



DOI:10.12404/j.issn.1671-1815.2405217

引用格式:余军军,严运兵,田茂帅.基于YOLOv8s改进的车辆前方障碍物轻量化检测算法[J].科学技术与工程,2025,25(14):5957-5966.

Yu Junjun, Yan Yunbing, Tian Maoshuai. Lightweight front vehicle obstacle detection algorithm based on improved YOLOv8s[J]. Science Technology and Engineering, 2025, 25(14): 5957-5966.

基于 YOLOv8s 改进的车辆前方障碍物 轻量化检测算法

余军军,严运兵*,田茂帅

(武汉科技大学汽车与交通工程学院,武汉 430065)

摘要 为解决自动驾驶感知域控制器中障碍物检测模型对高内存和高计算资源需求的问题,提出了一种基于 YOLOv8 改进的轻量化障碍物检测方法,使用内存访问和计算量更少的 FasterNet 重构 YOLOv8 主干网络。为弥补模型轻量化导致的精度下降以及对小目标检测能力的不足,主要在 3 个方面对 YOLOv8 进行改进:用 SPD-Conv (space-to-depth convolution) 替换颈部网络的传统跨步卷积,增强小目标特征提取能力;结合 IIoU (inner IoU) 和 PIoU (powerful IoU) 的思想,提出 IPIoU (inner powerful IoU) 作为边框回归损失,加快损失函数收敛并提高小目标检测性能;引入注意力机制 SimAM (simple attention module),进一步提高模型检测精度。实验结果表明,改进模型相比原模型在参数量、计算量和模型大小分别降低 29.1%、20.5% 和 28.8% 的情况下,检测精度提升了 1.2%。模型部署至自动驾驶车载控制器后,能够有效检测道路前方障碍物。

关键词 障碍物检测;YOLOv8;网络轻量化;FasterNet

中图分类号 TP391.41; **文献标志码** A

Lightweight Front Vehicle Obstacle Detection Algorithm Based on Improved YOLOv8s

YU Jun-jun, YAN Yun-bing*, TIAN Mao-shuai

(School of Automobile and Traffic Engineering, Wuhan University of Science and Technology, Wuhan 430065, China)

[Abstract] To solve the problem of high memory and computational resource demands in obstacle detection models within autonomous driving perception domain controllers, a lightweight obstacle detection method based on improved YOLOv8 was proposed. This method reconstructs the YOLOv8 backbone network using FasterNet, which utilizes less memory access and computational resources. To mitigate the accuracy decline and the insufficient detection capabilities for small objects caused by model lightweighting, three main improvements were made to YOLOv8: SPD-Conv (space-to-depth convolution) was used to replace traditional stride convolution in the neck network to enhance small object feature extraction. IPIoU (inner powerful IoU), combining the concepts of IIoU (inner IoU) and PIoU (powerful IoU), is introduced as the bounding box regression loss to accelerate loss convergence and improve small object detection performance. SimAM (simple attention module) was incorporated to further enhance model detection accuracy. Experimental results demonstrate that, compared to the original model, the improved model achieves a reduction of 29.1% in parameters, 20.5% in computational load, and 28.8% in model size, while increasing mAP@0.5 by 1.2%. Once deployed in autonomous driving vehicle controllers, the model effectively detects obstacles on the road ahead.

[Keywords] obstacle detection; YOLOv8; network lightweighting; FasterNet

基于深度学习的障碍物检测是实现自动驾驶技术的研究热点^[1]。目前,自动驾驶车辆常利用相机和雷达进行障碍物检测。基于雷达数据的深度学习目标检测已取得极大进展,但是雷达分辨率较低且语义表达能力偏弱,这可能导致对障碍物的误判或漏判。相较之下,视觉传感器在分辨率、检测范围及语义信息表达方面表现更优;此外,雷达价

格偏高,不利于其在自动驾驶中的产业化应用。

深度学习方法主要分为二阶段和单阶段检测算法。二阶段检测算法检测精度较高,但检测速率无法满足实时环境感知的要求^[2]。单阶段检测算法直接进行目标定位和识别,检测速率更高,平衡了效率与精度。常见单阶段检测算法有 SSD (single shot multiBox detector)、YOLO (you only look once) 系

收稿日期:2024-07-11; 修订日期:2025-02-27

基金项目:国家自然科学基金(51975428)

第一作者:余军军(1999—),男,汉族,四川达州人,硕士研究生。研究方向:无人驾驶环境感知。E-mail:1986546788@qq.com。

*通信作者:严运兵(1968—),男,汉族,湖北武汉人,博士,教授。研究方向:汽车动力学及其控制,无人驾驶感知及规划。E-mail:yyb@wust.edu.cn。

投稿网址:www.stae.com.cn

列^[3-4]、Centernet 等。其中, YOLO 系列算法已广泛应用于无人驾驶领域。郭克友等^[5]提出 Dim env-YOLO 车辆检测算法, 通过用 MobileNetv3 替换 YOLOv4 主干网络实现轻量化, 并用图像暗光增强技术提高低光照环境下的检测性能; Zaghari 等^[6]在 YOLOv3 基础上提出模糊逻辑非极大值算法, 缓解了重叠障碍物检测准确性较低的问题; Mahaur 等^[7]引入不同扩张率的空洞卷积到 SPP (spatial pyramid pooling) 模块, 并在特征金字塔网络中添加越层连接, 提高了 YOLOv5 对小目标的检测精度和速度。

截至目前, 在中外研究者的共同努力下, 基于视觉的交通目标检测已经达到了较高的精度。但是, 针对复杂交通应用场景下的道路障碍物检测仍存在以下不足。

(1) 在复杂交通环境中, 自动驾驶域控制器的计算资源和存储空间有限, 高精度障碍物检测网络因结构复杂和参数过多而需要高性能的芯片和足够大的存储空间; 而小规模网络存储和计算需求低但精度不足, 难以确保行车安全。因此, 兼顾精度和模型大小的障碍物检测网络具有重要的研究意义。

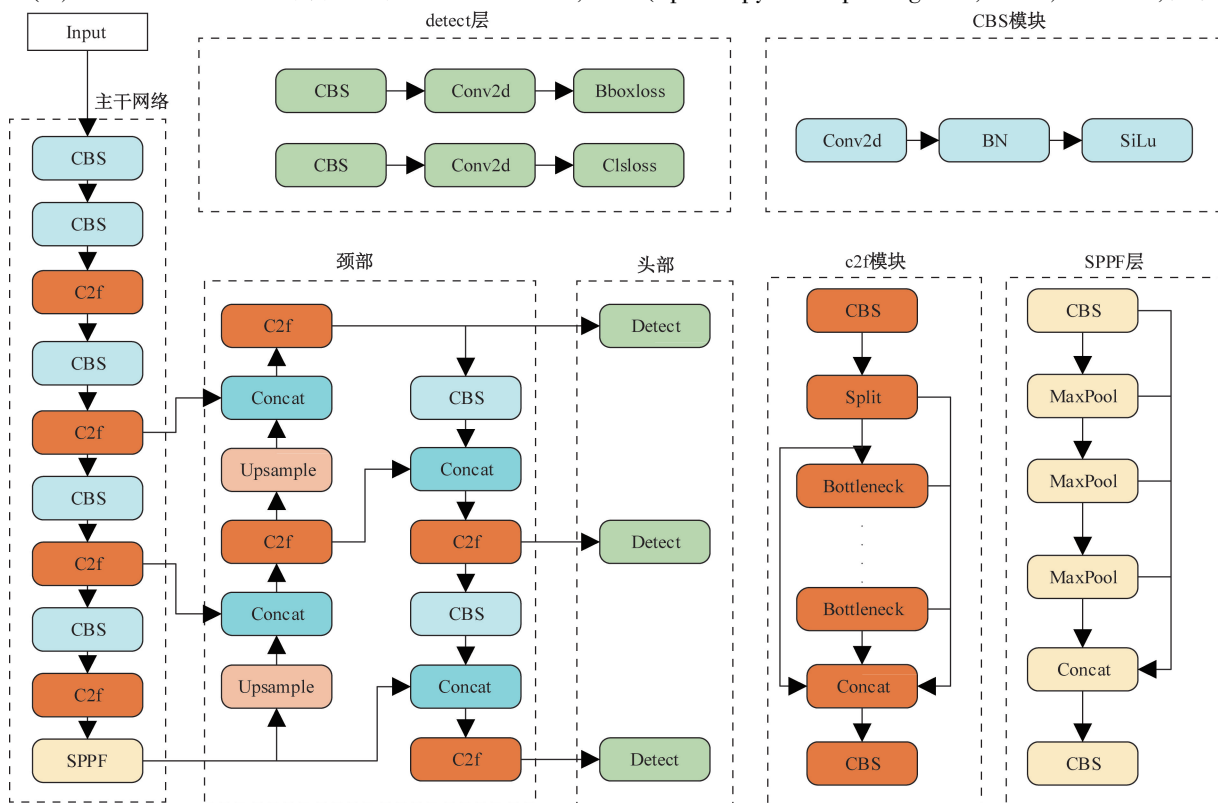
(2) 现有网络对小目标障碍物易漏检和误检,

而自动驾驶车辆需提前准确识别远处小目标以提升交通效率和行车安全性。故提升模型对小目标的检测能力至关重要。

YOLOv8 是 YOLO 系列中较为先进的模型, 做到了检测精度和速度的进一步平衡, 官方数据显示, YOLOv8 在各公开数据集上的检测精度相较于前作 YOLOv5 有不小提升。考虑到 YOLOv5 已广泛应用于自动驾驶 2D 目标检测^[8], YOLOv8 作为其改进版本, 精度和速度有显著提升, 对车辆前方障碍物的判断更为准确。为此, 现选择 YOLOv8 作为基础道路障碍物检测模型。一方面, 以模型轻量化、降低计算资源消耗为研究目标; 另一方面, 着重解决模型轻量化改进后精度下降以及对小目标障碍物检测精度不高的问题。

1 YOLOv8 网络结构

YOLOv8 是 YOLO 系列中较为先进的模型, 由主干网络 (Backbone)、颈部 (Neck) 和检测头 (Head) 组成, 其结构如图 1 所示。YOLOv8 的主干网络为 CSPDarknet, 由 CBS 模块、C2f 模块和 SPPF (spatial pyramid pooling fast, SPPF) 层构成, 负责提



Input 为输入层; Detect 为检测头; CBS 为二维卷积归一化层; C2f 为跨阶段融合层; SPPF 为快速空间金字塔池化层; Upsample 为上采样层; Concat 为特征图拼接层; Conv2d 为二维卷积层; Bboxloss 和 Clsloss 分别为边界框和分类损失; BN 为批归一化; SiLu 为激活函数; Split 为特征图分割; MaxPool 为最大池化层

图 1 YOLOv8 网络结构

Fig. 1 YOLOv8 network architecture

取目标的特征信息, CBS 模块包含 3×3 的卷积、批归一化和 SiLU 激活函数; C2f 模块在 C3 模块基础上增加了 Split 操作和越层连接, 以获得更为丰富的梯度流信息; SPPF 模块通过不同尺度的池化融合特征图, 提升特征提取能力。颈部采用 PAN (path aggregation network) + FPN (feature pyramid network) 特征融合方式, 是自顶向下和自下向上越层连接。检测头部分采用解耦头和无锚框策略, 解耦头将检测和分类分离, 减少了参数和计算复杂度; 无锚框策略直接预测目标位置, 提高了检测器的速度和灵活性^[9]。

为平衡精度和速度, 现选择 YOLOv8s 作为基础模型, 但应用过程中发现如下问题需要解决: 首先, 主干网络存在冗余卷积层, 结构复杂, 参数量和计算量过多, 不适用于资源受限的自动驾驶控制器; 其次, 颈部的跨步卷积可能丢失小目标障碍物的细节信息, 降低模型对小目标的检测能力; 最后, YOLOv8s 使用的 CIoU (complete IoU) 泛化能力较低, 并且忽略了真实框和预测框之间的形状差异, 从而导致损失函数收敛缓慢, 模型检测精度不佳。基于此, 提出了对 YOLOv8s 进行改进。

2 对 YOLOv8s 的改进

2.1 重构主干特征提取网络

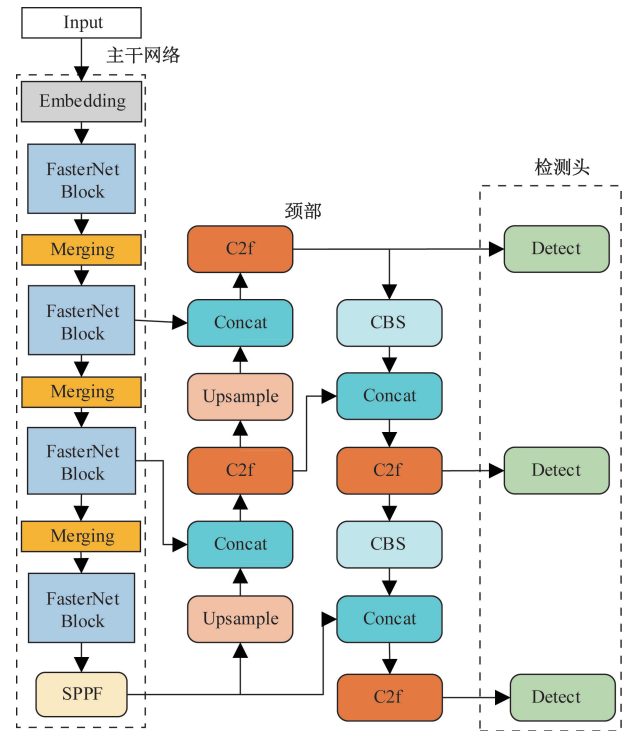
YOLOv8s 原始主干特征提取网络虽然通过局部跨阶段连接减少了模型计算量, 但仍存在冗余卷积和网络结构过于复杂的缺点, 不能适用于计算资源有限的自动驾驶域控制器。目前, 研究人员往往采用 MobileNet、ShuffleNet^[10]、EfficientNet 等轻量化网络替换原 YOLOv8s 主干网络。这些网络通常采用群混洗卷积 (grouped spatial convolution, GSConv)、幻影卷积 (ghost convolution, GC)、深度可分离卷积 (depthwise separable convolution, DSC)、分组卷积 (group convolution, GrC) 等来提取目标特征, 以此降低模型计算量。然而, 这些轻量化卷积会产生频繁的内存访问, 致使模型实时性变差。与此同时, 在连接、池化、混洗的过程中会引起额外的数据处理操作, 增加了模型推理时间, 使模型延迟增大。

鉴于此, 本文研究引入一种全新的轻量级网络 FasterNet^[11], 该网络在 GPU、ARM 和 CPU 等设备上对比其他轻量级网络推理延迟更低、检测精度更高, 图 2 为重构主干特征提取网络之后的 YOLOv8s 结构。

FasterNet 主要由嵌入层 (embedding layer)、融合层 (merging layer) 和 FasterNet Block 模块组成。嵌入层和融合层分别由 4×4 和 3×3 的卷积组成,

其作用是下采样和扩充通道数。如图 3 所示, FasterNet Block 模块由一个部分卷积 (partial convolution, PConv) 和两个逐点卷积 (point-wise convolution, PWConv) 组成, 并由这三部分组成一个倒置残差块, FasterNet Block 是负责特征提取的主要模块。

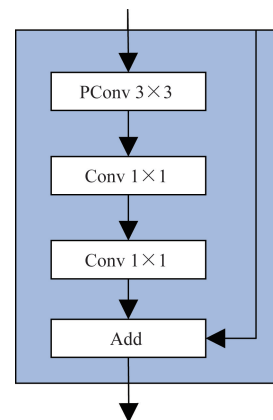
图 4 为部分卷积原理图, 部分卷积通过对输入特征图的最前段或最后段连续 c_p 个通道应用常规卷积来进行空间特征提取, 而其余通道保持不变。



Embedding 为嵌入层; Merging 为融合层; FasterNet Block 为特征提取模块

图 2 重构主干网络后的 YOLOv8s 结构图

Fig. 2 Reconstructed backbone network structure of YOLOv8s



PConv 3×3 为 3×3 部分卷积; Conv 1×1 为 1×1 卷积; Add 为特征图相加操作

图 3 FasterNet Block 结构

Fig. 3 FasterNet Block structure

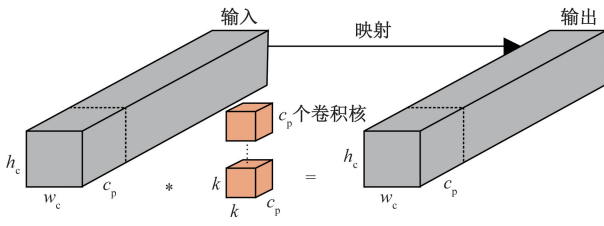


图4 部分卷积原理
Fig. 4 Principle of PConv

PConv 内存访问量 M 为

$$M = h_c w_c 2c_p + k^2 c_p^2 \quad (1)$$

式(1)中: h_c 和 w_c 为特征通道的高和宽; c_p 为参与常规卷积运算的通道数; k 为卷积核的大小。

若 c_p 为输入特征通道数的 1/4, 则部分卷积的内存访问量约为常规卷积的 1/4。

PConv 的计算量 F 为

$$F = h_c w_c k^2 c_p^2 \quad (2)$$

若 c_p 为输入特征通道数的 1/4, 则部分卷积的计算量仅为常规卷积的 1/16。因此, 采用 FasterNet 作为特征提取主干网络, 可以大幅削减模型的参数和计算量, 同时降低了模型的推理延迟。FasterNet 有多个不同版本, FasterNet_t1 在大幅减少模型参数和计算量的同时, 保证了模型的表达能力, 检测精度下降的幅度较小, 因此本文研究选择 FasterNet_t1 作为 YOLOv8 的主干网络。

2.2 用于小目标及低分辨率的 SPD-Conv

小目标易被漏检或误检, 威胁行车安全; 其低分辨率限制了模型学习小目标上下文信息的能力。为提升模型对数据集中远处小目标的检测性能, 在 YOLOv8s 的颈部网络引入了 SPD-Conv (space-to-depth convolution)^[12]。

SPD-Conv 模块由空间到深度 (space-to-depth, SPD) 层和非跨步卷积 (non-strided convolution, NSConv) 层构成, 原始输入图片在经过一系列特征提取后, 得到特征图 X , 特征图 X 经过 SPD 层, 转换为中间特征图 X' , 最后由 NSConv 进行滤波得到最终的特征图 X'' , 其原理如图 5 所示。

图 5 (a) 和图 5 (b) 表示在下采样因子 $scale = 2$ 的情形下, SPD 层将大小为 (S, S, C) 的输入特征图 X 按照一定规律沿 x 轴和 y 轴切割, 得到图 5 (b) 中 4 个大小为 $(S/2, S/2, C)$ 的子特征图, 这个过程相当于一次下采样操作。然后将这 4 个子特征图沿着通道维度拼接得到中间特征图 X' , X' 相较于原始特征图 X , 空间维度减少为原来的一半, 而通道维度增加至原来的 4 倍, 因此 SPD 层可以将原始输入大小为 (S, S, C) 的特征图转换为具有特征信息且大小为 $(S/2, S/2, 4C)$ 的特征图。图 5 (c)

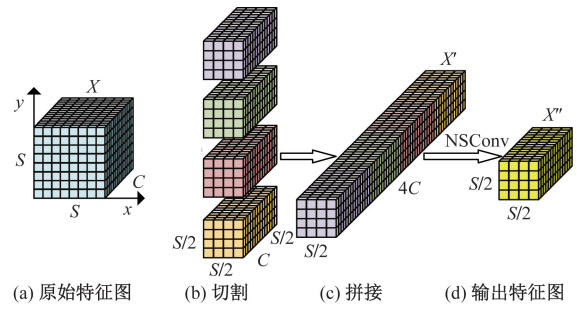


图5 SPD-Conv 结构
Fig. 5 SPD-Conv structure

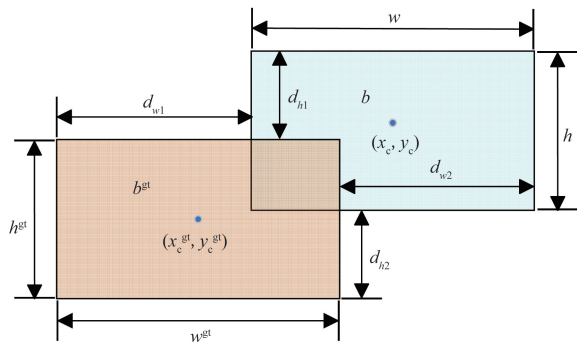
和图 5 (d) 为 NSConv 层操作, 即用一个步长为 1 的卷积层将 X' 转换为最终特征图 X'' , 其大小为 $(S/2, S/2, C_1)$ 。

在 YOLOv8s 颈部网络中, 位于 C2f 模块和 Concat 层之间的 CBS 模块 (卷积核步长为 2) 也可以实现特征图 X 到 X'' 的转变, 但不会生成中间特征图 X' , 进而导致特征信息丢失。SPD-Conv 可以在减小特征图尺寸的同时保留细粒度信息, 适用于低分辨率图像和小目标检测, 并避免了传统跨步卷积下采样丢失关键信息。因此, 将上述 CBS 模块中的传统卷积替换为 SPD-Conv (scale = 2, 卷积核大小为 3, 步长为 1) 模块, 改进后的 CBS 模块被命名为 SPD-CBS。

2.3 损失函数改进

YOLOv8s 损失函数由边框回归损失和分类损失组成, 边框回归的目的是利用检测器输出的预测框 b 进行细微调整而逐步逼近真实框 b^{gt} , 真实框与预测框的关系如图 6 所示。

交并比 (intersection over union, IoU) 为真实框和预测框之间的交集与并集之比, 现已成为目标检



b^{gt} 为真实框; h^{gt} 和 w^{gt} 分别为真实框的高和宽; x_c^{gt} 和 y_c^{gt} 分别真实框的中心点坐标; b 为预测框; h 和 w 分别为预测框的高和宽; x_c 和 y_c 分别为预测框的中心点坐标; d_{w1} 、 d_{w2} 和 d_{h1} 、 d_{h2} 为预测框和真实框对应边之间的距离

图6 真实框和预测框的关系图
Fig. 6 Relationship diagram between ground truth box and prediction box

测领域边框回归损失的主流评价标准, IoU 计算公式为

$$\text{IoU} = \frac{b \cap b^{\text{gt}}}{b \cup b^{\text{gt}}} \quad (3)$$

现今大多数边框回归损失函数都是在传统 IoU 的基础上添加新的损失项来加速边框回归以提升检测精度, 如 DIoU (distance IoU)、SIoU (soft IoU)、CIoU、GIoU (generalized IoU) 和 EIoU (efficient IoU) 等, YOLOv8s 采用的便是 CIoU, CIoU 计算公式为

$$L_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2(b, b^{\text{gt}})}{c^2} + \alpha v \quad (4)$$

式(4)中:

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{\text{gt}}}{h^{\text{gt}}} - \arctan \frac{w}{h} \right)^2 \quad (5)$$

$$\alpha = \frac{v}{v + (1 - \text{IoU})} \quad (6)$$

式中: $\rho^2(b, b^{\text{gt}})$ 为预测框 b 和真实框 b^{gt} 中心点之间的欧式距离的平方; c 为能够同时覆盖预测框和真实框的最小矩形的对角线距离; v 为两个矩形框的宽高比相似度; α 为 v 的影响因子; w 和 h 分别为预测框的宽度和高度; w^{gt} 和 h^{gt} 分别为真实框的宽度和高度。

CIoU 在 IoU 的基础上引入归一化的中心点距离来衡量真实框和预测框中心点的相对位置, 并通过宽高比的一致性来考虑形状差异, 检测效果得到有效提升。然而, CIoU 仍是在传统 IoU 的基础上通过添加新的损失项来加速收敛, 忽略了传统 IoU 本身在不同数据集和检测任务中泛化能力较低的问题。IIoU (inner IoU)^[13] 使用辅助边框来计算 IoU, 针对不同的数据集调整缩放因子 r , 以提高模型的泛化能力。IIoU 的计算公式为

$$\begin{cases} b_l^{\text{gt}} = x_c^{\text{gt}} - \frac{w^{\text{gt}}r}{2} \\ b_r^{\text{gt}} = x_c^{\text{gt}} + \frac{w^{\text{gt}}r}{2} \end{cases} \quad (7)$$

$$\begin{cases} b_t^{\text{gt}} = y_c^{\text{gt}} - \frac{h^{\text{gt}}r}{2} \\ b_b^{\text{gt}} = y_c^{\text{gt}} + \frac{h^{\text{gt}}r}{2} \end{cases} \quad (9)$$

$$\begin{cases} b_l = x_c - \frac{wr}{2} \\ b_r = x_c + \frac{wr}{2} \end{cases} \quad (9)$$

$$\begin{cases} b_t = y_c - \frac{hr}{2} \\ b_b = y_c + \frac{hr}{2} \end{cases} \quad (10)$$

$$\text{inter} = [\min(b_r^{\text{gt}}, b_r) - \max(b_l^{\text{gt}}, b_l)] \times [\min(b_b^{\text{gt}}, b_b) - \max(b_t^{\text{gt}}, b_t)] \quad (11)$$

$$\text{union} = (w^{\text{gt}}h^{\text{gt}})r^2 + (wh)r^2 - \text{inter} \quad (12)$$

$$\text{IoU}^{\text{inner}} = \frac{\text{inter}}{\text{union}} \quad (13)$$

式中: $r \in [0.5, 1.5]$, 是一个缩放因子, 用于控制辅助边框相对于真实边界框的尺寸; x_c 和 y_c 分别为预测框的中心点坐标; x_c^{gt} 和 y_c^{gt} 分别为真实框中心点的坐标。

从式(7)~式(13)可知, IIoU 先利用真实框和预测框的中心点坐标以及长宽计算出辅助边框角顶点坐标, 再分别计算出辅助边框的交集和并集, 进而得出 $\text{IoU}^{\text{inner}}$ 。

CIoU 新添的损失项未直接考虑两框之间形状差异对收敛速度的影响, 且未使用真实框尺寸作为损失计算, 导致预测框对自身尺寸调整不敏感, 影响小目标检测精度。Liu 等^[14] 提出 PIoU (powerful IoU), 利用与真实框相适应的惩罚因子 P 控制预测框大小, 引导其快速有效地回归, 提高对小目标及重叠目标的检测性能。PIoU 计算公式为

$$L_{\text{PIoU}} = 1 - \text{IoU} + (1 - e^{-P^2}) \quad (14)$$

式(14)中:

$$P = \frac{d_{w_1} + d_{w_2}}{4w_{\text{gt}}} + \frac{d_{h_1} + d_{h_2}}{4h_{\text{gt}}} \quad (15)$$

为解决不同质量预测框对边框回归速度的影响, PIoU 引入一个非单调注意力层, 加强对中等质量预测框的梯度关注, 减少对低质量预测框的梯度关注, 从而加快损失函数收敛。引入注意力函数的 PIoU_v2 公式, 即

$$L_{\text{PIoU}_v2} = u(\lambda q) L_{\text{PIoU}} \quad (16)$$

式(16)中:

$$u(\lambda q) = 3\lambda q e^{-(\lambda q)^2} \quad (17)$$

$$q = e^{-P}, q \in (0, 1] \quad (18)$$

式中: $u(\lambda q)$ 为注意力函数; q 为以 P 为变量的惩罚因子, 用于衡量预测框的质量; λ 为控制注意力函数行为的超参数。

因此, 将 IIoU 的思想结合 PIoU_v2, 提出 IPIoU (inner powerful IoU), ($r=0.78, \lambda=1.33$) 并替换原 YOLOv8s 中的 CIoU, IPIoU 的表达式为

$$L_{\text{IP-IoU}} = L_{\text{PIoU}_v2} + \text{IoU} - \text{IoU}^{\text{inner}} \quad (19)$$

2.4 引入 SimAM 注意力机制

现有的注意力模块如 SE、CBAM、ECA^[15] 等主要沿通道或空间维度优化, 生成的权重对所有通道或位置同等对待。通道注意力区分通道但同一通道内权重相同; 空间注意力关注位置但对所有

通道权重统一,这可能限制特征区分能力。SimAM^[16]采用三维注意力机制,为每个神经元分配唯一权重,增强对特征图的感知,能更好捕捉复杂特征的交互。SimAM 注意力模块结构如图 7 所示。

在 SimAM 中,为评估各神经元的重要性,每个神经元被定义了能量函数 e_i ,如式(20)所示,其解析式如式(21)和式(22)所示。

$$e_i(w_i, b_i, y, x_i) = (y_i - \hat{t})^2 + \frac{\sum_{i=1}^{M-1} (y_0 - \hat{x}_i)^2 + \lambda w_i^2}{M-1} \quad (20)$$

$$w_i = -\frac{2(t - \mu_i)}{(t - \mu_i)^2 + 2\sigma_i^2 + 2\lambda} \quad (21)$$

$$b_i = -\frac{1}{2}(t + \mu_i)w_i \quad (22)$$

$$\sigma_i^2 = \frac{1}{M-1} \sum_{i=1}^{M-1} (x_i - \mu_i)^2 \quad (23)$$

$$\mu_i = \frac{1}{M-1} \sum_{i=1}^{M-1} x_i \quad (24)$$

式中: w_i 和 b_i 为权重和偏置项; y 为目标神经元期望输出; x_i 为输入神经元; y_0 和 y_0 为目标神经元的期望输出和其他神经元的期望输出; \hat{t} 和 \hat{x}_i 为目标神经元和第 i 个其他神经元在相同通道上的输入特征 t 和 x_i 关于 w_i 和 b_i 的线性变换; M 为在该通道中的神经元总数; μ_i 和 σ_i^2 分别为目标神经元所在通道上除了目标神经元外所有神经元的平均值和方差; λ 为正则化系数。

SimAM 注意力模块的另一个优点是直接利用输入特征的内在特性计算注意力权重,无需学习额外参数,简化了模型复杂性,提高了效率和通用性。表 1 展示了不同注意力模块的参数量对比。

综上,改进后的网络结构如图 8 所示。

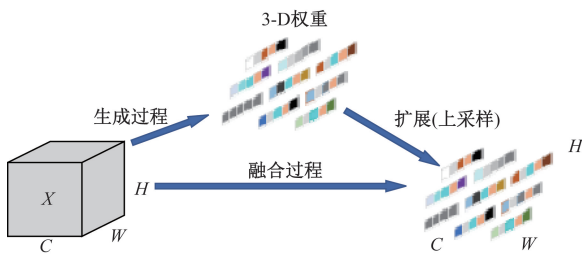
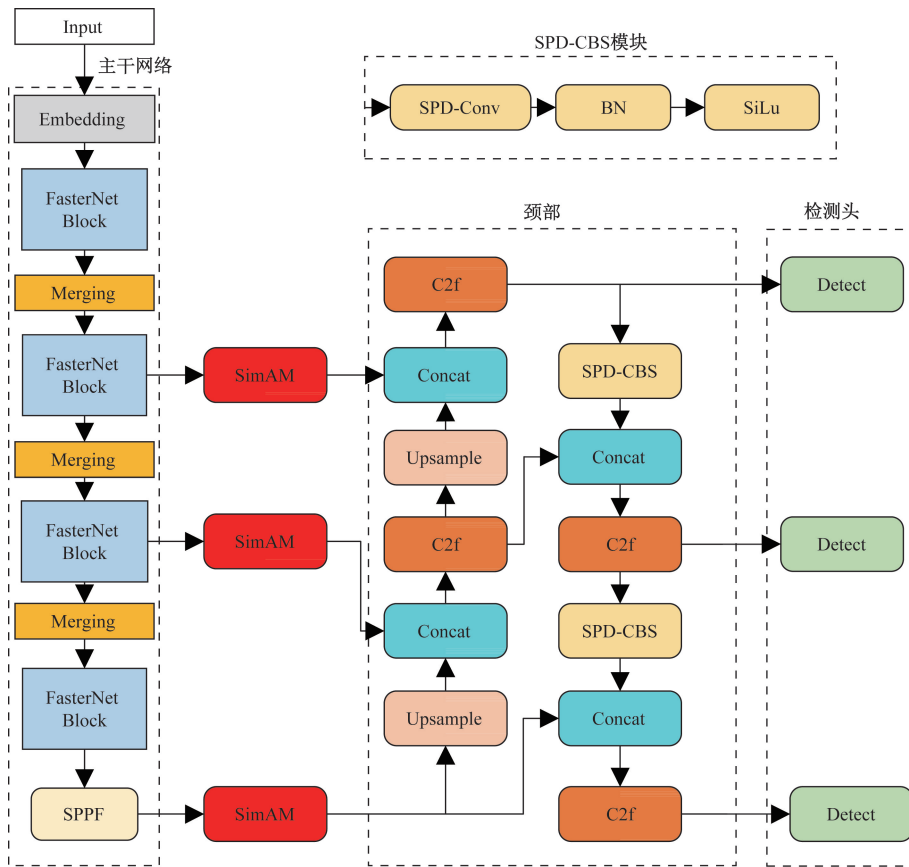


图 7 SimAM 结构
Fig. 7 SimAM Structure



SimAM 为三维注意力机制;SPD-CBS 为融合 SPD-Conv 的 CBS 模块

图 8 改进后的网络结构图

Fig. 8 Improved network structure diagram

表 1 不同注意力模块参数量

Table 1 Parameters of different attention modules

模块	SE	CBAM	ECA	SimAM
参数	$2C^2/r$	$2C^2/r$	$2C^2/r + 2k^2$	0

注: C 为当前通道数; k 和 r 分别为卷积核的数量和卷积的降维率。

3 实验验证及分析

3.1 实验数据集与实验环境

实验使用 KITTI 数据集, 将 Car、Van、Truck 和 Tram 合并为 Vehicle 类, Pedestrian 和 Pedestrian (sitting) 合并为 Pedestrian 类, 忽略 DontCare 类。由于 KITTI 数据集中 Cyclist 和 Pedestrian 较少, 实验中对这两类进行扩充, 减少漏检和误检风险。同时, 添加 cone-shaped-barrel 类应对特殊场景如校园和小区的交通锥需求。数据集共 9 651 张图片, 分为训练集、验证集和测试集, 按 8:1:1 比例划分, Vehicle、Pedestrian、Cyclist 和 cone-shaped-barrel 类实例个数分别为 27 842、2 998、4 821 和 2 943。

本文实验硬件环境为 Ubuntu20.4, CPU 采用 12th Gen Intel (R) Core (TM) i7-12700H, GPU 采用 NVIDIA GeForce RTX 3090; 软件环境为 Python3.9, CUDA11.2, Pytorch2.0.1。

实验超参数 batch size 为 32, epochs 为 150, 学习率为 0.001, 图片大小为 640, 动量为 0.937。

3.2 评价指标

本文研究以召回率 (recall, R)、精确率 (precision, P)、平均检测精度值 (mean average precision, mAP)、浮点计算量 (floating point operations, FLOPs)、模型参数量 (parameters, Params) 以及模型规模 (Size) 作为模型评价指标, 部分指标计算公式为

$$R = \frac{TP}{TP + FN} \quad (25)$$

$$P = \frac{TP}{TP + FP} \quad (26)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N \int_0^1 P(R) dR \quad (27)$$

式中: FN 为漏检的正类样本数量; TP 为正确预测的

正类样本数量; FP 为错误预测为正类的负类样本数量。

mAP 是对 N 个类别的 R 和 P 综合考虑, 反映整个模型检测的准确率, mAP@0.50 表示当 IoU 为 0.5 时平均检测精度值, 是目标检测常用的评价指标之一, 本文研究选取 IoU = 0.5 时的平均检测精度。

3.3 主流轻量化主干网络对比实验

为选取一款既能保证检测精度, 又不过多占用自动驾驶域控制器内存和计算资源的主干网络, 本文研究在 YOLOv8s 的基础上, 选取部分主流轻量化主干网络进行替换, 各主干网络实验对比结果如表 2 所示。从表 2 可知, FasterNet_t1 在参数量、计算量以及模型大小分别降低 33.2%、32.3% 和 32.6% 的情况下, 检测精度、精确率和召回率都保持较高水平, 在做到模型轻量化的同时兼顾了检测精度。Shufflenet2 虽然轻量化程度更高, 但检测精度降低了 3.2%, 道路障碍物检测精度太低, 危及行车安全。Mobilenet3_L 和 GhostNetv2 在模型规模及计算量方面与 FasterNet_t1 表现相近, 但检测精度都远低于 FasterNet_t1。因此, 选用 FasterNet_t1 替换 YOLOv8s 原主干网络, 在保证检测精度的同时可以大幅减少模型参数和计算量。

3.4 不同损失函数对比实验

为提高模型检测精度, 验证本文所提出 IP-IoU 的有效性, 在 YOLOv8s 的基础上采用不同边框回归损失计算方式, 即将原有 CIoU 分别替换为 DIoU、SIoU、GIoU、EIoU 以及本文所提出的 IPIoU, 其余部分保持不变。实验结果如表 3 所示。从表 3 可知, 将 CIoU 替换为 DIoU、GIoU 以及 EIoU 之后, 各项指标 (检测精度、精确率和召回率) 都呈不同程度下降。SIoU 和本文改进的 IPIoU 在检测精度上分别提高了 0.2% 和 0.6%, 而 IP-IoU 的精确率和召回率分别提高了 1.2% 和 0.3%。

图 9 为 CIoU 和 IP-IoU 的边框损失曲线对比图, 可以看出优化后的边框损失更低, 预测效果更好, 因此可以验证本文所提出 IP-IoU 的有效性。

表 2 主干网络对比实验结果

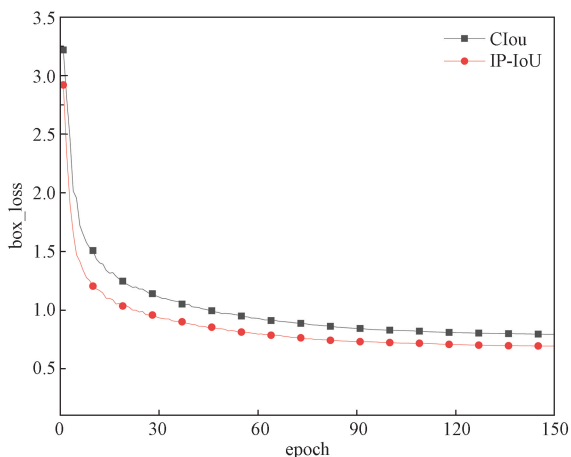
Table 2 Results of backbone networks experiment

基础模型	主干网络	检测精度	精确率	召回率	参数量/ 10^6	计算量/ 10^9	规模/MB
YOLOv8s	CSPDarknet	0.901	0.916	0.835	11.15	28.8	21.5
YOLOv8s	Shufflenet2	0.869	0.907	0.788	5.94	15.9	11.6
YOLOv8s	Mobilenet3_L	0.884	0.920	0.805	7.89	19.0	15.4
YOLOv8s	GhostNetv2	0.882	0.908	0.811	8.24	19.1	16.3
YOLOv8s	FasterNet_t1	0.894	0.915	0.825	7.45	19.5	14.5

表3 不同损失函数对比实验

Table 3 Comparison experiment of different loss functions

损失函数	检测精度	精确率	召回率
CIoU	0.901	0.916	0.835
DIoU	0.900	0.927	0.831
SIoU	0.903	0.934	0.833
GIoU	0.895	0.914	0.826
EIoU	0.898	0.921	0.826
IPIoU	0.907	0.928	0.838



box_loss 为边界框损失; epoch 为训练轮次

图9 改进前后边界框损失曲线图

Fig. 9 Before and after improvement of bounding box loss curve diagram

3.5 消融实验

为验证各改进策略的有效性,设计5组消融实验。在表4中,实验1为原模型;实验2将YOLOv8s主干网络重构为FasterNet_t1;实验3在实验2基础上在Neck部分引入SPD-Conv;实验4在实验3基础

表4 消融实验

Table 4 Ablation experiment

实验	FasterNet_t1	SPD-Conv	IP-IoU	SimAM	检测精度	精确率	召回率	参数量/ 10^6	计算量/ 10^9
1					0.901	0.926	0.837	11.15	28.8
2	✓				0.894	0.915	0.825	7.45	19.5
3	✓	✓			0.900	0.925	0.833	7.94	22.3
4	✓	✓	✓		0.906	0.934	0.834	7.94	22.4
5	✓	✓	✓	✓	0.913	0.942	0.838	7.91	22.9

根据表5显示,本文模型明显优于其他模型。与同级别精度的模型相比,本文模型的参数量和计算量都远低于其他模型;与同量级模型相比,检测精度远远高于其他模型。因此证明了本文所提出改进策略在道路障碍物检测应用场景的优势。

图10为模型改进前后的检测效果对比图。从图10(a)和图10(b)可以看出,改进后的模型对被遮挡和重叠的目标有了更为精确的识别,类别的置信度和检测框的位置更接近真实值;在远处小目标

上将原模型的CIoU替换为IPIoU,实验5在实验4基础上添加注意力模块SimAM。

实验2采用FasterNet_t1作为主干网络,模型参数量和计算量分别降低了33.2%和32.3%,大大削减了模型规模和计算量,但这也带来检测精度、精确率和召回率下降的弊端;实验3在实验2基础上引入SPD-Conv,改善了FasterNet_t1作为主干网络检测精度较低的问题,SPD-Conv增加了空间信息到深度信息的转换,致使模型参数量和计算量稍有增加,但仍低于原模型28.8%和22.6%,与实验2相比,检测精度、精确率和召回率分别提高了0.6%、1%和0.7%;实验4在实验3基础上采用IPIoU作为模型的边框回归损失函数,提高了边框回归速度及小目标检测能力,与实验3相比,检测精度、精确率和召回率分别提高了0.6%、0.9%和0.2%;最后,实验5融合了所有改进策略,即在实验4的基础上添加了SimAM注意力机制,与原YOLOv8s模型相比,改进后的模型参数量减少了 3.24×10^6 ,计算量降低了 5.9×10^9 ,同时检测精度、精确率和召回率分别提高了1.2%、1.6%和0.1%;实验5相较于实验2,弥补了轻量化改进带来的精度损失,实现了精度从下降0.7%到上升1.2%的转变。消融实验验证了各改进策略的有效性,FasterNet_t1对模型进行轻量化,SPD-Conv、IPIoU和SimAM则弥补了轻量化带来的检测精度下降,同时提高了小目标的检测能力。

3.6 不同模型对比实验

在完成轻量化及检测精度改进后,得到最终基于YOLOv8s的轻量化改进模型。为进一步验证本文方法在道路障碍物检测应用场景的优势,将本文方法与相关主流单阶段算法进行对比。

表5 不同模型对比实验

Table 5 Comparison Experiment of Different Models

模型	检测精度	参数量/ 10^6	计算量/ 10^9	规模/MB
v6s	0.889	16.45	44.9	31.3
v3-tiny	0.803	12.17	19.1	23.3
v5s	0.898	9.15	24.2	17.7
v7-tiny	0.861	6.02	13.0	12.3
v5 m	0.915	25.11	64.6	48.2
v8s	0.901	11.15	28.8	21.5
本文模型	0.913	7.91	22.9	15.3



图 10 检测效果对比

Fig. 10 Comparison of detection effects

检测方面,原模型相较于改进后的模型漏检了两处,小目标检测能力偏弱。

4 实车部署

为验证改进模型具有工程应用价值,将模型部署至无人驾驶车辆域控制器内,型号为 MIC-7700,车载相机型号为索尼 U2291,操作系统为 Ubuntu,并利用 GPU 加速运算,图 11 为实验所用车辆,车载相机位于车辆正前方支架上。实验所用无人车是以 ROS(robot operating system)为开发平台,通过发布/订阅消息模式实现数据共享和通信。如图 12 所示,建立/usb_cam 节点获取车载相机实时画面,并发布图像话题,节点/image_veiw 实时显示原始图像,节点/detect 实时显示改进模型检测效果。同时,也可以利用 ROS 的收发机制将前方道路障碍物检测信息发送至自动驾驶规划与控制模块,指导自动驾驶车辆安全行驶。

5 结论

针对已有道路障碍物检测网络对自动驾驶域控制器内存和计算资源占用较大的问题,提出了一种基于 YOLOv8s 改进的轻量化识别方法,兼顾了模型的大小和检测精度。通过重构主干特征提取网络有效降低了模型参数量和计算量;通过在颈部引入 SPD-Conv,并在颈部与主干网络之间加入



图 11 无人驾驶车

Fig. 11 Autonomous vehicle

SimAM 注意力机制,以及将原有的 CIoU 替换为本文提出的 IPIoU,这些方法有效弥补了轻量化改进带来的精度损失,使精度从下降 0.7% 转变为上升 1.2%。在实车验证中,通过 ROS 的通信机制,检测信息可发送至自动驾驶规划控制模块,指导车辆安全行驶。

参 考 文 献

[1] 邓亚平,李迎江. YOLO 算法及其在自动驾驶场景中目标检测综述[J]. 计算机应用, 2024, 44(6): 1949-1958.
Deng Yaping, Li Yingjiang. Review of YOLO algorithm and its applications to object detection in autonomous driving scenes [J]. Computer Applications, 2024, 44(6): 1949-1958.

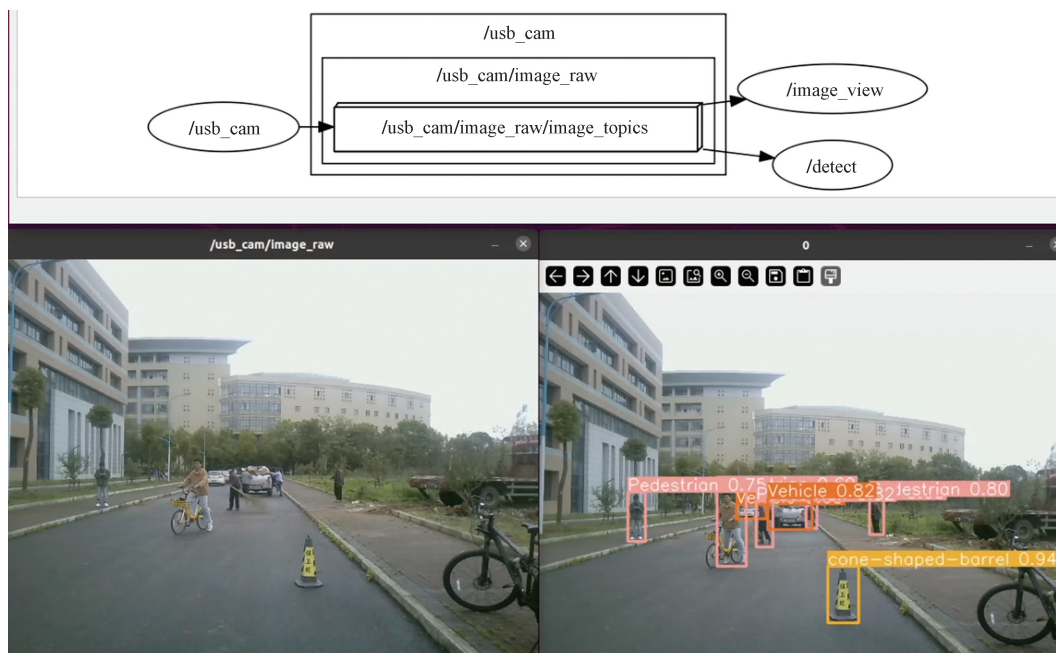


图 12 基于 ROS 的实时检测对比图

Fig. 12 Real-time detection comparison chart based on ROS

- [2] 张小俊, 奚敬哲, 史延雷, 等. 面向路侧视角目标检测的轻量级 YOLOv7-R 算法[J]. 汽车工程, 2023, 45(10): 1833-1844. Zhang Xiaojun, Xi Jingzhe, Shi Yanlei, et al. Lightweight YOLOv7-R algorithm for road-side view target detection[J]. Automotive Engineering, 2023, 45(10): 1833-1844.
- [3] Li C, Li L, Jiang H, et al. YOLOv6: a single-stage object detection framework for industrial applications[J]. arXiv preprint arXiv: 2209.02976, 2022.
- [4] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[J]. arXiv eprints arXiv: 2207.02696, 2022.
- [5] 郭克友, 王苏东, 李雪, 等. 基于 Dim env-YOLO 算法的昏暗场景车辆多目标检测[J]. 计算机工程, 2023, 49(3): 312-320. Guo Keyou, Wang Sudong, Li Xue, et al. Multi-target detection of vehicles in Dim scenes based on DIM env-YOLO algorithm[J]. Computer Engineering, 2023, 49(3): 312-320.
- [6] Zaghari N, Fathy M, Jameii S M, et al. The improvement in obstacle detection in autonomous vehicles using YOLO non-maximum suppression fuzzy algorithm[J]. The Journal of Supercomputing, 2021, 77(11): 13421-13446.
- [7] Mahaur B, Mishra K K. Small-object detection based on YOLOv5 in autonomous driving systems[J]. Pattern Recognition Letters, 2023, 168: 115-122.
- [8] 高昕, 甄国涌, 储成群, 等. 基于改进 YOLOv5 的自动驾驶目标检测[J]. 科学技术与工程, 2024, 24(16): 6757-6765. Gao Xin, Zhen Guoyong, Chu Chengqun, et al. Autonomous driving target detection method based on improved YOLOv5[J]. Science Technology and Engineering, 2024, 24(16): 6757-6765.
- [9] 张利丰, 田莹. 改进 YOLOv8 的多尺度轻量型车辆目标检测算法[J]. 计算机工程与应用, 2024, 60(3): 129-137. Zhang Lifeng, Tian Ying. Multi-scale lightweight vehicle detection algorithm YOLOv8[J]. Computer Engineering and Applications, 2024, 60(3): 129-137.
- [10] Zhang X, Zhou X, Lin M, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2018: 6848-6856.
- [11] Chen J, Kao S, He H, et al. Run, don't walk; chasing higher FLOPS for faster neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2023: 12021-12031.
- [12] Sunkara R, Luo T. No more strided convolutions or pooling: a new CNN building block for low-resolution images and small objects[J]. arXiv preprint arXiv: 2208.03641, 2022.
- [13] Zhang H, Xu C, Zhang S. Inner-IoU: more effective intersection over union loss with auxiliary bounding box[J]. arXiv preprint arXiv: 2311.02877, 2023.
- [14] Liu C, Wang K, Li Q, et al. Powerful-IoU: more straight forward and faster bounding box regression loss with a nonmonotonic focusing mechanism[J]. Neural Networks, 2024, 170: 276-284.
- [15] 张猛, 尹丽菊, 周辉, 等. 基于 SimAM-Ada YOLOv5 的太阳能电池表面缺陷检测[J]. 电子测量技术, 2023, 46(22): 17-25. ZhangMeng, Yin Liju, Zhou Hui, et al. Surface defect detection of solar cells based on SimAM-Ada YOLOv5[J]. Electronic Measurement Technology, 2023, 46(22): 17-25.
- [16] Yang L, Hang R Y, Li L, et al. Simam: a simple, parameter-free attention module for convolutional neural networks[C]//International Conference on Machine Learning. Cambridge: PMLR, 2021: 11863-11874.