激光与光电子学进展 先进成像

基于改进PointPillars的激光雷达三维目标检测

田枫，刘超，刘芳*，姜文文，徐昕，赵玲
东北石油大学计算机与信息技术学院，黑龙江 大庆 163318

摘要 针对目前基于点云的三维目标检测算法中小目标检测效果差的问题，提出了基于改进PointPillars模型的三维目标检测方法。首先，改进了PointPillars模型中的pillar特征网络，提出了一个新的pillar编码模块，在编码网络中引入了平均池化和注意力池化，充分考虑了每个pillar模块的局部详细几何信息，提高了每个pillar模块的特征表示能力，从而提升了模型的小目标检测性能。其次，基于ConvNeXt改进了骨干网络中的二维卷积下采样模块，使模型在网络特征提取阶段能够提取丰富的上下文语义信息和全局特征，从而增强了算法的特征提取能力。在公开数据集KITTI上进行验证，实验结果表明，所提方法具有更高的检测精度，相较于原网络，改进后的算法的平均检测精度提升了3.63个百分点，证明了该方法的有效性。

关键词 三维目标检测；PointPillars；小目标检测；注意力池化；ConvNeXt
中图分类号 TP391 **文献标志码** A **DOI**：10.3788/LOP231493

Laser Radar 3D Target Detection Based on Improved PointPillars

Tian Feng, Liu Chao, Liu Fang*, Jiang Wenwen, Xu Xin, Zhao Ling
School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, *Heilongjiang, China*

Abstract A 3D object detection method based on improved PointPillars model is proposed to address the problem of poor detection performance of small objects in current point cloud based 3D object detection algorithms. First, the pillar feature network in the PointPillars model is improved, and a new pillar encoding module is proposed. Average pooling and attention pooling are introduced into the encoding network, fully considering the local detailed geometric information of each pillar module, which improve the feature representation ability of each pillar module and further improve the detection performance of the model on small targets. Second, based on ConvNeXt, the 2D convolution downsampling module in the backbone network is improved to enable the model extract rich context semantic information and global features during feature extraction process, thus enhancing the feature extraction ability of the algorithm. The experimental results on the public dataset KITTI show that the proposed method has higher detection accuracy. Compared with the original network, the improved algorithm has an average detection accuracy improvement of 3.63 percentage points, proving the effectiveness of the method.

Key words 3D object detection; PointPillars; small target detection; attention pooling; ConvNeXt

1 引 言

随着自动驾驶、智能制造和机器人导航等领域的发展，三维目标检测的研究越来越受到人们的关注。三维目标检测从三维点云数据中检测出物体的类型、位置和姿态等信息，是实现自动驾驶[1-3]、智能机器人[4-5]和虚拟/增强现实[6-7]等任务的关键技术之一。近年来，随着深度学习技术的突破性进展，三维目标检测取得了长足进步，大量基于激光雷达的三维目标检测算法被相继提出。目前，基于点云的三维目标检测算法已经成为主流的研究方向之一，一些方法尝试将不规则点云转换为规则的体素，然后利用卷积神经网络(CNN)学习其相应的特征。VoxelNet模型[8]是一项开创性的工作，它对输入点云进行密集体素化，然后利用体素特征提取器(VFE)和3D CNN来学习其几何表示。其缺点是3D卷积的巨大计算负担造成推理速

收稿日期：2023-06-08；**修回日期**：2023-07-03；**录用日期**：2023-07-24；**网络首发日期**：2023-08-04
基金项目：黑龙江省自然科学基金(LH2021F004)
通信作者：*lfliufang1983@126.com

度相对较慢。为了节省内存成本，SECOND（sparsely embedded convolutional detection）模型[9]使用3D稀疏卷积来加速训练和推理。稀疏卷积仅对非空体素进行操作，这大大提高了计算和存储效率，但是仍然无法摆脱3D卷积层的巨大计算负担。PointPillars模型[10]将点划分为柱，对每个柱内的点云进行特征提取，形成伪BEV（battery electric vehicles）2D图像，然后使用2D图像目标检测方法进行目标提取。尽管该算法的检测速度得到了大幅度的提升，但检测精度会受到很多条件的限制。这些算法在一定程度上提高了模型的检测精度或速度，但模型的小目标检测性能有待提升。

虽然现有的基于点云的三维目标检测算法已经取得了一定的成果，但仍然存在许多问题。其中最主要的问题是小目标检测效果差，这是因为点云数据本身的稀疏性和复杂性会造成小目标的细节难以捕捉。

为了改善小目标检测的效果，对PointPillars模型中的pillar特征网络进行了改进，并提出了一个新的pillar编码模块。该编码模块充分考虑了局部详细几何信息的重要性，以提升每个pillar模块的特征表示能力。此外，针对PointPillars模型中骨干网络在特征提取方面的不足，采用了基于ConvNeXt的改进策略，对二维卷积下采样模块进行了优化，从而使模型在网络特征提取阶段能够捕捉更丰富的上下文语义信息和全局特征，进一步增强了算法的特征提取能力。

2　PointPillars算法

PointPillars通过一个微小的PointNet在X-Y平面上投影原始点云，产生一个稀疏的2D伪图像。它使用基于2D CNN的自上而下的网络来处理具有步长为1×、2×、4×卷积块的伪图像，然后将多尺度特征连接起来用于检测头。简单来说，PointPillars的处理思路是将三维信息映射到二维，在2D伪图像上进行目标检测。PointPillars的网络结构主要包括pillar特征网络、骨干网络、SSD（single shot multibox detector）检测头。

pillar特征网络对每个pillar中的点进行随机采样。随机采样可能会丢掉重要的特征信息，从而影响模型的检测性能。在最大池化过程中随机采样会导致细粒度信息，如局部详细几何信息和位置信息等的丢失，从而影响小目标的检测精度。通过改进PointPillars模型中的pillar特征网络，考虑每个柱的局部详细几何信息，使得点云局部几何结构和精准位置信息在该过程中得到保留，这有利于提高模型的小目标检测性能。

PointPillars模型的骨干网络使用了传统的2D CNN进行特征提取，这会忽略一些重要的局部特征和上下文信息，从而导致模型检测精度下降。基于ConvNeXt改进骨干网络中的二维卷积下采样模块，可以使模型在网络特征提取阶段提取丰富的上下文语义信息和全局特征，从而增强算法的特征提取能力。

3　改进PointPillars算法

PointPillars算法的优势是速度快，它仅使用2D CNN提取特征，大大提高了算法的运行效率。但其缺点是检测精度不高，因为PonitPillars算法在随机采样过程中可能会丢失一些有用的点，在最大池化过程中丢失细粒度信息。另外骨干网络的特征提取能力不够充分，从而影响了模型的检测精度。为了提高模型的检测性能，首先，改进了PointPillars模型中的pillar编码网络；其次，基于ConvNeXt改进了骨干网络中的二维卷积下采样模块。网络结构如图1所示，其中：C为特征维度；N为每个pillar中的点云个数；P为pillar个

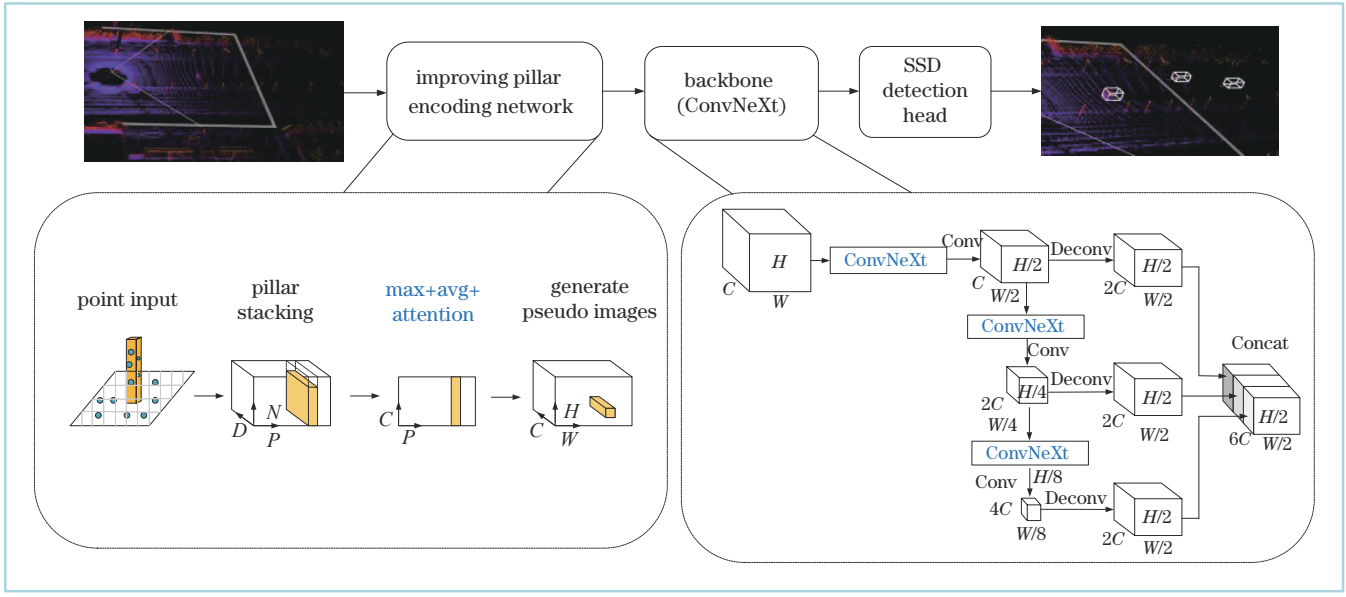

图1　网络结构图

Fig. 1　Network structure

数；H、W 分别为图像的高度和宽度；max+avg+attention 表示最大池化＋平均池化＋注意力池化操作。

3.1　改进 pillar 编码网络

在 PointPillars 模型的 pillar 特征网络中，仅使用最大池化来获取每个 pillar 中的最大点特征。然而，最大池化操作会导致细粒度信息的丢失，而这些信息对于基于 pillar 的目标检测模型，尤其是对于小目标的检测非常关键。原 PointPillars 模型的 pillar 特征网络结构如图 2 所示，其中，D 也是特征维度，与特征维度 C 不同，D 表示将点云由原来的 4 维再编码为 9 维，而 C 表示通过 MLP 将点云由 9 维映射到和原 PointPillars 模型相同的 64 维。所提算法以 PointPillars 为基础网络，改进了 PointPillars 模型中的 pillar 特征网络，在编码网络中引入了平均池化和注意力池化，将最大池化、平均池化和注意力池化相结合，考虑了每个 pillar 模块的局部详细几何信息，使得点云局部几何结构和精准位置信息在该过程中得到保留，有利于提升模型在伪图像生成过程中检测小目标（如行人等）的性能。如图 3 所示，改进的 pillar 编码网络由 4 个部分组成：1）点云编码模块；2）注意力池化编码模块；3）最大池化编码模块；4）平均池化编码模块。

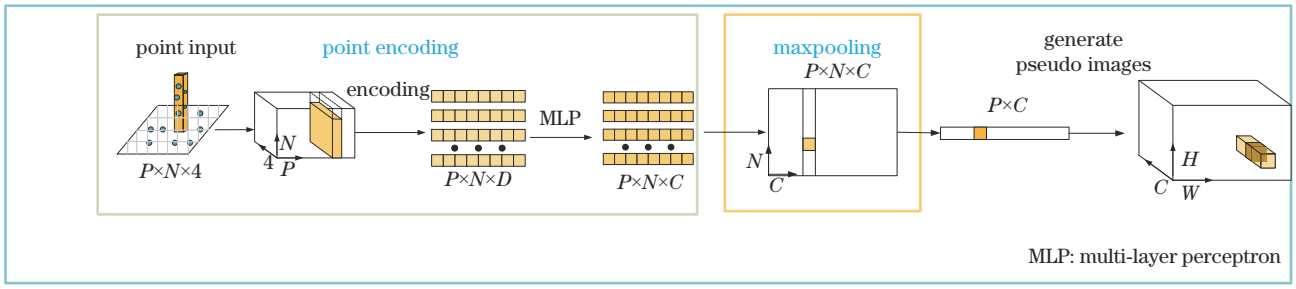

图 2　原 PointPillars 模型的 pillar 特征网络结构

Fig. 2　Structure of the pillar feature network of the original PointPillars model

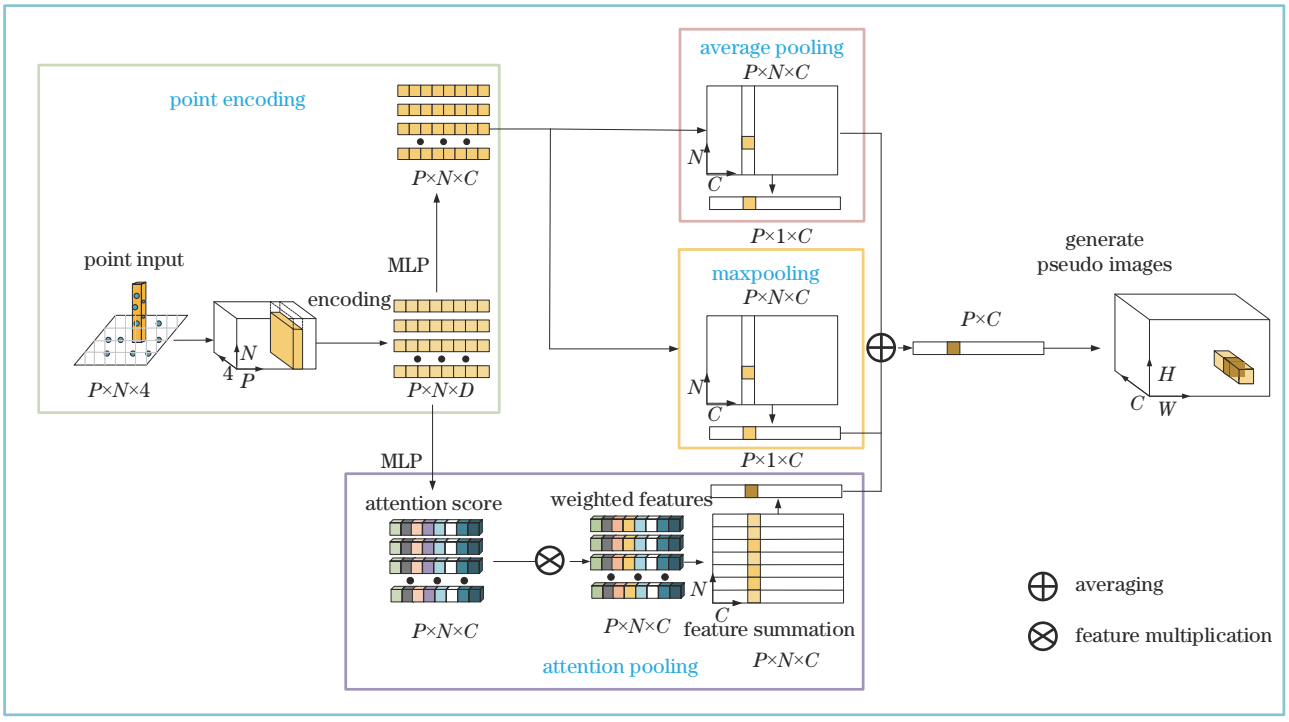

图 3　改进的 pillar 编码网络结构

Fig. 3　Structure of improved pillar encoding network

该模块的输入是包含点云坐标、反射强度等信息的原始点云。在点云编码模块中，首先利用柱中心和点云范围的信息对原始点进行增强，然后通过多层感知机（MLP）将增强的点特征映射为高维特征；在最大池化编码模块中，对每个 pillar 中的点特征进行最大池化操作，得到最大池化特征；在平均池化编码模块中，对每个 pillar 中的点特征进行平均池化操作，得到平均池化特征；在注意力池化编码模块中，对点云特征进行加权求和运算，得到注意力池化特征。然后对最大池化、平均池化和注意力池化特征求平均来获得当前的 pillar 特征。最后，把所有的 pillar 特征按照原始柱的位置组合堆积起来，形成尺寸为（C，H，W）的伪图像。

3.1.1 点云编码模块

沿用PointPillars的体素化方式，只在X-Y平面利用尺寸为$W\times H$的栅格进行离散化，将3D空间划分为尺寸相同的pillar网格。设$P=\{p_i=[x_i,y_i,z_i,r_i]\in R^{N\times 4}\}$是由$N$个点组成的非空柱，其中：$p_i$为柱中的第$i$个点，每个$p_i$的特征维度$D=4$；$[x_i,y_i,z_i]$为pillar中的原始点坐标；$r_i$为pillar中原始点的反射强度；$R$为每个柱中的点集合。

首先，在体素化为pillar的过程中对点云进行再编码，将每个柱中点的维度增加为$p_i=\{[x_i,y_i,z_i,r_i,x_{c,i},y_{c,i},z_{c,i},x_{r,i},y_{r,i}]\in R^{N\times 9}\}$，其中，$[x_{c,i},y_{c,i},z_{c,i}]$为$p_i$所在pillar的中心坐标；$[x_{r,i},y_{r,i}]$为$p_i$相对于当前pillar中心的$X$和$Y$两个维度的偏移量[10]，此时每个$p_i$的特征维度$D=9$。

在每个pillar中取消了对点的随机采样策略，因为这种操作可能会丢失有用的点，而使生成的伪图像丢失有用的信息。最后，通过MLP层将P内的张量化逐点特征p_i映射到和原PointPillars模型相同的64维特征，将升维后的逐点特征命名为$p_{e,i}$，此时$p_{e,i}$的特征维度$C=64$。

3.1.2 注意力池化编码模块

原PointPillars模型的最大池化操作仅保留了每个pillar中最大值的点特征，舍弃了大量的局部点特征信息，然而，丰富的局部特征信息同样有利于小目标检测的检测效果。为此，引入了注意力池化编码模块以保留重要的局部特征。首先，使用由MLP组成的函数$\mathrm{Ma}(\cdot)$来预测pillar中所有点的注意力得分，表示为

$$s_i=\mathrm{Ma}\left(p_{e,i},W_{\mathrm{Ma}}\right),\tag{1}$$

式中：$s_i\in R^{N\times C}$为注意力得分；W_{Ma}为MLP的可学习权重。由于每个点的特征维度为C，因此每个点的注意力得分的特征维度也为C。然后，将每个点在每个维度上的注意力分数和其对应特征相乘，再将所有维度的相乘结果累加得到加权特征。最后将所有点的加权特征累加得到加权求和特征，详细过程如图3注意力池化模块所示。加权求和特征$f_{\mathrm{a}}\in R^C$是最终的pillar注意力池化特征，表示为

$$f_{\mathrm{a}}=\sum_{i=1}^{i=N}\left(\sum_{i=1}^{i=C}s_i\cdot p_{e,i}\right)。\tag{2}$$

3.1.3 最大池化编码模块

pillar编码网络使用最大池化编码模块来提取pillar内部的局部特征[10]，从而获取pillar内所有点特征中最显著的特征，表示为

$$f_{\mathrm{m}}=\max\left(p_{e,i}\right),\tag{3}$$

式中：$\max(\cdot)$为pillar内点特征的最大池化操作；f_{m}为每个pillar最大池化后的结果特征。

3.1.4 平均池化编码模块

使用平均池化编码模块来提取pillar内部的平均特征，从而获取pillar内所有点特征的平均值，表示为

$$f_{\mathrm{v}}=\mathrm{avg}\left(p_{e,i}\right),\tag{4}$$

式中：$\mathrm{avg}(\cdot)$为pillar内点特征的平均池化操作；f_{v}为每个pillar平均池化后的结果特征。

3.1.5 特征融合

通过对学习到的pillar最大池化特征、平均池化特征、注意力池化特征求均值得到最终特征，表示为

$$f=\frac{f_{\mathrm{m}}+f_{\mathrm{v}}+f_{\mathrm{a}}}{3},\tag{5}$$

式中，$f\in R^C$为最终的pillar特征，其中，C为特征维度。最终的pillar特征包括pillar内的全局感知信息和局部感知信息。通过注意力池化和平均池化得到的是全局感知信息；通过最大池化得到的是局部感知信息。最大池化操作保留了每个pillar中的最大点特性，而注意力池化和平均池化操作保留了局部细粒度信息。通过结合这三个特征，可以有效地保留pillar中更丰富的点特征信息，以增强pillar表示。最后，将最终的柱状特征按照原始柱的位置组合堆积起来，形成尺寸为(C,H,M)的伪图像。尽管这是一种简单的方法，但改进的pillar编码网络可以显著提高模型的小目标检测效果。

3.2 基于ConvNeXt改进骨干网络

ConvNeXt[11]模块是目前特征提取能力较强的特征提取器。将ConvNeXt模块添加到PointPillars模型骨干网络的下采样模块中，可以提高骨干网络的特征提取能力，获得更丰富的特征信息，从而提高模型的检测精度。

PointPillars的骨干网络包括下采样降分辨率提取高维特征和上采样到相同尺度进行特征融合两个过程。ConvNeXt模块主要用于特征提取，它通过自上而下的下采样以及相应的上采样在多个尺度上聚集特征信息。

3.2.1 ConvNeXt模块

ConvNeXt是一个对原始ResNet[12]结构进行改进的模型，它通过借鉴Swin Transformer的设计思路，进一步提升了模型的表现力和效率[13]。ConvNeXt模块没有使用传统的卷积运算，而是使用深度可分离卷积。与传统卷积相比，深度可分离卷积具有参数量和工作量较小的优点。另外，MobileNetv2模型[14]中使用了一种中间大、末端小的反向瓶颈层结构，ConvNeXt模块使用该反向瓶颈层结构能够更好地捕捉特征之间的相关性，有效避免信息丢失，提高模型的特征表达能力。

ConvNeXt通过使用深度可分离卷积、LN（LayerNorm）、GeLU激活函数、更大的卷积核来替代原始下采样网络的特征提取，以获得更丰富的特征信息。所设计的ConvNeXt模块的结构如图4所示，其中：DConv为深度可分离卷积，卷积核大小为7×7；64-f表示卷积核通道数为64；LN为归一化层；GeLU为激活函数。

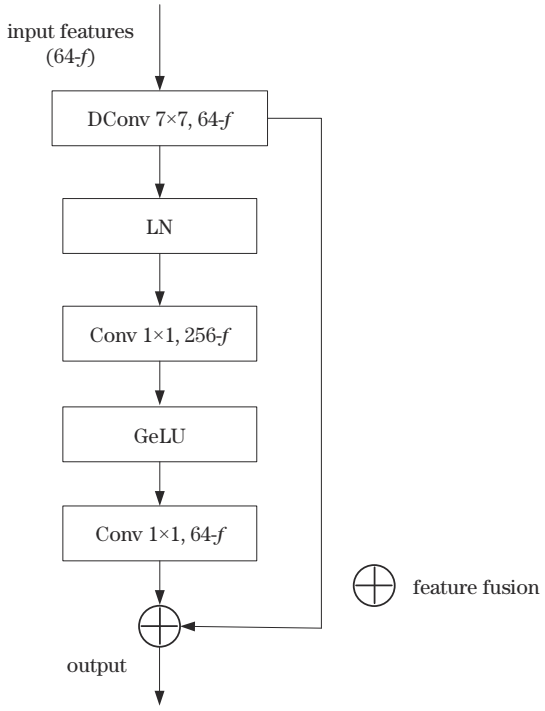

图 4　ConvNeXt模块结构

Fig. 4　Structure of ConvNeXt module

3.2.2　基于ConvNeXt模块的骨干网络

PointPillars模型的骨干网络使用了简单的CNN进行特征提取，特征提取不够充分，忽略了一些重要的局部特征和上下文信息，从而导致模型的检测精度下降。

将ConvNeXt模块和原始下采样模块的Conv串联起来重新构建骨干网络。ConvNeXt模块中只有点云特征的提取和融合，并没有进行最终的特征映射，因此需要再结合一个Conv来将特征映射到目标检测任务所需的特征维度。同时，这样做还能增加网络的深度和非线性表达能力，提高网络的特征提取能力。所设计的基于ConvNeXt模块的骨干网络结构如图5所示。

基于ConvNeXt改进骨干网络中的二维卷积下采样模块，使得网络在特征提取阶段能够提取丰富的上下文语义信息和全局特征，从而增强算法的特征提取能力。

3.3　目标朝向损失函数

在原始的PointPillars方法中，目标朝向损失函数为Softmax分类损失。然而，Softmax损失对于角度回归问题不是最优选择，因为它并不适用于多个类别（即多个离散角度）的情况。

为了改进目标朝向损失函数，使用SmoothL1损失函数，将目标朝向的估计看作是一个回归问题，从而获得更好的精度和鲁棒性[15]。另外引入余弦相似度来计算预测角度与真实角度之间的相似度，这样可以更好地处理角度的周期性。余弦相似度可以表示为

$$L_\theta = \cos\theta_i \cdot \cos\hat{\theta}_i + \sin\theta_i \cdot \sin\hat{\theta}_i, \tag{6}$$

式中：θ_i为真实角度；$\hat{\theta}_i$为预测角度。

综合以上两点改进，本算法的目标朝向损失函数定义为

$$L_{\mathrm{Dir}} = \frac{1}{N_{\mathrm{pos}}}\sum_{i=1}^{N_{\mathrm{pos}}}\mathrm{SmoothL\,1}(L_\theta - 1), \tag{7}$$

式中：N_{pos}为正样本数量；SmoothL 1(·)为SmoothL1损失函数。

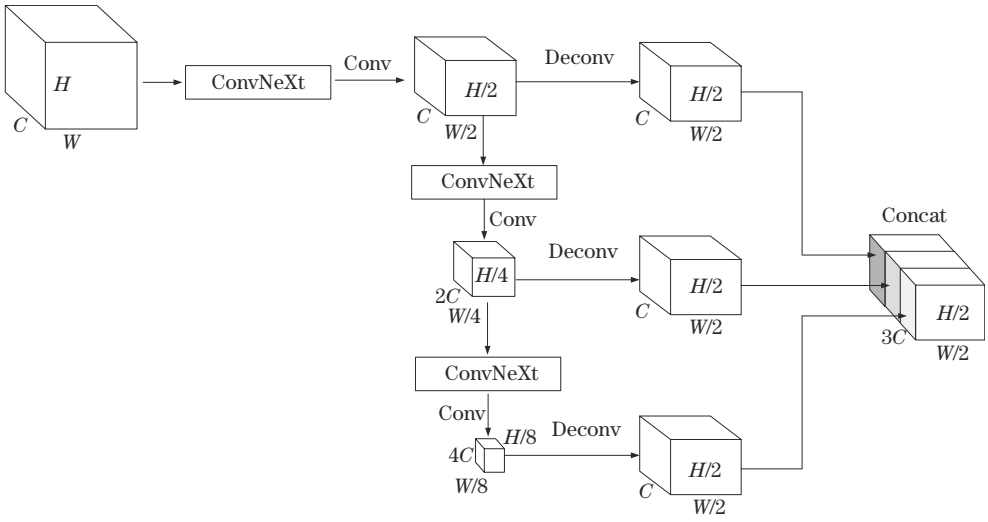

图 5　基于ConvNeXt模块的骨干网络结构

Fig. 5　Structure of backbone network based on ConvNeXt module

4　实　　验

4.1　数据集

使用大型公开数据集KITTI进行实验评估。KITTI数据集包含自动驾驶场景中的7481个训练样本和7518个测试样本[16]。遵循PointPillars算法将训练数据划分为3712帧图像的训练集和3769帧图像的验证集。使用训练集进行训练，使用验证集进行实验研究。KITTI数据集共包含三种分类：汽车、骑行者和行人。每个分类都有三个难度级别：容易、中等和困

难，难度级别取决于3D对象的大小、遮挡级别和截断级别等因素。

4.2 实验设置

实验基于OpenPCDet框架实现，利用这一框架可以对KITTI数据集和改进PointPillars算法进行实验。硬件环境为：NVIDIA Corporation GP102GL [Tesla P40] 24 GB显存图形处理器(GPU)、Intel Xeon Silver 4214中央处理器(CPU)、4.2 TB硬盘。软件环境为：Ubuntu 20.04 LTS、Python 3.7、Cuda 11.3、PyTorch 1.10。使用PyTorch框架实现网络结构，使用GPU进行训练和测试，使用Adam优化器实现端到端的训练，批大小为16，设置权重衰减值为0.01、动量值为0.9、学习率衰减值为0.1、最大迭代次数为120。

在训练过程中，随机抽取一些地面真值框，并将它们放置在样本中，以增加点云中地面真值框的数量，并模拟不同环境中的对象。根据[$-\pi/4,\pi/4$]范围内均匀分布的沿Z轴旋转的真实3D边界框内的点，可以确定点云的方向变化。类似地沿X轴或Y轴随机翻转3D框内的点云，在[0.95,1.05]范围内随机缩放全局点云。

在实验研究中，设置每个pillar的X-Y尺寸为(0.16, 0.16)，最大柱数(P)为12000，但不采用PointPillars算法中的随机采样策略来保持每个pillar中的点的数量相同。

4.3 实验结果及分析

使用所提算法在KITTI数据集上进行了实验，所有结果通过平均精度(AP)进行评估，具体评估方法参考KITTI官方测试服务器上的40个召回位置。汽车(car)的交并比(IoU)阈值设为0.7，行人(pedestrian)和骑行者(cyclist)的IoU阈值设为0.5。

为了评估改进PointPillars算法的性能，在KITTI数据集上对所提算法与基于LiDAR的3D目标检测相关算法进行测试对比。对比算法包括VoxelNet、SECOND、PointPillars、3D-GIoU[17]、TANet[18]、PointRCNN[19]、Point-GNN[20]和Part-A^2[21]。表1、表2、表3分别展示了在KITTI测试集汽车、行人和骑行者类别下，所提算法与其他算法的mAP的对比(mAP是类别在三种检测级别下的平均精度均值)。

表1 汽车类别下不同算法的mAP对比

Table 1 Comparison of mAP for different algorithms under car category

Model	AP /%			mAP /%
	Easy	Moderate	Hard	
VoxelNet	87.93	75.37	73.21	78.84
SECOND	88.61	78.62	77.22	81.48
PointPillars	87.50	77.01	74.77	79.76
3D-GIoU	87.83	77.91	78.84	82.60
TANet	88.17	77.75	75.31	80.41
PointRCNN	89.01	78.77	78.10	81.96
Point-GNN	89.33	79.47	78.29	82.36
Part-A^2	**89.56**	79.41	78.84	82.60
Ours	89.28	**79.56**	**79.62**	**82.82**

表2 行人类别下不同算法的mAP对比

Table 2 Comparison of mAP for different algorithms under the pedistrian category

Model	AP /%			mAP /%
	Easy	Moderate	Hard	
VoxelNet	67.81	63.52	58.87	63.40
SECOND	56.00	50.02	43.64	49.89
PointPillars	66.73	61.06	56.50	61.43
3D-GIoU	67.23	59.58	52.69	59.83
TANet	70.80	63.45	58.22	64.16
PointRCNN	62.69	55.36	51.60	56.55
Point-GNN	61.92	53.77	50.14	55.28
Part-A^2	65.69	60.05	55.45	60.40
Ours	**71.33**	**63.75**	**58.63**	**64.57**

表3 骑行者类别下不同算法的mAP对比

Table 3 Comparison of mAP for different algorithms under the cyclist category

Model	AP /%			mAP /%
	Easy	Moderate	Hard	
VoxelNet	77.69	58.72	51.63	62.68
SECOND	80.97	63.43	56.67	67.02
PointPillars	83.65	63.40	59.71	68.92
3D-GIoU	83.32	64.69	63.51	70.51
TANet	85.21	65.29	61.57	70.69
PointRCNN	84.48	65.37	59.83	69.89
Point-GNN	86.60	67.48	62.58	72.22
Part-A^2	85.50	**68.90**	**64.53**	72.98
Ours	**87.88**	68.75	64.26	**73.63**

由表1～3可知，所提改进的PointPillars算法相较于原始的PointPillars算法在检测KITTI数据集汽车、行人、骑行者类型时精度提升明显，其中：检测汽车类型时mAP提升了3.06个百分点；检测行人类型时mAP提升了3.14个百分点；检测骑行者类型时mAP提升了4.71个百分点。同时，在检测汽车、行人和骑行者类型时，所提算法相较于其他算法也取得了较优的检测精度。说明对PointPillars算法的pillar编码网络进行的改进，即取消每个pillar的点云采样策略，将最大池化提取的特征、平均池化提取的特征、注意力池化提取的特征相结合，有效地保留更丰富的细粒度信息，可以有效地提高模型的检测精度。同时，将所设计的ConvNeXt模块加入到PointPillars骨干网络中以获得更丰富的特征信息，大大提升了骨干网络的特征提取能力。

4.4 可视化结果与分析

在KITTI数据集上使用所提算法的检测效果如

图 6 所示，图 6 各分图中：上半部分为点云场景下的 3D 目标检测效果图；下半部分为真实场景下对应的相机图像。在 3D 目标检测结果中，绿色框为汽车的 3D 检测框，蓝色框为行人的 3D 检测框，黄色框为骑行者的 3D 检测框。实验中将点云场景下 3D 检测框中的点云染成了红色以便于观察可视化效果。

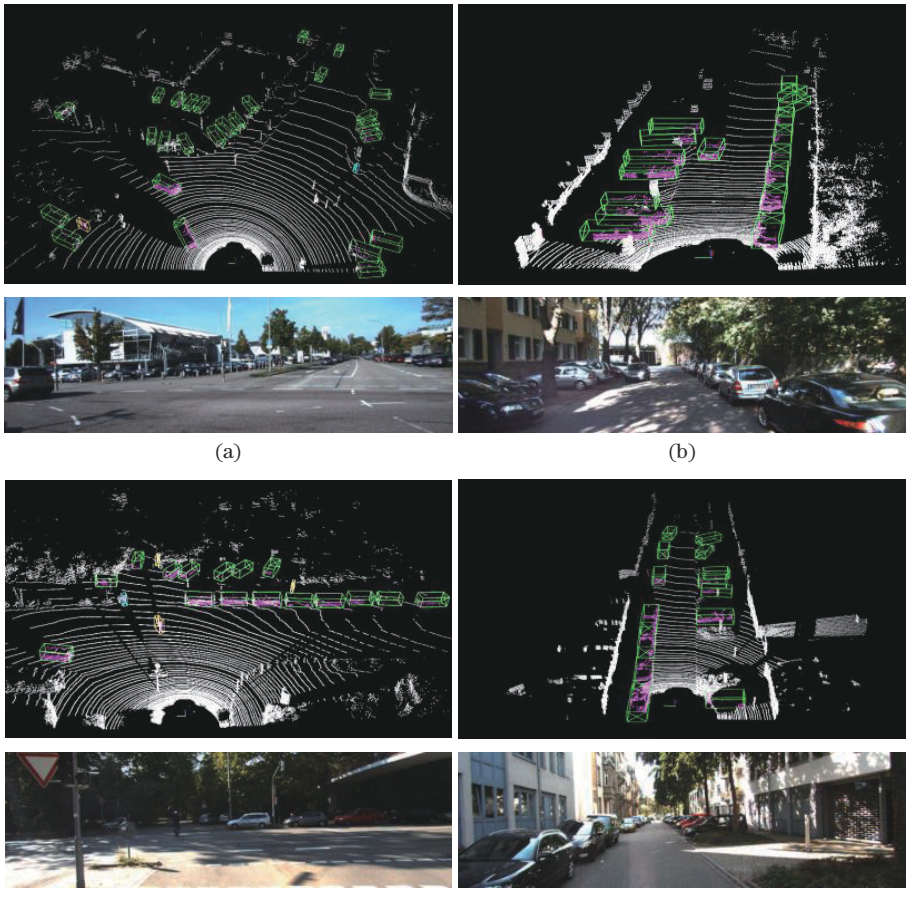

图 6　所提算法对不同场景的 3D 目标检测效果图和二维图像。(a)场景一；(b)场景二；(c)场景三；(d)场景四

Fig. 6　3D object detection renderings and 2D images of the proposed algorithm on different scenes. (a) Scene one; (b) scene two; (c) scene three; (d) scene four

从图 6(a)、(c)中可以看出，对于目标种类多、目标较为密集的场景，所提算法依旧可以进行较为准确的检测；从图 6(b)、(d)中可以看出，对于含有较为稀疏的远距离小目标的场景，所提算法也具有较好的检测结果。所提算法也存在着一定的误判情况[如，图 6(a)中，路边的小树被误检成了行人]，但避免了 PointPillars 算法中小目标误检和漏检较多的情况，因此检测精度有所提升。

所提算法和 PointPillars 算法对 KITTI 数据集的复杂场景和远距离场景的检测效果对比如图 7、图 8 所示，图 7、8 的各分图中：上半部分为点云场景下所提算法和 PointPillars 算法的 3D 目标检测效果图；下半部分为真实场景下对应的相机图像。

从图 7 中可以看出，PointPillars 算法在复杂场景小目标检测中存在较多的误检情况，用红色边框标识，其中：路边的栏杆、树木枝干、部分矮小的路灯等目标被错误地检测成了行人；连续且距离较近的栏杆被误识别为自行车；远距离的汽车检测也存在部分误检情况，如，方形的物体被误识别为汽车。所提算法也存在着一定的误检情况，例如场景 1 中路边矮小的树干被识别为行人，场景 2 中远距离的广告牌被识别成自行车。但所提算法避免了 PointPillars 算法检测时小目标误检和漏检较多的情况，取得了较好的可视化效果。

从图 8 中可以看出，PointPillars 算法在远距离小目标检测中存在汽车漏检的情况，如将树木枝干误检为行人；部分汽车也存在误检情况，即将方形的物体误识别为汽车。而所提算法避免了误检和漏检情况。

综上所述，所提算法在复杂场景的小目标检测和远距离检测上具有较好的检测效果，因此整体检测效果有所提升。

4.5　消融实验

为了验证所提算法各模块的有效性，在 KITTI 数据集的三种类型目标检测方面进行了消融实验，以对改进的 pillar 编码网络（平均池化、注意力池化）和 ConvNeXt 模块进行消融研究，分析它们对模型检测性能的影响。

表 4 详细展示了每个所提模块对算法检测性能的

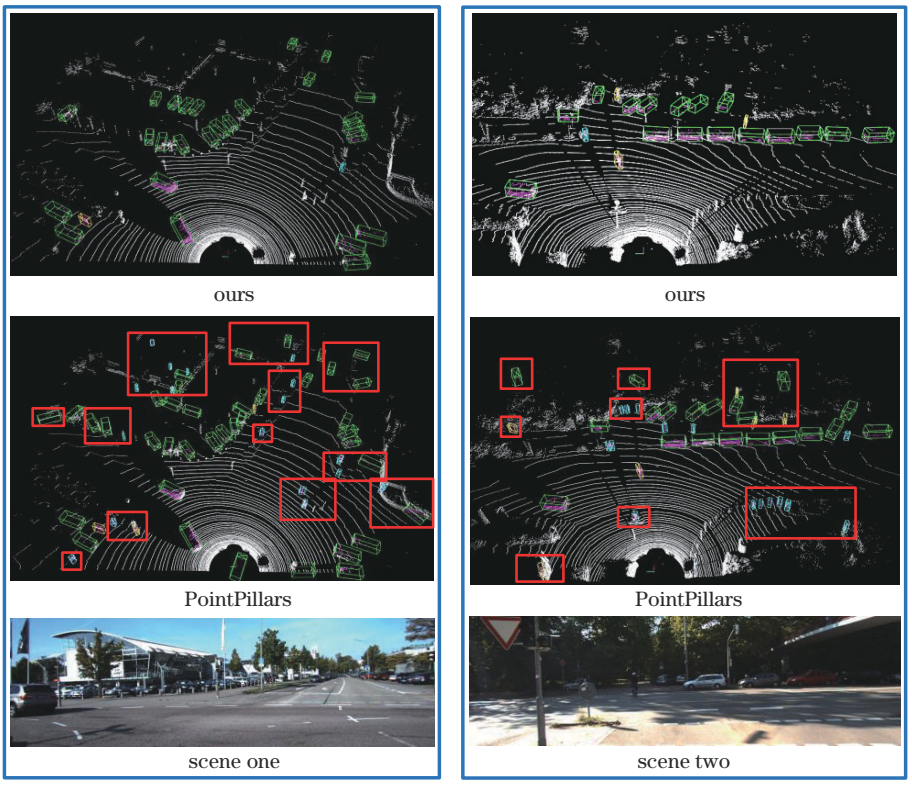

图 7　所提算法与 PointPillars 算法的检测效果对比——复杂场景

Fig. 7　Comparison of detection performance between proposed algorithm and PointPillars algorithm: complex scenes

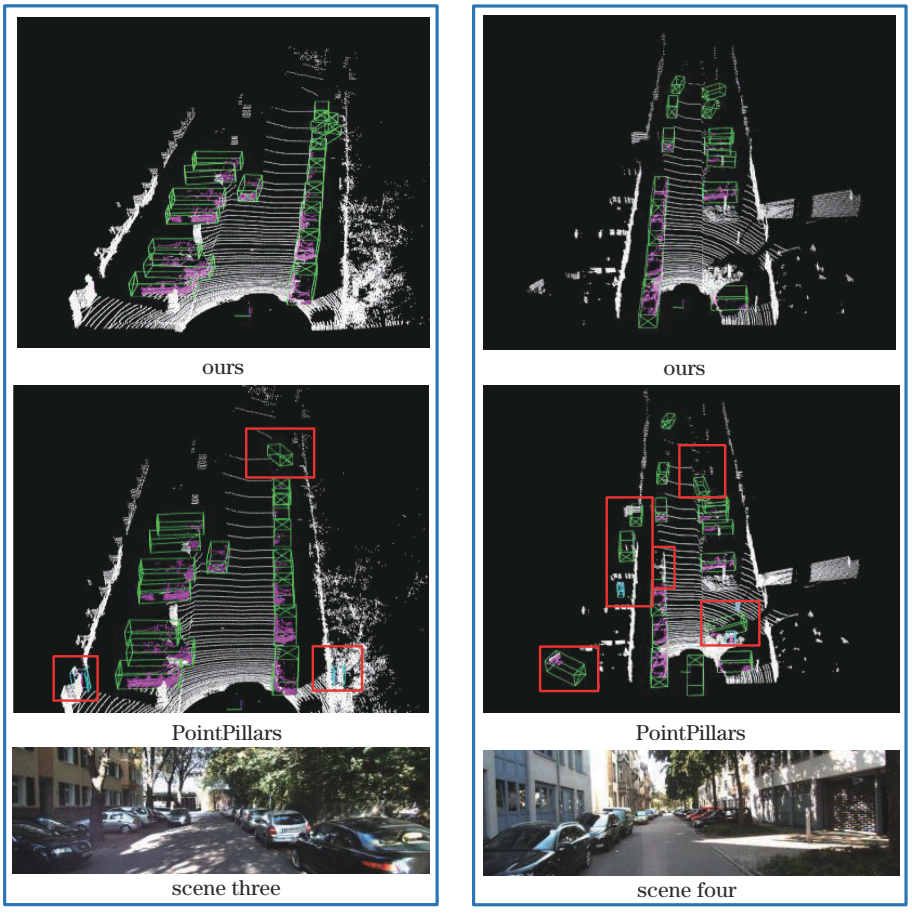

图 8　所提算法与 PointPillars 算法的检测效果对比——远距离场景

Fig. 8　Comparison of detection performance between proposed algorithm and PointPillars algorithm: long-distance scenes

影响，其中，帧每秒（FPS）用来评价模型的检测速度。用三个目标类型在所有检测难度级别下的mAP进行评估。基线模型（baseline）是PointPillars模型，其pillar编码网络使用的是最大池化操作。实验网络1（experiment 1）将最大池化和平均池化操作相结合，使汽车类型检测的AP从79.76%提高到了80.31%；行人类型检测的AP从61.43%提高到了62.03%；骑行者类型检测的AP从68.92%提高到了70.90%；mAP提高了1.04个百分点。实验网络2（experiment 2）将最大池化和注意力池化操作相结合，使汽车类型检测的AP从79.76%提高到了80.54%；行人类型检测的AP从61.43%提高到了62.43%；骑行者类型检测的AP从68.92%提高到了71.47%；mAP提高了1.44个百分点。实验网络3（experiment 3）将最大池化、平均池化、注意力池化操作相结合，使汽车类型检测的AP从79.76%提高到了81.26%；行人类型检测的AP从61.43%提高到了62.78%；骑行者类型检测的AP从68.92%提高到了72.39%；mAP提高了2.10个百分点。实验结果说明改进的pillar编码网络可以提高模型的检测性能。

表4 消融实验结果

Table 4 Results of ablation experiment

Method	Average pooling	Attention pooling	ConvNeXt	AP /%			mAP /%	FPS /(frame/s)
				Car	Pedestrian	Cyclist		
Baseline				79.76	61.43	68.92	70.04	42.2
Experiment 1	√			80.31	62.03	70.90	71.08	38.3
Experiment 2		√		80.54	62.43	71.47	71.48	36.4
Experiment 3	√	√		81.26	62.78	72.39	72.14	33.1
Ours	√	√	√	**82.82**	**64.57**	**73.63**	**73.67**	**26.1**

所提算法将最大池化、平均池化、注意力池化操作相结合，同时在骨干网络下采样过程中添加ConvNeXt模块，使汽车类型检测的AP从79.76%提高到了82.82%；行人类型检测的AP从61.43%提高到了64.57%；骑行者类型检测的AP从68.92%提高到了73.63%；mAP提高了3.63个百分点。说明在骨干网络中加入ConvNeXt模块带来了1.53个百分点的精度提升。综上所述，改进PointPillars的pillar编码模块，同时将ConvNeXt模块添加到骨干网络中，可以提高网络的整体检测性能，尤其是检测小目标的性能。

在pillar编码模块中引入平均池化和注意力池化操作，对模型的检测速度有一定的影响。从表4中可以看出：检测速度由42.2 frame/s降到了33.1 frame/s，检测时间由23.7 ms（1/42.2×1000≈23.7）增加到了30.2 ms（1/33.1×1000≈30.2）；但是mAP提高了2.10个百分点。在pillar编码模块中引入平均池化和注意力池化操作，同时在骨干网络下采样过程中引入ConvNeXt模块，检测速度会进一步受到影响，由33.1 frame/s降到了26.1 frame/s，检测时间由30.2 ms增加到了38.3 ms，但是mAP提高了3.63个百分点。改进PointPillars模型的检测速度虽然受到了影响，但仍然能够满足实时性要求，且检测精度有了明显的提高。

5 结 论

提出了一种基于改进PointPillars的三维目标检测方法，旨在解决基于点云的三维目标检测算法中小目标检测效果差的问题。首先，改进了PointPillars模型中的pillar特征网络，提高了每个pillar特征的表示能力，从而提高了小目标的检测性能；其次，基于ConvNeXt改进了骨干网络中的二维卷积下采样模块，增强了算法的特征提取能力。在KITTI数据集上进行了实验，并将实验结果与其他算法进行了比较。实验结果表明：所提算法将最大池化、平均池化、注意力池化操作相结合，同时在骨干网络下采样过程中添加ConvNeXt模块，使汽车类型检测的AP从79.76%提高到了82.82%；行人类型检测的AP从61.43%提高到了64.57%；骑行者类型检测的AP从68.92%提高到了73.63%；mAP提高了3.63个百分点。证明了所提算法在保证检测速度的同时具有更高的检测精度。

参 考 文 献

[1] Li B Y, Ouyang W L, Sheng L, et al. GS3D: an efficient 3D object detection framework for autonomous driving[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2020: 1019-1028.

[2] Arnold E, Al-Jarrah O Y, Dianati M, et al. A survey on 3D object detection methods for autonomous driving applications[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(10): 3782-3795.

[3] Wang Y, Chao W L, Garg D, et al. Pseudo-LiDAR from visual depth estimation: bridging the gap in 3D object detection for autonomous driving[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2020: 8437-8445.

[4] Kuo H Y, Su H R, Lai S H, et al. 3D object detection and pose estimation from depth image for robotic Bin picking[C]//2014 IEEE International Conference on

Automation Science and Engineering (CASE), August 18-22, 2014, New Taipei, China. New York: IEEE Press, 2014: 1264-1269.

[5] Minemura K, Liau H, Monrroy A, et al. LMNet: real-time multiclass object detection on CPU using 3D LiDAR [C]//2018 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), July 21-23, 2018, Singapore. New York: IEEE Press, 2018: 28-34.

[6] Talukder A, Goldberg S, Matthies L, et al. Real-time detection of moving objects in a dynamic scene from moving robotic vehicles[C]//Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453), October 27-31, 2003, Las Vegas, NV, USA. New York: IEEE Press, 2003: 1308-1313.

[7] Kim D I, Sukhatme G S. Semantic labeling of 3D point clouds with object affordance for robot manipulation[C]// 2014 IEEE International Conference on Robotics and Automation (ICRA), May 31-June 7, 2014, Hong Kong, China. New York: IEEE Press, 2014: 5578-5584.

[8] Zhou Y, Tuzel O. VoxelNet: end-to-end learning for point cloud based 3D object detection[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 4490-4499.

[9] Yan Y, Mao Y X, Li B. SECOND: sparsely embedded convolutional detection[J]. Sensors, 2018, 18(10): 3337.

[10] Lang A H, Vora S, Caesar H, et al. PointPillars: fast encoders for object detection from point clouds[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2020: 12689-12697.

[11] Liu Z, Mao H Z, Wu C Y, et al. A ConvNet for the 2020s[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 18-24, 2022, New Orleans, LA, USA. New York: IEEE Press, 2022: 11966-11976.

[12] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 770-778.

[13] Liu Z, Lin Y T, Cao Y, et al. Swin transformer: hierarchical vision transformer using shifted windows[C]// 2021 IEEE/CVF International Conference on Computer Vision (ICCV), October 10-17, 2021, Montreal, QC, Canada. New York: IEEE Press, 2022: 9992-10002.

[14] Sandler M, Howard A, Zhu M L, et al. MobileNetV2: inverted residuals and linear bottlenecks[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 4510-4520.

[15] 陈德江, 余文俊, 高永彬. 基于改进 PointPillars 的激光雷达三维目标检测[J]. 激光与光电子学进展, 2023, 60(10): 1028012.
Chen D J, Yu W J, Gao Y B. Lidar 3D target detection based on improved PointPillars[J]. Laser & Optoelectronics Progress, 2023, 60(10): 1028012.

[16] 胡杰, 安永鹏, 徐文才, 等. 基于激光点云的深度语义和位置信息融合的三维目标检测[J]. 中国激光, 2023, 50(10): 1010003.
Hu J, An Y P, Xu W C, et al. 3D object detection based on deep semantics and position information fusion of laser point cloud[J]. Chinese Journal of Lasers, 2023, 50(10): 1010003.

[17] Xu J, Ma Y X, He S H, et al. 3D-GIoU: 3D generalized intersection over union for object detection in point cloud [J]. Sensors, 2019, 19(19): 4093.

[18] Liu Z, Zhao X, Huang T T, et al. TANet: robust 3D object detection from point clouds with triple attention[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 11677-11684.

[19] Shi S S, Wang X G, Li H S. PointRCNN: 3D object proposal generation and detection from point cloud[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2020: 770-779.

[20] Shi W J, Rajkumar R. Point-GNN: graph neural network for 3D object detection in a point cloud[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 1708-1716.

[21] Shi S S, Wang Z, Shi J P, 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.