用于激光雷达目标检测的单阶段无锚框优化网络

朱望江[1,2)], 郭建伟[1,2)], 张吉光[1,2)], 孟维亮[1,3,2)*], 张晓鹏[1,2)]

[1)] (中国科学院自动化研究所多模态人工智能系统全国重点实验室 北京 100190)
[2)] (中国科学院大学人工智能学院 北京 100049)
[3)] (之江实验室 杭州 311121)
(weiliang.meng@ia.ac.cn)

摘 要: 激光雷达目标检测近年来开始借鉴图像目标检测的网络设计, 但依然存在计算低效无法满足实时应用以及网络结构简单导致性能不足的问题. 为此, 提出的网络采用了单阶段无锚框的简洁设计; 优化了激光点云体素化表达, 在提升计算效率的同时保留了一部分点云高程特征; 基于残差网络的思想, 设计了更深的主干网络结构用于提取深度特征; 引入特征金字塔来提升小目标的检测效果. 在公开数据集 KITTI 上, 所提网络的 mAP 指标在各类别目标的检测中均取得了领先的性能(提高了 1%~3%). 在自动驾驶计算平台上的运行时间测试表明, 所提网络能够达到 43 ms/帧的处理速度, 满足实时性需求.

关键词: 目标检测; 激光雷达; 自动驾驶; 单阶段
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023-00329

An Optimized Network for One-Stage Anchor-Free LiDAR Object Detection

Zhu Wangjiang[1,2)], Guo Jianwei[1,2)], Zhang Jiguang[1,2)], Meng Weiliang[1,3,2)*], and Zhang Xiaopeng[1,2)]

[1)] (*State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing* 100190)
[2)] (*School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing* 100049)
[3)] (*Zhejiang Lab, Hangzhou* 311121)

Abstract: In recent years, the realm of laser radar target detection has increasingly drawn inspiration from network designs employed in image target detection. Nevertheless, persistent challenges encompass inefficient computations impeding real-time applications and suboptimal performance resulting from overly simplistic network structures. The proposed network embraces a refined design characterized by a single-stage, anchor-free methodology. It strategically enhances the voxelization representation of laser point clouds, thereby bolstering computational efficiency while preserving crucial elevation features. Leveraging the principles of residual networks, a deeper backbone network structure is crafted to adeptly extract deep features. Additionally, the incorporation of a feature pyramid enhances the detection capabilities, especially for smaller targets. Across various object categories, the proposed network demonstrates outstanding performance on the KITTI public dataset, as reflected in the mAP indicator with 1% to 3% improvement. Runtime assessments on an autonomous driving computing platform have a processing speed of 43 ms per frame. validating the network's ability to meet real-time requirements.

收稿日期: 2023-06-16; 修回日期: 2023-12-22. **基金项目**: 国家自然科学基金(U21A20515, U22B2034, 61972459, 61971418, 62172416, 32271983); 之江实验室开放课题(2021KE0AB07). **朱望江**(1988—), 男, 博士, 主要研究方向为计算机视觉; **郭建伟**(1988—), 男, 博士, 副研究员, 硕士生导师, CCF 会员, 主要研究方向为计算机图形学、计算机视觉等; **张吉光**(1989—), 男, 博士, 副研究员, CCF 会员, 主要研究方向为计算机图形学、计算机视觉等; **孟维亮**(1981—), 男, 博士, 副研究员, 硕士生导师, CCF 杰出会员, 论文通信作者, 主要研究方向为计算机视觉、计算机图形学等; **张晓鹏**(1963—), 男, 博士, 研究员, 博士生导师, CCF 会员, 主要研究方向为计算机视觉、计算机图形学等.

Key words: object detection; light detection and ranging; autonomous driving; one-stage

激光雷达(light detection and ranging, LiDAR)是自动驾驶的重要传感器, 对于城市场景的 Robotaxi/RoboBus 几乎是必需的传感器. 这得益于 LiDAR 能提供精确的 3D 位置, 继而对场景中的车辆、行人等目标的运动意图判断提供可靠的依据.

传统的 LiDAR 目标检测[1-2]先去除背景点云, 然后对前景点云进行聚类和跟踪; 但是点云噪声很难去除干净, 导致误检比例较高, 且点云过分割和欠分割也会导致同一个目标分裂为多个, 或相邻目标被错误聚类到一起的问题, 使得模型鲁棒性较差, 且无法基于部分可见的点云推断目标的类型和准确轮廓.

随着深度学习和卷积神经网络的兴起, 研究人员开始探索将深度学习应用于 LiDAR 目标检测[3-7]. 近年来, 图像目标检测技术取得了很好的突破, 从二阶段[8]走向单阶段[9], 采用特征金字塔网络(feature pyramid networks, FPN)[10]解决不同目标尺度跨度大的问题, 以及无锚框的方式[11]让网络训练更加简洁高效. 还有一些工作很自然地开始迁移图像目标检测领域的成功做法, 将无序的点云先转化成规整的体素化表达[12], 然后在 BEV(bird eye view, BEV)空间进行类似图像的目标检测运算. 但是, 这些方法存在计算效率低[12]或点云高程信息编码不充分的问题[13]; 同时, 由于点云分辨率较高且缺乏可用的图像预训练网络, 因此点云目标检测的主干网络未能充分借鉴图像处理领域的研究进展.

本文提出 LiDAR 目标检测的单阶段无锚框优化网络. 通过探索点云体素化的不同方式发现, 简单的二值编码方式比复杂的点云特征抽取[14]检测效果更好, 并且二值编码方式的计算效率更高, 甚至可以在 CPU 上完成. 相反, PointNet[14]的计算效率低的原因是因为无序的点云处理是访存密集操作. 进一步, 本文还发现, 将 LiDAR 的反射率属性加入编码对于降低误检非常有帮助. 为了兼顾目标检测效果和检测范围, 体素化之后的伪图像分辨率一般较高, 导致无法使用深度网络 ResNet-50[15], 需要优化主干网络. 本文借鉴残差学习的思想, 在现有网络结构中加入直连连接, 显著地提升了检测效果. 因为本文要检测的目标的尺度变化非常大(从卡车到行人), 所以需要将 FPN 应用到 LiDAR 目标检测. 实验结果表明, 引入 FPN 后, 对于行人和两轮车等尺度较小的目标的检测效果提升非常明显. 最后在公开数据集 KITTI 上进行实验, 验证了本文网络可以超越文中对比的工作.

1 相关工作

(1) 图像目标检测. RCNN(rich feature convolutional neural network)[16]开创了将卷积神经网络应用于目标检测的先河; 紧随其后, Fast R-CNN[17]和 Faster R-CNN[8]对不同区域进行计算共享, 并通过第 1 阶段的神经网络高效实现区域提案, 使得基于卷积神经网络的目标检测走向实用. 上述工作激发了后续目标检测工作的大爆发: Focal Loss[18]有利于克服正负样本不均衡的问题, 将目标检测从二阶段的算法推向一阶段[9,19]的更加简洁高效的设计; 为了在同一个位置检出尺度不同的目标, Faster R-CNN[8]采取锚框设计, FPN[10]证明可以在不同尺度的特征上进行不同大小的目标预测, 不再需要手动设定超参数繁杂的锚框; 文献[11]进一步证明, 有锚框的设计和无锚框的设计在目标检测性能上差异不大. 因此, 图像目标检测收敛到单阶段的无锚框的方法上, 计算效率和训练难度都大幅优化.

(2) LiDAR 目标检测. LiDAR 点云中的目标检测是 3D 问题, 早期工作[20-21]直接探索 3D 卷积. 这些工作虽然提供了一个直观的框架, 但运行速度太慢, 一般帧率为 1~2 fps(frames per second), 难以满足自动驾驶这类场景的要求.

VoxelNet[12]首次将点云先进行体素化, 经过 3D 卷积后转换为 2D 的特征图, 再使用图像目标检测的方法, 虽然速度上有改观, 达到了 4 fps, 但依然无法满足实际自动驾驶系统的需求. Yan 等[22]利用 3D 体素的稀疏性, 提出一种稀疏卷积的高效实现方式 SECOND, 可以达到 20 fps, 但稀疏卷积访存不友好, 不利于跨平台迁移. 为了做到多平台友好, PointPillar[13]放弃了 3D 的点云体素化, 直接在 BEV 视角下将点云划分到 2D 网格中, 每个网格对应 3D 空间的一根柱子, 柱子内部不再进行高层上的划分; 显然, 这种做法会导致点云高程信息的损失. 本文提出体素化方法, 其计算简单, 可在 CPU 上实现, 并且没有使用访存不友好的稀疏卷积, 能有效地保留高程信息, 检测性能更高, 网络结构更加跨平台友好.

2　本文网络

得益于图像目标检测的发展，简洁的单阶段无锚框的方法被迁移到点云目标检测[23]. 受到文献[23]的启发，本文也采用单阶段无锚框的方案，并且在主干网络的结构以及 FPN 上进行了优化，显著地提升了目标检测性能.

本文提出的 LiDAR 目标检测网络结构如图 1 所示，由 4 部分构成: (1) 体素特征编码. 借鉴图像目标检测中的有效方法，将无序的点云转换成与图像类似的体素或像素表达. (2) 在 BEV 上的定义主干网络. 此时，点云的高层信息已经被压缩到特征图的通道中，在 BEV 空间下同类目标的尺度信息差异较小，对目标检测有利. (3) FPN. 针对大小不一的目标尺度图像进行检测的有效处理方法，广泛应用在当前目标检测任务中. 因为本文要检测的目标尺度跨度较大(从卡车到行人)，所以 FPN 也是非常必要的. (4) 检测头. 包含类别分类任务，以及目标的位置、尺寸等属性预测.

2.1　体素特征编码

为了能够迁移图像目标检测的方法，首先需要将不规则分布的点云转化成规则的体素或像素表达，这种表达称为伪图像.

(1) 体素划分. 将 LiDAR 周边一定范围内的 3D 空间划分为均匀的小格点. 假设选取 LiDAR 周边的 $D\times H\times W$ 区域，并且每个维度的体素网格大小为 $v_D\times v_H\times v_W$，则在体素化之后将会得到分辨率为 $D'=D/v_D$，$H'=H/v_H$ 和 $W'=W/v_W$ 的 3D 体素化表达. 对于 PointPillar[13]这类不进行高层区分的体素化做法，可以认为是 $D=v_D$ 情况下的一个特例.

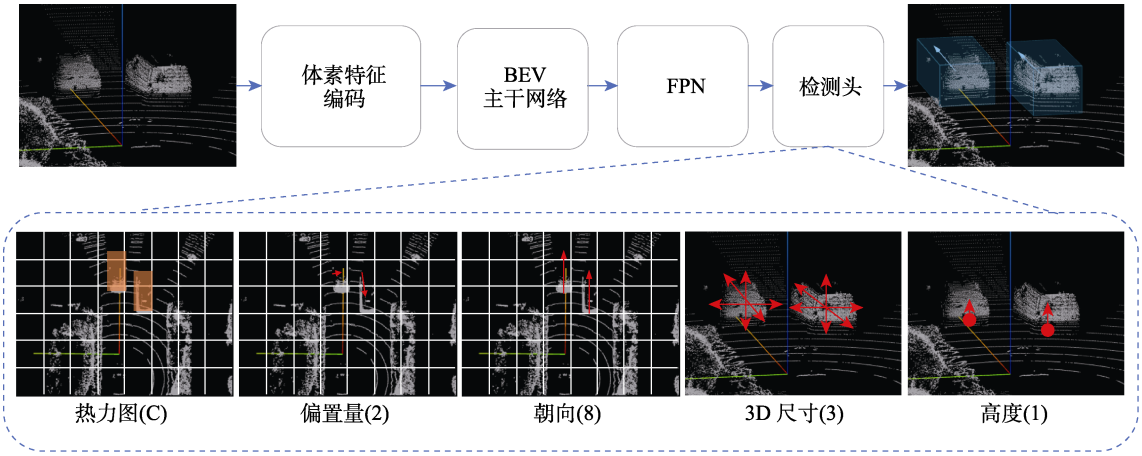


图 1　本文网络框架

(2) 体素特征编码. 给定一帧 LiDAR 点云，基于每个点的 3D 坐标，可以计算它所落入的体素网格. 由于点云集中在障碍物的表面，3D 空间中大部分位置是没有障碍物的，因此最终只有 0.5%~3.0%的网格是非空的[13,22].

用 $\boldsymbol{V}=\left\{\boldsymbol{p}_i=\left[x_i,y_i,z_i,r_i\right]^{\mathrm{T}}\in\mathbb{R}^4\right\}_{i=1,2,\cdots,t}$ 表示一个非空的体素网格，其由 t 个 LiDAR 点组成，$\boldsymbol{p}_i$ 表示其中第 i 个点，包含 3D 位置信息和反射率 r_i. 已有很多基于这 4D 基础信息构建每个体素的特征表达的工作:

VoxelNet[12]将 4D 特征增广到 7D,

$$\boldsymbol{V}_{\mathrm{in}}=\left\{\hat{\boldsymbol{p}}_i=\left[x_i,y_i,z_i,r_i,x_i-v_x,y_i-v_y,z_i-v_z\right]^{\mathrm{T}}\in\mathbb{R}^7\right\}_{i=1,2,\cdots,t};$$

其中，(v_x,v_y,v_z) 表示落在该网格内所有点的坐标均值. 然后通过类似 PointNet[14]的多层感知器进行特征变换. VoxelNet[12]定义多层体素特征编码(voxel feature encoding, VFE)进行深度特征抽取，每层 VFE 由一个全连接层、一个批归一化层[24]和一个修正线性单元(rectified linear unit, ReLU)[25]构成.

在 VoxelNet 的 7D 特征的基础上，PointPillar[13]继续增加每个点在 XY 方向上相对体素网格中心点的偏移量，将 7D 特征增广到 9D；然后同样采用 VFE 的做法进行深度特征编码.

通常，落入近处的体素网格中的点云数量会比较多，从实际效果和计算量的角度出发，对于点云数量较多的网格一般会做一些随机采样，保证每个网格最多 T 个点.

本文提出更简单有效的 2 种体素表达方式:

(1) 二值表达. 通过网格中是否落入 LiDAR 点云，将该网格做二值标记: 有点云的体素网格标记为 1，无点云的体素网格标记为 0，最后形成一

个 $D'\times H'\times W'$ 的 3D 二值矩阵，将其输入后续的主干网络做卷积操作. 这样做的好处是计算量大大缩减，甚至可以在 CPU 上完成，无须占用神经网络专用计算单元(如 NVIDIA Orin 的 GPU 或者 DLA)的算力. 事实上，神经网络专用计算单元在处理这种访存密集且不连续的点云预处理上并不占优势，对于需要实时运行的自动驾驶系统非常不友好.

(2) 反射率表达. 该方式是在二值表达方式的基础上增加反射率的信息. 由于自动驾驶场景中的交通牌、车辆、行人这些不同类型的目标的反射率差异较大，因此加上这部分信息有益于神经网络区分不同类型的障碍物，减少误检.

本文提出的这 2 种体素编码方式，计算上更加高效简洁，占据网格的生成可以快速地在 CPU 上实现，跨平台友好，并且由于结合了 VoxelNet[12]的高层分辨率的优势，目标检测的效果上比 PointPillar[13]效果更好.

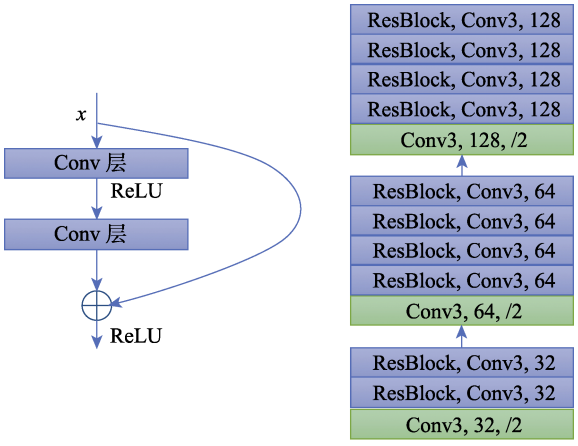
2.2 主干网络

对于一个实用的典型自动驾驶系统，LiDAR 的感知范围至少需要覆盖前后各 80 m，左右各 40 m(为了应对过路口的情况)；而为了检出行人这类俯视图上尺寸较小的目标，体素化采用的网格尺寸又不能太小，导致分辨率巨大. 如果分辨率设定为 $v_D=10$ cm，$v_H=7$ cm 和 $v_W=7$ cm，那么送入主干网络的特征图的空间分辨率会达到 2 400×1 200 像素，而常规图像目标检测[18]的分辨率在 512×512 像素左右.

受限于网络空间分辨率，在 LiDAR 目标检测上只能使用较浅的神经网络层数：VoxelNet[12]采用了 3 层 3D 卷积，SECOND[22]共采用 13 层 2D 卷积，PointPillar[13]共采用 19 层 2D 卷积，AFDet[23]共采用 17 层 2D 卷积，这与图像检测任务上动辄使用 ResNet-50[15]或者 ResNet-101 相差非常大.

虽然无法直接使用 ResNet-50 这样的大网络，本文还是借鉴 ResNet 的思想，增加直连连接方式进行残差学习.

图 2a 所示为 ResNet[15]中设计的残差单元，本文没有直接复制原始的 ResNet-50 那样的结构，而是通过残差单元搭建图 2b 所示的主干网络结构：有 3 个阶段构成，每个阶段先做 2 倍下采样(见绿色框)降低空间分辨率；然后采用多个残差单元进行堆叠，随着空间分辨率的降低，逐步增大通道数量(蓝色框最后一个数字是通道数)来抽取更加丰富的高层语义特征.



a. 残差单元[14] b. 本文主干网络

图 2 残差单元和本文的主干网络结构

2.3 FPN

FPN[10]将传统的图像金字塔拓展到了特征图上，使得同一个输入可以生成多种不同分辨率的特征图，并在不同分辨率的特征图上各自检出相匹配大小的目标，有效地解决目标尺度变化范围过大的问题. FPN 在图像识别中被证明效果显著，几乎成为标准配置.

在 BEV 视角的 LiDAR 目标检测中，同一类型的目标尺度变化相对较小，不同类型的目标尺度差异巨大，如卡车和行人的尺寸会差几十倍.

已有工作[12-13,22-23]都是针对车辆和行人训练不同的模型，甚至输入尺寸都是不一样的，这对于算力和功耗受限的实际自动驾驶系统是难以接受的. 因此，本文在网络结构中使用 FPN[10]，使得不同类型的目标能够共享主干网络，同时又可以在最合适分辨率的特征图上进行检测. 类似文献[9]，本文在图 2 所示的 8 倍下采样的基础上增加 4 倍下采样的特征，用于行人和两轮车这类小尺度的目标检测.

2.4 检测头

本文选择无锚框的方法设计检测头，这类方法省去了烦琐的锚框超参数设定，在训练和推理阶段的效率也大大加快，形式上简洁、效果上和细致调参的有锚框的方案持平[11].

如图 1 所示，本文设计的检测头由类别预测热力图、中心点相对于特征像素坐标的偏移量、目标 3D 框的朝向、目标长宽高尺寸和目标中心点的高度 5 个部分组成.

(1) 热力图. 热力图的预测结果是 $\hat{\boldsymbol{M}}\in\mathbb{R}^{W\times H\times C}$，用于表征特征像素所在的位置是否存在本文定义类型的目标；其中，C 表示目标类型的数量，由于本文关注卡车、小汽车、行人和两轮车，因此取 $C=4$. 如果要检测更多的类型，可以非常方便地

增加 C 的数量.

采用类似 FCOS[9]的做法计算热力图的真值, 将落在真值框内的像素都设为前景, 其他区域设为背景. 对于热力图上的每一点 (x, y), 可以通过其所在层的下采样倍数换算到原始体素化网格的坐标 (sx, sy), 其中, s 表示下采样倍数; 然后根据 (sx, sy) 是否落在真值框中对真值热力图赋值, 公式为

$$\boldsymbol{M}_{x,y,c}=\begin{cases}1, & \text{if } (sx,sy)\in \text{前景像素}\\ 0, & \text{else}\end{cases}.$$

Focal Loss[18]能够很好地解决正负样本不均衡的问题, 抑制大量简单背景样本的损失, 因此大量使用在单阶段目标检测工作中. 本文也使用 Focal Loss 来定义热力图的损失函数

$$L_{\text{heat}}=-\frac{1}{N}\cdot \sum_{x,y,c}\begin{cases}\left(1-\hat{\boldsymbol{M}}_{x,y,c}\right)^{\alpha}\times\ln\left(\hat{\boldsymbol{M}}_{x,y,c}\right), & \text{if } \boldsymbol{M}_{x,y,c}=1\\ \left(\hat{\boldsymbol{M}}_{x,y,c}\right)^{\alpha}\times\ln\left(1-\hat{\boldsymbol{M}}_{x,y,c}\right), & \text{else}\end{cases}.$$

类似于 AFDet[23], 本文设定 $\alpha=2$.

(2) 偏置量. 在每个像素的位置估计当前像素到目标中心点的偏置量 $\hat{\boldsymbol{O}}\in\mathbb{R}^{W\times H\times 2}$. 偏置量的预测一方面可以弥补因为点云体素化带来的量化误差, 另一方面可以把不同位置的预测结果引导到同一个位置, 使得最终的预测结果的误检漏检更少. 因为 Smooth L_1 损失函数[17]能够避免误差较大时传导的梯度过大, 有利于学习过程的稳定性, 所以本文采取 Smooth L_1 损失函数对偏置量进行估计

$$L_{\text{offset}}=\frac{\sum_{x,y,c}\boldsymbol{M}_{x,y,c}\cdot\text{Smooth}L_1\left(\hat{\boldsymbol{O}}_{x,y}-\boldsymbol{O}_{x,y}\right)}{\sum_{x,y,c}\boldsymbol{M}_{x,y,c}}\quad(1)$$

本文仅对有真值框定义的区域, 即 $\boldsymbol{M}_{x,y,c}=1$ 的区域进行偏置量的计算, 其他背景区域不会计算此项损失. 式(1)中, $\boldsymbol{O}_{x,y}$ 表示当前像素到目标中心点偏置量的真值.

(3) 3D 尺寸. 对于每个前景像素, 预测其所在目标的长宽高, 类似于偏置量, 也采用 Smooth L_1 损失函数. 记网络预测的尺寸为 $\hat{\boldsymbol{S}}\in\mathbb{R}^{W\times H\times 3}$, 人工标注框映射到各个像素的真值为该项损失函数, 定义为 $\boldsymbol{S}\in\mathbb{R}^{W\times H\times 3}$. 则该项损失函数定义为

$$L_{\text{size}}=\frac{\sum_{x,y,c}\boldsymbol{M}_{x,y,c}\cdot\text{Smooth}L_1\left(\hat{\boldsymbol{S}}_{x,y}-\boldsymbol{S}_{x,y}\right)}{\sum_{x,y,c}\boldsymbol{M}_{x,y,c}}.$$

(4) 中心点高度. 因为开放道路自动驾驶场景中的路面有起伏, 为了能够比较准确地知道每个目标的海拔高度, 还需要在 BEV 上预测每个目标的实际高度 $\hat{\boldsymbol{Z}}\in\mathbb{R}^{W\times H\times 1}$. 同样地, 使用 Smooth L_1 损失函数

$$L_{\text{altitude}}=\frac{\sum_{x,y,c}\boldsymbol{M}_{x,y,c}\cdot\text{Smooth}L_1\left(\hat{\boldsymbol{Z}}_{x,y}-\boldsymbol{Z}_{x,y}\right)}{\sum_{x,y,c}\boldsymbol{M}_{x,y,c}}.$$

(5) 朝向角. 自动驾驶场景中非常关注目标的朝向角, 因为这会直接影响到该目标未来的预测轨迹. 记目标 k 的朝向角为 $\theta^{(k)}$, 表示目标绕着垂直于地面的轴在 BEV 平面上相对于水平轴转过的角度. 文献[26-27]证明, 直接回归 360°的角度效果较差, 本文提出 MultiBin 方法: 先对角度进行粗颗粒度的分类, 然后在被分到的区间内部再进行相对于中心角度的偏差回归.

本文将角度划分成 2 个大的区间: $\psi_1=\left[-\frac{7\pi}{6},\frac{\pi}{6}\right]$ 和 $\psi_2=\left[-\frac{\pi}{6},\frac{7\pi}{6}\right]$, 它们有一定的重叠区域. 对于每个区间, 预测 $\hat{\mu}_i^{(k)}\in\mathbb{R}^2$ 用于 Softmax 分类, $\hat{v}_i^{(k)}\in\mathbb{R}^2$ 用于预测相对于区间中心角度 γ_i 偏角的正弦和余弦值. 最终的朝向角损失函数定义为

$$L_{\text{ori}}=\frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{2}\text{Softmax}\left(\hat{\mu}_i^{(k)},\eta_i^{(k)}\right)+\frac{1}{N}\sum_{k=1}^{N}\sum_{i=1}^{2}\eta_i^{(k)}\left|\hat{v}_i^{(k)}-v_i^{(k)}\right|.$$

其中,

$$\eta_i^{(k)}=\begin{cases}1, & \text{if } \theta^{(k)}\in\psi_i\\ 0, & \text{else}\end{cases},$$

$$v_i^{(k)}=\left(\sin(\theta^{(k)}-\gamma_i),\ \cos(\theta^{(k)}-\gamma_i)\right).$$

得到这些分类和回归的预测结果后, 最终通过 arctan2 函数恢复绝对角度.

(6) 总体损失函数, 即最终的损失函数是上述所有项的加权和

$$L=L_{\text{heat}}+\lambda_{\text{offset}}L_{\text{offset}}+\lambda_{\text{size}}L_{\text{size}}+\lambda_{\text{altitude}}L_{\text{altitude}}+\lambda_{\text{ori}}L_{\text{ori}}.$$

通过实验, 最终选定权重系数 $\lambda_{\text{offset}}=1.0$, $\lambda_{\text{size}}=0.3$, $\lambda_{\text{altitude}}=1.5$, $\lambda_{\text{ori}}=1.0$.

3　实验及结果分析

3.1　数据集

目前, KITTI 数据集[28]常用于 LiDAR 目标检测中, 其包含 7 481 帧有标注的训练数据. 本文实

验中采取与文献[23,29]相同的做法，将数据分为 3 712 帧的训练集和 3 769 帧的验证集.

为了搭建本文的自动驾驶系统，针对中国特色的道路环境进行数据采集和标注，构建 121 375 帧训练集和 62 319 帧验证集，包含不同天气、不同道路环境的丰富场景，很好地支撑了实际的高速和城区环境的自动驾驶.

3.2 实验设置

(1) 训练参数. 除非特别说明，实验中使用 AdamW[30]优化器并使用 one-cycle[31]策略. 网络初始学习率设定为 3×10^{-3}，衰减率为 2，动量范围是 0.95~0.85，固定衰减系数为 0.01.

(2) 数据增强. 与文献[13,23]类似，在训练阶段对点云数据进行增强，包括放置虚拟目标、点云整体旋转、点云镜像. 其中，放置虚拟目标是将其他点云帧中的目标点云整体复制到当前点云帧，放置到空白区域上，并作一定的角度和位置的随机变换，对目标朝向角按照分布 $U\left(-\frac{\pi}{20},\frac{\pi}{20}\right)$ 进行扰动，对目标位置按照分布 $N(0,0.25)$ 进行扰动；点云整体旋转是将同一帧点云的所有点按照分布 $U\left(-\frac{\pi}{4},\frac{\pi}{4}\right)$ 整体旋转一个角度，模拟自车不同朝向时 LiDAR 扫描到的点云；点云镜像则是对整体场景进行左右手的镜像.

(3) 体素化分辨率. 在实际的自动驾驶系统中，为了满足复杂场景的检测距离和性能要求，采取比较细的体素化分辨率，$v_D=10$ cm，$v_H=7$ cm 和 $v_W=7$ cm；并且覆盖较大的检测范围，自车前后各 84 m，左右各 42 m，高度方向 4 m，体素化后分辨率为 $2\,400\times1\,200\times40$ 像素. 对于行人和两轮车，检测前后各 63 m，左右各 42 m. 注意：虽然检测范围不同，但不同类别共享一个网络，仅在训练和推理时去掉检测范围之外的结果.

对于 KITTI 数据集[28]，为了公平比较，本文采用与其他方法相同的检测范围和分辨率设置. 对于 KITTI 数据集上的车辆检测，输入范围为自车前方 70.4 m，左右各 40 m，高度方向 4 m，$v_D=10$ cm，$v_H=16$ cm 和 $v_W=16$ cm，体素化后的分辨率为 $416\times480\times40$ 像素.

3.3 方法对比

在 KITTI 数据集[28]上，采用平均精度均值(mean average precision, mAP)指标，将本文方法与其他高性能检测方法进行实验，包括 VoxelNet[12]，SECOND[22]，PointPillar[13]和 AFDet[23]方法，结果如表 1 所示. 可以看出，本文方法全面超过其他方法，证明了其有效性. 图 3 展示了本文方法检测结果的定性可视化，其中红色 3D 框代表小汽车，绿色 3D 框代表两轮车，淡蓝色 3D 框代表行人，亮蓝色 3D 框代表卡车.

表 1 在 KITTI 数据集上 5 种方法检测 3 种点云目标的 mAP 对比 %

方法	车辆			行人			两轮车		
	简单	中等	困难	简单	中等	困难	简单	中等	困难
VoxelNet[12]	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55
SECOND[22]	88.07	79.37	77.95	55.10	46.27	44.76	73.67	56.04	48.78
PointPillar[13]	88.35	86.10	79.83	58.66	50.23	47.19	79.14	62.25	56.00
AFDet[23]	88.91	84.69	79.83						
本文	**90.93**	**87.24**	**82.14**	**60.32**	**52.42**	**50.26**	**80.24**	**63.72**	**58.37**

注. 粗体表示最优值.

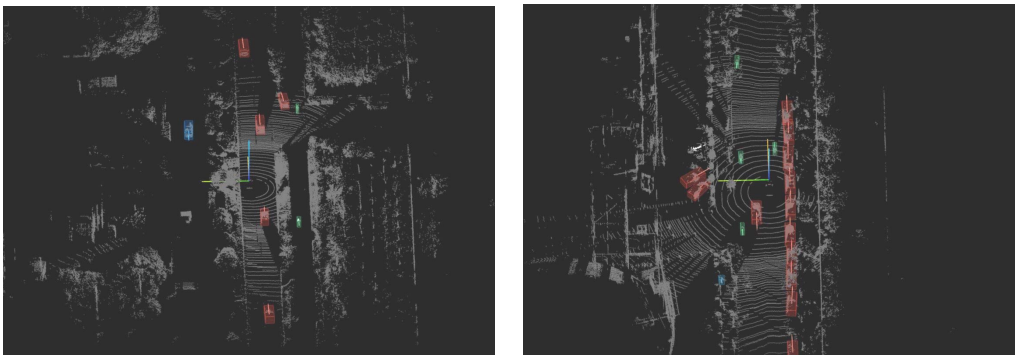


图 3 检测结果可视化

3.4 消融实验

对不同网络配置下的检测性能进行消融实验, 结果如表 2 所示. 可以看出: (1) 第 1 列中, 简单的二值表达的检测性能全面好于使用复杂的 PointNet[14]编码, 可能是后续的主干网络更多地提取的是体素在 3D 空间的几何信息, 每个体素内的点的分布信息相对太过细节, 网络无法在多层卷积后抽取出有意义的信息; 进一步, 将二值表达换成反射率表达, 性能上微弱上升, 通过一些定性的示例发现, 反射率特性对于区分场景中的高反交通牌和其他目标效果非常显著, 能够改善特定场景下的误检漏检. (2) 第 2 列中, 引入 FPN 后, 行人和两轮车这类尺度较小的目标的检测性能显著提升. (3) 第 3 列中, 引入残差设计后, 同样层数的神经网络的检测性能在各个类别上都有提升.

表 2　不同网络配置下消融实验结果　　%

VFE	FPN	残差	汽车	卡车	行人	两轮车
PointNet[14]	×	×	90.60	87.27	82.03	84.30
二值	×	×	90.69	87.63	84.63	85.48
反射率	×	×	91.53	87.64	85.45	85.75
反射率	√	×	92.06	87.76	87.75	86.90
反射率	√	√	92.54	88.19	89.92	88.46

3.5 运行时间

本文网络在 NVIDIA 2080Ti GPU 上的推理时间为 43 ms, 完全满足实际自动驾驶系统实时运行的需求.

3.6 方法局限

本文的体素化方法虽然计算效率较高, 但在 2D 卷积运算中依然未充分地提取点云的高程特征, 后续将会探索如何充分地利用点云的稀疏性来节省算力, 以及如何充分提取点云的高程特征来提升检测性能.

4 结　语

本文提出一种用于 LiDAR 点云目标检测的单阶段无锚框优化网络, 对点云的体素化方法、主干网络的结构和 FPN 的引入都进行了优化实验, 最终给出一个检测性能优秀、跨平台友好, 能实时运行在自动驾驶车端的网络结构. 在公开数据集 KITTI 上, 证明了本文网络能够取得领先的检测性能.

参考文献(References):

[1] Leonard J, How J, Teller S, *et al*. A perception-driven autonomous urban vehicle[J]. Journal of Field Robotics, 2008, 25(10): 727-774

[2] Himmelsbach M, Müller A, Lüttel T, *et al*. LIDAR-based 3D object perception[C] //Proceedings of 1st International Workshop on Cognition for Technical Systems. 2008: Article No.1

[3] Yang B, Luo W J, Urtasun R. PIXOR: real-time 3D object detection from point clouds[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 7652-7660

[4] Ku J, Mozifian M, Lee J, *et al*. Joint 3D proposal generation and object detection from view aggregation[C] //Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Los Alamitos: IEEE Computer Society Press, 2018: 1-8

[5] Chen X Z, Ma H M, Wan J, *et al*. Multi-view 3D object detection network for autonomous driving[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6526-6534

[6] Qi C R, Liu W, Wu C X, *et al*. Frustum PointNets for 3D object detection from RGB-D data[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 918-927

[7] Shin K, Kwon Y P, Tomizuka M. RoarNet: a robust 3D object detection based on RegiOn approximation refinement[C] //Proceedings of the IEEE Intelligent Vehicles Symposium (IV). Los Alamitos: IEEE Computer Society Press, 2019: 2510-2515

[8] Ren S Q, He K M, Girshick R, *et al*. Faster R-CNN: towards real-time object detection with region proposal networks[C] //Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2015: 91-99

[9] Tian Z, Shen C H, Chen H, *et al*. FCOS: fully convolutional one-stage object detection[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 9626-9635

[10] Lin T Y, Dollár P, Girshick R, *et al*. Feature pyramid networks for object detection[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 936-944

[11] Zhang S F, Chi C, Yao Y Q, *et al*. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2020: 9756-9765

[12] Zhou Y, Tuzel O. VoxelNet: End-to-end learning for point cloud based 3D object detection[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2018: 4490-4499

[13] Lang A H, Vora S, Caesar H, *et al*. PointPillars: fast encoders for object detection from point clouds[C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2019: 12689-12697

[14] Charles R Q, Su H, Mo K C, *et al*. PointNet: deep learning on point sets for 3D classification and segmentation[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern

Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 77-85

[15] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 770-778

[16] Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2014: 580-587

[17] Girshick R. Fast R-CNN[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2015: 1440-1448

[18] Lin T Y, Goyal P, Girshick R, *et al*. Focal loss for dense object detection[C] //Proceedings of the IEEE International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2017: 2999-3007

[19] Duan K W, Bai S, Xie L X, *et al*. CenterNet: keypoint triplets for object detection[C] //Proceedings of the IEEE/CVF International Conference on Computer Vision. Los Alamitos: IEEE Computer Society Press, 2019: 6568-6577

[20] Engelcke M, Rao D, Wang D Z, *et al*. Vote3Deep: fast object detection in 3D point clouds using efficient convolutional neural networks[C] //Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Los Alamitos: IEEE Computer Society Press, 2017: 1355-1361

[21] Li B. 3D fully convolutional network for vehicle detection in point cloud[C] //Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Los Alamitos: IEEE Computer Society Press, 2017: 1513-1518

[22] Yan Y, Mao Y X, Li B. SECOND: sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): Article No.3337

[23] Ge R Z, Ding Z Z, Hu Y H, *et al*. AFDet: anchor free one stage 3D object detection[OL]. [2023-06-16]. https://arxiv.org/abs/2006.12671

[24] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift[C] //Proceedings of the 32nd International Conference on International Conference on Machine Learning. PMLR, 2015: 448-456

[25] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C] //Proceedings of the 27th International Conference on International Conference on Machine Learning. 2010: 807-814

[26] Mousavian A, Anguelov D, Flynn J, *et al*. 3D bounding box estimation using deep learning and geometry[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 5632-5640

[27] Zhou X Y, Wang D Q, Krähenbühl P. Objects as points[OL]. [2023-06-16]. http://export.arxiv.org/abs/1904.07850

[28] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2012: 3354-3361

[29] Chen X Z, Kundu K, Zhu Y K, *et al*. 3D object proposals for accurate object class detection[C] //Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2015: 424-432

[30] Loshchilov I, Hutter F. Decoupled weight decay regularization[OL]. [2023-06-16]. https://openreview.net/forum?id=Bkg6RiCqY7

[31] Smith L N, Topin N. Super-convergence: very fast training of neural networks using large learning rates[C] //Proceedings of the SPIE 11006, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications. Bellingham: Society of Photo-Optical Instrumentation Engineers, 2019: 369-386