

doi: 10.3969/j.issn.1671-7775.2020.03.004

基于注意力机制的 PointPillars + 三维目标检测

詹为钦¹, 倪蓉蓉², 杨彪¹

(1. 常州大学 信息科学与工程学院, 江苏 常州 213164; 2. 常州纺织服装职业技术学院 能源管理科, 江苏 常州 213164)

摘要: 针对自动驾驶场景下的周边车辆及行人等目标的识别和定位问题,提出了一种基于注意力机制的 PointPillars + 三维目标检测算法. 算法将完整空间按等尺寸柱均匀划分,从所有柱中提取内部点云的特征进而生成伪图;引入2种注意力机制,实现对伪图中特征信息的放大和抑制;使用卷积神经网络对注意力模块输出结果进一步处理,并使用 SSD (single shot multibox detector) 算法进行三维目标检测. 结果表明:基于并行注意力机制的 PointPillars + 检测算法取得了良好的检测效果,相对于 PointPillars 算法,鸟瞰图下,平均均值精度(mAP)的中等难度均值(mAP_m)从66.19增加到69.95,汽车的 mAP 从86.10增加到87.73;三维模式下, mAP_m 从59.20增加到62.55,汽车的 mAP 从74.99增加到76.25.

关键词: 三维目标; 激光雷达; 注意力机制; 卷积神经网络; 点云

中图分类号: TP391.4 **文献标志码:** A **文章编号:** 1671-7775(2020)03-0268-06

引文格式: 詹为钦,倪蓉蓉,杨彪. 基于注意力机制的 PointPillars + 三维目标检测[J]. 江苏大学学报(自然科学版),2020,41(3): 268-273.

An attention-based PointPillars + 3D object detection

ZHAN Weiqin¹, NI Rongrong², YANG Biao¹

(1. School of Information Science and Engineering, Changzhou University, Changzhou, Jiangsu 213164, China; 2. Department of Energy Management, Changzhou Vocational Institute of Textile and Garment, Changzhou, Jiangsu 213164, China)

Abstract: To accurately recognize and locate the surrounding vehicles and pedestrians, an attention-based PointPillars + 3D target detection algorithm was proposed. The entire space was uniformly divided into pillars with a given resolution, and a pseudo-image was generated by extracting point-based features from all pillars. Two attention modules were introduced to highlight and restrain the information in the pseudo-image. A convolution neural network was used to process the output of the attention module, and the single shot multibox detector(SSD) was used for 3D object detection. The evaluation results show that the parallel attention-based PointPillars achieves good performance. Compared with the traditional PointPillars, the mAP_m is increased from 66.19 to 69.95 in the bird's eye view, and the car mAP is increased from 86.10 to 87.73. In the 3D mode, the mAP_m is increased from 59.20 to 62.55, and the car mAP is increased from 74.99 to 76.25.

Key words: 3D object; lidar; attention mechanism; convolutional neural network; point cloud

深度神经网络在计算机视觉领域取得了突破性进展,以传统机器学习算法为主导的二维目标检测

算法正被深度神经网络算法^[1]所取代. 基于深度神经网络的目标检测算法在检测精度和识别率方面都

收稿日期: 2019-09-29

基金项目: 江苏省高校自然科学研究面上项目(18KJB520003); 江苏省道路货运工具新技术应用重点实验室开放课题(BM20082061708)

作者简介: 詹为钦(1990—),男,江苏扬州人,硕士研究生(augmare@qq.com),主要从事计算机视觉、量化交易研究.

杨彪(1987—),男,江苏常州人,讲师(通信作者,yb6864171@cczu.edu.cn),主要从事模式识别、机器学习的理论及应用研究.

优于传统目标检测算法. 人们也在尝试将神经网络应用在其他检测领域. 自动驾驶是三维目标检测领域的前沿应用方向, 车辆在自动驾驶过程中, 需要对周围的车辆、行人等目标进行检测和行为预测.

传统二维图片检测算法, 使用车载相机作为数据来源, 依赖外部光源, 且无法精确定位目标车辆、人物的距离、位置、深度和角度等信息. 由车载激光雷达所生成的三维点云数据, 则包含了目标对象的位置、距离、深度和角度等信息, 数据构成更符合真实情况. 车载激光雷达具有测距准、无需可见光等优点. 基于点云数据集的目标检测算法可以为自动驾驶系统提供目标对象的准确空间方位、速度等信息. 因此研究基于点云数据的三维目标检测十分必要.

三维目标检测算法可分为单步检测算法和双步检测算法. 单步检测算法通过使用卷积神经网络对鸟瞰图提取特征来实现. 具有实现简单、检测速度快等优点. 但在将点云数据转化为鸟瞰图时, 会造成目标空间特征信息丢失. 文献[2]将点云数据划分为一定数量的体素, 使用体素特征编码层提取局部特征, 再使用三维卷积层和区域候选网络对目标分类和回归. 文献[3]通过在文献[2]网络中添加稀疏卷积层的方法, 提高了网络的整体检测速度以及检测效果. 文献[4]将检测最小单元从文献[2]中的体素改为柱, 通过柱特征网络提取柱内点云特征并生成伪图, 最后通过区域候选网络和单步多尺度检测网络实现目标检测任务. 单步检测速度较快, 精度适中, 在车载嵌入式领域有广泛的应用. 文献[5]对 RGB 图像使用检测算法以生成若干个目标建议区, 接着在对应点云数据上延轴线生成若干个截锥体块, 最后对每个截锥体块采用文献[6]的方法提取特征并输出检测结果. 双步检测算法由于需要同时检测 RGB 图像和点云数据, 因此检测速度通常慢于单步检测算法, 而检测精度一般高于单步检测算法.

注意力机制可以对网络的输入数据按不同部分赋予权重, 抽取其中关键信息, 抑制不重要信息. 帮助网络学习数据中的重要信息, 以做出更加准确的判断. 注意力模块计算要求低, 不会给算法的训练、运行、计算和存储带来太多额外开销. 文献[7]对通道重新加权, 模块首先对输入($H \times W \times C_1$)使用压缩操作, 输出大小为($1 \times 1 \times C_1$)的特征图; 接着使用提取操作, 另参数 ω 学习 C_1 个通道间的相关性, 生成($1 \times 1 \times C_2$)的特征图; 最后输出堆叠后的特征图($H \times W \times C_2$), C_2 即为加权后通道值. 文献[8]结合通道注意力和空间注意力2种注意力机制, 重新

加权通道和空间位置, 并通过自主学习的方式学习每个特征空间的重要性程度. 目前注意力模块被广泛应用于计算机视觉和自然语言处理等相关领域.

为进一步提取伪图中特征信息, 笔者在文献[4]检测算法基础上提出一种基于注意力机制的 PointPillars + 三维目标检测算法, 算法引入空间注意力模块及通道注意力模块. 使用卷积神经网络对注意力模块输出结果进行处理, 并使用 SSD (single shot multibox detector)^[9] 算法进行三维目标检测.

1 检测网络模型

PointPillars + 注意力模型如图1所示, 网络输入点云数据集, 计算并得出目标的预测结果. 网络主要分为4个部分: ①柱特征网络, 在三维空间中对点云数据按等尺寸柱均匀划分, 并提取柱内点云特征生成伪图; ②注意力模块, 对伪图特征数据更进一步加工(特征的放大和抑制); ③Backbone 模块, 将经过注意力模块处理过的伪图进行多尺度的特征提取; ④检测模块, 负责目标对象的检测, 三维边界框的生成和回归.

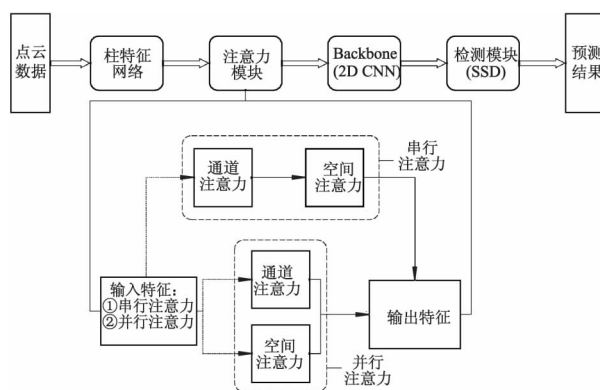


图1 PointPillars + 注意力模型

1.1 柱特征网络

首先对点云数据进行划分, 点云数据中的点 L 用 (x, y, z) 表示. 柱特征网络将点云数据均匀分布在基于 Oxy 平面的网格中. 由这些网格组成1组柱集合, 即柱 Pillars. 柱 Pillars 在 z 轴上没有高度限制. 点 L 添加 $r_l, x_m, y_m, z_m, x_p, y_p$ 信息来增强表示, r_l 为反射率; x_m, y_m, z_m 分别为点 L 到柱中所有点的算术平均值距离; x_p, y_p 分别为点 L 距柱中心 (x, y) 的方向偏离. 因此网络中的1个点 L 由9维特征表示. 点云数据集具有稀疏性, 空间中大部分的柱都为空柱, 少数非空柱也只有很少的点. 点云数据使用0~97%的稀疏度, 通过对每个样本的非空柱数 (P) 和每个柱

的点数(N)实施限制来利用这种稀疏度(D),以此产生尺度为(D, P, N)的疏密度张量. 如果样本或柱中保留过多数据,则通过随机采样保留数据以适应张量. 相反,如果样本或柱中数据过少,则使用0填充张量. 接着使用类似 PointNet^[6]的网络,使每个点通过线性层、BatchNorm 层和 ReLU 层,输出大小为(C, P, N)的张量. 对通道 C 使用最大化操作,得到大小为(C, P)的张量. 线性层可以用卷积核为 1×1 的卷积层替代,这种替代方法可以提高计算效率. 最后,特征被按照原始柱的位置组合堆积起来,形成大小为(C, H, W)的伪图,其中 H 和 W 分别为伪图的高度和宽度. 生成伪图后,送入注意力模块,对伪图中特征进行处理.

1.2 注意力模块

注意力模块的主要功能: 期望通过添加注意力机制来增加数据的表征能力,使网络学习伪图特征中的重要信息并抑制不重要的信息. 注意力机制是受人的行为特点启发所发明,当人们需要做出决定时,会有选择性地使用数据中重要的部分作为判断的主要依据. 与之类似,人工模型在处理问题时,不会平等地处理所有数据,而是只关注其中重要数据,这一过程称为注意力机制. 当处理的问题是输入本身时,通常称为自我注意力. 自我注意力对于网络学习给定任务十分有意义. 例如,在分类任务下,自我注意力以关注信息特征的形式表现出来. 假设输入特征图 $F \in \mathbf{R}^{C \times H \times W}$,参考 CBAM^[8] 注意力模型将 F 分为一维的通道注意力 $M_c \in \mathbf{R}^{C \times 1 \times 1}$ 和二维的空间注意力 $M_s \in \mathbf{R}^{1 \times H \times W}$. 整个注意力处理过程如下:

$$F' = M_c(F) \otimes F, \quad (1)$$

$$F'' = M_s(F') \otimes F', \quad (2)$$

式中: F' 为经过通道注意力加工后的数据; F'' 为经过空间注意力加工后的数据; \otimes 为逐元素乘法.

1.2.1 通道注意力

模块采用通道内部间的特征关系来产生通道注意力,如图2所示. 由于每个通道的特征图都被看作特征探测器,因此通道注意力主要试图找出输入数据中“有意义”的部分. 为了提高计算通道注意力的效率,对输入数据的空间维度进行了压缩. 为了汇总空间信息,通道注意力同时采用了最大池化和平均池化来聚合特征图的空间信息,分别用 F_{\max} , F_{avg} 表示. 接着将上述信息转入共享网络中以产生通道注意力 $M_c \in \mathbf{R}^{C \times 1 \times 1}$. 共享网络由单隐层的多层感知机组成,为了减少参数开销,隐藏层激活大小设置为 $\mathbf{R}^{(C/r) \times 1 \times 1}$,其中 r 为缩减比例. 在这之后使用逐元素

相加法,输出合并后的特征向量. 通道注意力^[8]为

$$M_c(F) = \sigma\{MLP[AvgPool(F) + MaxPool(F)]\} = \sigma\{W_1[W_0(F_{\text{avg}})] + W_1[W_0(F_{\max})]\}, \quad (3)$$

式中: MLP 为多层感知机; $AvgPool$ 为平均池化; $MaxPool$ 为最大池化; σ 为 sigmoid 函数; W_0 , W_1 均为多层感知机的可学习参数, $W_0 \in \mathbf{R}^{(C/r) \times C}$, $W_1 \in \mathbf{R}^{C \times (C/r)}$.

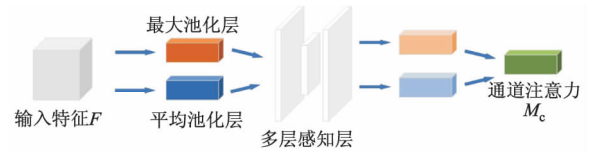


图2 通道注意力

1.2.2 空间注意力

模块利用特征的内部空间关系生成空间注意力,如图3所示. 与通道注意力不同,空间注意力更关注信息的具体位置,是对通道注意力的补充. 为了计算空间注意力,在通道注意力之后依次进行平均池化和最大池化,通过2个池化操作,依次生成2个二维特征图: F_{avg} , F_{max} . 接着将两者连接起来输入卷积层,生成 $M_s \in \mathbf{R}^{1 \times H \times W}$.

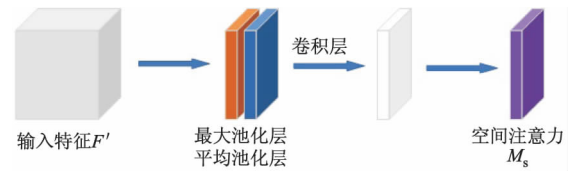


图3 空间注意力

空间注意力^[8]计算如下:

$$M_s(F) = \sigma\{f^{7 \times 7}[AvgPool(F); MaxPool(F)]\} = \sigma\{f^{7 \times 7}[F_{\text{avg}}; F_{\text{max}}]\}, \quad (4)$$

式中: $f^{7 \times 7}$ 为 7×7 的卷积核卷积操作.

1.3 Backbone 模块

Backbone 模块分为下采样网络和上采样连接网络. 下采样网络可由一系列(S, H, O)的块表示, 其中: S 为步长; O 为输出通道数; H 为 3×3 的二维卷积层层数. 每个通道后都接有 BatchNorm 层和 ReLU 层, 层内的第1个卷积具有步幅 S/S_{in} , 以确保网络层在接收到步长 S_{in} 的输入后, 仍保持为 S . 层中的后续卷积步长均为1, 3层的通道数分别为64, 128, 256. 下采样网络可以产生依次减小的空间分辨率. 上采样连接网络 $Up(S_{\text{in}}, S_{\text{out}}, S_F)$ 从初始步幅 S_{in} 到最后步幅 S_{out} 并得到最终特征 S_F , 和下采样网络相同, 上采样网络后都接有 BatchNorm 层和 ReLU 层. 最终输出来自不同步长的所有模块连接.

1.4 检测模块

检测模块采用了 SSD 进行目标检测, SSD 是典型的单步检测算法, 检测速度快, 精度高. SSD 网络中引入了锚的思想, 可以适应多尺度的目标检测任务, 较为符合点云数据尺度变换较大的特点. SSD 主要分为 6 个模块, 第 1 个模块由 VGG16 的前 5 层 Conv1-Conv5 卷积层组成, 接着将 VGG16 中的 FC6, FC7 全连接层转变为 Conv6, Conv7 卷积层为第 2 模块, 在此基础上, 又添加了 Conv8, Conv9, Conv10 和 Conv11 卷积层 4 个模块, 以此提取不同尺度下的目标信息, 最后进行目标分类检测和非极大值抑制回归位置操作. 非极大值抑制使用了二维 IoU^[10], 将先验边界框与真实目标相匹配. 提出的 SSD 检测算法可以替换成其他检测算法, 如 Faster R-CNN^[1] 等.

1.5 损失函数

参考文献[4]设置损失函数. 真实目标的边界框由 $(x, y, z, w, l, h, \theta)$ 表示, 其中: x, y, z 为边界框中心坐标; w, l, h 分别为边界框的宽度、长度和高度; θ 为边界框绕 z 轴的偏航旋转角度. 目标和锚点之间的线性回归残差定义为

$$\begin{cases} \Delta x = \frac{x_{gt} - x_a}{d_a}, \Delta y = \frac{y_{gt} - y_a}{d_a}, \Delta z = \frac{z_{gt} - z_a}{h_a}, \\ \Delta w = \log \frac{w_{gt}}{w_a}, \Delta l = \log \frac{l_{gt}}{l_a}, \Delta h = \log \frac{h_{gt}}{h_a}, \\ \Delta \theta = \sin(\theta_{gt} - \theta_a), \end{cases} \quad (5)$$

式中: 下标是 gt 的变量为目标真实值; 下标是 a 的变量为锚点值; $d_a = \sqrt{w_a^2 + l_a^2}$.

因此总的定位损失函数为

$$L_{loc} = \sum_{b \in (x, y, z, w, l, h, \theta)} SmoothL1(\Delta b), \quad (6)$$

式中: $SmoothL1$ 为 $L1$ 平滑函数.

因为定位损失无法区分边界框是否翻转, 因此使用 L_{dir} 在离散方向上学习边界框方向. 分类损失使用 $focal loss$ 损失函数, 即

$$L_{cls} = -\alpha_a (1 - P_a)^\gamma \log P_a, \quad (7)$$

式中: P_a 为锚点的概率值; $\alpha_a = 0.25$; $\gamma = 2$.

总的损失函数为

$$L = \frac{1}{N_{pos}} (\beta_{loc} L_{loc} + \beta_{cls} L_{cls} + \beta_{dir} L_{dir}), \quad (8)$$

式中: N_{pos} 为正概率锚数; $\beta_{loc} = 2$; $\beta_{cls} = 1$; $\beta_{dir} = 0.2$.

损失函数使用的是 Adam 优化器, 初始学习率为 2×10^{-4} , 每 15 个周期衰变为当前学习率的 0.8 倍, 试验参数均来自文献[4], 并使用 KITTI^[11] 官方点云数据集, 通过两折交叉验证法证实参数有效性.

2 试验相关

2.1 数据集

试验数据来自 KITTI^[11] 官网, 它是目前自动驾驶领域最重要的数据集之一. 由德国卡尔斯鲁厄理工学院和丰田美国技术研究院联合建立. KITTI 数据集包含点云和图像数据. 数据集内包含市区、乡村和高速公路等真实驾驶场景的数据图像. 每张图像中最多包括 15 辆车和 30 个行人, 以及不同程度的遮挡物等. 在训练过程中只使用 KITTI 点云数据. 根据惯例, 将数据集划分为 7 481 个训练集和 7 518 个测试集. 在训练过程中, 又将训练集分为 3 712 个训练样本和 3 769 个验证样本.

2.2 试验细节

试验环境为 Ubuntu 16.04 操作系统, Python 3.7, Pytorch 1.1. 处理器为 Intel i5 CPU, 显卡为 Nvidia 1080 GPU. 默认参数: 柱的 x, y 分辨率都为 0.16 m; 最大柱数 (P) 为 12 000 个; 柱内最大点数 (N) 为 100 个. 每个类的锚点由宽度、长度、高度和 z 中心组成, 具有 0° 和 90° 这 2 个方向. 二维 IoU 将锚点与真实目标匹配时, 正匹配通常选择标记值中最高的或大于正匹配阈值的. 负匹配则选低于负阈值的. 匹配时使用轴对齐的非最大抑制方法, 重叠阈值为 0.5 倍 IoU. 汽车的 x, y, z 阈值分别为 $(0, 70.4), (-40.0, 40.0), (-3.0, 1.0)$ m. 锚宽、长度和高度分别为 1.60, 3.90, 1.50 m, z 中心为 -1.00 m. 匹配正负阈值为 0.60 和 0.45. 行人和骑行者的 x, y, z 阈值分别为 $(0, 48.0), (-20.0, 20.0), (-2.5, 0.5)$ m. 行人锚宽、长度和高度分别为 0.60, 0.80, 1.73 m, 骑行者锚宽、长度和高度分别为 0.60, 1.76, 1.73 m, 两者 z 中心均为 -0.60 m.

2.3 数据增强

数据增强通过增加数据样本提高网络的泛化能力及检测效果. 首先为所有类创建基于真实目标的三维边界框查找表以及边界框内的点云. 对每个样本, 采用随机选择 15 辆汽车及 8 位骑车者的真实样本, 将它们放入当前点云数据中参与网络训练. 接下来逐个增加真实目标的边界框, 对每个边界框分别旋转 $[-\pi/20, \pi/20]$ 以及平移操作, 其中框的 x, y, z 坐标按 $N(0, 0.25)$ 的正态分布进行转换, 模型通过以上操作实现训练样本集数量的增加. 最后, 对所有边界框沿 x 轴进行镜像翻转并按 $N(0, 0.20)$ 的正态分布绘制 x, y, z 全局平移, 模拟定位噪声.

3 结果分析

3.1 定量分析

定量分析评测分为鸟瞰图、二维、三维和平均方向相似性几种方法. KITTI 官方数据集分为简

单、中等和困难 3 种难度, KITTI 官网排行榜按中等难度数据集排名. 鸟瞰图模式、三维模式、AOS 模式验证结果分别如表 1-3 所示, 其中: mAP 为平均均值精度; mAP_m 为汽车、行人以及骑行者中等难度下 mAP 的均值; 检测算法均为单步目标检测算法.

表 1 鸟瞰图模式验证结果

检测算法	汽车的 mAP			行人的 mAP			骑行者的 mAP			mAP_m
	简单	中等	困难	简单	中等	困难	简单	中等	困难	
VoxelNet ^[2]	89.35	79.26	77.39	46.13	40.74	38.11	66.70	54.76	50.55	58.25
SECOND ^[3]	88.07	79.37	77.95	55.10	46.27	44.76	73.67	56.04	48.78	60.56
PointPillars ^[4]	88.35	86.10	79.83	58.66	50.23	47.19	79.14	62.25	56.00	66.19
PointPillars + 串	89.94	86.88	84.38	41.53	39.07	36.12	66.79	49.48	46.75	58.48
PointPillars + 并	90.26	87.73	85.06	65.21	60.38	56.97	81.02	61.76	57.97	69.95

表 2 三维模式验证结果

检测算法	汽车的 mAP			行人的 mAP			骑行者的 mAP			mAP_m
	简单	中等	困难	简单	中等	困难	简单	中等	困难	
VoxelNet ^[2]	77.47	65.11	57.73	39.48	33.69	31.50	61.22	48.36	44.37	49.05
SECOND ^[3]	83.13	73.66	66.20	51.07	42.56	37.29	70.51	53.85	46.90	56.69
PointPillars ^[4]	79.05	74.99	68.30	52.08	43.53	41.49	75.78	59.07	52.92	59.20
PointPillars + 串	83.48	75.59	69.70	33.32	31.31	29.03	64.13	45.74	43.24	50.88
PointPillars + 并	85.47	76.25	70.13	57.24	53.53	48.81	78.54	57.88	54.30	62.55

表 3 AOS 模式验证结果

检测算法	汽车的 mAP			行人的 mAP			骑行者的 mAP			mAP_m
	简单	中等	困难	简单	中等	困难	简单	中等	困难	
SECOND ^[3]	87.84	81.31	71.95	51.56	43.51	38.78	80.97	57.20	55.14	54.53
PointPillars ^[4]	90.19	88.76	86.38	58.05	49.66	47.88	82.43	68.16	61.96	68.86
PointPillars + 串	90.48	88.32	86.51	46.24	46.21	45.60	67.07	48.93	46.31	61.15
PointPillars + 并	90.68	88.92	86.90	46.53	47.31	46.84	80.20	61.82	58.25	66.01

从表 1, 2 可以看出: PoinPillars + 并行注意力在所有类检测结果中均取得良好的结果, 相对于 PointPillars 算法, 鸟瞰图下, mAP_m 从 66.19 增加到 69.95, 汽车的 mAP 从 86.10 增加到 87.73; 三维模式下, mAP_m 从 59.20 增加到 62.55, 汽车的 mAP 从 74.99 增加到 76.25. 除骑行者类中等难度以外, PoinPillars + 并行注意力在鸟瞰图和三维 2 种方法下, 均优于表中其他检测模型, 骑行者类中等难度检测结果也与第 1 名相差不大. 证明了 PoinPillars + 并行注意力模型的有效性, 以及注意力机制的有效性.

从表 3 可以看出: 行人和骑行者的 mAP 检测结果略有下降, 并不理想, 考虑到平均方向相似性模式和三维模式的主要检测区别在于是否判别目标方向, 推测是模型在提高了检测结果后, 丢失了一部分的方向信息. 通过观察三维模式和鸟瞰图模式中行人检测结果的 mAP 大幅提升, 可以说明这个问题.

3.2 定性分析

在训练过程中仅使用点云数据集, 为了便于研究、观察和解释说明, 分别展示了鸟瞰图 and RGB 图像的边界框预测结果. 正例试验结果如图 4 所示.

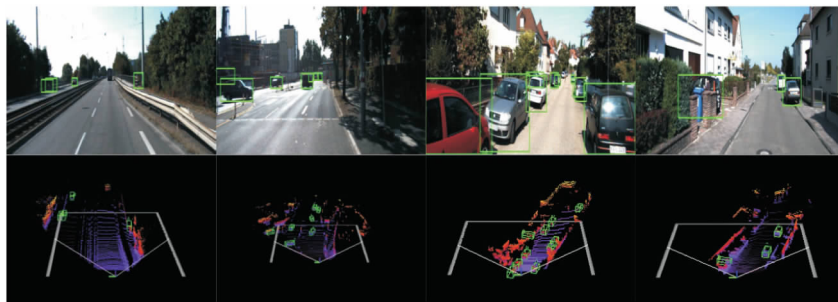


图 4 正例试验结果

从图4可以看出: 检测结果具有紧密且定向的三维边界框. 汽车的预测结果也较准确, 此外鸟瞰图的三维边界框与RGB图像中的边界框的对应关系也很明显, 没有发现误报、错报等错误现象. 反例试验结果如图5所示, 所展示的为一些常见的检测失败结果. 主要包括对困难样本(部分遮挡或距离较

远)或相似类别(卡车与大型客车)误报. 此外检测行人和骑行者更加困难, 行人和骑自行车者通常被错误分类, 行人和桌子组合被误判为骑行者. 此外, 行人也容易被误判为杆、树干之类的物体, 此处也更进一步说明了平均方向相似性模式下, 模型行人检测率较低的原因.

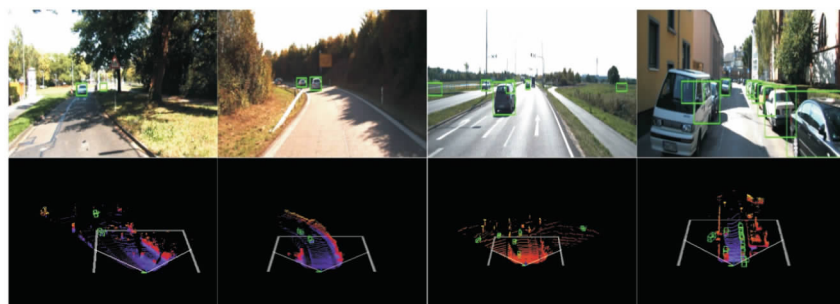


图5 反例试验结果

4 结 论

提出了一种基于注意力机制的 PointPillars + 注意力机制的目标检测模型, 用来检测真实交通场景下的汽车、行人和骑行者检测. 将训练的模型结果与以往模型的结果相比较可以发现, PointPillars + 并行注意力检测模型在预测结果方面有较大优势. 2组注意力机制对比结果表明: 并行注意力机制更适用于本模型, 注意力机制的内部结构, 也对模型检测结果的准确率有十分重要的影响.

参考文献(References)

- [1] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137 – 1149.
- [2] ZHOU Y, TUZEL O. Voxelnet: end-to-end learning for point cloud based 3d object detection [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2018: 4490 – 4499.
- [3] YAN Y, MAO Y X, LI B. Second: sparsely embedded convolutional detection [J]. Sensors, 2018, 18(10): 3337 – 1 – 17.
- [4] 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. Piscataway, USA: IEEE, 2019: 12697 – 12705.
- [5] WANG Z X, JIA K. Frustum convnet: sliding frustums to aggregate local point-wise features for amodal 3d object detection [J]. arXiv preprint arXiv: 1903.01864v2, 2019.
- [6] QI C R, SU H, MO K, 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. Piscataway, USA: IEEE, 2017: 652 – 660.
- [7] HU J, SHEN L, SUN G, et al. Squeeze-and-excitation networks [C] // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Piscataway, USA: IEEE, 2018: 7132 – 7141.
- [8] WOO S, PARK J, LEE J Y, et al. Cbam: convolutional block attention module [C] // Proceedings of the European Conference on Computer Vision, 2018: 3 – 19.
- [9] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: single shot multibox detector [C] // European Conference on Computer Vision, 2016: 21 – 37.
- [10] EVERINGHAM M, VAN Gool L, WILLIAMS C K I, et al. The pascal visual object classes (VOC) challenge [J]. International Journal of Computer Vision, 2010, 88(2): 303 – 338.
- [11] GEIGER A, LENZ P, STILLER C, et al. Vision meets robotics: the KITTI dataset [J]. The International Journal of Robotics Research, 2013, 32(11): 1231 – 1237.

(责任编辑 贾国方)