

Doi: 10.20062/j.cnki.CN 37-1453/N.2024.03.010

基于改进 YOLOv5s 的轻量级车辆检测系统

林小涵^a, 侯典立^a, 郑红霞^b

(鲁东大学 a.信息与电气工程学院; b.交通学院, 山东 烟台 264039)

摘要: 基于深度学习的车辆检测在智慧交通中起着重要作用, 现有模型结构复杂、计算量大, 难以布置在嵌入式系统的边缘设备。本文提出一种基于 YOLOv5s 的轻量级车辆检测算法, 通过 GhostNet 和剪枝优化策略对 YOLOv5s 进行了改进, 实现系统的轻量化和检测的实时性; 损失函数 Focal-EIoU Loss 的引入解决了样本不平衡和纵横比模糊定义问题, 从而提高目标检测的性能。在 UA-DETRAC 数据集上的实验结果表明, 所提算法与原 YOLOv5s 算法相比参数量、模型体积和浮点运算次数分别减少了 63%、50.6% 和 64.7%, 检测速度提升了 50%, 同时保持了较高精度、召回率, 为空间、能源、资源有限的边缘检测设备提供了一种实时性算法选择。

关键词: YOLOv5s; GhostNet; 车辆检测; 轻量级模型; EIoU

中图分类号: TP391.41 **文献标志码:** A **文章编号:** 1673-8020(2024)03-0261-08

车辆检测是计算机视觉领域中一个重要的应用, 旨在从图像或视频中准确地检测和确定车辆的位置。如今在交通系统中, 路网监控依然存在提取能力有限、实时性较差等问题, 难以满足对车流量监控和智能引导的需求, 通过深度学习方式进行车辆检测成为研究热点。

车辆检测算法主要有 3 大类: 基于传统特征的方法、基于滑动窗口的检测和基于深度学习的方法。基于传统特征的方法^[1]和滑动窗口方法^[2]对遮挡和复杂背景比较敏感。基于深度学习的目标检测算法为车辆检测的首选方案, 分为两阶段法和单阶段法。基于候选区域的两阶段法包含 R-CNN^[3]、Fast R-CNN^[4]、Faster R-CNN^[5]等, 在 R-CNN 基础上通过多任务等方式进行了陆续改进、优化, 但冗余计算过多, 测试速度仍然较慢, 无法实时更新。Redmon 等^[6]为降低 R-CNN 及其变体的时间复杂度、提高检测速度, 提出了一种新的端到端的检测方法 YOLO, 将检测任务转化为回归问题, 直接在图像上预测边界框和类别。Lu 等^[7]提出了一种基于 YOLO 深度学习算法的航空影像车辆检测模型, 能够很好地检测一些小型、紧凑和密集物体。YOLOv2^[8]采用了一种联合训练方法, 可以根据检测和分类数

据对目标检测器进行训练, 但检测准确率仍然较低。Sang 等^[9]采用改进的 YOLOv2 模型 YOLOv2_Vehicle 来提高车辆检测的精度、速度和泛化能力, 在两个车辆数据集上进行了实验, 可以加速车辆数据集的收敛, 更好地满足车辆检测任务。Wang 等^[10]提出了一种基于改进的 tiny-YOLOv3 的车辆检测算法, 以解决复杂道路环境下识别率低、实时检测性能低的问题, 可以将检测速度提高到 144 帧 \cdot s⁻¹。YOLO 后续版本陆续发布, YOLOv4 在数据处理、主干网络、网络训练、激活函数、损失函数等方面都有不同程度的优化; YOLOv5 加入了 focus 框架, 提高了训练速度; YOLOv6 主要是为了更加适应 GPU 设备; YOLOv7 主要针对模型结构重参化和动态标签分配问题进行了优化; YOLOv8 主要在结构算法、命令行界面、Python API 等方面进一步优化。YOLOv8s 与 YOLOv5s 相比, 在 CSPLayer 上的更改提高了检测精度, 模型中的可学习参数数量从 7.2 M 增加到了 11.2 M, 浮点预算次数 FLOPs@640(B) 由 16.5 G 增加到了 28.6 G, 速度有所下降。YOLOv5s 是众多版本中适用于边缘设备的一个经典版本, 对深度学习的环境配置要求较低, 且相对于新版本代码中的缺陷更少。

收稿日期: 2023-11-21; 修回日期: 2024-03-28

基金项目: 山东省高等学校科学技术计划项目(J16LN31); 烟台市智慧城市创新实验室科研课题项目(SDGP37060000202302000504)

通信作者简介: 侯典立(1974—), 男, 副教授, 硕士研究生导师, 博士, 研究方向为无线功率传输。E-mail: hdianli@163.com

基于边缘设备计算基础上车辆检测精度和速度的均衡,本文提出一种改进的 YOLOv5s 车辆检测算法:将 YOLOv5s 的网络替换为轻量级网络 GhostNet,以解决传统卷积方法中计算冗余的问题;通过剪枝优化策略剪除对于检测效果影响不大的冗余通道,大大降低模型的体积和计算能力;利用损失函数 Focal-Elou Loss^[11] 替代原始的 CIoU Loss 作为目标边界框的回归损失函数,以减少训练时间、缓解样本不平衡问题,同时有效提高车辆检测的回归精度。

1 轻量级 YOLOv5s 目标检测算法

1.1 YOLOv5s 算法

YOLOv5s 是 YOLOv5 系列中的较轻量级模型,旨在保持较高检测精度的同时提高模型的计算速度,适用于边缘设备和实时目标检测。YOLOv5s 的主要架构由主干网络、特征融合模块、检测头和损失函数 4 部分组成。1) CSPDarknet53 作为主干网络,是一种改进的 Darknet53 网络结构,采用了 CSP (cross stage partial connections) 结构,以提高特征提取能力和网络的计算效率。2) PANet (path aggregation network) 特征融合模块作为颈部,通过自底向上和自顶向下的路径聚合,将来自不同层级的特征进行融合,以提取更丰富的语义信息。3) 检测头采用多尺度预测策略,在网络的不同层级上进行预测,以捕捉不同尺度目标的特征,YOLOv5s 默认预测 3 个不同尺度的目标框,分别对应较大、中等和较小的目标。4) 损失函数采用了 CIoU Loss,用于衡量模型预测与真实标签之间的差异,以及反向传播和模型参数的更新。当输入图像大小为 640×640 ,颈部网络分别执行 8 次、16 次和 32 次下采样,对应检测层特征图的维度为 80×80 、 40×40 和 20×20 ,分别检测小目标、中目标和大目标。

1.2 基于 GhostNet 的 YOLOv5s

Darknet53 是一个相对轻量的网络,其依赖于深度可分离卷积来减小参数量。当结合 CSP 构成 CSPDarknet53 后进一步提升了效率,计算量低于未经优化的 Darknet53,但参数量和计算量在同等条件下与轻量级神经网络相比仍然较高。

MobileNet、ShuffleNet、GhostNet 是具有代表性

的轻量级神经网络的 3 个系列。MobileNet 系列^[11-14] 采用了深度可分离卷积 (DSConv) 模块,将标准卷积分解为深度卷积和点卷积,网络参数的数量显著减少。ShuffleNet 系列^[15-16] 添加了基于 DSConv 的组卷积模块和通道混洗模块,引入了通道拆分结构,减少网络参数的数量。为了改进大量冗余特征,GhostNet 系列中采用了 GhostConv 模块^[17-18],GhostConv 模块网络结构如图 1 所示。GhostConv 将一个大的卷积核的特征提取过程拆分成两个步骤:首先通过一个较小的卷积核得到一部分特征映射;然后通过线性变换将这部分特征映射扩展到与原设计的大卷积核所对应的特征映射数量相同。通过此种方式,模型能够以相对较低的计算成本获取接近于大卷积核提取的特征效果,从而降低模型复杂度,提升运算速度,并在一定程度上保持甚至提升模型性能。

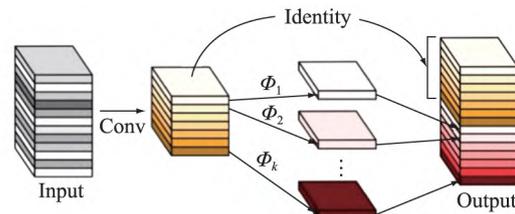


图 1 GhostConv 模块网络结构

Fig.1 The network structure of GhostConv model

根据其设计原则,GhostNet 通过 GhostConv 模块生成额外特征图时主要依赖 1×1 卷积,在输入和输出特征图数量相同的情况下,GhostConv 模块的计算成本远低于普通卷积。给定输入数据 $X \in \mathbf{R}^{c \times h \times w}$, c 表示输入通道的数量, h 和 w 分别表示输入数据的高度和宽度,输出特征 $Y \in \mathbf{R}^{h' \times w' \times n}$, n 为卷积核数量, h' 和 w' 分别表示输出数据的高度和宽度。GhostConv 模块与标准卷积的参数数量压缩比为:

$$r_c = \frac{n \cdot c \cdot k \cdot k}{n \cdot c \cdot k \cdot k / s + (s - 1) \cdot n \cdot d \cdot d / s} \approx \frac{s \cdot c}{s + c - 1} \approx s, \quad (1)$$

其中,卷积核大小为 $k \times k$, s 为冗余特征数量,线性变换的大小为 $d \times d$ 。

GhostConv 模块通过拼接基础特征与廉价特征来增加通道数,避免了传统卷积中大量乘加运算。GhostConv 模块计算量远小于普通卷积,两者比值为:

$$r_s = \frac{n \cdot h' \cdot w' \cdot c \cdot k \cdot k}{n \cdot h \cdot w \cdot c \cdot k \cdot k / s + (s - 1) \cdot n \cdot h' \cdot w' \cdot d \cdot d / s} \approx \frac{s \cdot c}{s + c - 1} \approx s, \quad (2)$$

基于 GhostNet 的 YOLOv5s 主要组成如图 2 所示。利用 GhostConv 模块替换 YOLOv5s 中 Conv 卷积模块, C3Ghost 替换 BottleneckC3, 基于 GhostNet 的 YOLOv5s 模型架构如图 3 所示。

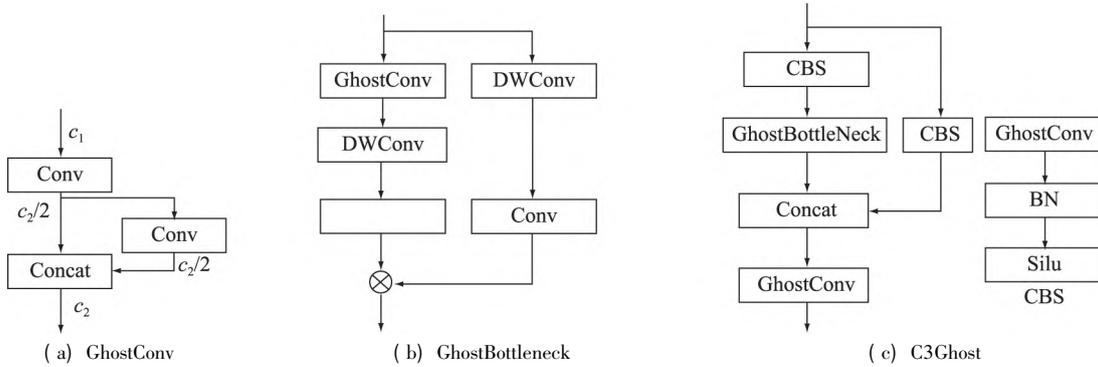


图 2 基于 GhostNet 的 YOLOv5s 主要组成
Fig.2 The main components of YOLOv5s based on GhostNet

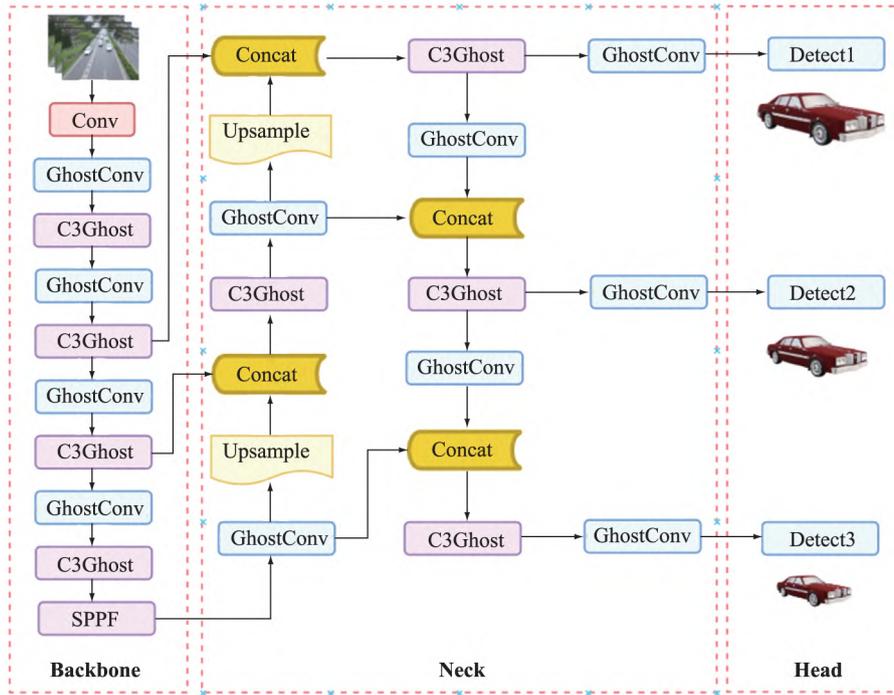


图 3 基于 GhostNet 的 YOLOv5s 模型架构
Fig.3 The model architecture of YOLOv5s based on GhostNet

1.3 优化损失函数

YOLOv5s 的损失函数计算公式为:

$$L = L_{obj} + L_{cls} + L_{bbox}, \quad (3)$$

其中 L 代表损失函数, L_{obj} 、 L_{cls} 、 L_{bbox} 分别为置信度损失函数、分类损失函数和边界框回归损失函数。

YOLOv5s 的边界框回归函数为 CIoU Loss, 主要考虑边界框之间的重叠区域、中心点距离和纵横比, 可以降低遮挡重叠对目标检测精度的干扰,

预测框回归算法的评估也变得更加准确; 但其对纵横比描述的是相对值, 存在一定的模糊, 未考虑难易样本的平衡问题。EIoU Loss 在 CIoU Loss 的基础上分别计算宽高的差异值, 取代了纵横比, 明确衡量了重叠区域、中心点和边长这 3 个几何因素的差异。EIoU Loss 包含 3 个部分: 重叠损失 L_{IoU} 、中心距离损失 L_{dis} 和宽高损失 L_{asp} , 考虑更全面, 收敛更快, 精确度更高。惩罚项 L_{EIoU} 公式如下:

$$L_{\text{EIoU}} = L_{\text{IoU}} + L_{\text{dis}} + L_{\text{asp}}, \quad (4)$$

$$L_{\text{IoU}} = 1 - R_{\text{IoU}}, \quad (5)$$

$$L_{\text{dis}} = \frac{\rho^2(b, b_{\text{gt}})}{c^2}, \quad (6)$$

$$L_{\text{asp}} c_h^2 = \frac{\rho^2(w, w_{\text{gt}})}{c_w^2} + \frac{\rho^2(h, h_{\text{gt}})}{c_h^2}, \quad (7)$$

式中: R_{IoU} 表示目标区域与预测区域之间的交并比; b 和 b_{gt} 分别代表预测框和真实框; $\rho^2(w, w_{\text{gt}})$ 为预测框与真实框中心点的欧氏距离; c 为两框的最小外接矩形框的对角线距离; c_w 、 c_h 分别表示该矩形框的宽度和高度; w 和 h 分别表示预测框的宽度及高度; w_{gt} 及 h_{gt} 分别表示真实框的宽度及高度。

为了让 EIoU Loss 更关注高质量样本, Zhang 等^[11] 在 EIoU Loss 的基础上引入 Focal Loss, 以解决样本不平衡问题, 公式如下:

$$L_{\text{FEIoU}} = R_{\text{IoU}}^\tau L_{\text{EIoU}}, \quad (8)$$

式中 τ 为控制异常值抑制程度的参数, L_{FEIoU} 表示 Focal-EIoU 损失。在 $R_{\text{IoU}} \in (0, 0.8]$ 时, 损失值会被拉低, 在 $R_{\text{IoU}} \in (0.8, 1]$ 时, 损失值基本保持不变, Focal-EIoU 损失函数能够让网络更关注简单样本。

考虑到质量较差样本产生的过大梯度将影响训练过程, 为了在梯度上更好地区分高质量锚框和低质量锚框, 本文采用 Focal-EIoU Loss 作为损失函数, 可以有效提高模型的收敛速度, 优化边界框回归任务中的样本不平衡问题, 改善检测精度和质量。

1.4 通道剪枝

YOLO 系列神经网络通常过度参数化, 具有显著冗余, 而剪枝可以在保持准确性低损失前提下消除冗余权重。剪枝方式主要分为结构性和非结构性剪枝。结构性剪枝涉及到删除模型中的整个神经元、过滤器(即卷积层中通道), 或者完全移除整个层, 会导致模型结构发生改变, 可能需要重新设计和实现剪枝后的模型结构以保持其完整性。非结构性剪枝则侧重于逐元素或逐通道删除权重而不改变模型的整体结构。结构性剪枝在具有常见卷积的大型模型上效果很好, 但当涉及到基于 GhostNet 深度卷积的模型时剪枝率很小, 且准确性也会急剧下降, 故本文采用非结构性剪枝。

以训练好的模型为基础, 通过敏感度分析计算每个卷积层通道的权重量化指标, 确定哪些通道对输出的影响较小, 即敏感度较低, 从而采取不同的剪枝策略; 再对剪枝后的模型进行微调, 通过重新训练使模型适应剪枝后的结构, 完成剪枝工作。稀疏化剪枝是一种有效减少模型复杂度、提升计算效率和降低内存占用的技术, 尤其适用于 YOLO 系列这样的大型神经网络。为了促进权重稀疏化, 对原始网络中 BN(batch normalization) 层的参数添加 L1 正则, 对输入特征进行标准化。假设 x 是 BN 层的输入, μ 是输入的均值, σ^2 是输入的方差, 规范化后的输出为:

$$\hat{x} = (x - \mu) / \sqrt{\sigma^2 + \varepsilon}, \quad (9)$$

式中 ε 是一个很小的正值, 用于防止分母为零的情况。

BN 层通过 γ 和 β 对规范化后的输出进行缩放和平移, β 参数则负责对规范化后的特征进行偏移, 即

$$y = \gamma \cdot \hat{x} + \beta. \quad (10)$$

使用带有 L1 正则化的损失函数进行网络训练, 训练过程中, L1 正则化将引导模型自我剪枝, 特别是 BN 层的 γ 参数, 使得一些通道的重要性降低。针对 GhostNet-YOLOv5s 中 BN 层统计规范化后特征的参数 γ 添加 L1 正则化, 使得系数稀疏化。训练结束后, 统计所有 BN 层中 γ 参数的绝对值大小, 按照预先设定的剪枝率筛选待剪枝的通道, 移除 γ 值低于设定阈值的对应通道, 剪掉不重要的特征图通道。

2 实验配置

2.1 数据集

本文实验使用 UA-DETRAC 数据集, 该数据集是一个广泛应用于目标检测和跟踪领域的计算机视觉数据集, 特别专注于交通监控和车辆识别应用。该数据集主要包含拍摄于中国北京和天津各种场景的大约 1400 个视频剪辑, 总时长超过 100 h, 涵盖正常道路、十字路口、立交桥等, 对象类型、比例和姿势差异大, 有助于提高车辆检测系统的鲁棒性。为了避免由于重复学习同一特征而导致的过拟合现象, 有效减少模型的训练时间, 本文从原始视频每 5 帧提取一次, 选择 16 542 张图

像作为新的训练集,从测试集中选择5617张图像作为新的测试集,其中包括晴天、阴天和雨天等天气条件下的图像。使用LabelImg进行标注,将车辆分为4类,即轿车、公共汽车、货车和其他车辆,训练集和测试集样本分布如表1所示。

表1 训练集和测试集样本数量分布
Tab.1 Sample number distribution of training and test sets

类型	训练集	测试集
轿车	13 990	4694
货车	1586	496
公共汽车	866	410
其他车辆	100	17

2.2 实验环境和训练参数设置

为了验证YOLOv5s在车辆检测中的准确度和稳定性,边缘设备采用旭日X3,X3处理器采用ARM Cortex-A53作为主控CPU,L1 Cache大小为32 KB,L2 Cache大小为512 KB,支持FPU、NEON加速,典型工作频率为1.2 GHz,并支持软件层面进行频率控制。处理器共有4个CPU Core,BPU为基于地平线伯努利二代架构打造的加速IP核,内置了2个IP核心,典型工作频率1.2 GHz,等效算力高达5 TOPS。

本实验基于Ubuntu 20.01、Python 3.8和PyTorch 1.9.0搭建了深度学习框架。模型的初始学习率lor为0.01,周期学习率lrf为0.1,优化器为SGD,迭代批量设置为32,采用学习率余弦退火衰减和Mosaic数据增强。针对车辆目标进行了训练与验证,预训练采用YOLOv5s.pt,按照顺序网络权重、硬件计算资源量和识别精度递增,计算量采用浮点运算次数表示。

2.3 评估标准

为评估模型精度,将实验结果分为TP、FP、FN、TN 4类。TP表示模型正确地将正例样本预测为正例;FP表示模型错误地将负例样本预测为正例;FN表示模型错误地将正例样本预测为负例;TN表示模型正确地将负例样本预测为负例。本文采用精确度 P 、召回率 R 、平均精度 P_a 及平均精度均值 M_{ap} 作为评价指标。 P_a 是指PR曲线下的面积,反映了模型在不同召回水平下的平均精确率;在多类别分类任务中, M_{ap} 表示所有类别 P_a 的平均值,全面反映模型的检测精度,是目标检测

算法最重要的评价指标, $M_{ap}@0.5$ 代表当IoU阈值为0.5时的平均精度。具体公式如下:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad (12)$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad (13)$$

$$P_a = \int_0^1 P dR, \quad (14)$$

$$M_{ap} = \sum_{i=1}^n P_a(i) / n, \quad (15)$$

式中: N_{TP} 、 N_{FP} 、 N_{FN} 、 N_{TN} 分别表示TP、FP、FN和TN 4类样本数; $P_a(i)$ 表示第 i 类别的 P_a ; n 为类别数。

3 实验结果和分析

3.1 消融实验分析

为了验证本研究中改进策略对模型性能的影响,在相同条件下对各个模块进行了评估, F_EIoU_L 表示采用Focal-EIoU Loss,YOLOv5s_GB表示利用GhostNet只对YOLOv5s主干网络更换,YOLOv5s_GA表示利用GhostNet对YOLOv5s进行全部更换,消融实验结果如表2所示。

1) 更换为GhostNet对模型性能影响

由表2数据可以看出:更换为GhostNet后性能相对于原始YOLOv5s性能变化较小,基本持平;YOLOv5s_GB只对主干网络更换参数量减少约27.5%,模型大小减小约26.8%,浮点运算减少约32.5%;YOLOv5s_GA完全采用GhostNet参数量减少约47.5%,模型大小减小约45.5%,浮点运算减少约48.8%。YOLOv5s_GA在保持模型准确率的同时,大大减轻模型的负担,更适合边缘计算设备。

2) 更换为Focal-EIoU Loss对模型性能影响

将YOLOv5s的边界框回归损失函数由CIoU Loss改进为Focal-EIoU Loss,由改进后的模型YOLOv5s_GAF与YOLOv5s_GA实验结果(表2)对比可以看出:前者 $M_{ap}@0.5$ 、召回率分别提高0.5%、0.54%,其它参数基本不变。通过Focal-EIoU Loss有效区分了高质量和低质量锚框,解决了训练集中样本不平衡问题,提高了回归精度的同时克服了CIoU Loss中纵横比的模糊定义,改善了样本整体质量。

表2 消融实验结果对比
Tab.2 Comparisons of ablation experiments results

模型	GhostNet	F_EIoU_L	通道 剪枝	$M_{ap}@0.5$ /%	召回率 /%	检测时 间/ms	参数量 / 10^6	模型大小 /MB	浮点运算 次数/ 10^9
YOLOv5s	—	—	—	92.9	92.3	66.0	7.03	13.8	15.9
YOLOv5s_GB	✓	—	—	92.5	93.1	48.6	5.10	10.1	10.8
YOLOv5s_GA	✓	—	—	92.1	92.8	42.6	3.70	7.5	8.2
YOLOv5s_GAF	✓	✓	—	92.5	93.3	42.9	3.71	7.5	8.2
YOLOv5s_GAFP	✓	✓	✓	91.8	92.7	34.8	2.60	6.8	5.6

注 “✓”表示增加了相应的改进策略,“—”表示没有增加相应改进策略。

3) 剪枝对模型性能影响

在 YOLOv5s_GAF 基础上通过剪枝优化策略剪枝后可得模型 YOLOv5s_GAFP,由表 2 数据可以看出,剪枝率基本为 30%,其 $M_{ap}@0.5$ 、召回率分别下降 0.7%、0.6%,检测时间缩短了 8.3%。YOLOv5s_GAFP 通过剪枝优化策略进一步减小了模型的大小和计算负担,同时保证模型在车辆识别中保持较高准确率。

4) 模型整体改进前后对比分析

由表 2 数据可以看到,YOLOv5s_GAFP 与原来的 YOLOv5s 相比,改进后的模型 YOLOv5s_GAFP 的 $M_{ap}@0.5$ 降低约 1.1%,召回率分别提高了 0.4%。参数量、模型体积和浮点运算次数分别减少了 63%、50.6%和 64.7%。在测试环境中,YOLOv5s_GAFP 的检测速度提升约 50%。总体

而言,改进后的 YOLOv5s 算法进一步提升了检测性能,同时实现了轻量性和准确性之间的平衡,解决了现有模型结构复杂、无法在小型移动设备上使用等问题。

3.2 对比实验

为了评估所提出算法的有效性,本文在相同实验条件下将 YOLOv5s_GAFP 与主流目标检测模型进行了比较,其它检测模型包括单阶段检测模型 SSD、YOLOv3、YOLOv4、YOLOv6s、YOLOv7-tiny 和 YOLOv8s,以及双阶段检测模型 Faster-RCNN。鉴于有些模型较大,检测时间较长,嵌入式系统难于运行,对比实验采用 Windows 系统计算机,采用 RTX3060TI 双显卡。不同目标检测模型检测结果比较如表 3 所示。

表3 不同目标检测模型检测结果比较

Tab.3 Comparisons of detection results by different object detection models

模型	$M_{ap}@0.5$ /%	召回率/%	检测时间/ms	参数量/ 10^6	模型大小/MB	浮点运算次数/ 10^9
SSD	90.3	90.7	113.1	24.0	91.6	116.0
Faster RCNN	89.7	93.0	234.8	136.8	521.7	252.7
YOLOv3	91.5	90.3	65.2	61.5	234.8	65.6
YOLOv4	92.3	92.7	117.6	52.5	100.6	119.8
YOLOv6s	91.6	92.1	39.3	17.2	36.3	44.1
YOLOv7-tiny	89.3	90.1	16.5	6.0	12.3	13.1
YOLOv8s	92.1	92.2	44.7	11.2	78.9	28.6
YOLOv5s_GAFP	91.8	92.7	11.6	2.6	6.8	5.6

由表 3 可知,YOLOv4 模型取得了最高的 $M_{ap}@0.5$,两级网络 Faster RCNN 召回率最高,但其模型体积和浮点运算都比较高,不适合应用于边缘设备,无法满足移动设备上的轻量级部署要求。与其他模型相比,本文提出的改进模型 YOLOv5s_GAFP 在模型体积和浮点运算次数方面具有明显优势,其 $M_{ap}@0.5$ 虽略低于 YOLOv4、YOLOv8s 模型,但召回率与 YOLOv4 模型相当,能够满足智能交通的精度要求;其模型体积和浮点运算远低于其它模型,更适用于边缘设备布置。总体而言,本文提出的 YOLOv5s_GAFP 模型在保持较高检

测精度的同时具有较低的模型体积和计算量,更宜用于计算资源有限的小型移动设备。

3.3 现场视频测试

为了验证 YOLOv5s_GAFP 可行性,对模型进行了现场视频测试,边缘设备采用旭日 X3,场景为烟台市红旗路与南迎祥路路口,不同光照环境下基于 YOLOv5s_GAFP 的检测示例如图 4 所示。由图 4 可以看出,本文改进的算法具有较高识别准确性,表现出较强的鲁棒性,能够满足交通路口车辆检测需要。



(a) 光线充足的白天

(b) 光线较弱的傍晚

图4 基于YOLOv5s_GAFP的检测示例

Fig.4 Detection examples based on YOLOv5s_GAFP

4 结语

针对现有模型体积过大,难于满足边缘设备布置的问题,本文基于YOLOv5s提出了改进的轻量级YOLOv5s_GAFP算法。首先,通过GhostNet对YOLOv5s中算法进行替换,减小了模型体积和运算量;其次,Focal-ElIoU Loss的引入进一步区分高质量锚框和低质量锚框,改善训练集存在的样本不平衡问题,加快检测速度、提高回归精度;最后,通过剪枝优化策略对模型进行了剪枝,对模型体积进一步压缩。通过基于UA-DETRAC数据集和现场视频的测试分析,改进后的模型适合于边缘设备布置,且能够满足智慧交通实时性要求。在后续的研究中,希望进一步减小模型的大小和计算量,提高精确度,使其更好地适用于低功耗的应用中。

参考文献:

- [1] TEOH S S, BRAUNL T. Performance evaluation of HOG and gabor features for vision-based vehicle detection [C] // Proceedings of 2015 IEEE International Conference on Control System Computing and Engineering (ICCSCE), 2015: 66-71.
- [2] ARUNMOZHI A, PARK J. Comparison of HOG, LBP and haar-like features for on-road vehicle detection [C] // Proceedings of IEEE International Conference on Electro / Information Technology (EIT) 2018: 362-367.
- [3] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition 2014: 580-587.
- [4] GIRSHICK R. Fast R-CNN [C] // Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV) 2015: 1440-1448.
- [5] 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 and Machine Intelligence 2017, 39(6): 1137-1149.
- [6] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified real-time object detection [C] // Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 779-788.
- [7] LU J Y, MA C, LI L, et al. A vehicle detection method for aerial image based on YOLO [J]. Journal of Computer and Communications 2018, 6(11): 98-107.
- [8] JOSEPH R, ALI F. A YOLO9000: better, faster, stronger [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017: 7263-7271.
- [9] SANG J, WU Z Y, GUO P, et al. An improved YOLOv2 for vehicle detection [J]. Sensors 2018, 18(12): 4272-4280.
- [10] WANG X L, WANG S, CAO J Q, et al. Data-driven based tiny-YOLOv3 method for front vehicle detection inducing SPP-net [J]. IEEE Access 2020(8): 110227-110236.
- [11] ZHANG Y F, REN W Q, ZHANG Z, et al. Focal and efficient IOU loss for accurate bounding box regression [J]. Neurocomputing 2022, 506(Sep.28): 146-157.
- [12] SANDLER M, HOWARD A G, ZHU M L, et al. MobileNetV2: inverted residuals and linear bottlenecks [C] // Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 2018: 4510-4520.
- [13] HOWARD A G, ZHU M L, CHEN B, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications [J]. arXiv 2017: 1-9.

- [14] HOWARD A ,SANDLER M ,CHU G ,et al. Searching for mobilenet v3 [C] // Proceedings of the 2019 IEEE/ CVF International Conference on Computer Vision , 2019: 1314-1324.
- [15] ZHANG X Y ,ZHOU X Y ,LIN M X ,et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices [C] // Proceedings of the 2018 IEEE/ CVF Conference on Computer Vision and Pattern Recognition ,2018: 6848-6856.
- [16] MA N ,ZHANG X ,ZHENG H T ,et al. ShuffleNet v2: practical guidelines for efficient CNN architecture design [J]. arXiv ,2018: 1-9.
- [17] HAN K ,WANG Y H ,TIAN Q ,et al. Ghostnet: more features from cheap operations [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition ,2020: 1580-1589.
- [18] TANG Y H ,HAN K ,GUO J Y ,et al. GhostNet v2: enhance cheap operation with long-range attention [J]. Advances in Neural Information Processing Systems. 2022(35) : 9969-9982.

Lightweight Vehicle Detection System Based on the Improved YOLOv5s

LIN Xiaohan^a , HOU Dianli^a , ZHENG Hongxia^b

(a.Faculty of Information and Electrical Engineering; b.Faculty of Transportation ,Ludong University ,Yantai 264039 ,China)

Abstract: Vehicle detection based on deep learning plays an important role in smart transportation. The existing model structure which is complex and computationally intensive makes it difficult to deploy on edge devices of embedded systems. This paper proposed a lightweight vehicle detection algorithm based on YOLOv5s. YOLOv5s was improved through GhostNet and pruning optimization strategies to achieve lightweight system and real-time detection; and the introduction of the loss function Focal-EIoU Loss solved the problem of sample imbalance and aspect ratio blur define the problem ,thereby improving the performance of object detection. Experimental results on the UA-DETRAC data set are as follows. Compared with the original YOLOv5s algorithm ,the proposed algorithm reduces the number of parameters ,model volume and FLOPs by 63% ,50.6% and 64.7% respectively ,and the detection speed increases by 50% ,while maintaining high precision and recall rate. This paper provides a real-time algorithm choice for edge detection equipment with limited space ,energy ,and resources.

Keywords: YOLOv5s; GhostNet; vehicle detection; lightweight model; EIoU

(责任编辑 李秀芳)