改进 YOLOv10n 的轻量化道路裂缝检测模型

王井阳[1], 徐勇超[1], 张　波[2], 王　珏[3], 黄　敏[1]

(1.河北科技大学 信息科学与工程学院,河北 石家庄 050018;2.河北工程技术学院 网络空间安全学院,河北 石家庄 050091;3.中国电信股份有限公司 石家庄分公司,河北 石家庄 050035)

摘　要:针对现有道路裂缝检测模型不能有效平衡检测精度、计算复杂度与检测速度,实际应用效果差的问题,提出了一种基于改进 YOLOv10n 的轻量级道路裂缝检测模型 YOLO-CGVE。首先,利用坐标注意力(CA)模块替换部分自注意力(PSA)模块,从而更好地捕捉空间上的局部和全局关系,增强特征提取能力;其次,通过使用轻量级的 GSConv 替换主干网络和颈部网络中的部分标准卷积,降低了计算复杂度;再次,在颈部网络采用 VoV-GSCSP 模块替换 C2f 模块,实现对不同阶段的特征图的有效融合,在保证精度的同时进一步降低计算复杂度;最后,使用 ECIoU 代替原损失函数,提高检测框定位精度和收敛速度。在 RDD2022_China 数据集上的实验结果表明,相较于 YOLOv10n,YOLO-CGVE 的 *mAP*@0.5 提高了 2.4 百分点,达到了 75.9%,参数量与计算量分别减少了 11.1%和 9.8%,同时保持了较高的检测速度。YOLO-CGVE 可以更好地满足在计算资源有限环境下的应用需求。

关键词:道路裂缝检测; YOLOv10n; 注意力机制; 损失函数; 轻量化

中图分类号: TP391.4;U418.6　　**文献标志码**:A　　**doi**:10.13705/j.issn.1671-6833.2026.04.007

道路裂缝是道路病害中最严重的问题之一,对交通安全和道路可靠性具有重大影响。裂缝如果不能被及时发现,就很有可能扩展甚至造成路面断裂、塌陷,对车辆、行人构成潜在威胁,并显著缩短道路的使用寿命。因此,裂缝检测不仅是道路养护的重要组成部分,更是预防因道路裂缝导致交通事故的重要手段[1-2]。

传统的检测方法存在检测准确率低、漏检率高、检测效率低等不足,对道路养护工作造成了一定程度的困扰。近年来,基于深度学习的目标检测方法已成为道路裂缝检测的主流方法。根据算法流程,基于深度学习的目标检测算法大致分为两类:两阶段目标检测算法和一阶段目标检测算法[3]。两阶段目标检测算法的代表性算法有 R-CNN[4]和 Faster R-CNN[5]等,它们具有较高的检测精度,但检测速度较慢,且需要较为庞大的计算资源,不适合部署到计算资源有限的设备上[6]。一阶段目标检测算法主要有 YOLO[7]系列和 SSD[8]等,具有较快的检测速度,计算资源需求量较小,在实时应用场景下具有更大优势[9-10]。两种目标检测算法,近年来都有被研究人员应用在道路裂缝检测模型上的成功案例。Li 等[11]提出了一种基于改进 Faster-RCNN 算法的道路破损检测方法,使用可变形卷积替换特征提取的卷积,并将特征金字塔网络最后一层操作替换为空洞卷积,扩大了感受野,增强了模型对特征的表达能力,提升了检测精度,但该方法的计算复杂度较高。Wang 等[12]提出了一种改进的 YOLOv8s 道路缺陷检测模型,利用 EMA Faster Block 来替换 C2f 模块中的 Bottleneck,并引入 SimSPPF 代替 SPPF,还使用 Detect-Dyhead 替换了原来的检测头,提升了检测精度,但模型的计算量较高。Youwai 等[13]提出了基于 YOLOv9 的路面损伤检测模型 YOLO9tr,通过融入部分注意力模块,增强了模型的特征提取能力和复杂场景下的检测性能,但模型的参数量和计算量较大。此外,近年来兴起的基于 Transformer 的 DETR[14]及其改进版本 RT-DETR[15]等模型也被用于道路裂缝

收稿日期:2025-10-24;**修订日期**:2025-11-28

基金项目:国防科技重点实验室基金项目(6142205240201)

作者简介:王井阳(1971—),男,河北迁西人,河北科技大学教授,主要从事人工智能、计算机视觉研究,E-mail:ever211@163.com。

通信作者:张波(1985—),女,河北石家庄人,河北工程技术学院副教授,主要从事深度学习、计算机视觉研究,E-mail:zhangbo8503@hebuet.edu.cn。

检测。Luo 等[16]提出一种基于 RT-DETR 的多尺度信息交叉融合的道路损伤检测模型 MMR-DETR,引入了 M2SA 和 MCF 模块增强特征表达和检测性能,通过优化检测框来提高定位精度,但该模型的参数量和计算量远超过 YOLO 模型。

尽管已有许多道路裂缝检测模型被提出,也取得了一定的成效,但在实际的道路裂缝检测任务中,检测模型在参数量与计算量、检测精度以及检测速度方面往往难以取得较好的平衡。同时,对于 YOLO 模型,由于更新迭代速度较快,在检测精度和轻量化上仍有一定的提升空间。因此,针对以上问题,本文在 YOLOv10n 的基础上提出了一种改进的 YOLOv10n 轻量化道路裂缝检测模型 YOLO-CGVE,旨在提高道路裂缝检测精度,改善漏检情况以及提高在计算资源受限环境中的适用性和实用性。

1 YOLO-CGVE 模型

Wang 等[17]提出的 YOLOv10 在实时目标检测领域取得了显著成效,提供了更高的效率和精度,适用于多种实际应用场景。YOLOv10 提供多种模型版本以满足不同应用需求,其中 YOLOv10n 推理速度较快,对设备要求最低,适合在计算资源有限的设备上部署,应用较为广泛,故本文在 YOLOv10n 的基础上进行改进。改进模型 YOLO-CGVE 的网络结构如图 1 所示,其中实线框部分是本文改进的部分。

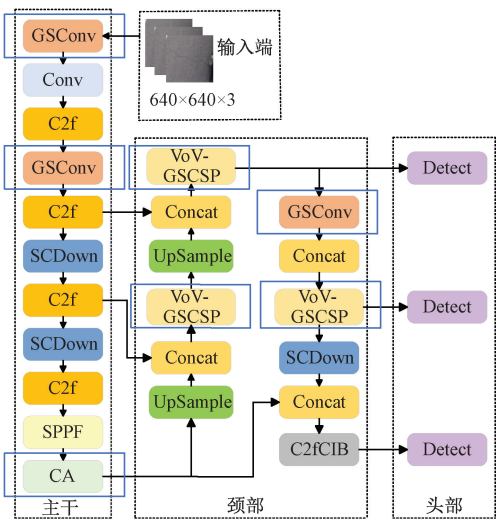


图 1 YOLO-CGVE 网络结构

Figure 1 Network framework of YOLO-CGVE

1.1 CA 模块

为增强复杂背景环境下道路裂缝检测的准确性,提高模型的特征提取能力,引入了 CA[18]模块,结构如图 2 所示。

与部分自注意力 PSA 相比,CA 不仅考虑输入

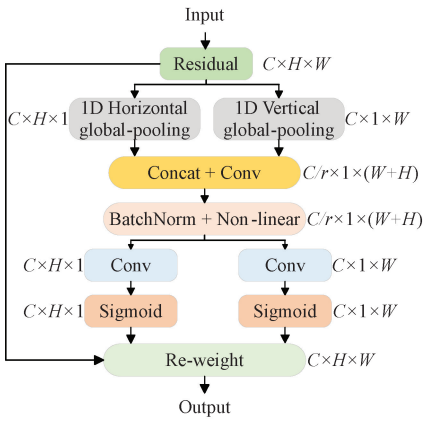


图 2 CA 模块结构

Figure 2 Structure of CA module

的特征信息,还考虑每个像素点的位置信息,具备对各通道特征图进行自适应加权的能力,既能提升重要特征通道的表达能力,也能有效抑制非相关信息带来的干扰;将水平和垂直方向上的信息编码到通道注意力中,使得模型能够更好地捕捉空间上的局部和全局关系,这对模型检测具有不规则、曲折特点的裂缝目标更加有利。同时,CA 灵活性强,避免了较大的计算开销,不增加模型的复杂度,更适合轻量化网络结构的设计。

当给定一个大小为 $C \times H \times W$(通道数为 C,高为 H,宽为 W)的输入特征图 $\boldsymbol{x}$,CA 模块首先对 $\boldsymbol{x}$ 在水平和垂直方向上的各个通道分别使用大小为 $(H,1)$ 和 $(1,W)$ 的池化核进行编码,获得一对大小分别为 $C \times H \times 1$ 和 $C \times 1 \times W$ 的方向感知特征图 $\boldsymbol{z}_c^h$ 和 $\boldsymbol{z}_c^w$,计算公式如下所示:

$$\boldsymbol{z}_c^h = \frac{1}{W}\sum_{0 \leqslant i < W} \boldsymbol{x}_c(h,i); \tag{1}$$

$$\boldsymbol{z}_c^w = \frac{1}{H}\sum_{0 \leqslant j < H} \boldsymbol{x}_c(j,w)。 \tag{2}$$

式中:c 表示第 c 个通道;i 和 j 为 $\boldsymbol{x}$ 的空间维度(宽度和高度)上的位置索引;$\boldsymbol{x}_c(h,i)$ 和 $\boldsymbol{x}_c(j,w)$ 分别表示在 (h,i) 和 (j,w) 处的特征;$\boldsymbol{z}_c^h$ 和 $\boldsymbol{z}_c^w$ 分别表示第 c 个通道在高度 h 处和宽度 w 处的输出。

然后将获得的两个不同方向的特征图拼接在一起,经过一个 1×1 卷积将通道数减至 C/r,其中 r 表示控制减少率的系数。再进行批量归一化和应用非线性激活函数,得到大小为 $C/r \times 1 \times (W+H)$ 的新的特征图。计算公式为

$$\boldsymbol{f} = \delta(F_1([\boldsymbol{z}^h, \boldsymbol{z}^w]))。 \tag{3}$$

式中:δ 为非线性激活函数;F_1 表示 1×1 卷积;$[\boldsymbol{z}^h, \boldsymbol{z}^w]$ 表示沿空间维度的拼接操作;$\boldsymbol{f}$ 为输出特征图。

接下来将特征图在水平和垂直方向上进行特征

分离,得到两个方向上独立的张量 $\boldsymbol{f}^h$ 和 $\boldsymbol{f}^w$。然后通过 1×1 卷积将通道数调整至 C,再通过 sigmoid 激活函数得到两个方向上的权重 $\boldsymbol{g}^h$ 和 $\boldsymbol{g}^w$。$\boldsymbol{g}^h$ 和 $\boldsymbol{g}^w$ 的计算公式如下所示:

$$\boldsymbol{g}^h = \sigma(F_h(\boldsymbol{f}^h)); \tag{4}$$

$$\boldsymbol{g}^w = \sigma(F_w(\boldsymbol{f}^w))。 \tag{5}$$

式中:σ 表示 sigmoid 激活函数;F_h 和 F_w 分别为水平方向上和垂直方向上的 1×1 卷积操作。

最后,将 $\boldsymbol{g}^h$ 和 $\boldsymbol{g}^w$ 拓展为与 (i,j) 位置处的输入特征 $\boldsymbol{x}_c(i,j)$ 相同的维度,并与输入特征按位相乘得到加上注意权重的输出特征 $\boldsymbol{y}_c(i,j)$,计算公式为

$$\boldsymbol{y}_c(i,j) = \boldsymbol{x}_c(i,j) \times \boldsymbol{g}_c^h(i) \times \boldsymbol{g}_c^w(j)。 \tag{6}$$

1.2 GSConv 模块

为了能够在保证模型性能的前提下同时减少计算开销,使用 GSConv[19] 替换部分标准卷积 Conv。与标准卷积相比,GSConv 的计算成本更低,同时保持了具有竞争力的性能。GSConv 和 Conv 的计算成本以及两者的比值 $ratio_c$ 的计算公式[19]如下所示:

$$Cost_{\mathrm{GSConv}} = \frac{1}{2}WHK_1K_2(C_1 + 1)C_2; \tag{7}$$

$$Cost_{\mathrm{Conv}} = WHK_1K_2C_1C_2; \tag{8}$$

$$ratio_c = \frac{Cost_{\mathrm{GSConv}}}{Cost_{\mathrm{Conv}}} = \frac{C_1 + 1}{2C_1}。 \tag{9}$$

式中:$Cost_{\mathrm{GSConv}}$ 和 $Cost_{\mathrm{Conv}}$ 分别为 GSConv 和 Conv 的计算成本;K_1 和 K_2 均为卷积核的大小;C_1 和 C_2 分别为输入特征图的通道数和输出特征图的通道数。

GSConv 的结构如图 3 所示。GSConv 的基本思想是利用混洗(shuffle)操作将标准卷积 Conv 产生的信息与深度可分离卷积(depth-wise separable convolution,DWConv)产生的信息进行混合,各通道彼此间可达成局部特征信息的均匀交换,以此保留更丰富的通道间连接,进而降低模型的计算复杂度。

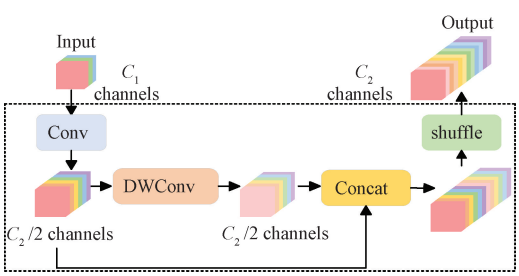


图 3 GSConv 结构

Figure 3 Structure of GSConv

GSConv 的实现过程如下。将输入通道数为 C_1 的特征图通过一个标准卷积得到 $C_2/2$ 通道数的特征图;将此特征图经过 DWConv 得到通道数相同的新的特征图,通过这一操作过程构建了信息的协同交互与有效融合机制,进一步强化特征图的表征强度及信息传递性能;将得到的两组通道数相同的特征图进行拼接;进行混洗操作,用较低的计算成本使得不同通道间的信息进行交互;最终形成通道数为 C_2 的输出特征图。

GSConv 对于轻量级模型效果更明显,但如果将模型中的标准卷积 Conv 都替换成 GSConv,则模型层数会变得很多,增加推理时间。因此,本文将主干网络中的部分 Conv 以及颈部网络中的 Conv 进行替换,尽可能在提升检测精度的同时降低计算成本,帮助模型更好地适配实时检测任务。

1.3 VoV-GSCSP 模块

为进一步降低模型的计算量和网络结构复杂性,在保持模型轻量化的前提下,保证对不同阶段的特征图进行有效的信息融合,采用 VoV-GSCSP[19] 模块替换颈部网络中的 C2f 结构。

VoV-GSCSP 是在 GSConv 的基础上引入 GS 瓶颈(GS bottleneck)[19],然后采用一次性聚合的方法设计出的跨阶段部分网络(cross stage partial network,CSP)模块。该模块通过改进瓶颈层,提高了特征的非线性表达能力,提升了特征复用率并减少了计算复杂度。此外,还通过跨阶段部分网络结构在特征传递中保留了较多原始信息,并通过分流后再融合的方式,增强了深层网络的梯度流传递能力,减少了冗余计算,进一步优化了计算效率。VoV-GSCSP 在性能上接近 C2f,但更为轻量化。GS bottleneck 与 VoV-GSCSP 模块的结构见图 4。

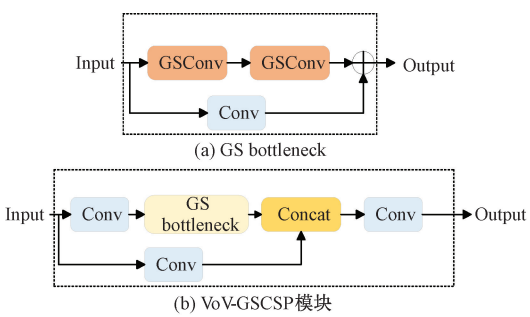


图 4 GS bottleneck 与 VoV-GSCSP 模块结构

Figure 4 Structures of GS bottleneck and VoV-GSCSP module

算法 1:VoV-GSCSP 模块实现过程。

输入:输入特征图 $\boldsymbol{X}$(通道数为 C_1);

输出:输出特征图 $\boldsymbol{Y}$(通道数为 C_2)。

① 模块初始化:将参数 C_1,C_2,$e = 0.5$ 输入模块,其中 e 表示通道缩放因子,用于控制隐藏通道数与输出通道数的比例关系;

② 主干分支:先通过 1×1 卷积将 $\boldsymbol{X}$ 的通道数从 C_1

压缩至隐藏通道数 $C\_ = \text{int}(C_2 \times e)$,得到特征图 $\boldsymbol{X}_1$;然后通过 GS bottleneck 模块处理特征得到特征图 $\boldsymbol{X}_2$;

③ 残差分支:通过 1×1 卷积将 $\boldsymbol{X}$ 的通道数从 C_1 压缩至 $C\_$,得到 $\boldsymbol{X}_3$;

④ 拼接两路特征:将 $\boldsymbol{X}_3$ 与 $\boldsymbol{X}_2$ 沿通道维度进行拼接(Concat),得到通道数为 $2C\_$ 的特征图 $\boldsymbol{X}_4$;

⑤ 通道恢复至 C_2:通过 1×1 卷积将特征图 $\boldsymbol{X}_4$ 的通道数从 $2C\_$ 恢复至 C_2,得到输出特征图 $\boldsymbol{Y}$。

1.4 ECIoU 损失函数

道路裂缝具有不同的尺寸和形状,这种多样性和复杂性会对模型的边界框预测带来挑战。同时,由于直接对模型进行一系列的轻量化处理可能导致检测性能下降,为此对损失函数进行改进。

YOLOv10 采用 CIoU 作为回归损失函数,具体计算公式[20]如下所示:

$$\text{IoU} = \frac{A \cap B}{A \cup B}; \tag{10}$$

$$\alpha = \frac{v}{(1 - \text{IoU}) + v}; \tag{11}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{\text{gt}}}{h^{\text{gt}}} - \arctan\frac{w}{h}\right)^2; \tag{12}$$

$$L_{\text{CIoU}} = 1 - \text{IoU} + (\rho^2(b, b^{\text{gt}})/d^2) + \alpha v。 \tag{13}$$

式中:IoU 为预测框和真实框的交并比;α 表示权重系数;v 为用来衡量长宽比一致性的参数;w^{gt} 和 h^{gt} 分别为真实框的宽和高;w 和 h 分别为预测框的宽和高;L_{CIoU} 为 CIoU 损失函数;ρ 为两个框中心点之间的距离;b 和 b^{gt} 为预测框和真实框;d 为能够同时包含两框的最小闭合区域的对角距离。

CIoU 使用的是宽高比而非真实宽高,可能导致预测框长宽比难以收敛,影响性能。EIoU[21]通过拆分长宽差异,弥补了 CIoU 的缺陷,但当遇到具有极大差异的边时,EIoU 的计算可能会变得相对较慢,且不一定能提前收敛。为此,Chen 等[22]提出了增强型损失函数 ECIoU 以进一步优化性能。ECIoU 结合了 CIoU 和 EIoU 的优势,这种组合策略不仅提高了预测框的精度,还加快了回归任务的收敛速度,使得模型在处理具有挑战性的场景时能够表现出更好的性能。ECIoU 的计算公式[22]为

$$L_{\text{ECIoU}} = 1 - \text{IoU} + \alpha v + \frac{\rho^2(b^{\text{gt}}, b)}{d^2} + \frac{\rho^2(h^{\text{gt}}, h)}{d_h^2} + \frac{\rho^2(w^{\text{gt}}, w)}{d_w^2}。 \tag{14}$$

式中:L_{ECIoU} 为 ECIoU 损失函数;d_h 和 d_w 分别为最小闭合区域的高度和宽度。

2 实验结果与分析

2.1 数据集

实验使用的数据集为 RDD2022[23]中来自中国的 4 878 张道路图像,其中包含 2 401 张由无人机拍摄的道路图像;2 477 张由固定在摩托车上的智能手机拍摄的道路图像。为了方便实验记录,将该数据集命名为 RDD2022_China。鉴于该数据集存在明显的类别不平衡问题,对数据集进行了数据清洗工作,去除了数量极其有限的坑洼(D40)标签和无关的修补(Repair)标签,保留本文重点关注的 3 类道路裂缝标签,分别为纵向裂缝(D00)、横向裂缝(D10)和网状裂缝(D20)。最终按照 8∶1∶1重新将数据集划分为训练集、验证集和测试集。

2.2 实验环境

实验所使用的 GPU 为 NVIDIA Tesla V100 S-PCIE,显存为 8 GB,CUDA 版本为 11.8,编程语言为 Python 3.10,深度学习框架采用 Pytorch 2.0.1。模型训练过程中的参数设置如表 1 所示。

表 1 训练过程中的参数设置

Table 1 Parameter settings in the training process

参数	设定
数据增强方法	Mosaic
输入图片尺寸	640 像素×640 像素
批次大小	16
训练轮数	200
优化器	AdamW
动量参数	0.937
初始学习率	0.001 49
权重衰减	0.000 5

2.3 消融实验

为了验证每种改进策略对模型性能的影响,进行了消融实验。实验结果如表 2 所示。

由表 2 的实验结果可知,当使用 CA 替换 PSA 后,*Precision* 提高了 4.1 百分点,*mAP*@0.5 提高了 1.2 百分点,参数量和计算量都有所减少,说明 CA 能够在避免较大计算开销的前提下自适应地对不同通道特征图进行加权,增强了重要特征通道的表达能力,在降低计算复杂度的同时提升了检测精度。当使用 GSConv 替换部分 Conv 后,*Precision* 提高了 6.5 百分点,*mAP*@0.5 提高了 0.5 百分点,参数量和计算量均有所减少,说明 GSConv 通过利用 shuffle 操作将 Conv 产生的信息与 DWConv 产生的信息进行混合,用较低的计算成本使得不同通道间的信息进行了交互,在提升性能的同时降低了计算复杂度。采用 VoV-GSCSP 改进策略后,*Precision* 提高了 2.8

表 2　消融实验结果

Table 2　Results of ablation experiments

CA	GSConv	VoV-GSCSP	ECIoU	*Precision*/%	*Recall*/%	*mAP*@0.5/%	参数量/10^6	计算量/GFLOPs	帧率/(帧·s^{-1})
				68.9	70.3	73.5	2.7	8.2	118
√				73.0	70.9	74.7	2.6	8.0	125
	√			75.4	68.1	74.0	2.6	8.1	108
		√		71.7	68.9	73.8	2.6	7.7	95
			√	71.5	71.1	74.8	2.7	8.2	123
√	√			72.9	71.0	74.7	2.4	7.9	111
√		√		71.5	68.2	74.2	2.4	7.5	102
√			√	73.6	68.2	74.1	2.6	8.0	100
	√	√		74.8	64.8	73.3	2.5	7.6	103
	√		√	72.2	71.1	74.2	2.6	8.1	119
		√	√	73.2	70.8	74.8	2.6	7.7	101
√	√	√		75.5	70.4	74.3	2.4	7.4	118
√	√		√	69.1	71.3	73.8	2.4	7.9	116
√		√	√	75.3	69.8	75.1	2.4	7.5	103
	√	√	√	75.5	64.7	74.2	2.5	7.6	100
√	√	√	√	77.2	67.2	75.9	2.4	7.4	102

百分点,*mAP*@0.5 提高了 0.3 百分点,参数量减少了 3.7%,计算量减少了 6.1%,表明 VoV-GSCSP 模块通过引入 GS bottleneck 以及特殊的结构设计,提高了特征的非线性表达能力,提升了特征复用率并且减少了计算复杂度。当采用 ECIoU 损失函数后,*Precision* 提高了 2.6 百分点,*mAP*@0.5 提高了 1.3 百分点,帧率也有所增加。

在两种改进策略组合的消融实验中,当使用 GSConv 和 VoV-GSCSP 组合时,参数量和计算量分别降低了 7.4%和 7.3%,*Precision* 提高了 5.9 百分点,但 *Recall* 和 *mAP*@0.5 略有降低,说明频繁地对模型结构进行轻量化改进会对部分指标带来负面影响。

在 3 种改进策略组合的消融实验中,CA、VoV-GSCSP 和 ECIoU 组合时的 *mAP*@0.5 最高,达到了 75.1%,提高了 1.6 百分点,同时参数量和计算量分别降低了 11.1%和 8.5%。

最终改进的 YOLO-CGVE 模型相比于原始模型 YOLOv10n,*Precision* 达到了 77.2%,提高了 8.3 百分点,*mAP*@0.5 达到了 75.9%,提升了 2.4 百分点,参数量减少了 11.1%,计算量减少了 9.8%,同时帧率达 102 帧/s,能够满足实际需要。每种改进策略均提升了模型的整体性能,验证了每种改进策略均是有效的。

此外,为直观呈现损失函数 ECIoU 在收敛速度方面的优越性,对比 CIoU 与 ECIoU 训练过程中损失值的变化曲线,如图 5 所示。

从图 5 可以看出,相较于 CIoU,ECIoU 的损失值下降得更为迅速,并且更早达到平稳状态,说明 ECIoU 的收敛速度更快,效果更好。

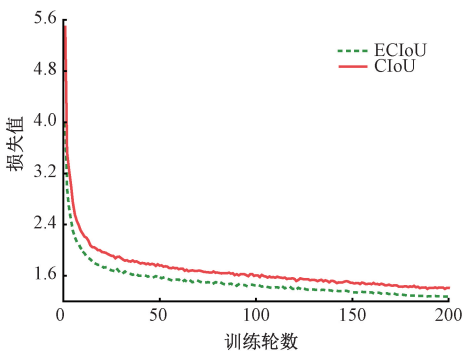


图 5　ECIoU 与 CIoU 的效果对比

Figure 5　Performance comparison of ECIoU and CIoU

2.4　对比实验

本文将 YOLO-CGVE 模型与其他几种主流检测模型以及其他研究人员提出的模型在相同的实验条件下进行了对比实验,以验证 YOLO-CGVE 模型性能的优越性。实验结果如表 3 所示。

结果表明,与 Faster R-CNN 相比,YOLO-CGVE 在各个方面都有非常明显的优势。与 SSD 相比,YOLO-CGVE 的 *mAP*@0.5 高出了 3.2 百分点,在参数量、计算量和检测速度方面也更有优势。对比 YOLOv8n,YOLO-CGVE 的 *Precision* 和 *mAP*@0.5 分别高出了 2.5 百分点和 3.4 百分点,在参数量和计算量方面也都更有优势。与 YOLOv9t 相比,YOLO-CGVE 的 *Precision* 和 *mAP*@0.5 分别高出了 4.9 百分点和 2.7 百分点,检测速度也更快,但在参数量和计算量方面,YOLOv9t 具有微弱优势。与 YOLOv11n

表 3　在 RDD2022_China 数据集上的对比实验结果

Table 3　Results of comparative experiments on RDD2022_China dataset

模型	*Precision*/%	*Recall*/%	*mAP*@0.5/%	参数量/10^6	计算量/GFLOPs	帧率/(帧·s^{-1})
Faster R-CNN	64.2	44.6	61.8	137.1	201.1	37
SSD	88.8	49.0	72.7	23.9	137.0	53
YOLOv8n	74.7	62.3	72.5	3.0	8.1	158
YOLOv9t	72.3	69.2	73.2	2.1	6.7	89
YOLOv10n	68.9	70.3	73.5	2.7	8.2	118
YOLOv11n	76.1	66.2	72.9	2.6	6.3	131
YOLOv12n	73.1	69.1	72.8	2.6	6.3	82
YOLO9tr[13]	73.7	69.9	74.2	10.1	40.2	40
MMR-DETR[16]	75.2	74.0	74.4	19.5	51.1	67
YOLO-CGVE	77.2	67.2	75.9	2.4	7.4	102

和 YOLOv12n 相比,YOLO-CGVE 的 *Precision* 分别高出了 1.1 百分点、4.1 百分点,*mAP*@0.5 分别高出了 3.0 百分点和 3.1 百分点。与 YOLO9tr 相比,YOLO-CGVE 的 *Precision* 和 *mAP*@0.5 分别高出了 3.5 百分点和 1.7 百分点,参数量和计算量更低。与 MMR-DETR 相比,YOLO-CGVE 的 *Precision* 和 *mAP*@0.5 分别高出了 2.0 百分点和 1.5 百分点,在参数量和计算量方面更具优势,检测速度也更快。综上所述,本文提出的改进模型在保持轻量化的同时,还具备较高的检测精度,更具优势。

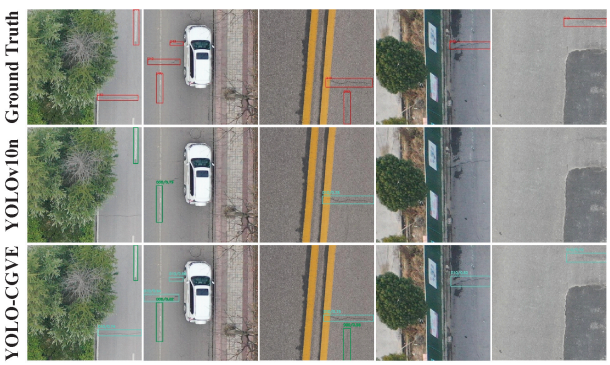
2.5　可视化分析

如图 6 所示,为了更直观地展示模型改进前后检测效果的比较,选取了数据集中的 10 张样本图像(图 6(a)中的原始图像由无人机拍摄,图 6(b)中的原始图像由智能手机拍摄)来进行检测效果验证,其中 Ground Truth 是指用来评估模型检测准确性的数据的真实标签。

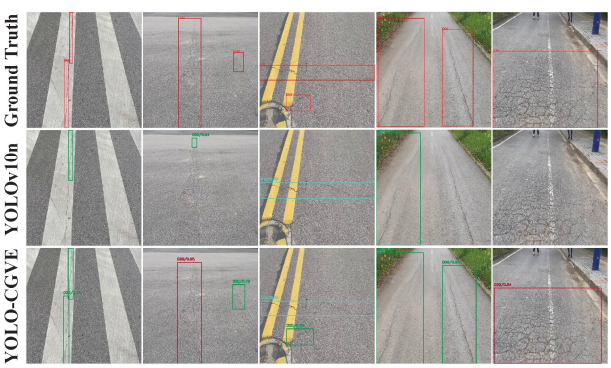
从图 6 的检测效果来看,YOLOv10n 在检测时易受到道路标线、绿化带或车辆等因素的影响,降低了对道路裂缝的关注度,从而导致了漏检情况的发生,针对较小和难以识别的目标,也存在着漏检和误检的情况,从而影响了检测精度。相较于 YOLOv10n,本文提出的改进模型 YOLO-CGVE 对不同环境下不同类型的裂缝都有更为出色的检测效果,漏检情况也得到了改善,说明本文对模型的改进提升了对裂缝目标的关注度以及复杂环境下的检测能力,从而提高了模型对道路裂缝的识别准确率。

2.6　泛化实验

为了探究 YOLO-CGVE 的泛化能力,利用该模型在 CRACK500[24-25] 数据集上的实验结果来分析,并与其他主流模型进行了对比,实验结果如表 4 所示。结果表明,与原始模型 YOLOv10n 相比,YOLO-CGVE 在检测性能上显著提升,其中 *Precision* 提高



(a) 在无人机拍摄图像上的检测效果



(b) 在智能手机拍摄图像上的检测效果

图 6　YOLOv10n 与 YOLO-CGVE 检测效果对比

Figure 6　Comparison of detection effects between YOLOv10n and YOLO-CGVE

了 9.1 百分点,达到了 71.1%,*mAP*@0.5 提高了 4.3 百分点,达到了 68.2%。然而 YOLO-CGVE 的 *Recall* 相对较低,这主要是由于模型为提升 *Precision* 以抑制道路裂缝检测中的误检情况,对“疑似裂缝正例”的判定标准更严格,导致部分边缘的裂缝目标或者低对比度的裂缝目标被误判为负例,最终造成 *Recall* 较 YOLOv10n 有所下降,但该下降幅度较小,未能抵消 *Precision* 与 *mAP*@0.5 提升带来的整体性能增加。与其他检测模型相比,YOLO-CGVE 仍表现出了更好的检测性能。实验结果表明,本文改进的模型在其他道路裂缝数据集上也表现良好。

表 4　在 CRACK500 数据集上的泛化实验结果

Table 4　Results of generalization experiments on CRACK500 dataset

模型	*Precision*/%	*Recall*/%	*mAP*@0.5/%	参数量/10^6	计算量/GFLOPs	帧率/(帧·s^{-1})
Faster R-CNN	28.6	85.9	57.4	137.1	201.1	46
SSD	63.2	40.9	52.6	23.9	137.0	37
YOLOv8n	64.4	68.7	61.1	3.0	8.1	138
YOLOv9t	65.0	64.4	65.5	2.1	6.7	81
YOLOv10n	62.0	63.3	63.9	2.7	8.2	105
YOLOv11n	61.1	68.0	65.6	2.6	6.3	123
YOLOv12n	60.8	62.0	62.1	2.6	6.3	86
YOLO9tr[13]	64.1	64.0	67.0	10.1	40.2	55
MMR-DETR[16]	70.4	66.7	67.2	19.5	51.1	57
YOLO-CGVE	71.1	62.0	68.2	2.4	7.4	92

3　结论

本文提出了一种轻量化的道路裂缝检测模型 YOLO-CGVE,通过使用 CA 模块替换 PSA 模块,增强了特征提取能力;通过使用 GSConv 替换部分 Conv,在保证模型检测性能的前提下,减少了计算开销;通过使用 VoV-GSCSP 替换颈部网络中的 C2f 模块,进一步降低计算复杂度;通过采用 ECIoU 代替 CIoU 损失函数,提升了检测精度和模型训练过程中的收敛速度。实验结果表明,改进后的模型与原始模型相比,*mAP*@0.5 达到了 75.9%,提高了 2.4 百分点,参数量与计算量分别减少了 11.1%和 9.8%,同时保持了较高的检测速度,并改善了漏检情况,泛化能力较强,满足实际应用的任务需求。

未来将进一步扩充道路裂缝数据集,特别是包含网状裂缝和复杂背景的图像。同时,改进模型的检测速度还有待提升,后续研究工作将探索使用知识蒸馏和剪枝技术对模型进行优化。

参考文献:

[1] SUN Z, ZHU L X, QIN S, et al. Road surface defect detection algorithm based on YOLOv8[J]. Electronics, 2024, 13(12): 2413.

[2] 王启涵, 刘超. 改进 YOLOv7-Tiny 的道路裂缝检测算法[J]. 计算机工程与应用, 2025, 61(10): 372-380.
WANG Q H, LIU C. Improved YOLOv7-tiny road crack detection algorithm[J]. Computer Engineering and Applications, 2025, 61(10): 372-380.

[3] 李军, 周科宇, 邹军, 等. 基于改进 YOLOv8n 的施工场景下防护装备佩戴检测算法[J]. 郑州大学学报(工学版), 2025, 46(3): 19-25, 104.
LI J, ZHOU K Y, ZOU J, et al. Protective equipment wearing detection algorithm in construction scenarios based on YOLOv8n[J]. Journal of Zhengzhou University (Engineering Science), 2025, 46(3): 19-25, 104.

[4] 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.

[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] YANG Z Y, LAN X, WANG H. Comparative analysis of YOLO series algorithms for UAV-based highway distress inspection: performance and application insights[J]. Sensors, 2025, 25(5): 1475.

[7] 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.

[8] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot MultiBox detector[C]// Computer Vision-ECCV 2016. Cham: Springer, 2016: 21-37.

[9] 王雪秋, 高焕兵, 郝泽萌. 改进 YOLOv8 的道路缺陷检测算法[J]. 计算机工程与应用, 2024, 60(17): 179-190.
WANG X Q, GAO H B, JIA Z M. Improved road defect detection algorithm based on YOLOv8[J]. Computer Engineering and Applications, 2024, 60(17): 179-190.

[10] 胡凤阔, 叶兰, 谭显峰, 等. 一种基于改进 YOLOv8 的轻量化路面病害检测算法[J]. 图学学报, 2024, 45(5): 892-900.
HU F K, YE L, TAN X F, et al. A refined YOLOv8-based algorithm for lightweight pavement disease detection[J]. Journal of Graphics, 2024, 45(5): 892-900.

[11] LI S B, HUANG Y J. Damage detection algorithm based on Faster-RCNN[C]//2023 5th International Conference on E-

lectronics and Communication, Network and Computer Technology (ECNCT). Piscataway: IEEE, 2023: 177-180.
[12] WANG J L, MENG R F, HUANG Y H, et al. Road defect detection based on improved YOLOv8s model[J]. Scientific Reports, 2024, 14: 16758.
[13] YOUWAI S, CHAIYAPHAT A, CHAIPETCH P. YOLO9tr: a lightweight model for pavement damage detection utilizing a generalized efficient layer aggregation network and attention mechanism[J]. Journal of Real-Time Image Processing, 2024, 21(5): 163.
[14] CARION N, MASSA F, SYNNAEVE G, et al. End-to-end object detection with transformers[C]//Computer Vision-ECCV 2020. Cham: Springer, 2020: 213-229.
[15] ZHAO Y A, LV W Y, XU S L, et al. DETRs beat YOLOs on real-time object detection[C]//2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2024: 16965-16974.
[16] LUO X L, LIU R C, HE X B, et al. A road damage detection model based on improved RT-DETR for complex environments[J]. IEEE Transactions on Instrumentation and Measurement, 2025, 74: 5037413.
[17] WANG A, CHEN H, LIU L H, et al. YOLOv10: real-time end-to-end object detection[EB/OL]. (2024-08-30)[2025-10-13]. https://arxiv.org/abs/2405.14458.
[18] HOU Q B, ZHOU D Q, FENG J S. Coordinate attention for efficient mobile network design[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2021: 13708-13717.
[19] LI H L, LI J, WEI H B, et al. Slim-neck by GSConv: a lightweight-design for real-time detector architectures[J]. Journal of Real-Time Image Processing, 2024, 21(3): 1-11.
[20] ZHENG Z H, WANG P, REN D W, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation[J]. IEEE Transactions on Cybernetics, 2022, 52(8): 8574-8586.
[21] ZHANG Y F, REN W Q, ZHANG Z, et al. Focal and efficient IOU loss for accurate bounding box regression[J]. Neurocomputing, 2022, 506: 146-157.
[22] CHEN X L, LIAN Q W, CHEN X L, et al. Surface crack detection method for coal rock based on improved YOLOv5[J]. Applied Sciences, 2022, 12(19): 1-18.
[23] ARYA D, MAEDA H, GHOSH S K, et al. RDD2022: a multi-national image dataset for automatic road damage detection[J]. Geoscience Data Journal, 2024, 11(4): 846-862.
[24] YANG F, ZHANG L, YU S J, et al. Feature pyramid and hierarchical boosting network for pavement crack detection[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(4): 1525-1535.
[25] ZHANG L, YANG F, DANIEL ZHANG Y, et al. Road crack detection using deep convolutional neural network[C]//2016 IEEE International Conference on Image Processing (ICIP). Piscataway: IEEE, 2016: 3708-3712.

Improved YOLOv10n Lightweight Road Crack Detection Model

WANG Jingyang[1], XU Yongchao[1], ZHANG Bo[2], WANG Jue[3], HUANG Min[1]

(1. School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China; 2. School of Cyberspace Security, Hebei University of Engineering Science, Shijiazhuang 050091, China; 3. Shijiazhuang Branch, China Telecom Corporation Limited, Shijiazhuang 050035, China)

Abstract: Aiming at the problem that the existing road crack detection model cannot effectively balance the detection accuracy, computational complexity and detection speed and has poor practical application effect, a lightweight road crack detection model YOLO-CGVE based on improved YOLOv10n was proposed. Firstly, the coordinate attention (CA) module was used to replace the partial self-attention (PSA) module to better capture the local and global relationships in space and improve the capacity to extract features. Secondly, the computational complexity was reduced by using lightweight GSConv to replace some standard convolutions in the backbone and neck networks. Thirdly, the original C2f structure in the neck network was replaced by VoV-GSCSP, which allowed for the efficient merging of feature maps from various stages and further minimized computing complexity while maintaining accuracy. Finally, the ECIoU loss function was used to replace the original loss function to improve the detection box positioning accuracy and convergence speed. The experimental results on RDD2022_China dataset showed that compared with YOLOv10n, while keeping a high detection speed, the *mAP*@0.5 of YOLO-CGVE was improved by 2.4 percentage points, reaching 75.9%, and the number of parameters and the amount of computation were decreased by 11.1% and 9.8%, respectively. YOLO-CGVE could better meet the application needs in environments with limited computing resources.

Keywords: road crack detection; YOLOv10n; attention mechanism; loss function; lightweight