

# 基于点云鸟瞰图的实时车辆目标检测\*

吴庆 彭育辉 黄炜 陈泽辉 姚宇捷

(福州大学,福州 350116)

**【摘要】**针对目前基于三维点云的车辆检测算法实时性差的问题,提出了一种基于点云鸟瞰图的实时车辆目标检测算法。首先,将原始车辆三维点云转换成二维点云RGB特征图;其次,基于YOLOv4-tiny网络添加车辆偏航角度预测分支,实现车辆的准确定位,并通过添加改进空间金字塔池化(SPPF)模块提升网络的目标定位能力;最后,通过在主干网络中引入双注意力机制和优化损失函数提高目标检测精度。试验结果表明:所提出算法的车辆检测平均精度达到96.92%,与YOLOv4-tiny相比提升了2.94个百分点,同时在KITTI鸟瞰图验证集中等难度条件下的平均检测精度达到87.73%,且整体网络检测速度达到100帧/s,可满足实时性要求。

**关键词:**YOLOv4-tiny 点云RGB特征图 角度预测 双注意力机制

**中图分类号:**TP391.4 **文献标识码:**A **DOI:** 10.19620/j.cnki.1000-3703.20221071

## Real-Time Vehicle Target Detection Based on Bird's-Eye View of Point Cloud

Wu Qing, Peng Yuhui, Huang Wei, Chen Zehui, Yao Yujie

(Fuzhou University, Fuzhou 350116)

**【Abstract】**To improve the performance of vehicle detection algorithm based on 3D point clouds, this paper proposed a real-time vehicle target detection algorithm based on the bird's-eye view of point cloud. First, the original 3D vehicle point cloud was converted into the 2D point cloud RGB feature map. Second, the vehicle's yaw angle prediction branch was added to achieve accurate vehicle localization based on YOLOv4-tiny network, the target localization capability of the network was improved by adding an improved Spatial Pyramid Pooling-Fast (SPPF) module. Finally, the target detection precision was improved by introducing a dual-attention mechanism and optimizing the loss function in the backbone network. The test results show that the average vehicle detection precision of the proposed algorithm reaches 96.92%, which is 2.94 percentage points higher than YOLOv4-tiny, the average detection accuracy of the proposed algorithm reaches 87.73% in the moderate difficulty of KITTI bird's-eye view validation set, detection rate reaches 100 frames per second, which is able to meet the real-time requirements.

**Key words:** YOLOv4-tiny, Point cloud RGB feature map, Angle prediction, Dual-attention mechanism

**【引用格式】**吴庆,彭育辉,黄炜,等.基于点云鸟瞰图的实时车辆目标检测[J].汽车技术,2023(9):35-42.

WU Q, PENG Y H, HUANG W, et al. Real-Time Vehicle Target Detection Based on Bird's-Eye View of Point Cloud[J]. Automobile Technology, 2023(9): 35-42.

## 1 前言

近年来,基于车载激光雷达的三维目标检测备受关注,但是存在三维点云数据量大、目标检测算法计算量大和实时性差的问题<sup>[1-3]</sup>,将三维点云构建成鸟瞰图输入二维目标检测网络是一种有效的处理方法。根据信

息来源不同,基于鸟瞰图的目标检测算法可以分为两类,一类融合激光雷达与相机采集的信息进行检测,另一类仅基于激光雷达采集的点云信息进行检测。在基于激光雷达与相机的信息融合的目标检测算法方面,多视角3D(Multi-View 3D, MV3D)网络<sup>[4]</sup>采用3个输入,即RGB图像、鸟瞰图投影图像和前视图投影图像,利用深

\*基金项目:福建省引导性科技计划项目(2022H0007);福建省自然科学基金项目(2021J01559)。

通讯作者:彭育辉(1975—),男,教授,博士,主要研究方向为车辆智能驾驶、车辆工况等, pengyuhui@fzu.edu.cn。

度融合(ContFusion)网络对3个输入进行感兴趣区域(Region Of Interest, ROI)提取操作,输出融合特征图,最后细化3D检测框;深度融合网络<sup>[5]</sup>分别对图像和点云进行特征提取,建立激光雷达点云源数据投影关系,提取融合特征,但计算效率较低。在基于纯点云的鸟瞰图目标检测算法中:BirdNet网络<sup>[6]</sup>将三维点云信息处理成鸟瞰图,将其输入快速区域卷积神经网络(Fast Region Convolutional Neural Network, Fast R-CNN)<sup>[7]</sup>中完成检测任务;BirdNet++网络<sup>[8]</sup>在BirdNet网络的基础上添加两级对象检测器和特殊回归分支,可以实现由鸟瞰图检测回归到三维边界框;目标分解网络(Matting Objective Decomposition Network, MODet)<sup>[9]</sup>不考虑三维点云的反射强度,直接对三维点云进行鸟瞰图投影,通过残差结构提取高层语义信息,融合多层次特征图,网络检测速度达到20帧/s;3D网格(GridNet-3D)<sup>[10]</sup>将2D网格映射用于预处理原始点云,直接输入到区域提议网络模块完成检测,检测频率达到21.5 Hz;BEVDetNet<sup>[11]</sup>利用关键点检测目标中心,边界框预测和方向预测使用装箱分类在一个更简单的鸟瞰图上表示,检测时间达到0.06 s。FS<sup>2</sup>3D网络<sup>[12]</sup>针对鸟瞰图投影后的稀疏特征信息,采用网格前景分割而不采用基于锚框的方法来预测对象,检测网络运行速度达55.1帧/s。

综上,基于激光雷达与相机的信息融合的目标检测算法利用点云鸟瞰图信息以及其他传感器信息进行特征提取,但是检测效率低,而基于纯点云的鸟瞰图目标检测算法只通过点云鸟瞰图提取特征信息,检测效率有所提升,但是网络的特征信息提取能力较弱,导致检测精度较低。为此,本文提出一种基于点云鸟瞰图的实时车辆目标检测算法,将原始点云的高度、强度和密度信息进行编码,构建RGB特征图,通过添加车辆目标检测旋转框的检测尺度、添加改进空间金字塔池化(Spatial Pyramid Pooling-Fast, SPPF)模块、引入双注意力机制和优化损失函数对YOLOv4-tiny网络进行改进,在实现车辆目标检测网络轻量化的同时保证车辆的检测精度。

## 2 点云RGB特征图构建

目前,自动驾驶领域成熟的点云数据库有KITTI数据库、Apollo Scape自动驾驶数据库、Waymo数据库等,为网络模型的训练及验证提供支持。KITTI数据库中的车辆数据丰富,包含检测的各种场景,且标注完备,故选择KITTI数据库作为本文算法研究的数据支撑。

Chen等<sup>[4]</sup>提出MV3D网络对三维点云的高度、强度和密度分别进行编码,将投影点云离散化为分辨率为

0.1 m的二维网格,然后对每个网格计算高度、强度、密度。但是,投影点云离散化成二维网格中的分辨率对点云特征信息的保留十分重要,MV3D的二维网格的分辨率过大,导致丢失大量的点云特征信息,不利于后期的目标检测。

YOLOv4-tiny网络输入图片的分辨率为608×608,使二维网格变小,可改善三维点云投影到二维网格中车辆点云特征信息丢失的情况。考虑长度 $x$ 、宽度 $y$ 、高度 $z$ 的取值范围分别为 $[0,50]$  m、 $[-25,25]$  m、 $[-2.73,1.27]$  m的点云集合,YOLOv4-tiny输入的特征图尺寸高度和宽度 $H \times W$ 为608×608,对单帧点云进行区域划分,获得感兴趣区域 $\Omega$ 内的点云数据 $P_\Omega$ 满足:

$$P_\Omega = \{P_i = (x, y, z) | x \in [0, 50], y \in [-25, 25], z \in [-2.73, 1.27], i = 1, 2, \dots, n\} \quad (1)$$

对 $P_\Omega$ 中的任意一点 $P_i(x_i, y_i, z_i)$ 进行鸟瞰图的投影,投影到对应的像素点 $P_i(m_i, n_i)$ 上:

$$m_i = \frac{x_i}{x_{\max} - x_{\min}} \times H$$

$$n_i = \frac{y_i}{y_{\max} - y_{\min}} \times W + \frac{W}{2} \quad (2)$$

式中, $x_{\min}$ 、 $x_{\max}$ 分别为点云感兴趣区域 $\Omega$ 内 $x$ 坐标的最小值和最大值; $y_{\min}$ 、 $y_{\max}$ 分别为点云感兴趣区域 $\Omega$ 内 $y$ 坐标的最小值和最大值。

根据 $P_\Omega$ 内的点云数据计算每个像素的RGB通道值,实现三维点云的点云鸟瞰图转换,计算过程为:

$$R = \min(1, \frac{\log(N+1)}{\log(64)}), N = |P_g|$$

$$G = (\max(P_g \cdot [0, 0, 1]^T)) / z_{\max}$$

$$B = \max(I(P_g)) \quad (3)$$

式中, $N$ 为一个立体栅格中三维点云的数量; $g$ 为网格单元的大小; $P_g$ 为大小为 $g$ 的网络单元内点云数据矩阵; $z_{\max}$ 为大小为 $g$ 的网络单位内点云 $z$ 坐标的最大值; $I(P_g)$ 为大小为 $g$ 的网络单位内点云强度数据矩阵; $R$ 为映射到图像上的归一化密度,体现单元网格内的点云分布情况; $G$ 为点云归一化最大高度,体现单元网格内的点云高度情况; $B$ 为点云归一化最大强度,体现单元网格内的点云强度情况。

将 $R$ 、 $G$ 、 $B$ 3个通道合并,生成点云RGB特征图,如图1所示。网络的输入特征图尺寸 $H \times W \times C$ 为608×608×3。

## 3 目标检测网络

YOLO(You Only Look Once)网络的核心思想是将目标检测任务作为回归问题处理,以达到快速检测的目的。相较于YOLOv4网络模型,YOLOv4-tiny精简了特

征提取网络,提高了检测速度。网络由输入(Input)、骨干网络(Backbone)、颈(Neck)和检测头(Head)组成。骨干网络主要用于特征提取,采用 CSPDarknet53-tiny 网络结构;颈主要用于特征融合,采用特征金字塔网络 (Feature Pyramid Network, FPN) 结构;检测头根据提取到的特征结果进行最后的预测。

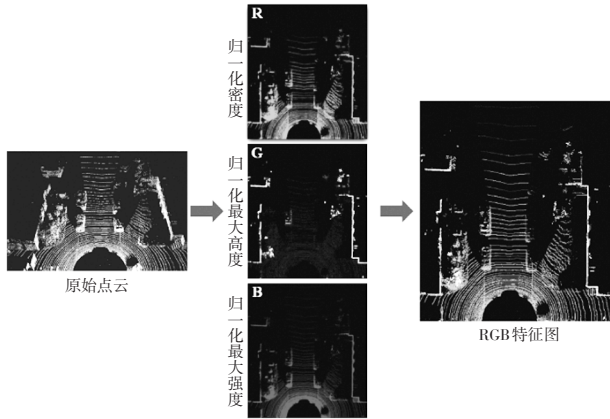


图1 点云 RGB 特征图构建示意

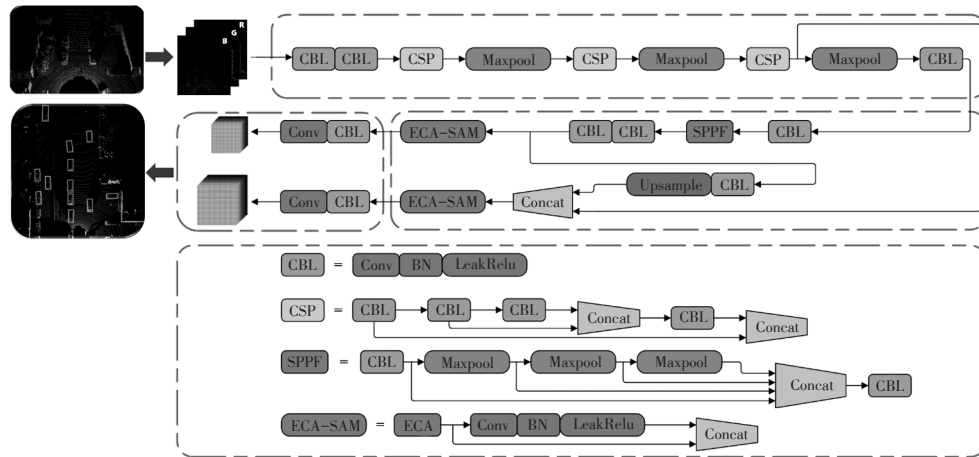


图2 改进 YOLOv4-tiny 的车辆点云鸟瞰图实时目标检测算法

根据实际的检测场景,设定一个目标检测框的参数为  $(b_x, b_y, b_w, b_l, b_\phi)$ , 其中  $(b_x, b_y)$  为点云鸟瞰图中目标车辆的中心点位置坐标,  $b_w, b_l$  分别为点云鸟瞰图中目标车辆的宽度和长度,  $b_\phi$  为点云鸟瞰图中目标车辆的偏航角:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_l &= p_l e^{t_l} \\ b_\phi &= \arctan_2(t_{\sin}, t_{\cos}) \end{aligned} \quad (4)$$

式中,  $t_x, t_y, t_w, t_l$  分别为目标检测框中心点的横、纵坐标、宽度和长度的预测参数;  $t_{\sin}, t_{\cos}$  分别为目标检测框的偏航角的正弦值和余弦值的预测参数;  $c_x, c_y$  分别为当前网格与网格原点的距离;  $p_w, p_l$  分别为先验框的宽度和长度;  $\sigma()$  为 Sigmoid 函数, 将预测参数限制在  $(0,1)$  范围内;

对 YOLOv4-tiny 网络的改进体现在: 添加车辆偏航角度预测分支, 使其适用于车辆在点云鸟瞰图上的检测场景, 实现车辆的精确定位; 添加 SPPF 模块, 融合点云鸟瞰图的全局特征和局部特征, 提升网络的目标定位能力; 引入双注意力机制, 分别加强学习图像的通道信息和空间位置信息, 获得更加丰富的特征, 提升网络的目标检测能力; 优化目标框位置损失函数, 添加目标预测偏航角损失函数, 提升网络的检测能力。整个网络的目标检测框架如图 2 所示, 其中卷积块 (CBL) 由卷积层 (Convolution)、批量归一化层 (Batch Normalization) 和 LeakReLU 激活函数组成, CSP (Cross Stage Partial) 模块为跨阶段局部模块, Concat 为拼接操作, Upsample 为下采样模块。

### 3.1 目标检测框设计

采用点云 RGB 特征图作为 YOLOv4-tiny 网络输入, 由于车辆在点云鸟瞰图中呈任意方向分布, 故在 YOLOv4-tiny 的原有目标框基础上, 添加车辆偏航角度预测分支。

$\arctan_2()$  为四象限反正切函数。

相较于直接回归车辆的偏航角度, 车辆目标偏航角通过正弦值和余弦值计算, 一方面可以避免奇点, 另一方面对模型泛化存在积极影响, 检测框参数示意如图 3 所示。

### 3.2 SPPF 结构

为了更好地融合点云鸟瞰图的全局特征和局部特征, 学习多尺度特征信息, 在 YOLOv4-tiny 的骨干网络后引入空间金字塔池化 (Spatial Pyramid Pooling, SPP) 结构。SPP 由 He 等<sup>[13]</sup>提出, 目的是解决计算机视觉中输入图像尺寸不同造成图像失真的问题, 如图 4a 所示, YOLOv4 中的 SPP 模块由 3 个最大池化层 (Maxpool) 构成, 池化层的大小分别为  $13 \times 13$ 、 $9 \times 9$  和  $5 \times 5$ , 步长为 1。YOLOv4 通过 SPP 模块增强了网络的感受野, 更加充分

地利用特征层的信息。如图4b所示,本文在YOLOv4-tiny的骨干网络后引入SPPF模块,SPPF也由3个最大池化层构成,通过串行3个5×5大小的最大池化层实现多尺度融合,相较于SPP模块,SPPF模块在保证与SPP模块具有相同作用的同时,可实现更高的计算效率。

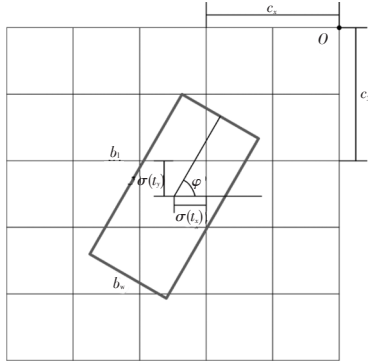


图3 目标检测框回归参数示意

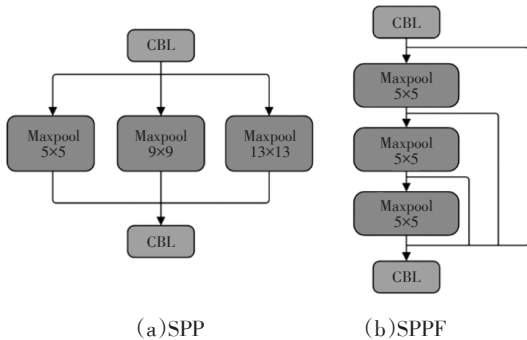


图4 空间金字塔池化结构

### 3.3 双注意力机制

相较于YOLOv4, YOLOv4-tiny采用了更加轻量化的网络结构,虽然大幅提高了目标检测效率,但是目标检测精度存在一定程度降低。YOLOv4-tiny有限的参数量限制了目标检测的效果,故在网络的主干部分添加双注意力机制,以充分利用有限的参数量,加强对通道和空间特征信息的关注。双注意力机制由高效通道注意力(Efficient Channel Attention, ECA)机制与空间注意力模块(Spatial Attention Module, SAM)串联构成。

ECA机制<sup>[14]</sup>在挤压和激励(Squeeze-and-Excitation, SE)注意力机制<sup>[15]</sup>的基础上改进而来,结构如图5所示,该机制避免降维,适当的跨通道交互对学习高性能和高效率的通道注意力较为重要。通过ECA机制可以加强特征通道的局部跨通道信息融合,根据全局特征信息对通道权重进行校准,使得网络更加关注重要的特征通道信息,提升网络检测的性能。

如图5所示,ECA模块首先经过全局平均池化获得一个C×1×1的特征向量,接着用一个权值共享的一维卷积来学习特征通道中的权重,其中一维卷积核k为模块

局部跨通道信息交互率,可随通道的变化动态调整:

$$k = \varphi(C) = \left\lfloor \frac{\log_2 C}{\gamma} + \frac{b}{\gamma} \right\rfloor_{\text{odd}} \quad (5)$$

式中,C为总通道数量;|a|<sub>odd</sub>为a最近的奇数;b=1;γ=2。

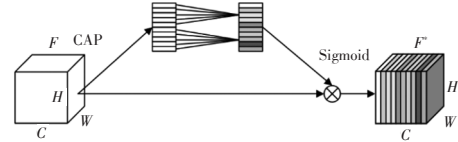


图5 ECA机制

ECA机制通过使网络更加关注特征通道信息来提升网络检测性能,在ECA模块后加入点注意力机制。本文的点注意力机制是基于空间注意力机制<sup>[16]</sup>的简化,通过一个1×1大小的卷积核与ECA模块输出的特征图进行卷积,通过网络的前、后传播来自适应调整空间特征信息的权重,输出的最终特征为该权重特征层与原始特征的加权和,从而提高网络对重要空间特征信息的关注,减少对无关空间特征信息的关注,如图6所示。

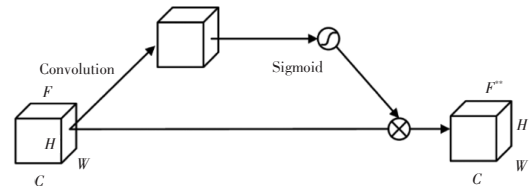


图6 简化的空间注意力机制

### 3.4 损失函数设计

设计的损失函数L<sub>tot</sub>由目标框预测损失L<sub>loc</sub>、置信度损失L<sub>conf</sub>、分类损失L<sub>cls</sub>和预测偏航角损失L<sub>φ</sub>组成:

$$L_{\text{tot}} = L_{\text{loc}} + L_{\text{conf}} + L_{\text{cls}} + L_{\varphi} \quad (6)$$

常见的目标框预测损失函数有GIoU(Generalized IOU)、DIoU(Distance-IoU)、CIoU(Complete-IoU)以及基于交并比(Intersection over Union, IoU)损失的统一雾化α-IoU<sup>[17]</sup>,其中α-IoU对目标框的回归更加稳定,收敛速度更快。α-IoU损失函数的表达式为:

$$L_{\text{loc}} = 1 - s_{\text{IoU}}^{\alpha} + \frac{\rho^{2\alpha}(b, b^{\text{gt}})}{c^{2\alpha}} + (\beta v)^{\alpha} \quad (7)$$

其中:

$$\begin{cases} \beta = \frac{v}{1 - s_{\text{IoU}} + v} \\ v = \frac{4}{\pi^2} \left( \arctan \frac{w^{\text{gt}}}{h^{\text{gt}}} - \arctan \frac{w}{h} \right)^2 \end{cases} \quad (8)$$

式中,s<sub>IoU</sub>为预测框与真实框的交并比;ρ()为欧式距离;b、b<sup>gt</sup>分别为预测框、真实框的中心点;c为能同时包含预测框和真实框的最小闭包区域的对角线长度;β为权重参数;v为衡量长宽比相似性的参数;w<sup>gt</sup>/h<sup>gt</sup>、w/h分别为真实框、预测框的宽高比;α为相对梯度权重参数,当α>

1时,可提升IoU较高的对象的损失和梯度加权,提高预测框的回归精度,且不会引入额外参数,也不会增加训练、推理时间。

分类损失和置信度损失函数采用交叉熵损失函数,有利于更新模型参数,同时可以加快模型的收敛。

只考虑正样本的目标预测偏航角损失,使用Smooth  $L_1$ 损失函数:

$$\text{Smooth } L_1 = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & \text{其他} \end{cases} \quad (9)$$

式中, $x$ 为误差。

同时在目标预测偏航角后加入惩罚项以保证偏航角的物理意义,目标偏航角损失函数表达式为:

$$L_\varphi = \text{Smooth } L(\sin \varphi - \sin \hat{\varphi}) + \text{Smooth } L_1(\cos \varphi - \cos \hat{\varphi}) + \text{Smooth } L_1(1 - \sin^2 \hat{\varphi} - \cos^2 \hat{\varphi}) \quad (10)$$

式中, $\varphi$ 、 $\hat{\varphi}$ 分别为目标偏航角的真实值和预测值。

## 4 试验验证

### 4.1 数据集

采用KITTI数据集进行算法的性能评估。KITTI数据集包含相机图像、激光雷达点云数据以及对应的标签文件,其训练集有7 481帧数据,测试集有7 518帧数据。KITTI数据集包含多个场景连续帧的数据,故直接划分训练集和验证集会包含相近帧的数据,无法直接通过验证集来评估模型的泛化能力,故按照采集时间分组随机划分数据。由于KITTI未公布测试集的标签文件,故本文将KITTI训练集按照6:2:2的比例划分为训练集、验证集和测试集;在KITTI官方的鸟瞰图性能评估试验中,将KITTI训练集随机拆分成3 712帧训练集和3 769帧数据验证集,用于模型网络的训练和本地评估。

### 4.2 试验环境与参数

应用PyTorch深度学习框架和Python编程语言实现检测网络,试验环境为Ubuntu 16.04操作系统,训练的服务器配置为Intel Xeon Silver 4108处理器、NVIDIA GTX 1080Ti显卡、32 GB运行内存。整个改进YOLOv4-tiny的目标检测网络采用表1的参数训练。

表1 网络参数

参数	数值
最大训练次数/次	300
批量大小	4
初始学习率	0.001
权重衰减系数	0.000 5
动量参数	0.949

### 4.3 数据增强

KITTI数据集中训练数据有限,本文通过数据增强增加训练数据量和环境噪声,提升网络的鲁棒性和泛化能力。数据增强具体的步骤为:收集所有真值目标框内的点云作为样本池;对每一帧投入训练的点云,随机抽取样本池中一定数量的样本,从而提高每一帧点云的正样本数量,同时进行碰撞测试,避免违反物理规律;对每一帧点云的目标边界框进行扩充,对增加的真值框进行随机旋转、平移和缩放,引入旋转和平移噪声,同时随机将样本池中的样本进行裁剪(Cutout)操作。通过以上数据增强操作,有效提高了网络训练的泛化能力,同时提高了网络整体的检测精度。

### 4.4 试验结果及分析

#### 4.4.1 评估指标

网络模型的评价指标为准确率(Precision)、召回率(Recall)、平均精度(Average Precision)、算法耗时(Time)及检测帧率。其中,准确率 $P$ 、召回率 $R$ 和平均精度 $S_{AP}$ 的表达式为:

$$P = n_{TP} / (n_{TP} + n_{FP}) \quad (11)$$

$$R = n_{TP} / (n_{TP} + n_{FN}) \quad (12)$$

$$S_{AP} = \int_0^1 P(R) dR \quad (13)$$

式中, $n_{TP}$ 为模型正确预测的数量; $n_{FP}$ 为预测错误检测的数量; $n_{FN}$ 为未检测到的目标数量; $S_{AP}$ 为PR曲线与水平轴围成的面积。

#### 4.4.2 网络改进对比

采用原始的YOLOv4-tiny网络对KITTI数据集进行训练和测试,模型在测试集上的平均精度达到93.98%;采用改进后的YOLOv4-tiny网络对KITTI数据集进行训练和测试,模型在测试集的平均精度达到96.92%,相较于原始网络提升了2.94个百分点,如图7所示。

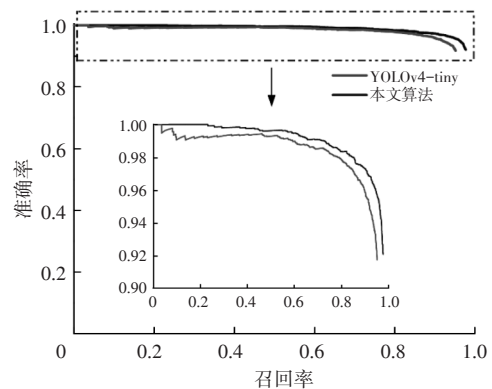


图7 模型改进前、后PR曲线对比

为了验证各改进模块的有效性,通过消融试验逐步验证本文基于YOLOv4-tiny改进的网络结构改变所引

起的性能的变化。表2所示为消融试验对比结果,由表2可知:YOLOv3网络和YOLOv4网络的检测速率较低;YOLOv4-tiny原网络的检测精度较低,检测速率较快;改进模型1引入SPPF模块,对应的AP@0.5(置信度阈值为0.5时的平均精度)提高了1.04个百分点,表明空间金字塔池化可以通过增强网络的感受野,充分利用特征信息提升网络的检测精度;改进模型2只引入双注意力机制,对应的AP@0.5提高了1.33个百分点,表明双注意力机制促进网络更加关注重要的通道特征信息和空间特征信息,降低对无关通道和空间特征信息的关注,对提升检测精度有积极影响;改进模型3同时引入SPPF模块和双注意力机制,对应的AP@0.5提高了2.07百分点;改进模型4在改进模型3的基础上,采用 $\alpha$ -IoU作为边界框的回归损失函数,相较于改进模型3,AP@0.5提高了0.4个百分点,表明采用 $\alpha$ -IoU作为边界框回归损失函数能够提高网络的检测精度。本文模型在以上改进措施条件下,使用马赛克(Mosaic)数据增强进行训练,相较于使用前AP@0.5提高了0.47个百分点。本文改进的模型在提升整体网络的检测精度的同时,仍能保证网络的检测速率达到100帧/s。

表2 消融试验

模型	SPPF	ECA-SAM	$\alpha$ -IoU	Mosaic	AP@0.5/%	检测速率/帧·s <sup>-1</sup>
YOLOv3					96.48	40
YOLOv4					97.11	30
YOLOv4-tiny					93.98	110
改进模型1	✓				95.02	100
改进模型2		✓			95.31	110
改进模型3	✓	✓			96.05	100
改进模型4	✓	✓	✓		96.45	100
本文模型	✓	✓	✓	✓	96.92	100

4.4.3 不同网络对比验证

为了将本文网络模型与其他使用鸟瞰图作为输入的网络模型进行比较,使用KITTI的官方性能评价标准,设置IoU阈值为0.7进行测试。KITTI数据集根据检测目标的最小边界框高度、遮挡程度和最大截断程度,将检测目标按照简单、中等、困难3个等级进行划分。由于KITTI官网的测试集未公布标注文件,故不同网络对比试验为采用KITTI鸟瞰图验证集在不同评价标准下的性能比较,检测结果如表3和图8所示。

由表3和图8可知,本文提出的车辆目标检测网络的算法耗时为0.01 s,检测速度可达到100帧/s。相较于BirdNet、MODet、GridNet-3D等使用点云鸟瞰图信息

的网络,本文在检测精度上具有优势,说明改进后的YOLOv4-tiny的网络能够较好地提取车辆点云的特征信息;相较于MV3D、F-PointNet(Frustum PointNet)、ContFusion等使用点云鸟瞰图信息和图像信息的网络,本文在检测速度上具有明显优势,说明将三维点云转换成点云RGB特征图以及采用轻量化的单阶段目标检测网络能够有效提高检测速度。以KITTI鸟瞰图数据集中等难度下的平均精度和帧速率作为评价指标,本文算法在检测精度和检测速度上具有优势,说明改进的YOLOv4-tiny网络能够更好地提取点云RGB特征图的特征信息,如图9所示。同时,将本文算法在KITTI官方测试集上进行检测,其部分检测结果如图10所示。

表3 KITTI数据集下不同网络性能比较

方法	使用数据	算法耗时/s	不同目标难度下的平均精度/%		
			简单	中等	困难
BirdNet <sup>[6]</sup>	点云	0.110	90.43	71.45	71.34
VoxelNet <sup>[18]</sup>	点云	0.230	89.60	84.81	78.57
PointPillars <sup>[19]</sup>	点云	0.016	88.35	86.10	79.83
MODet <sup>[9]</sup>	点云	0.050	89.89	88.57	82.13
GridNet-3D <sup>[10]</sup>	点云	0.047	90.06	85.81	80.13
FS <sup>2</sup> 3D <sup>[12]</sup>	点云	0.018	90.04	87.19	85.95
MV3D <sup>[4]</sup>	点云+图像	0.360	86.55	78.10	76.67
F-PointNet <sup>[20]</sup>	点云+图像	0.170	88.16	84.92	76.44
ContFusion <sup>[5]</sup>	点云+图像	0.060	95.44	87.34	82.43
本文算法	点云	0.010	94.48	87.73	85.75

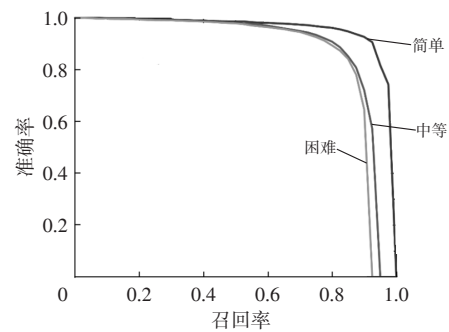


图8 KITTI验证集不同目标难度下的PR曲线

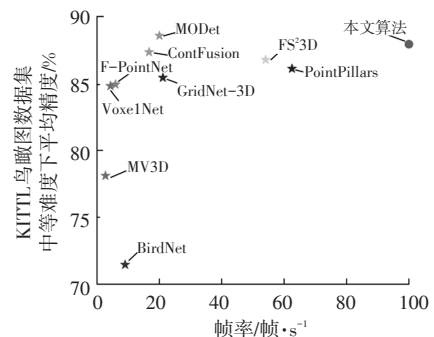


图9 不同网络下的性能比较

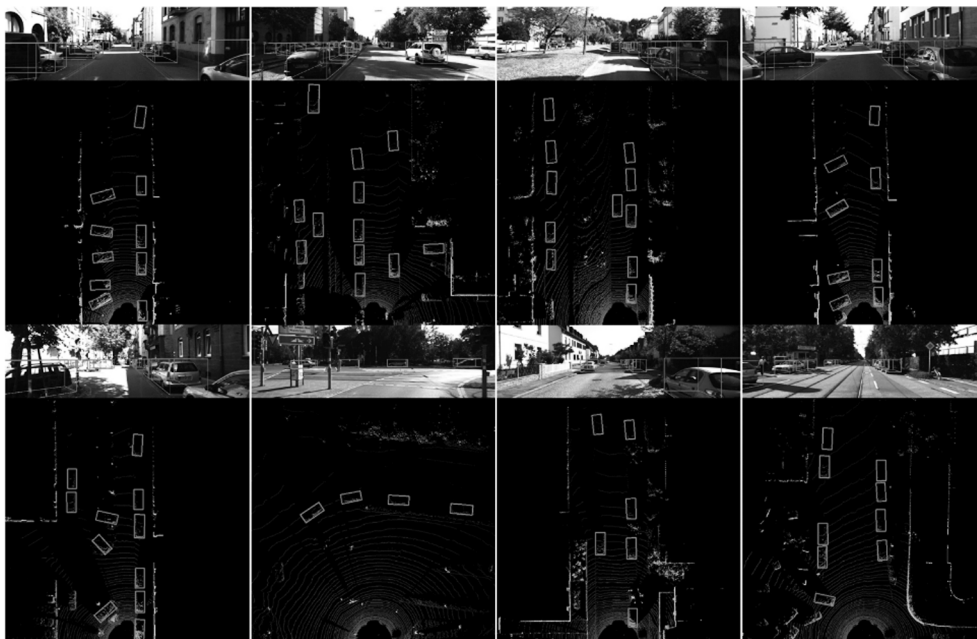


图10 在KITTI测试集上的部分检测结果

## 5 结束语

针对基于三维点云的车辆检测算法存在实时性差的问题,本文提出基于点云鸟瞰图的实时车辆目标检测算法,该算法通过将三维点云进行鸟瞰图投影,根据点云的高度、强度和密度信息制作点云RGB特征图,在保证车辆的特征信息的同时,有效降低了原始点云的数据量;其次,在YOLOv4-tiny网络上进行改进优化,添加车辆偏航角度预测分支来实现车辆的精确定位,引入空间金字塔池化结构、双注意力机制和采用 $\alpha$ -IoU边界框回归损失函数来提高车辆目标的检测精度。试验结果表明:基于YOLOv4-tiny网络的改进优化措施相较于原网络具有更高的检测精度;与其他网络相比较,本文网络在KITTI数据集下具有更高的检测速度,达到100帧/s,能够满足目标检测的实时性要求。

### 参 考 文 献

- [1] 伍锡如, 薛其威. 基于激光雷达的无人驾驶系统三维车辆检测[J]. 光学精密工程, 2022, 30(4): 489-497.  
WU X R, XUE Q W. 3D Vehicle Detection for Unmanned Driving System Based on Lidar[J]. Optics and Precision Engineering, 2022, 30(4): 489-497.
- [2] 尹慧琳, 伍淑莉, 王亚伟, 等. 基于层次因子图的智能车环境感知和态势认知模型[J]. 控制与决策, 2020, 35(10): 2528-2534.  
YIN H L, WU S L, WANG Y W, et al. An Intelligent Vehicle Environment Perception and Situation Cognition Model Based on Hierarchical Factor Graph[J]. Control and Decision, 2020, 35(10): 2528-2534.

- [3] 张燕咏, 张莎, 张昱, 等. 基于多模态融合的自动驾驶感知及计算[J]. 计算机研究与发展, 2020, 57(9): 1781-1799.  
ZHANG Y Y, ZHANG S, ZHANG Y, et al. Multi-Modality Fusion Perception and Computing in Autonomous Driving[J]. Journal of Computer Research and Development, 2020, 57(9): 1781-1799.
- [4] CHEN X Z, MA H M, WAN J, et al. Multi-View 3D Object Detection Network for Autonomous Driving[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE, 2017: 6526-6534.
- [5] LIANG M, YANG B, WANG S L, et al. Deep Continuous Fusion for Multi-Sensor 3D Object Detection[C]// Proceedings of the European Conference on Computer Vision (ECCV). Munich, Germany: Springer, 2018: 641-656.
- [6] BELTRAN J, GUINDEL C, MORENO F M, et al. BirdNet: A 3D Object Detection Framework from Lidar Information[C]// 2018 21st International Conference on Intelligent Transportation Systems (ITSC). Maui, HI, USA: IEEE, 2018: 3517-3523.
- [7] GIRSHICK R. Fast R-CNN[C]// Proceedings of the IEEE International Conference on Computer Vision. Santiago, Chile: IEEE, 2015: 1440-1448.
- [8] BARRERA A, GUINDEL C, BELTRAN J, et al. BirdNet++: End-to-End 3D Object Detection in Lidar Bird's Eye View [C]// 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). Rhodes, Greece: IEEE, 2020: 1-6.
- [9] 张易, 项志宇, 乔程昱, 等. 基于3维点云鸟瞰图的高精度实时目标检测[J]. 机器人, 2020, 42(2): 148-156.  
ZHANG Y, XIANG Z Y, QIAO C Y, et al. High-Precision

- Real-Time Object Detection Based on Bird's Eye View from 3D Point Clouds[J]. Robot, 2020, 42(2): 148-156.
- [10] YUE Y C, CAI Y F, WANG D S. GridNet-3D: A Novel Real-Time 3D Object Detection Algorithm Based on Point Cloud[J]. Chinese Journal of Electronics, 2021, 30(5): 931-939.
- [11] MOHAPATRA S, YOGAMANI S, GOTZIG H, et al. BEV-DetNet: Bird's Eye View LiDAR Point Cloud Based Real-Time 3D Object Detection for Autonomous Driving[C]// 2021 IEEE International Intelligent Transportation Systems Conference(ITSC). Indianapolis, US: IEEE, 2021: 2809-2815.
- [12] WANG B, ZHU M, LU Y, et al. Real-Time 3D Object Detection from Point Cloud Through Foreground Segmentation[J]. IEEE Access, 2021, 9: 84886-84898.
- [13] HE K M, ZHANG X Y, REN S Q, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916.
- [14] WANG Q, WU B, ZHU P, et al. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks [C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). Seattle, WA, USA, 2020: 11531-11539.
- [15] HU J, SHEN L, SUN G. Squeeze-and-Excitation Networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(8): 2011-2023.
- [16] ZHU X Z, CHENG D Z, ZHANG Z, et al. An Empirical Study of Spatial Attention Mechanisms in Deep Networks [C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. South Korea: IEEE, 2019: 6688-6697.
- [17] HE J, ERFANI S, MA X, et al. Alpha-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression[J]. Advances in Neural Information Processing Systems, 2021, 34.
- [18] 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 (CVPR). Salt Lake City, USA: IEEE, 2018: 4490-4499.
- [19] 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. Long Beach, California, USA: IEEE, 2019: 12697-12705
- [20] QI C R, LIU W, WU C, et al. Frustum PointNets for 3D Object Detection From RGB-D Data[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018: 918-927.

(责任编辑 斛 畔)

修改稿收到日期为2023年1月30日。

## 《汽车工程师》征稿启事

《汽车工程师》于1974年创刊,是由中国第一汽车集团有限公司主管、中国第一汽车股份有限公司主办的、国内外公开发行的汽车应用技术类学术期刊,国家新闻出版广电总局第一批认定的学术期刊。

《汽车工程师》以报道汽车产品开发领域的成果为主,专注于技术开发与应用,提供汽车设计创新解决方案。《汽车工程师》将把握电动化、智能化、网联化、共享化的汽车技术主流发展趋势,努力在电池技术、电驱技术、电控技术、高压技术、补能技术、燃料电池技术、智慧控制、智能驾驶、智享座舱、智能悬架、线控转向、线控制动、NVH、功能安全、预期功能安全、信息安全、数据安全、被动安全、高效动力、高效传动、智能管理、低风阻、低滚阻、轻量化领域吸收优质稿源,为我国汽车工程技术创新能力提升贡献力量。

热忱欢迎汽车行业的专家学者不吝赐稿,反映国家重点扶持项目、自然科学基金项目和其他重点项目等研究成果的稿件将优先发表,我们期待与您共同践行“把论文写在祖国大地上”的指示精神,为强大中国汽车工业作贡献!

《汽车工程师》编辑部