

文章编号:1671-6833(2024)02-0012-08

基于轻量化 YOLOv5 的交通标志检测

张震¹, 王晓杰¹, 晋志华¹, 马继骏²

(1. 郑州大学 电气与信息工程学院, 河南 郑州 450001; 2. 河南省交通调度指挥中心, 河南 郑州 450001)

摘要: 为了提高道路交通标志的检测速度, 提出一种基于轻量化 YOLOv5 的改进模型。首先, 使用 Ghost 卷积和深度分离卷积(DWConv)构建新的主干模块, 减少计算量和参数量; 引入加权特征融合网络(BiFPN)结构, 增强特征融合能力; 将 CIoU 损失函数替换为 Siou 损失函数, 关注真实锚框与预测的角度信息, 提升检测精度。其次, 对 TT100K 数据集进行优化, 筛选出标签个数大于 200 的交通标志图片和标注信息共 24 类。最后, 实验结果取得 84% 的准确率、81.2% 的召回率和 85.4% 的所有类别平均精确率的平均值 $mAP@0.5$, 相比原始 YOLOv5, 参数量减少 29.0%, 计算量减少 29.4%, $mAP@0.5$ 仅下降 0.1 百分点, 检测帧率提升了 34 帧/s。使用改进后的模型进行检测, 检测速度有了明显提升, 基本达到了在保持检测精度的基础上压缩模型的目的。

关键词: 交通标志检测; 轻量化 YOLOv5; Siou 损失函数; Ghost 卷积; TT100K; BiFPN

中图分类号: TP311.5

文献标志码: A

doi: 10.13705/j.issn.1671-6833.2023.05.014

交通标志蕴含丰富的语义信息, 这些信息或提醒或警告或禁令行驶在道路上的司机, 保障行人和司机的安全。并且随着自动驾驶技术的快速发展, 行驶在道路上的车辆只有快速、准确、高效地识别车道线、交通信号灯、交通标志等关键信息, 并综合分析其他来源信息, 才能提高自动驾驶的安全性。所以, 交通标志检测的研究对于智能驾驶相关技术具有重大意义。

随着算力水平的提高, 基于深度学习的目标检测算法已经成为科研人员的主流选择。两阶段目标检测算法和单阶段目标检测算法相继被研究学者提出。Girshick 等^[1]提出的 R-CNN(region convolutional neural networks)被认为是两阶段检测算法的先驱, 该算法先提取候选区域, 再使用卷积神经网络和 SVM 分类器进行分类预测。在此基础上, 科研人员相继提出 Fast R-CNN^[2]、Faster R-CNN^[3]、Mask R-CNN^[4]等更高效的两阶段检测算法, 但囿于其特点, 两阶段算法很难达到实时检测的要求。另一类目标检测算法为 YOLO(you only look once)系列, YOLO 算法由 Redmon 等^[5]提出, 该算法创新性地将检测视为回归问题, 使用预设锚框的方式代替候选区域选取。

此后, YOLO9000^[6]、YOLOv3^[7]、YOLOv4^[8]等 YOLO 的算法以及 SSD(single shot multibox detector)^[9]算法相继被提出, 在精度和速度方面的表现也越来越优秀。

国内外学者针对交通标志的检测探索出不同的研究方案。Sudha 等^[10]通过使用新的随机梯度动量序列和形状特征提取方法进行检测, 然后使用卷积神经网络(CNN)分类器对训练输出标签进行分类, 最后在训练与测试阶段将交通标志转换为音频信号, 帮助视障人士解决问题。Han 等^[11]提出一种针对小目标交通标志检测的 Faster R-CNN, 首先使用小的区域候选网络(region proposal network, RPN)提取建议框, 然后将 Faster R-CNN 与在线硬示例挖掘(online hard example mining, OHEM)算法相结合来提高定位小目标的能力。Wan 等^[12]使用改进的 YOLOv3 进行模型剪枝和多尺度预测, 在减小模型参数量的同时提高检测速度, 但是在实时检测时性能不高。尹靖涵等^[13]使用改进 YOLOv5 对雾霾场景下的交通标志进行识别, 运用削减特征金字塔深度、限制最高下采样倍数、调整残差模块的特征传递深度等方法提升检测精度。张毅等^[14]在

收稿日期: 2023-04-16; 修订日期: 2023-06-01

基金项目: 国家重点研发计划重点专项(2018XXXXXXXXX03); 河南省交通运输厅科技项目(2019G3)。

作者简介: 张震(1966—), 男, 河南郑州人, 郑州大学教授, 博士, 博士生导师, 主要从事计算机视觉研究, E-mail: zhang-zhen66@126.com。

引用本文: 张震, 王晓杰, 晋志华, 等. 基于轻量化 YOLOv5 的交通标志检测[J]. 郑州大学学报(工学版), 2024, 45(2): 12-19. (ZHANG Z, WANG X J, JIN Z H, et al. Traffic sign detection based on lightweight YOLOv5[J]. Journal of Zhengzhou University(Engineering Science), 2024, 45(2): 12-19.)

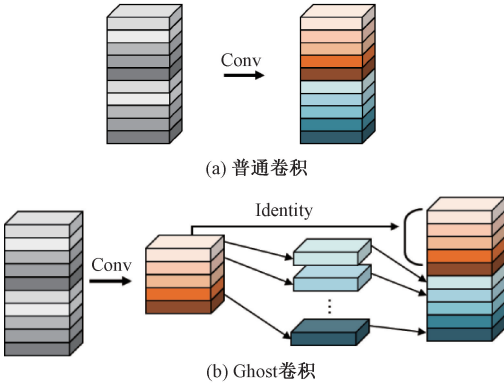


图2 普通卷积与 Ghost 卷积

Figure 2 Normal convolution and Ghost convolution

Ghost 卷积操作的计算量可以表示为

$$\frac{n}{s} \times h' \times w' \times c \times k \times k + (s-1) \times \frac{n}{s} \times h' \times w' \times d \times d. \quad (2)$$

式中: s 表示本征图参与线性变换的数量; d 表示深度可分离卷积的卷积核大小。

因为 $d \times d$ 的卷积核与 $k \times k$ 的卷积核大小相似, 且 $s \ll c$, 因此普通卷积与 Ghost 卷积计算量的比值 $r_s \approx s$, 证明 Ghost 卷积相比普通卷积更节省计算量。

普通卷积的参数量可以表示为

$$n \times c \times k \times k. \quad (3)$$

Ghost 卷积的参数量可以表示为

$$\frac{n}{s} \times c \times k \times k + (s-1) \times \frac{n}{s} \times d \times d. \quad (4)$$

同理, 普通卷积与 Ghost 卷积的比值 $r_c \approx c$, 证明 Ghost 卷积比普通卷积也能在参数量上得到削减。

GhostBottleneck 模块就是以 Ghost 模块为基础构建的一款骨干网络, Han 等^[18] 给出了 2 种骨干网络, 如图 3 所示。步长为 1 (图 3(a)) 与步长为 2 (图 3(b)) 都采用残差结构, 且图 3(b) 在 Ghost 模块中

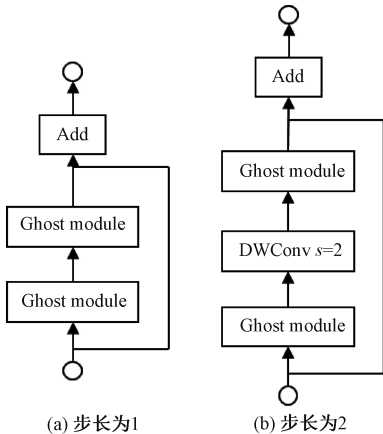


图3 GhostBottleneck 模块

Figure 3 GhostBottleneck module

间多了 1 个深度卷积, 并且残差路径由 1 个下采样层和 1 个步长为 2 的深度卷积实现。

本文中采用步长为 2 的 GhostBottleneck 结构替换 YOLOv5 主干网络中的 C3 结构的 Bottleneck, 从而达到进一步降低计算量和参数量的目的, 如图 4 所示, 图 4 中的 CBS 模块指的是卷积+批量归一化+SiLU 激活函数。

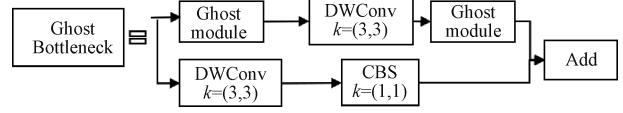


图4 本文使用的 GhostBottleneck

Figure 4 GhostBottleneck in this article

基于 GhostBottleneck 构建的 C3-Ghost 结构如图 5 所示, 用该结构替换 YOLOv5 网络中除第 1 个 C3 外的其他 C3 模块。

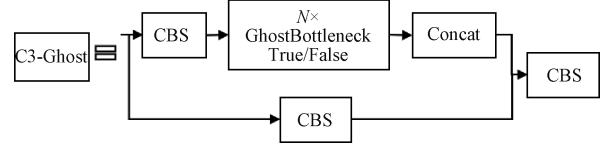


图5 基于 C3 构建的 C3-Ghost 模块

Figure 5 C3-Ghost module based on C3

2.2 加权特征融合网络 BiFPN

由于使用 Ghost 卷积降低参数量和计算量的同时也造成了检测精度下降, 因此选择将 BiFPN 引入 YOLOv5 的 Neck 网络部分, 构建 Concat 操作替换普通 Concat 层, 通过可学习权重进一步增强网络特征融合能力。

BiFPN^[19] 是谷歌大脑提出的一种加权双向特征金字塔网络, 由于特征金字塔网络 (feature pyramid network, FPN) 仅通过一条自顶向下的路径传递特征, 可能会遇到特征信息丢失的情况, 精度不高, 结构如图 6(a) 所示。PANet 是在上述网络的基础上增加一条相反的路径, 用于传达丢失的位置信息, 如图 6(b) 所示。BiFPN 则是在 PANet 的基础上删除只有一个输入的节点, 在不丢失重要信息的前提下简化网络, 同时若输入节点和输出节点位于同一层则添加一条额外的边, 以融合更多特征, 其结构如图 6(c) 所示。更重要的是, BiFPN 提出使用带权重的特征融合机制, 给每条路径分配一个可学习的权重, 通过对数据特征的学习不断更新权重, 从而获取更重要的信息。以中间层 P_6 层为例, 其最终输出可以表示为

$$P_6^{\text{out}} = \text{Conv} \left(\frac{w'_1 \cdot P_6^{\text{in}} + w'_2 \cdot P_6^{\text{id}} + w'_3 \cdot \text{Resize}(P_5^{\text{out}})}{w'_1 + w'_2 + w'_3 + \varepsilon} \right). \quad (5)$$

式中: w'_1, w'_2, w'_3 分别为各自层输入的权重; P_6^{in} 为 P_6 层的输入; P_5^{out} 为 P_5 层的输出; ε 取 0.000 1,作用是避免分母为 0; P_6^{id} 为 P_6 中间层的输出,可由式(6)表示。

$$P_6^{\text{id}} = \text{Conv} \left(\frac{w_1 \cdot P_6^{\text{in}} + w_2 \cdot \text{Resize}(P_7^{\text{in}})}{w_1 + w_2 + \varepsilon} \right) \quad (6)$$

式中: w_1, w_2 为各自可学习权重; P_7^{in} 为 P_7 层的输入。

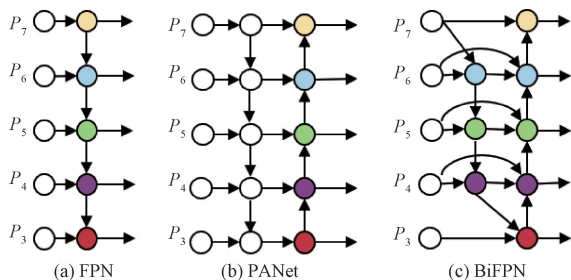


图6 FPN、PANet与BiFPN结构

Figure 6 Structures of FPN, PANet and BiFPN

2.3 SIOU 损失函数

YOLOv5 的损失函数包含 3 个部分,分别为定位损失 box_loss 、置信度损失 obj_loss 及分类损失 cls_loss 。定位损失用于计算预测框与真实框的误差;置信度损失用于计算网络的置信度;分类损失用于计算锚框与对应的标定分类是否正确。

针对定位损失,Gevorgyan^[20]认为应该进一步考虑预测框与真实框的角度信息,从而提出了新的损失函数 SIOU 损失函数,该损失函数由预测框与真实框的角度、距离和形状以及传统 IoU 4 部分损失组成,其中角度损失可以表示为

$$\Lambda = 1 - 2\sin^2 \left(\arcsin x - \frac{\pi}{4} \right) \quad (7)$$

式中: $x = \frac{c_h}{\sigma}$, σ 表示真实框与预测框中心点连线与水平轴的夹角, c_h 表示真实框中心点与预测框中心点之

间的垂直距离,二者计算公式为

$$\sigma = \sqrt{(b_{c_x}^{\text{gt}} - b_{c_x})^2 + (b_{c_y}^{\text{gt}} - b_{c_y})^2} \quad (8)$$

$$c_h = \max(b_{c_x}^{\text{gt}}, b_{c_x}) - \min(b_{c_y}^{\text{gt}}, b_{c_y}) \quad (9)$$

式中: $b_{c_x}^{\text{gt}}$ 与 $b_{c_y}^{\text{gt}}$ 表示真实框的坐标; b_{c_x} 与 b_{c_y} 表示预测框的坐标。

距离损失可以表示为

$$\Delta = \sum_{t=x,y} (1 - e^{-\gamma \rho_t}) \quad (10)$$

式中: $\rho_x = \left(\frac{b_{c_x}^{\text{gt}} - b_{c_x}}{c_w} \right)^2$; $\rho_y = \left(\frac{b_{c_y}^{\text{gt}} - b_{c_y}}{c_h} \right)^2$, c_w 和 c_h 为预测框与真实框最小外接矩形框的宽和高; $\gamma = 2 - \Lambda$ 。

形状损失可以表示为

$$\Omega = \sum_{t=w,h} (1 - e^{-\omega_t})^\theta \quad (11)$$

式中: $\omega_w = \frac{|w - w^{\text{gt}}|}{\max(w, w^{\text{gt}})}$; $\omega_h = \frac{|h - h^{\text{gt}}|}{\max(h, h^{\text{gt}})}$, $w, h, w^{\text{gt}}, h^{\text{gt}}$ 分别表示预测框和真实框的宽高信息; θ 负责控制对形状损失的关注程度,一般取[2, 6]。

最后一项为传统的交并比损失,因此 SIOU 损失函数的最终损失可以表示为

$$L_{\text{box}} = 1 - \text{IoU} + \frac{\Delta + \Omega}{2} \quad (12)$$

因此,为了提高因 Ghost 卷积丢失的检测精度,本文选择将用 SIOU 损失代替 Clou 损失,更加关注真实框与预测框之间的角度位置信息,提高锚框精度。

2.4 改进后模型

改进后的轻量化 YOLOv5 模型如图 7 所示,使用 C3-Ghost 替换原始 YOLOv5 中的 C3 模块,在颈部网络引入 BiFPN 结构,以及使用 SIOU 损失函数。

改进后模型的算法流程如下。

步骤 1 图片经过输入层的处理,计算预设锚框与真实锚框的差值并聚类得到新的一组预设锚框。

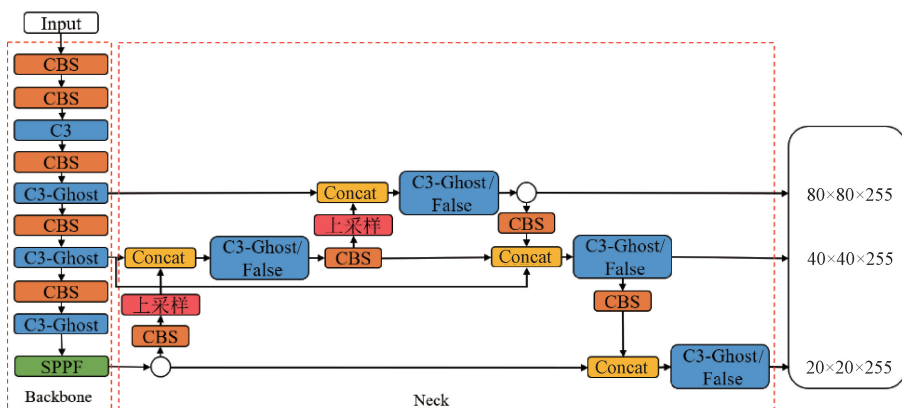


图7 轻量化YOLOv5结构

Figure 7 Structure of lightweight YOLOv5

步骤 2 将图片送入新的主干网络,利用加入 Ghost 卷积的模块和卷积层提取特征。

步骤 3 图片经过在颈部网络设计的 BiFPN 结构增加短接和跳跃连接,加强特征融合能力。

步骤 4 图片到达输出层,通过使用 SiLU 计算真实框与预测框之间的误差,降低定位损失。

3 实验与结果分析

3.1 数据集预处理

目前开源的和针对中国实际道路的交通标志数据集有 CCTSDB,以及腾讯和清华大学使用街景图像构建的 TT100K^[21]数据集等。CCTSDB 数据集仅仅将数据分为指示、警告和禁止标志 3 类,没有详细内容的区分。而 TT100K 数据集对实地采集且有详细分类,因此选择 TT100K 数据集作为本次实验的数据集。

TT100K 数据集一共包括了上百个交通标志类别,但大多数标志实例个数较少。数据集分布不均衡,有些实例的个数在 3 000 以上,有些实例数量仅为个位数,并且包含上百个种类,因此需要先对数据集进行筛选,将得到的交通标志实例个数在 200 个以上的类别进行实验,避免由于数据集的缺陷无法得到最优模型。以下为预处理步骤。

步骤 1 读取 json 标注文件,遍历标注信息,保存实例个数大于等于 200 的标签。

步骤 2 对原始 json 标注文件中的标注信息按照训练集与验证集为 8 比 2 的比例划分为 2 个 json 文件。

步骤 3 将划分好的 json 格式标注信息转换成 YOLOv5 训练需要的 TXT 格式进行保存。

步骤 4 根据划分好的标注文件将原始图片分开存放。

经过上述步骤后得到 24 个常见交通标志类别,共 8 521 张图片,其中训练集 6 801 张,验证集 1 720 张,具体类别如图 8 所示。



图 8 24 个交通标志类别

Figure 8 24 traffic sign categories

3.2 评价指标

准确率 P 指预测正确的正样本数量占有所有预测为正样本数量的百分比;召回率 R 指预测正确的正样本占有所有正样本的比值;平均精确率 AP 表示 PR 曲线与坐标轴所围成的图形面积; $mAP@0.5$ 表示在 IoU 阈值为 0.5 时所有类别 AP 的平均值; $mAP@0.5:0.95$ 表示不同 IoU 阈值(从 0.5 到 0.95,步长 0.05)上的平均 mAP ;参数量用来衡量模型大小,参数量和计算量越小说明模型越轻量;帧率表示每秒刷新的图片数,可以用来评估模型在检测时的效率。

3.3 实验平台

本次实验使用的硬件平台 GPU 为 NVIDIA RTX3090,显存 24 GB,CPU 为 Intel(R) Xeon(R) Platinum 8350C@ 2.60 GHz,内存 45 GB。系统环境为 Linux,CUDA 版本 11.1,Pytorch 版本 1.8.1,Python 版本 3.8, $batch-size$ 设置为 16,进行 200 个 $epoch$ 迭代。超参设置为系统默认,即初始学习率设置为 0.01,使用 SGD 优化器,初始动量设置为 0.937,权重衰减系数设置为 0.000 5。

3.4 实验结果及分析

训练过程如图 9 所示,由图 9 可以看出,改进后的 YOLOv5 大概在 130 个轮次之后达到峰值并趋于稳定,而原始 YOLOv5 则在 160 个轮次之后才趋于稳定状态,并且改进后的 YOLOv5 训练过程更平稳,极少出现大幅度变化过程,说明改进后的 YOLOv5 具有更良好的性能。

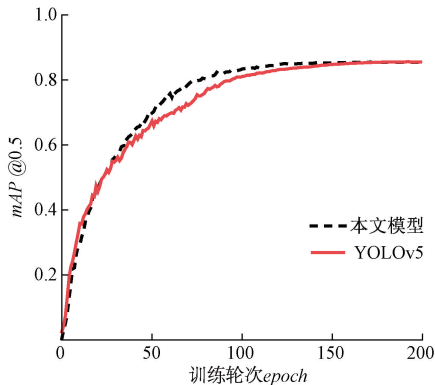


图 9 训练过程曲线

Figure 9 Training process curve

经过实验分析,原始 YOLOv5 模型与改进后的 YOLOv5 模型在 TT100K 数据集上的结果对比如表 1 所示。

由表 1 可以看出,改进后的 YOLOv5 模型的 P 下降了 0.8 百分点, R 提高了 2.8 百分点,在参数量减少 29%、计算量减少 29.4% 的情况下, $mAP@0.5$

表 1 改进后的 YOLOv5 模型与原始 YOLOv5 模型在 TT100K 数据集上的结果对比

Table 1 Comparison of results between the improved YOLOv5 model and the original YOLOv5 model on the TT100K dataset							
模型	$P/\%$	$R/\%$	$mAP@0.5/\%$	$mAP@0.5:0.95/\%$	参数量/ 10^6	计算量/ GFLOPs	帧率/ (帧 $\cdot s^{-1}$)
YOLOv5	84.8	78.4	85.5	63.9	7.07	16.15	175
本文模型	84.0	81.2	85.4	62.9	5.02	11.40	209

仅减少了 0.1 百分点, $mAP@0.5:0.95$ 仅减少了 1.0 百分点,但检测的速率提升了 34 帧。达到了几乎不损失精确率的前提下,简化 YOLOv5 网络的目的。

其中每个类别的 AP 值如表 2 所示,由表 2 可以看出,有 50% 的标签在改进后网络中的平均精确率高于原始网络,说明尽管参数量和计算量有所下降,但是由于 BiFPN 结构和 SIoU 损失的加入,也使得某些特征的学习能力得到加强。

表 2 每个类别 AP 值

Table 2 AP value of each category			%		
类别	YOLOv5	本文模型	类别	YOLOv5	本文模型
i2	85.2	86.8	pl100	93.4	88.6
i4	93.1	91.8	pl120	91.4	83.3
i5	95.0	96.4	pl30	78.4	80.5
il60	98.0	97.8	pl40	79.6	78.1
il80	90.7	90.5	pl5	86.3	86.4
io	82.3	84.8	pl50	73.2	80.3
ip	81.9	85.4	pl60	76.0	67.5
p10	77.2	76.7	pl80	82.5	84.3
pll	77.9	81.4	pn	92.3	93.4
p23	87.0	83.6	pne	95.5	97.0
p26	89.2	87.7	po	77.3	75.5
p5	95.7	94.2	w57	72.3	78.9

为了进一步研究各个模块对网络的影响,本文设置了消融实验证明改进后网络的各模块带来的影响,具体数据如表 3 所示。

表 3 消融实验

Table 3 Ablation experiment			
模型	$mAP@0.5/\%$	参数量/ 10^6	计算量/ GFLOPs
YOLOv5	85.5	7.07	16.15
YOLOv5+Ghost	83.2	5.03	11.30
YOLOv5+BiFPN	86.1	7.14	16.20
YOLOv5+SIoU	85.7	7.07	16.15
YOLOv5+Ghost+ BiFPN+SIoU	85.4	5.02	11.40

由表 3 可以看出,单独增加 Ghost 模块后,整体参数量和计算量都会大幅减少,但同时也会导致 $mAP@0.5$ 降低 2.3 百分点;单独增加 BiFPN 结构会在稍微提高参数量和计算量的同时提高 0.6 百分点的 $mAP@0.5$;单独增加 SIoU 损失函数则

不会对参数量和计算量有所影响,却能带来 0.2 百分点的 $mAP@0.5$ 提升。当 3 种改进同时添加到 YOLOv5 模型之后,在参数量降低 29.0%、计算量降低 29.4% 的同时仅仅牺牲 0.1 百分点 $mAP@0.5$ 。所以该改进方法是可行的。

本文使用 SSD 模型、Faster R-CNN 模型、YOLOv3-tiny 模型、YOLOv4 模型以及本文提出的模型进行对比,实验结果如表 4 所示。

表 4 5 种算法性能对比

Table 4 Performance of five algorithms			
模型	$mAP@0.5/\%$	帧率/ (帧 $\cdot s^{-1}$)	权重大小/ MB
SSD	79.1	28.4	159.0
Faster R-CNN	92.9	14.2	100.2
YOLOv3-tiny	83.4	120.8	33.8
YOLOv4	87.8	20.7	246.0
本文模型	85.5	209.0	10.0

由表 4 可知,相比其他几项传统的目标检测算法,本文提出的轻量化模型在保持较高的检测精度的基础上,检测速度最快、权重最小。

为了验证轻量化模型在推理过程中的优势,本文设计了推理过程实验,考虑到现实使用时很少拥有高性能 GPU 资源,因此推理过程使用中等偏下配置的平台。推理所使用的 GPU 为 GeForce 1660super,CPU 为 Intel i5-10400,软件环境与训练一致。使用原始 YOLOv5 与轻量化模型进行对比,以模型大小和推理时间作为评价指标,实验结果如表 5 所示。

表 5 推理实验

Table 5 Interface experiments			ms
模型	GPU 推理用时	CPU 推理用时	
YOLOv5	26.3	167.6	
本文模型	18.6	110.5	

由表 5 可知,在推理实验中,本文提出的轻量化模型在使用 GPU 推理时间相比原始模型减少了 7.7 ms,使用 CPU 推理时间减少了 57.1 ms。实验证明改进后的轻量化模型在 GPU 和 CPU 上的推理时间均有所降低。

图 10 为使用改进前后模型检测不同图片的结

果。图 10(a)、10(b) 是有多个目标的图片检测结果对比,可以看出除了“限低速 60”的检测精度下降之外,其标志的检测结果均有不同幅度的提升;而“限低速 100”两次都没被检测出来是因为本次训练集中未包含该标签。图 10(c)、10(d) 则是只包含

单个检测目标的图片检测结果对比,图 10(c) 发现有误检结果,将远处一个医院的红十字错误检测成“限速 50”的标志,经改进后的网络检测结果如图 10(d) 所示,并没有误检现象出现,且“禁止掉头”的检测精度有略微提升。

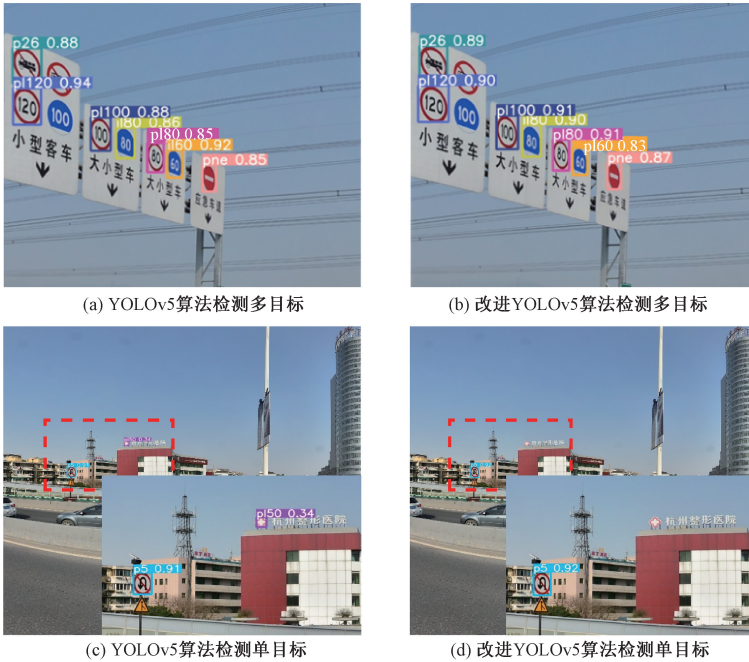


图 10 改进前后检测结果

Figure 10 Results before and after improvement

4 结论

改进后的 YOLOv5 在参数量降低 29.0%、计算量降低 29.4% 的情况下, $mAP@0.5$ 仅下降了 0.1 百分点,同时检测帧率提升了 34 帧/s,取得较好的效果。

(1)使用参数量大幅减少的 Ghost 卷积和深度可分离卷积构建 GhostBottleneck,替换主干网络和颈部中的部分 Bottleneck,简化网络参数和计算量。

(2)使用 BiFPN 替换原始的 PANet 结构,增强特征融合能力,提高检测准确率。

(3)将原始 CIoU 损失替换成更关注预测框与真实框角度位置的 SIoU 损失,优化损失函数,提升检测精度。

下一步将考虑在进一步降低参数量和计算量的基础上提高检测精度,加入注意力机制等改进方案,使模型能够部署到移动设备进行实际运用。

参考文献:

[1] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // 2014 IEEE Conference on

Computer Vision and Pattern Recognition. Piscataway: IEEE, 2014: 580–587.
[2] GIRSHICK R. Fast R-CNN[C]//2015 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2016: 1440–1448.
[3] 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.
[4] HE K M, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]//2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2017: 2980–2988.
[5] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway:IEEE, 2016: 779–788.
[6] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2017: 6517–6525.
[7] REDMON J, FARHADI A. YOLOV3: An incremental improvement[C]//Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition. Pisca-

taway: IEEE, 2018: 1–6.

[8] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: optimal speed and accuracy of object detection [EB/OL]. (2020–04–23) [2023–12–18]. <https://arxiv.org/abs/2004.10934>.

[9] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector [C] // European Conference on Computer Vision. Cham: Springer, 2016: 21–37.

[10] SUDHA M, GALDIS PUSHPARATHI D V P. Traffic sign detection and recognition using RGSM and a novel feature extraction method [J]. Peer-to-Peer Networking and Applications, 2021, 14(4): 2026–2037.

[11] HAN C, GAO G Y, ZHANG Y. Real-time small traffic sign detection with revised faster-RCNN [J]. Multimedia Tools and Applications, 2019, 78(10): 13263–13278.

[12] WAN J X, DING W, ZHU H L, et al. An efficient small traffic sign detection method based on YOLOv3 [J]. Journal of Signal Processing Systems, 2021, 93(8): 899–911.

[13] 尹靖涵, 瞿绍军, 姚泽楷, 等. 基于 YOLOv5 的雾霾天气下交通标志识别模型 [J]. 计算机应用, 2022, 42(9): 2876–2884.

YIN J H, QU S J, YAO Z K, et al. Traffic sign recognition model in haze weather based on YOLOv5 [J]. Journal of Computer Applications, 2022, 42(9): 2876–2884.

[14] 张毅, 龚致远, 韦文闻. 基于改进 Faster R-CNN 模型的交通标志检测 [J]. 激光与光电子学进展, 2020, 57(18): 173–181.

ZHANG Y, GONG Z Y, WEI W W. Traffic sign detection based on improved faster R-CNN model [J]. Laser & Optoelectronics Progress, 2020, 57(18): 173–181.

[15] 李宇琼, 周永军, 蒋淑霞, 等. 基于注意力机制的交通标志识别 [J]. 电子测量技术, 2022, 45(8): 116–120.

LI Y Q, ZHOU Y J, JIANG S X, et al. Traffic sign recognition based on attention mechanism [J]. Electronic Measurement Technology, 2022, 45(8): 116–120.

[16] 郭继峰, 孙文博, 庞志奇, 等. 一种改进 YOLOv4 的交通标志识别算法 [J]. 小型微型计算机系统, 2022, 43(7): 1471–1476.

GUO J F, SUN W B, PANG Z Q, et al. Improved traffic sign recognition algorithm for YOLOv4 [J]. Journal of Chinese Computer Systems, 2022, 43(7): 1471–1476.

[17] LIU S, QI L, QIN H F, et al. Path aggregation network for instance segmentation [C] // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018: 8759–8768.

[18] HAN K, WANG Y H, TIAN Q, et al. GhostNet: more features from cheap operations [C] // 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 1577–1586.

[19] TAN M X, PANG R M, LE Q V. EfficientDet: scalable and efficient object detection [C] // 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 10778–10787.

[20] GEVORGYAN Z. SiIoU loss: more powerful learning for bounding box regression [EB/OL]. (2022–05–25) [2023–02–18]. <https://arxiv.org/abs/2205.12740>.

[21] ZHU Z, LIANG D, ZHANG S H, et al. Traffic-sign detection and classification in the wild [C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2016: 2110–2118.

Traffic Sign Detection Based on Lightweight YOLOv5

ZHANG Zhen¹, WANG Xiaojie¹, JIN Zhihua¹, MA Jijun²

(1. School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China; 2. Henan Province Transportation Dispatching Command Center, Zhengzhou 450001, China)

Abstract: In order to improve the detection speed of road traffic signs, an improved model based on lightweight YOLOv5 was proposed. Firstly, Ghost convolution and depthwise convolution were used to build a new Bottleneck, which could reduce the amount of computation and parameters. Then the BiFPN structure was introduced, which could enhance the feature fusion ability. CIoU loss function was replaced by SiIoU loss function, which focused on the angle information of ground true box and prediction one, so that it would improve the detection accuracy. Secondly, the TT100K dataset was optimized, and 24 categories of traffic sign pictures and labels with more than 200 were screened out. Finally, the experiment achieved 84% accuracy, 81.2% recall and 85.4% *mAP@0.5*. Compared with the original YOLOv5 model, the number of parameters was reduced by 29.0%, the amount of computation was reduced by 29.4%, but the *mAP@0.5* was only reduced by 0.1 percentages, and the detection frame rate was improved by 34 frames/s. Using the improved model for detection, the detection speed could be significantly improved, could basically achieve the goal of compression model on the basis of maintaining the detection accuracy.

Keywords: traffic sign detection; lightweight YOLOv5; SiIoU loss function; Ghost convolution; TT100K; BiFPN