算法检测车辆目标的效果差异。

KITTI 数据集上,改进 PointRCNN 在远处具有较好的表现效果,也极少会有误检情况,但对于大货车等大目标仍有漏检情况;同时,改进算法在远距离目标的检测上也得到了提升。而 PointPillars、PartA2 的远距离检测效果相对较好,但其误检较多。在 BLVD、ApolloScape 数据集上的检测效果也与之类似。这也说明了改进算法在不同数据集上稳定的检测能力。

3.4 算法数据对比

实验结果中,表 3 所示为原始 PointRCNN 及其优化算法的 AP 值对比;表 4 所示为优化后的算法与当前主流算

表 1 改进算法在 KITTI 数据集上的测试结果可视化

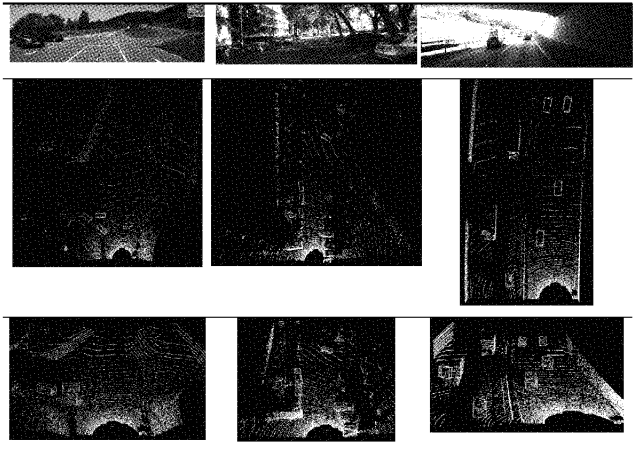


表 2 其他主要算法检测结果俯视图可视化对比

数据集	PointRCNN	PointPillars	PartA2	本文
KITTI				
ApolloScape				
BLVD				

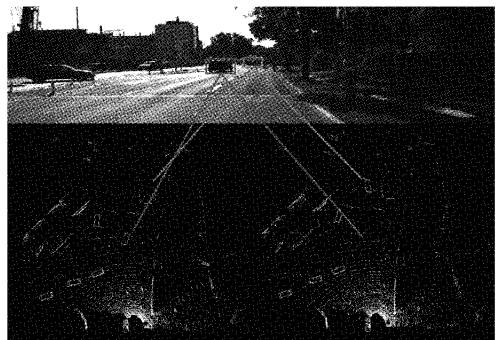


图 9 改进后算法与原始算法可视化效果差异对比 (左侧为原始算法、右侧为改进算法)

法的 AP 值对比;表 5 所示为优化后算法与其他主要算法对于行人等小目标的检测 AP 值;这都在一定程度上反映了改进算法的检测效果以及优化工作的有效性;表 6 所示为改进算法针对原始算法的平均精度值提升情况。

3.5 消融实验结果

表 7 所示为不同优化策略对算法整体效果的影响。

表3 PointRCNN 及其优化工作在 KITTI 数据集上对“car”类别的检测结果 AP 值对比

算法	3D			BEV		
	易	中	难	易	中	难
文献[3]	88.57	78.32	76.40	89.96	86.49	84.16
文献[9]	87.92	77.40	76.03	—	—	—
文献[10]	84.88	80.77	77.44	—	—	—
文献[11]	88.91	78.92	77.07	90.05	87.65	86.51
文献[12]	88.26	78.13	76.66	—	—	—
本文	88.87	78.61	77.73	90.11	87.74	86.74

注:本文中的原始 PointRCNN 的 AP 值基于 OpenPCDet 框架测试

表4 典型点云目标检测算法在 KITTI 数据集上对“car”类别的检测结果 AP 值对比

算法	3D			BEV		
	易	中	难	易	中	难
F-PointNet ^[24]	81.20	70.39	62.19	88.70	84.00	75.33
VoxelNet	81.97	65.46	62.85	89.60	84.81	78.57
SECOND	83.13	73.66	66.20	88.07	79.37	77.95
PointPillars	79.05	74.99	68.30	88.35	86.10	79.83
PointRCNN	88.57	78.32	76.40	89.96	86.49	84.16
PointGNN ^[25]	88.33	79.47	72.29	93.11	89.17	83.90
SE-SSD ^[26]	91.49	82.54	77.15	95.68	91.84	86.72
CIA-SSD ^[27]	89.59	80.28	72.87	93.74	89.84	82.39
本文	88.87	78.61	77.73	90.11	87.74	86.74

表5 改进算法在 KITTI 数据集上对“Pedestrian”类别的检测结果 AP 值对比

算法	3D			BEV		
	易	中	难	易	中	难
PointPillars	50.42	44.57	40.73	54.70	49.67	46.03
PartA2 ^[28]	60.75	53.75	49.43	63.36	57.93	53.15
PointRCNN	64.50	55.41	51.55	65.82	60.43	53.84
本文	63.33	56.24	51.24	64.90	60.66	54.10

表6 改进算法与原始算法的平均精度值比较

算法	3D	BEV
原始 PointRCNN	81.10	86.87
改进 PointRCNN	81.74	88.20

表7 不同优化模块对改进算法“car”类别的影响对比

Point-Focus	Point-CSPNet	3D			BEV		
		易	中	难	易	中	难
×	×	88.57	78.32	76.40	89.96	86.49	84.16
✓	×	88.59	78.44	76.67	89.98	86.79	84.34
×	✓	88.65	78.59	77.23	90.04	87.12	85.95
✓	✓	88.87	78.61	77.73	90.11	87.74	86.74

3.6 实验总结

经过实验,对比展示了改进算法与原始 PointRCNN 的检测效果,同时还引入了 PointPillars、PartA2 等算法的检测效果对比。

改进 PointRCNN 相较于其他算法,具有更少的误检和更好的表现效果。在 KITTI 数据集的检测基础上,还使用了 ApolloScape、BLVD 数据集对算法模型进行进一步的测试检验。实验结果说明了改进 PointRCNN 的鲁棒效果和在其它数据集上进行检测的泛化能力,且从可视化角度来看,在原始 PointRCNN 稳定的基础上,进一步提高了其在较远距离目标上的检测效果。

从数据上看,在 3D 场景下改进算法精度提升不是很明显,其 mAP 值仅从 81.10% 提升至 81.74%;在 BEV 场景下平均精度值则从 86.87% 提升至 88.20%,具有一定程度的提升效果。在当前主流算法的横向比较中也具有较为强势的表现效果,在困难场景下的 AP 值超过了当前 AP 值最高的点云目标检测算法,但其他场景略低于 SE-SSD。而对于行人等小目标物体,改进算法也在 BEV 视角的困难场景下得到了一定的提升,这也验证了改进方法在较小目标检测上的有效性。

4 结论

三维点云目标检测技术作为智能驾驶技术中的重要部分,对智能驾驶的发展具有重要作用。本文在基于 PointRCNN 的基础上,主要贡献在于:

1) 提出 Point-Focus 结构有效的减少数据量并保留重要的点云信息特征,提升原始算法在较远距离上的检测效果;2) 提出 Point-CSPNet 结构进行特征拼接,提高网络的整体学习能力,进而提高目标检测精度;3) 引入 ApolloScape、BLVD 数据集进行测试实验,进一步验证改进算法在不同数据集上的泛化性能。

经过实验,改进算法在 KITTI、ApolloScape、BLVD 等多个点云数据集上具有较好的表现效果,其中 KITTI 数据集的 3D 场景下平均精度较原始 PointRCNN 从 81.10% 提升到 81.74%;BEV 场景下的平均精度较原始 PointRCNN 从 86.87% 提升到 88.20%。

参考文献

- [1] GUO Y, WANG H, HU Q, et al. Deep learning for 3D point clouds: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(12): 4338-4364.
- [2] 李晖晖,周康鹏,韩太初.基于 CReLU 和 FPN 改进的 SSD 舰船目标检测[J].仪器仪表学报,2020,41(4): 183-190.
- [3] BELLO S A, YU S, WANG C, et al. Review: Deep learning on 3D point clouds[J]. Remote Sensing, 2020, 12(11): 1729-1763.
- [4] SHI S, WANG X, LI H, et al. PointRCNN: 3D

- object proposal generation and detection from point cloud[C]. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2019; 770-779.
- [5] YANG Z, SUN Y, LIU S, et al. STD: Sparse-to-dense 3D object detector for point cloud[C]. IEEE/CVF International Conference on Computer Vision, 2019; 1951-1960.
- [6] XIE J, ZHENG Z, GAO R, et al. Generative VoxelNet: Learning energy-based models for 3D shape synthesis and analysis [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(5): 2468-2484.
- [7] YAN Y, MAO Y X, BO L. Second: Sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.
- [8] LANG A H, VORA S, CAESAR H, et al. PointPillars: Fast encoders for object detection from point clouds[C]. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2019; 12689-12697.
- [9] 徐兵. 无人驾驶汽车车辆障碍物检测及换道决策研究[D]. 杭州:浙江大学,2020.
- [10] YAN M, LI Z, YU Z, et al. An end-to-end deep learning network for 3D object detection from RGB-D data based on hough voting[J]. IEEE Access, 2020, 8; 138810-138822.
- [11] 梁小芳. 基于点云和深度学习的目标检测方法研究[D]. 荆州:长江大学,2021.
- [12] LI J, HU Y. A density-aware PointRCNN for 3D object detection in point clouds[C]. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2021, DOI: 10.48550/arXiv.2009.05307.
- [13] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014; 580-587.
- [14] BOCHKOVSKIY A, WANG C Y, LIAO H. YOLOv4: Optimal speed and accuracy of object detection[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, DOI: 10.48550/arXiv.2004.10934.
- [15] 王文胜,李继旺,吴波,等. 基于 YOLOv5 交通标志识别的智能车设计[J]. 国外电子测量技术, 2021, 40(10): 158-164.
- [16] WANG C Y, MARK L, WU Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]. IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPR), 2020; 390-391.
- [17] CHARLES Q R, YI L, SU H, et al. PointNet++: Deep hierarchical feature learning on point sets in a metric space[C]. Conference on Neural Information Processing Systems, 2017, 30; 5099-5108.
- [18] 徐晓光,李海. 多尺度特征在 YOLO 算法中的应用研究[J]. 电子测量与仪器学报,2021,35(6):96-101.
- [19] TEAM O D. OpenPCDet: An open-source toolbox for 3D object detection from point clouds[J]. <https://github.com/open-mmlab/OpenPCDet>, 2020.
- [20] GEIGER A, LENZ P, STILLER C, et al. Vision meets robotics: The KITTI dataset[J]. International Journal of Robotics Research, 2013, 32 (11): 1231-1237.
- [21] HUANG X, CHENG X, GENG Q, et al. The apolloscape dataset for autonomous driving [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018; 954-960.
- [22] XUE J, FANG J, LI T, et al. BLVD: Building a large-scale 5D semantics benchmark for autonomous driving [C]. IEEE International Conference on Robotics and Automation, 2019; 6685-6691.
- [23] 毕雪婷,刘小军,邵文远. 基于聚类方法的自动驾驶场景下的三维目标检测[J]. 电子测量技术, 2021, 44(6): 103-107.
- [24] CHARLES Q R, LIU W, WU C, et al. Frustum PointNets for 3D object detection from RGB-D data[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2018; 918-927.
- [25] SHI W, RAGUNATHAN R. Point-GNN: Graph neural network for 3D object detection in a point cloud[C]. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2020; 1711-1719.
- [26] ZHENG W, TANG W, JIANG L, et al. SE-SSD: Self-ensembling single-stage object detector from point cloud[C]. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2021; 14494-14503.
- [27] ZHENG W, TANG W, CHEN S, et al. CIA-SSD: Confident IoU-aware single-stage object detector from point cloud[C]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(4): 3555-3562.
- [28] SHI S, WANG Z, SHI J, et al. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(8): 2647-2664.

作者简介

王庆林, 硕士研究生, 主要研究方向为人工智能与机器视觉。

E-mail: wqlandjay@163.com

李辉, 硕士研究生, 主要研究方向为数字信号处理。

E-mail: 1163200071@qq.com

谢礼志, 硕士研究生, 主要研究方向为嵌入式系统与物联网。

E-mail: 1254916118@qq.com

谢剑斌, 博士, 教授, 主要研究方向为人工智能与机器视觉。

E-mail: jbxie@126.com

彭石林, 博士, 教授, 主要研究方向为核磁共振与物联网。

E-mail: pengsl2002@sina.com