

基于 YOLOv5 的多任务自动驾驶环境感知算法^①



张凯祥, 朱 明

(中国科学技术大学 信息科学技术学院, 合肥 230027)

通信作者: 张凯祥, E-mail: zhx48@mail.ustc.edu.cn

摘 要: 自动驾驶任务是当前深度学习研究的热门领域, 环境感知作为自动驾驶中最重要的模块之一, 是一项极具挑战性并具有深远意义的任务, 包括目标检测、车道线检测、可行驶区域分割等. 传统的深度学习算法通常只解决环境感知中的一个检测任务, 无法满足自动驾驶同时感知多种环境因素的需求. 本文使用 YOLOv5 作为骨干网络及目标检测分支, 结合实时语义分割网络 ENet 进行车道线检测和可行驶区域分割, 实现了多任务自动驾驶环境感知算法, 损失计算时采用 α -IoU 提高回归精度, 对噪声有更好的鲁棒性. 实验表明, 在 BDD100K 数据集上, 本文提出的算法结构优于当前现有的多任务深度学习网络, 并且在 GTX1080Ti 上可达到 76.3 FPS 的速度.

关键词: 多任务; 环境感知; 目标检测; 车道线检测; 可行驶区域分割; YOLOv5

引用格式: 张凯祥, 朱明. 基于 YOLOv5 的多任务自动驾驶环境感知算法. 计算机系统应用. <http://www.c-s-a.org.cn/1003-3254/8698.html>

Environmental Perception Algorithm for Multi-task Autonomous Driving Based on YOLOv5

ZHANG Kai-Xiang, ZHU Ming

(School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: Autonomous driving tasks are a popular field of deep learning research. As one of the most important modules in autonomous driving, environmental perception includes object detection, lane detection, and drivable area segmentation, which is extremely challenging and has far-reaching significance. Traditional deep learning algorithms usually only solve one detection task in environmental perception and cannot meet the needs of autonomous driving to simultaneously perceive multiple environmental factors. In this study, YOLOv5 is used as the backbone network and object detection branch for lane detection and drivable area segmentation in combination with the real-time semantic segmentation network ENet. Therefore, the environmental perception algorithm for multi-task autonomous driving is achieved, and α -IoU is employed for loss calculation to improve the regression accuracy, which is greatly robust against noise. Experiments show that on the BDD100K data set, the proposed algorithm structure is better than the existing multi-task deep learning networks and can reach a speed of 76.3 FPS on GTX1080Ti.

Key words: multitasking; environment perception; object detection; lane detection; drivable area segmentation; YOLOv5

随着卷积神经网络的快速发展, 自动驾驶越来越多地出现在我们的视野中, 成为研究人员重点研究的领域. 在自动驾驶中, 环境感知是至关重要的, 可以为控制系统提供不可或缺的环境信息, 保证车辆的正常行驶. 自动驾驶的环境感知一般包括目标检测、车道

线检测以及可行驶区域分割. 目标检测主要用于检测行驶道路上的道路交通标志、周边车辆、行人等, 使得车辆可以遵守道路交通规则, 避让行人. 车道线检测是用于提取道路的边缘线, 用于自动驾驶车辆的定位和纠偏, 是平稳安全行驶的重要保障之一. 可行驶区域

^① 基金项目: 科技创新特区计划 (20-163-14-LZ-001-004-01)

收稿时间: 2021-12-24; 修改时间: 2022-01-24; 采用时间: 2022-02-15; csa 在线出版时间: 2022-05-31

分割是将自动驾驶车辆行驶的道路进行全景分割,并检测其中可行驶的区域,前方道路阻塞,障碍物挡道等情况下,可保证车辆行驶安全。

近年来,人工智能技术的革新进一步推动了自动驾驶领域的发展。对于单任务的环境感知方法,目标检测算法通常分为两阶段目标检测和单阶段目标检测^[1],两阶段目标检测算法首先对输入图像提取候选框,再根据第一阶段选定的候选框拟合最后的检测结果,可以更好地处理类别不均衡的问题,检测精度更高,但由于需要进行两阶段计算,速度较慢,无法满足实时检测场景;单阶段目标检测则同时进行分类和检测,直接生成物体类别概率和位置坐标,提高了检测速度,但精度有所下降。基于卷积神经网络的车道线检测通常作为语义分割任务对输入图片进行分类,通过后处理拟合出最后的车道线,但由于车道线像素较少,造成了类别不平衡的问题,降低了检测的准确率。可行驶区域分割一般也是作为语义分割任务,对输入像素进行二分类,从而分割出背景和可行驶区域。

然而自动驾驶场景需要同时获取道路上的多种信息,单个任务的环境感知方法无法满足需求,因此多任务自动驾驶环境感知算法成了开发人员的迫切需要。多任务的环境感知算法通常结合多个任务的特点,设计符合要求的骨干网络,多个分支之间共享参数,从而减少计算量。

基于 YOLOv5 网络^[2],本文提出了一个用于多任务自动驾驶环境感知算法,如图 1 所示,可根据单张输入图片端到端地输出目标检测,车道线检测和可行驶区域分割的结果。编码部分以改进的 YOLOv5 作为骨干网络,解码时目标检测头仍使用 YOLOv5 检测头,车道线检测和可行驶区域分割作为两个语义分割任务共享骨干网络,在解码部分采用实时语义分割网络 ENet 头^[3]进行解码,由于分割任务和目标检测任务存在一定的差异,因此本文为分割任务单独增加一个 FPN 结构^[4],以获得更好的分割效果。



图 1 多任务环境感知算法处理结果

1 相关工作

1.1 单任务环境感知方法

基于卷积神经网络 (CNN) 的目标检测根据有无候选框的生成分为两阶段目标检测和单阶段目标检测,对两阶段目标检测算法, Girshick 等人首次提出用含有 CNN 特征区域进行目标检测的算法 R-CNN^[5],效果显著优于基础传统图像特征的方法,但由于存在大量冗余的特征计算,速度比较低。Lin 等人提出了空间金字塔池化网络^[4],使得 CNN 只需要计算一次特征图,避免了反复计算卷积特征。Ren 等人提出 Faster-R-CNN 检测算法^[6],引入了区域生成网络,是第一个端到端的实时目标检测网络,但其网络的检测部分仍存在冗余计算。对于单阶段目标检测算法 Redmon 等人首次提出单阶段目标检测算法 YOLOv1^[7],完全抛弃了提取候选框再验证的检测范式,而是将图像分割成多个区域,同时预测每个区域的边界框和检测分类,但其对于小物体的定位效果较差。Redmon 等人先后发布了 YOLOv2^[8], YOLOv3^[9],通过引入类似于 ResNet 的残差结构以及 FPN 等操作,提高了检测精度。Bochkovskiy 等人提出 YOLOv4^[10],以 CSPDarknet53 作为骨干网络,引入空间金字塔池化 (SPP) 模块等,大幅度提高了检测的速度和精度。同年, Ultralytics 提出 YOLOv5^[2],通过基于训练数据集自动适应锚点框等不仅减少了模型大小,还大大提高了训练收敛的速度,根据网络不同的深度和宽度实现了 s, l, m, x 四个版本,本文在 YOLOv5s v6.0 的基础上进行相关算法的实现和改进。

可行驶区域分割是以语义分割任务进行处理的,语义分割是计算机视觉的另一个关键任务,Long 等人首次提出全卷积网络 FCN^[11]用于语义分割。随后 Chen 等人^[12]提出 Deeplab 系列的分割网络,提出了空洞卷积扩大了感受野,在空间维度上实现金字塔型的空间池化。Zhao 等人提出 PSPNet^[13],新增金字塔池化模块用于聚合背景并采用了附加损失。由于以上语义分割的速度都达不到实时,Paszke 等人提出实时语义分割网络 ENet^[3],使用 bottleneck 模块以更少的参数量获得更高的精度。

车道线检测方面,基于深度学习的方法常分为基于语义分割的方法和基于行分类的方法。Pan 等人提出基于分割的车道线检测网络 SCNN^[14],设计了一个特殊的 3D 操作,能够更好地获取全局特征。Hou 等人提出基于知识蒸馏的车道线检测算法 SAD,增强了多种

轻量级车道线检测模型如 ENet-SAD^[15], 并在多个数据集取得 SOTA 效果. 为了提高实时性, 降低计算量, Qin 等人提出基于分类的车道线检测算法^[16], 转换为逐行分类任务, 提高了检测速度.

1.2 多任务环境感知方法

多任务环境感知是单个深度学习模型能够同时完成两个或两个以上的检测任务, 不同任务之间通过共享参数等减少计算量, 并且有关联的检测任务相互促进可以达到更好的拟合效果. Teichmann 等人提出基于编解码结构的多任务网络 MultiNet^[17], 可同时处理目标检测, 路面分割和场景分类 3 个任务, 3 个任务共享编码结构, 使用 3 个不同的解码器得到 3 个任务的结果, 但其在目标检测解码阶段缺乏尺度信息, 对于不同大小的物体检测效果一般, 分割部分采用 FCN 结构存在较多冗余计算, 速度较慢. Qian 等人提出 DLT-Net^[18], 在子任务解码器之间构建了上下文张量共享参数, 能够同时处理目标检测, 车道线检测和可行驶区域分割 3 个任务, 但由于这种共享方式也导致该网络在车道线稀疏的场景表现效果不佳. Wu 等人提出实时多任务网络 YOLOP^[19], 也是同时处理 DLT-Net 中的 3 类任务,

只是在参数共享部分, 3 个任务仅共享了编码阶段, 在解码时单独处理, 该网络在 BDD100K 数据集^[20]上取得了 SOTA 效果, 但由于未考虑到分割任务和目标检测任务的差异性, 导致在车道线检测方面的准确率较低.

1.3 本文工作

为了在自动驾驶中同时进行目标检测, 车道线检测和可行驶区域分割, 本文贡献如下: 1) YOLOv5s 作为骨干网络, ENet 作为分割分支设计了多任务自动驾驶环境感知网络; 2) 在车道线检测任务和可行驶区域分割任务前设计独立的 FPN 结构, 以提高车道线检测和可行驶区域分割的准确率; 3) 损失计算时使用更优越的 $\alpha\text{-IoU}$ ^[21] 计算方法. 实验表明, 本文所提算法优于当前流行的多任务深度学习网络 YOLOP 等.

2 算法设计和实现

本节从网络结构, 损失函数两部分进行详细介绍本算法结构.

2.1 网络结构

整个网络的结构如图 2 所示, 网络由特征提取模块, 空间特征池化模块, 检测模块组成.

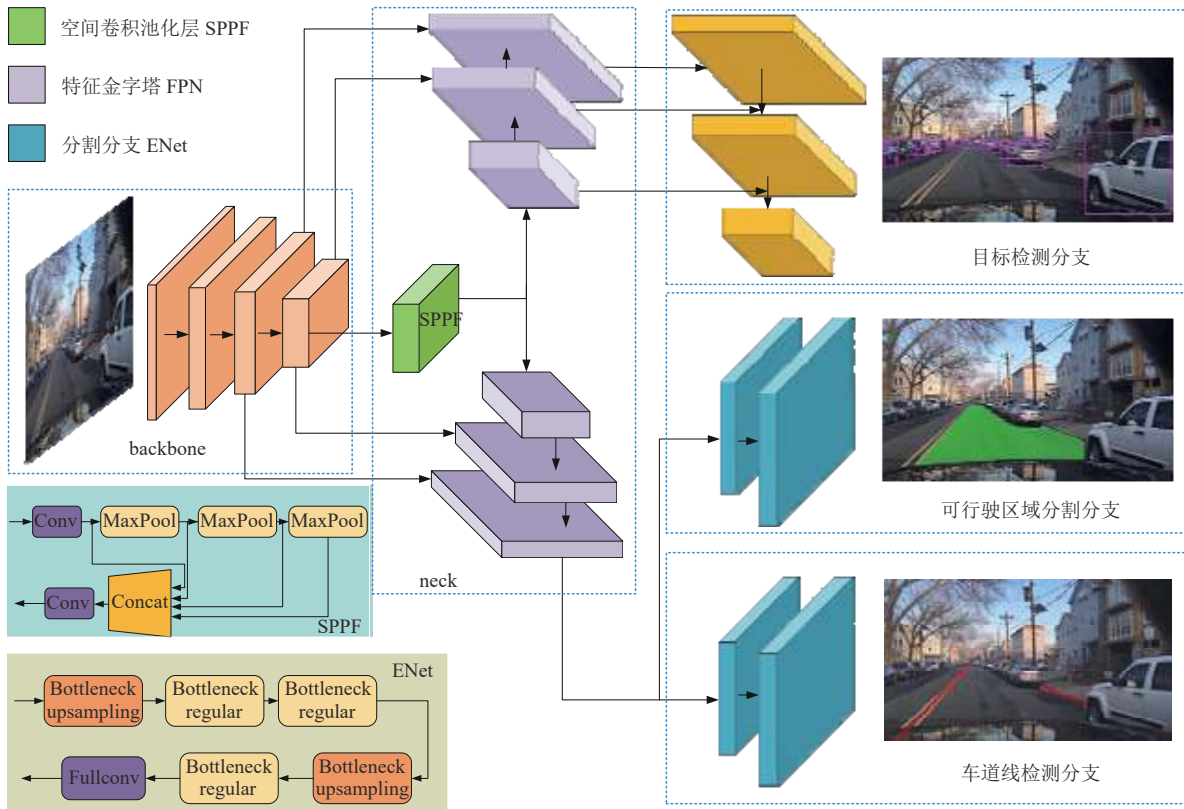


图 2 基于 YOLOv5 的多任务自动驾驶环境感知算法网络结构

(1) 特征提取模块

本文使用 YOLOv5s^[2] 的 CSPDarknet 作为特征提取模块, 以 3 通道的图片作为输入, 经过多层卷积后得到的特征图为原图的 1/32, v6.0 版本 YOLOv5 的特征提取模块有以下特点. 第一是使用一个等效的卷积层替换了 Focus 层, 不仅提高了运算速度, 更方便导出模型; 第二是使用模块 C3 取代之前的 BottleneckCSP, 降低了卷积层的深度, 也取消了 Concat 层之后的卷积层, 以达到更快的速度. 第三是标准卷积中封装了三个模块, 包括卷积 (Conv2d)、批量标准化 (BatchNorm2d)、激活函数, 标准卷积的激活函数使用 SiLU 函数, 该函数的数学表达式如下所示:

$$SiLU(x) = x \cdot Sigmoid(x) = x \frac{1}{1 + e^{-x}} \quad (1)$$

Sigmoid 函数在趋向无穷的地方, 容易造成梯度消失, 不利于深度神经网络的反向传播, 而 SiLU 函数的一阶导数则更加平缓, 更适合提取更深层网络的特征.

(2) 空间特征池化模块

在编码器和解码器之间有一层空间特征池化层, 该层包括快速空间卷积池化层 (SPPF) 和特征金字塔层 (FPN), 如图 2 所示, SPPF 层由两个标准卷积层和 3 个 MaxPool 层以及相加层组成, 相对于原 YOLOv5 的 SPP 层, 将 MaxPool 层的 kernel_size 统一设置为 5, 通过级联的方式连接 3 个 MaxPool 层, 不仅精度未下降, 还大大提高了推理速度. SPPF 层主要用于提取不同尺寸的特征图, 将不同感受野的特征融合, 有利于检测不同尺度的物体. FPN 实现的具体步骤如表 1 所示, 表中以输入大小为 640×640 的 3 通道图片为例, FPN 层的输入是 SPPF 层的输出, 为原尺寸的 1/32, 经过多层卷积以及尺度参数为 2 的上采样层, 经过 Concat 层与编码器不同尺度的特征融合, 最终得到特征金字塔, 用于下游的检测和分割任务. 本文在实验中发现目标检测任务和其他两类分割任务存在明显差异, 目标检测任务关注的更多是全局特征, 而分割任务则更关注像素级特征, 应设计独立的 FPN 结构, 以辅助分割分支更好地获取输入图像特征, 拟合最终的结果.

(3) 检测模块

本文将 3 个任务分别用 3 个检测头来完成, 目标检测任务的检测头和 YOLOv5 保持一致, 将车道线检

测视为分割任务, 与可行驶区域分割任务一样, 都采用 ENet 检测头. 如图 2 所示, 目标检测检测头的输入是自顶向下的 FPN 结构的不同尺度的特征, 再经过一个自底向上的路径聚合网络 (PAN), 最终经过检测头输出不同尺度的计算结果. 为了达到实时检测的效果, 本文的分割任务采用和 ENet 类似的检测头, Bottleneck upsample 层是将卷积和上采样层聚合的模块, Bottleneck regular 层由三层卷积和激活函数以及 Dropout 层构成, Dropout 层用于提高模型泛化能力, 防止过拟合, ENet 的检测头部的输入是 FPN 的最后一层的输出特征, 经过多层 Bottleneck 层后通过全卷积层输出分割结果, 由于本文都是二分类因此最后的特征图大小是 (2, W, H), 分别代表背景类像素和目标像素的概率.

表 1 FPN 网络结构具体实现

输入	结构细节	输出
(512, 20, 20)	BottleneckCSP	(512, 20, 20)
(512, 20, 20)	Conv(kernel_size=1, stride=1)	(256, 20, 20)
(256, 20, 20)	Upsample(scale_factor=2)	(256, 40, 40)
(256, 40, 40)	Concat	(512, 40, 40)
(256, 40, 40)	BottleneckCSP	(256, 40, 40)
(256, 40, 40)	Conv(kernel_size=1, stride=1)	(128, 40, 40)
(128, 40, 40)	Upsample(scale_factor=2)	(128, 80, 80)
(128, 80, 80)	Concat	(256, 80, 80)

2.2 损失函数

本文的损失函数计算包含 3 个分支的部分, 目标检测分支的损失记为 L_{det} , 该部分的损失包含分类损失 L_{class} , 回归损失 L_{box} 和置信度损失 L_{object} , 三类损失加权后构成了目标检测部分的损失, 如下所示:

$$L_{det} = \alpha_1 L_{class} + \alpha_2 L_{box} + \alpha_3 L_{object} \quad (2)$$

分类损失使用二分类交叉熵损失, 计算公式如下:

$$y_i = Sigmoid(x_i) = \frac{1}{1 + e^{-x_i}} \quad (3)$$

$$L_{class} = - \sum_{n=1}^N y_i^* \log(y_i) + (1 - y_i^*) \log(1 - y_i) \quad (4)$$

以上公式考虑到一个物体可能属于多个类别的问题, N 表示类别总数, x_i 表示当前类别的预测值, y_i 代表经过激活函数后的当前类别的概率, y_i^* 表示当前类别的真实值 (0 或 1).

边界框回归损失本文在 YOLOv5 的基础上进行改进, 原 YOLOv5 的回归损失采用的是 $CIoU$, 计算公式

如下所示:

$$CIoU(B, B_{gt}) = IoU(B, B_{gt}) - \frac{\rho^2(B, B_{gt})}{c^2} - \beta v \quad (5)$$

$$v = \frac{4}{\pi} \left(\arctan \frac{\omega^{gt}}{h^{gt}} - \arctan \frac{\omega}{h} \right)^2 \quad (6)$$

$$\beta = \frac{v}{1 - IoU(B, B_{gt}) + v} \quad (7)$$

其中, $\rho(B, B_{gt})$ 为目标框和真实框中心点之间的距离, c 为包住目标框和真实框的最小外接框的对角线长度, v 用来度量目标框宽高比的一致性, β 为权重函数. 本文在 YOLOv5s v6.0 的基础上, 使用 α -IoU 取代 $CIoU$ 计算边界框回归损失, 计算公式如下:

$$L_{\alpha-IoU} = 1 - IoU^\alpha, \alpha \neq 0 \quad (8)$$

$$L_{box} = 1 - IoU^\alpha + \frac{\rho^{2\alpha}(B, B_{gt})}{c^{2\alpha}} + (\beta v)^\alpha \quad (9)$$

通过调节 α , 可以使检测器在不同水平的边界框回归方面具有更大的灵活性, α 大于 1 时有助于提高 IoU 较大的目标回归精度.

车道线检测损失 L_{ll} 和可行驶区域分割损失 L_{da} 都包含交叉熵损失, 借鉴 YOLOP 的损失计算方法, 本文在车道线检测任务中也加入了 IoU 损失, 以提高车道线的检测精度, 最后总体的损失由 3 个任务加权后的损失组成, 计算如下所示:

$$L_{all} = \gamma_1 L_{det} + \gamma_2 L_{da} + \gamma_3 L_{ll} \quad (10)$$

3 实验分析

3.1 数据集

本文测试数据集是 BDD100K, 该数据集是目前最大规模, 内容最具有多样性的公开自动驾驶数据集, 共有 100k 张图片, 其中训练集 70k 张, 验证集 10k 张, 测试集 20k 张, 包含晴天、阴天、雪天、雨天、多云、有雾 6 种不同天气, 以及城市道路、隧道、高速公路、住宅、停车场、加油站 6 种不同场景.

3.2 实现细节

本文实验中训练时的 warmup 设置为 3 个 epoch, 初始的学习率设置为 0.001, 优化器使用 ADAM, $\beta = (0.9, 0.999)$, IoU 阈值为 0.2, anchor 阈值设置为 4.0, 损失函数计算时, α 设置为 3, α_1 - α_3 , γ_1 - γ_3 设置为 1. 因为 BDD100K 的测试集并未完全公开 3 类任务的标签, 因此本文在 70k 张训练集图片上训练, 并在验证集上

验证模型效果. 数据增强方面, 本文主要考虑到光照畸变和几何畸变, 光照畸变本文通过随机改变图片的色调、饱和度和亮度进行增强, 几何畸变本文通过随机旋转、缩放、平移和左右翻转等几何操作进行增强数据集, 以提高模型在不同环境中的鲁棒性. 本实验过程中测试 FPS 时使用的硬件配置如表 2 所示.

表 2 实验所用硬件配置

类型	型号	参数
系统	Ubuntu	16.04.12LTS
CPU	Intel(R) Core(TM) i7-7800X	6核
显存	NVIDIA GeForce GTX 1080Ti	11 GB
内存	DDR4	32 GB

3.3 实验结果

为了验证本算法的有效性, 本文分别从 3 个任务的检测结果与其他多种多任务环境感知算法对比, 本文和当前最好的多任务自动驾驶环境感知算法 YOLOP 的结果对比图如图 3 所示.

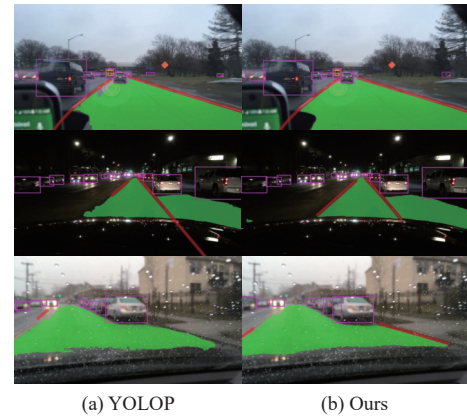


图 3 实验结果对比图

从图 3 中我们可以看到, 在白天, 夜晚以及雨天的情景下, 我们的算法处理结果均优于当前最好的多任务网络 YOLOP. 在第 1 张图片中, 本文的目标检测边界框回归更精准, 而 YOLOP 还出现了误检的情况; 第 2 张图片中, YOLOP 的车道线检测出现了缺失, 过长等情况, 本文结果车道线拟合更准确; 第 3 张图片在雨天的情况下, 本文的结果仍比 YOLOP 的处理结果精度更高.

目标检测方面如表 3 所示, 本文对比了当前流行的多任务深度学习网络 MultiNet^[17], DLT-Net^[18], 以及目前表现 SOTA 的 YOLOP, 除此之外笔者还对比了单

任务目标检测网络 Faster R-CNN^[6] 和 YOLOv5s^[2], 关于速度的测试都是基于显卡 NVIDIA GeForce 1080Ti 进行的, 表中对比 3 个目标检测评价标准召回率、平均精度和帧率, 可以看出本文工作结果在多任务中取得了 SOTA 效果, 检测准确率比当前最好的多任务网络提高了 0.9%, 速度也提高了 0.9, YOLOv5s 的速度虽然比本文快, 但是该网络没有处理分割任务的能力。

表 3 目标检测结果对比

网络 (*为多任务)	Recall (%)	mAP50 (%)	FPS
MultiNet*	81.3	60.2	15.7
DLT-Net*	89.4	68.4	17.1
Faster R-CNN	77.2	55.6	9.7
YOLOv5s	86.8	77.2	150.4
YOLOP*	89.2	76.5	75.2
Ours*	89.3	77.4	76.3

可行驶区域分割的结果如表 4 所示, 在对比时除了 3 个多任务网络还对比了单任务分割网络 PSPNet^[13], 使用平均交并比 (*mIoU*) 来评价网络, 从结果可以看出本文结果在 BDD100K 和其他网络相比表现最佳。

表 4 可行驶区域分割结果对比

网络 (*为多任务)	<i>mIoU</i> (%)	FPS
MultiNet*	71.6	15.7
DLT-Net*	71.3	17.1
PSPNet	89.6	9.7
YOLOP*	91.5	75.2
Ours*	92.2	76.3

车道线检测的结果如表 5 所示, 实验对比了多任务网络 YOLOP 以及一些传统的单任务车道线检测网络 ENet^[3], SCNN^[14], ENet-SAD^[15], 使用准确率和 *IoU* 评价网络的效果, 准确率的定义如下:

$$Accuracy = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

其中, $Precision = \frac{TP}{TP + FP}$, $Recall = \frac{TP}{TP + FN}$, *FP* 是假阳性, *FN* 是假阴性, *TP* 是真阳性. 从表中可以看出本文结果在准确率上提高了 6.91%, 表现最佳。

表 5 车道线结果对比 (%)

网络 (*为多任务)	<i>Accuracy</i>	<i>IoU</i>
ENet	34.12	14.64
SCNN	34.79	15.84
ENet-SAD	36.56	16.02
YOLOP*	70.50	26.20
Ours*	77.41	26.42

为验证本文算法对于复杂场景的有效性, 本文对 BDD100K 数据集中的不同场景、不同时段图片分别计算 3 个任务的检测性能, 结果如表 6 所示, 从表中可以看出本算法在各个场景下均优于 YOLOP。

表 6 不同场景下结果对比 (%)

不同场景	mAP50		<i>mIoU</i>		<i>Accuracy</i>	
	YOLOP	Ours	YOLOP	Ours	YOLOP	Ours
城市道路	76.7	77.8	91.2	92.1	70.2	77.9
隧道	80.2	85.7	91.4	92.5	58.6	67.8
高速公路	71.5	72.8	91.6	92.4	69.3	77.3
住宅	79.6	80.8	91.2	92.4	70.7	78.6
停车场	72.0	72.7	87.5	89.4	44.8	50.2
加油站	65.2	67.4	79.1	81.2	61.5	74.0
白天	77.7	78.5	91.5	92.5	70.8	78.6
夜晚	72.6	74.4	91.1	91.8	68.3	76.2
黄昏	77.2	77.7	91.1	92.1	70.9	79.0

3.4 消融实验

为了验证本文所提网络中用到的模块的有效性, 本文做了如下消融实验, 采用普通的 *IoU* 计算方式和去除分割分支的独立 FPN 结构进行实验, 实验结果如表 7 所示。

表 7 消融实验结果对比 (%)

网络	mAP	<i>mIoU</i>	<i>Accuracy</i>
baseline	76.0	91.5	68.50
baseline+ α - <i>IoU</i>	77.6	91.3	70.52
baseline+ α - <i>IoU</i> +FPN	77.4	92.2	77.41

表 7 中网络 baseline 在损失函数计算时使用一般的 *CIoU* 计算方式, 分割分支和目标检测分支共用一个特征金字塔结构, 目标检测使用 mAP 评价, 可行驶区域分割使用 *mIoU* 评价, 车道线检测使用准确率进行评价. 从表 7 中结果我们可以看到, α -*IoU* 提升了目标检测准确率, 而独立 FPN 结构则有效提升了分割分支的检测效果。

4 结论与展望

本文基于 YOLOv5 设计了一个多任务自动驾驶环境感知算法, 可以同时处理目标检测, 可行驶区域分割, 车道线检测 3 个任务, 并且在 1080Ti 显卡上达到了 76.3 FPS, 在 BDD100K 数据集上与其他多任务网络相比取得了最佳的效果. 下一步的研究工作将在本文的基础上结合车道线检测的特点, 增加一些后处理操作, 进一步提高车道线检测的准确率, 构建一个工业级的多任务自动驾驶环境感知算法。

参考文献

- 1 Zhao ZQ, Zheng P, Xu ST, *et al.* Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 2019, 30(11): 3212–3232. [doi: [10.1109/TNNLS.2018.2876865](https://doi.org/10.1109/TNNLS.2018.2876865)]
- 2 Github. YOLOv5. <https://github.com/ultralytics/yolov5>.
- 3 Paszke A, Chaurasia A, Kim S, *et al.* ENet: A deep neural network architecture for real-time semantic segmentation. arXiv: 1606.02147, 2016.
- 4 Lin TY, Dollár P, Girshick R, *et al.* Feature pyramid networks for object detection. *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 936–944. [doi: [10.1109/CVPR.2017.106](https://doi.org/10.1109/CVPR.2017.106)]
- 5 Girshick R, Donahue J, Darrell T, *et al.* Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus: IEEE, 2014. 580–587. [doi: [10.1109/CVPR.2014.81](https://doi.org/10.1109/CVPR.2014.81)]
- 6 Ren SQ, He KM, Girshick R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems*. Montreal: MIT Press, 2015. 91–99.
- 7 Redmon J, Divvala S, Girshick R, *et al.* You only look once: Unified, real-time object detection. *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas: IEEE, 2016. 779–788. [doi: [10.1109/CVPR.2016.91](https://doi.org/10.1109/CVPR.2016.91)]
- 8 Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 6517–6525. [doi: [10.1109/CVPR.2017.690](https://doi.org/10.1109/CVPR.2017.690)]
- 9 Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv: 1804.02767, 2018.
- 10 Bochkovskiy A, Wang CY, Liao HYM. Yolov4: Optimal speed and accuracy of object detection. arXiv: 2004.10934, 2020.
- 11 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition*. Boston: IEEE, 2015. 3431–3440. [doi: [10.1109/CVPR.2015.7298965](https://doi.org/10.1109/CVPR.2015.7298965)]
- 12 Chen LC, Papandreou G, Kokkinos I, *et al.* DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 40(4): 834–848. [doi: [10.1109/TPAMI.2017.2699184](https://doi.org/10.1109/TPAMI.2017.2699184)]
- 13 Zhao HS, Shi JP, Qi XJ, *et al.* Pyramid scene parsing network. *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 6230–6239. [doi: [10.1109/CVPR.2017.660](https://doi.org/10.1109/CVPR.2017.660)]
- 14 Pan XG, Shi JP, Luo P, *et al.* Spatial as deep: Spatial CNN for traffic scene understanding. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. Palo Alto: AAAI, 2018. 7276–7283.
- 15 Hou YN, Ma Z, Liu CX, *et al.* Learning lightweight lane detection CNNs by self attention distillation. *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision*. Seoul: IEEE, 2019. 1013–1021.
- 16 Qin ZQ, Wang HY, Li X. Ultra fast structure-aware deep lane detection. *Proceedings of the 16th European Conference on Computer Vision*. Glasgow: Springer, 2020. 276–291. [doi: [10.1007/978-3-030-58586-0_17](https://doi.org/10.1007/978-3-030-58586-0_17)]
- 17 Teichmann M, Weber M, Zöllner M, *et al.* MultiNet: Real-time joint semantic reasoning for autonomous driving. 2018 IEEE Intelligent Vehicles Symposium (IV). Changshu: IEEE, 2018. 1013–1020.
- 18 Qian YQ, Dolan JM, Yang M. DLT-Net: Joint detection of drivable areas, lane lines, and traffic objects. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 21(11): 4670–4679. [doi: [10.1109/TITS.2019.2943777](https://doi.org/10.1109/TITS.2019.2943777)]
- 19 Wu D, Liao MW, Zhang WT, *et al.* YOLOP: You only look once for panoptic driving perception. arXiv: 2108.11250, 2021.
- 20 Yu F, Xian WQ, Chen YY, *et al.* Bdd100k: A diverse driving video database with scalable annotation tooling. arXiv: 1805.04687, 2018.
- 21 He J, Erfani S, Ma X, *et al.* Alpha-IoU: A family of power intersection over union losses for bounding box regression. *Proceedings of the 35th Conference on Neural Information Processing Systems*. 2021. 34.

(校对责编: 牛欣悦)