

•智能交叉科学与工程•

DOI:10.12454/j.jsuese.202400302



本刊网刊

## MFF-YOLO:多尺度特征融合的轻量级道路缺陷检测算法

侯涛,张田明\*,牛宏侠

(兰州交通大学自动化与电气工程学院,甘肃兰州730070)

**摘要:**道路缺陷检测是道路养护的前提与基础,对道路安全保障非常重要。针对现有道路缺陷检测算法未能有效平衡检测精度与运算复杂度,从而难以应用于移动终端设备的问题,本文在YOLOv7-tiny的基础上提出了一种多尺度特征融合(multiscale feature fusion, MFF)的轻量级道路缺陷检测算法MFF-YOLO。首先,设计多尺度特征融合模块(MFFBlock)与下采样模块(DSB),并在此基础上构建高效的骨干多尺度特征提取网络(MFENet),以增强多尺度特征的提取能力。然后,颈部特征融合网络采用Slim-Neck设计范式,即利用GSCov与VoV-GSCSPC模块实现颈部特征的聚合,降低计算复杂度,轻量化网络。同时,颈部特征融合网络采用FSF-PAFPN结构以实现对多尺度特征的高效融合,提升算法对道路缺陷的定位与分类能力。最后,利用K-Means算法对道路缺陷数据集进行聚类,获取更符合道路缺陷目标形状特点的先验框,降低算法的训练难度,提升检测精度。在数据集RDD2022上的实验结果表明,相较于YOLOv7-tiny, MFF-YOLO轻量化显著,参数量与运算量分别降低了约25.1%与约25.8%。此外, MFF-YOLO的平均精确率均值达到了60.1%,较YOLOv7-tiny提升了2.3个百分点,为多种对比算法中的最高值。同时, MFF-YOLO在检测效果方面表现出色,能够对道路缺陷进行精准定位并分类,且其检测帧率达到81帧/s,表现出较高的实时性。MFF-YOLO实现了检测精度与计算复杂度的有效平衡,为在移动终端设备上实现道路缺陷检测提供参考。

**关键词:**道路缺陷;目标检测;YOLOv7-tiny;轻量化

中图分类号:U446.2

文献标志码:A

文章编号:2096-3246(2026)01-0303-10

道路作为最重要的基础设施之一,与经济发展和人民生活幸福密切相关。然而在长时间的使用下,道路路面结构会因恶劣天气、车辆反复碾压、老化等原因发生恶化,由此产生的道路缺陷将降低道路的使用寿命<sup>[1]</sup>。坑洞、裂缝、龟裂等道路缺陷如果没得到及时修缮,会给交通运输带来安全隐患,并造成不可估量的经济损失。及时检测并修护路面,有助于保证交通安全,也可为后续路面修复和养护提供依据。因此,道路缺陷快速检测具有非常重要的理论意义与应用价值。

早期的道路缺陷检测工作很大程度上依赖于道路检测人员的经验,通过人工检测和车载多传感器检测系统来了解路面的健康状况<sup>[2]</sup>。但这类方法往往耗时、低效且妨碍交通,不适用于大范围的道路路面检测。随着计算机性能不断提升和计算机视觉技术的快速发展,基于深度学习的目标检测技术得到越来越

多的研究和应用,这在很大程度上推动了路面缺陷检测技术的进步<sup>[3-4]</sup>。目前,基于深度学习的目标检测算法可根据是否生成候选目标区域分为两阶段算法与单阶段算法,这两类算法均在道路缺陷检测应用中显示了有效性,但它们表现出不同的特点。典型的两阶段算法,如Faster R-CNN<sup>[5]</sup>、Mask R-CNN<sup>[6]</sup>、Cascade R-CNN<sup>[7]</sup>,在被应用于道路缺陷检测时,具有较高的检测精度,但因其庞大的运算量表现出较慢的检测速度<sup>[8-10]</sup>。相比之下,典型的单阶段算法,如YOLO系列算法<sup>[11-15]</sup>、SSD<sup>[16]</sup>、RetinaNet<sup>[17]</sup>,检测精度略低,但具有较高的实时性<sup>[18-20]</sup>。

在实际道路缺陷检测中,由于受到工作环境的限制,移动终端设备更适合执行检测任务。然而,移动终端设备有限的计算能力,导致难以采用计算复杂度较高的检测算法,为此,许多研究者开始研究轻量

收稿日期:2024-05-06 修回日期:2024-08-14 网络出版日期:2024-10-16

基金项目:甘肃省重点研发计划—工业类项目(23YFGA0049);甘肃省自然科学基金项目(22JR5RA358)

作者简介:侯涛(1975—),男,教授,博士。研究方向:智能控制与智能信息处理。E-mail:ht\_houtao@163.com

\*通信作者:张田明, E-mail: 12221494@stu.lzjtu.edu.cn

级道路缺陷检测算法。Du 等<sup>[21]</sup>提出了一种基于 YOLOv5s 的轻量级道路缺陷检测算法,该算法采用双向特征金字塔网络(BIFPN)网络结构进行特征融合,并利用 Varifocal Loss 优化样本不平衡问题,结果表明该算法在检测速度和精度上均优于 YOLOv5s。Wu 等<sup>[22]</sup>提出了一种适用于移动设备的轻量级道路缺陷检测算法 YOLO-LWNet,它通过使用设计的轻量化卷积(LWC)作为基本单元构建骨干网络与特征融合网络,并优化注意力机制与激活函数,实验结果表明, YOLO-LWNet 在平衡检测精度与算法计算复杂度方面均优于对比检测算法。Diao 等<sup>[23]</sup>在 YOLOv5s 的基础上提出了一种轻量级道路缺陷检测算法 LE-YOLOv5,该算法结合 MobileNetV3 与 GSAM 注意力模块构建轻量骨干特征提取网络,并利用 GSConv(ghost shuffle 卷积)、VCACSP 模块与无参数注意特征融合模块 PAFF 重构特征融合网络,与 YOLOv5s 相比,LE-YOLOv5 不仅有出色的检测性能,还大幅降低了计算复杂度,算法更加轻量化。上述研究将 YOLO 系列算法应用于道路缺陷检测领域,为轻量级道路缺陷检测算法提供了参考,但其在轻量化程度与检测精度的平衡方面仍有进一步优化的空间。

YOLOv7-tiny 是 YOLOv7<sup>[15]</sup> 中的一个轻量级版本,它采用了更紧凑的网络架构和优化的训练策略,更易部署于嵌入式设备和移动终端。因 YOLOv7-tiny 具有出色的性能且有效地平衡了精度和速度,故本文在 YOLOv7-tiny 的基础上提出了一种以多尺度特征融合(multiscale feature fusion, MFF)为特点的轻量级道路缺陷检测算法 MFF-YOLO,以减少算法参数量与计算复杂度,同时提高检测平均精度,实现对二者更好的平衡。本文的工作可概括为以下 4 点:

1)设计了一种高效的骨干多尺度特征提取网络(multiscale feature extraction network, MFENet),增强算法提取多尺度特征的能力,并减少有效特征的丢失。

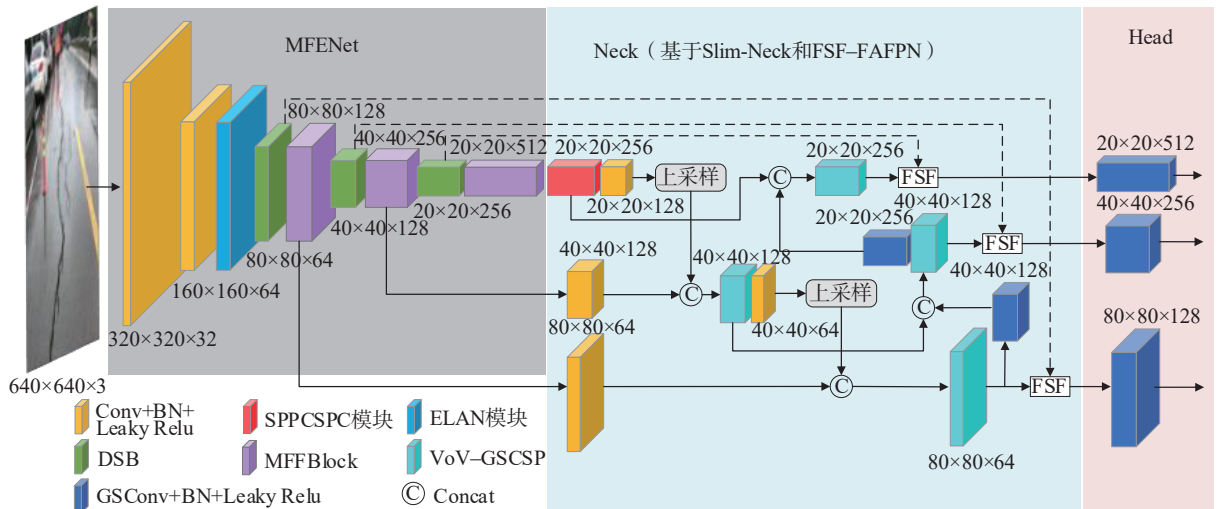
2)颈部特征融合网络采用 Slim-Neck<sup>[24]</sup>设计范式,以实现更高的计算成本效益,在降低网络计算复杂度的同时,确保特征的聚合效果。

3)提出一种特征选择性融合路径聚合特征金字塔网络(feature selective fusion-PAFPN, FSF-PAFPN),其对同一分辨率层级的深浅层特征进行选择融合,实现简单有效的多尺度特征融合。

4)利用 K-Means 算法重聚类后获取先验框,使先验框更符合道路缺陷目标特点,以降低训练难度,提升检测精度。

## 1 MFF-YOLO 算法

MFF-YOLO 算法的网络结构如图 1 所示,包括 MFENet、颈部特征融合网络(图 1 中简称 Neck)和头部预测网络(图 1 中简称 Head)3 部分。其中, MFENet 由多尺度特征融合模块(MFFBlock)与下采样模块(DSB)构建而成,能提升算法对于多尺度特征的提取能力,同时减少有效特征的丢失。颈部特征融合网络采用 Slim-Neck 设计范式,以轻量级模块 GSConv 和 VoV-GSCSP 实现特征的聚合,在保证不丢失重要特征的前提下,轻量化整体算法。此外,颈部特征融合网络采用 FSF-PAFPN 结构进行多尺度特征的融合,增强算法对于特征的表达能力,提升检测精度。MFF-YOLO 沿用 YOLOv7-tiny 头部预测网络的架构,接收来自颈部特征融合网络的三尺度特征金字塔输出,通过分层的先验框机制,实现多尺度目标检测。



注:GSConv(ghost shuffle-convolution)为一种混合卷积,BN(batch normalization)为批归一化层,LeakyRelu为激活函数,Concat为拼接。

图 1 MFF-YOLO 算法的网络结构

Fig. 1 Network structure of MFF-YOLO algorithm

### 1.1 骨干多尺度特征提取网络

#### 1.1.1 MFFBlock 的设计

受 MS-Block<sup>[25]</sup> 的启发,本文设计了一种采用分层特征提取与融合策略的 MFFBlock。图 2 为 MFFBlock 结构。MFFBlock 通过通道扩张与分割(Split)的方式将输入特征  $X(X \in \mathbb{R}^{H \times W \times C}, H$  为高度,  $W$  为宽度,  $C$  为通道数)分为 3 个分支  $X_i(i \in \{1, 2, 3\})$ , 每个分支采用不同方式的卷积进行特征提取。第 1 个分支利用核大小为  $1 \times 1$  的卷积( $\text{Conv}_{1 \times 1}$ )着重提取局部特征信息,而第 2 个与第 3 个分支则采用反向瓶颈结构(IB)<sup>[26]</sup> 以实现深度卷积(DWConv)大卷积核(卷积核大小为  $K_m \times K_m, m \in \{1, 2\}$ )的高效利用,即  $\text{IB}_{K_1}$  和  $\text{IB}_{K_2}$ , 在增大感受野的同时降低计算成本。此外,第 3 个分支的输入特征在进入  $\text{IB}_{K_2}$  之前会与第 2 个分支的输出特征融合,进一步增大了感受野。通过此方式,每个分支的输出  $Y_i$  都被编码了不同尺度的特征信息,其数学表达如下。

$$Y_1 = \text{Conv}_{1 \times 1}(X_1) \quad (1)$$

$$Y_2 = \text{IB}_{K_1}(X_2) \quad (2)$$

$$Y_3 = Y_2 + \text{IB}_{K_2}(X_3) \quad (3)$$

将 3 个分支的输出进行拼接与通道数调整,并利用挤压激励注意力模块(SE)<sup>[27]</sup> 学习通道间的相互依赖性,最终融合经核大小为  $1 \times 1$  的卷积的原始输入特征,得到输出特征。

#### 1.1.2 DSB 的设计

道路缺陷特征往往具有背景复杂与噪声大的特点,使用最大池化(MaxPooling)进行下采样虽能保留主要特征,但容易丢失一些细节特征信息,尤其是在深度网络中,大量有效特征会被过滤。为了最大限度地保留有效道路缺陷特征,同时保持运算量的稳定,本文提出了一种 DSB。图 3 为 DSB 结构。DSB 将输入特征图沿着通道维度分割为两部分,一部分通过步长  $S$  为 2 的卷积进行下采样,另一部分则通过最大池化,将两个来自不同下采样路径的特征图进行融合拼接,再经过正则化(BN)与 Mish 激活函数后得到输出特征图。

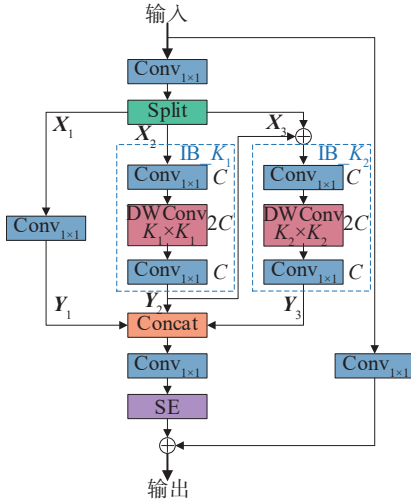


图 2 MFFBlock 结构

Fig. 2 Structure of MFFBlock

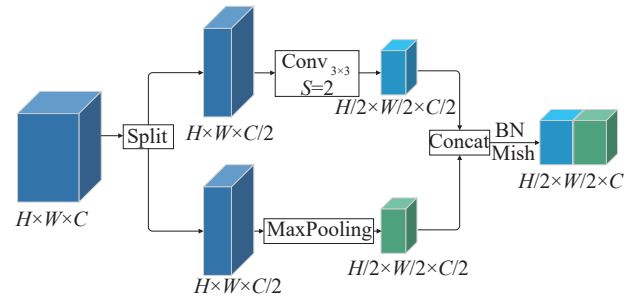


图 3 DSB 结构

Fig. 3 Structure of DSB

#### 1.1.3 MFENet 的构建

基于 MFFBlock 与 DSB 构建 MFENet, 图 4 为 MFENet 结构。图 4 中, P1~P5 为特征层。在特征层 P3



640×640×3  
输入

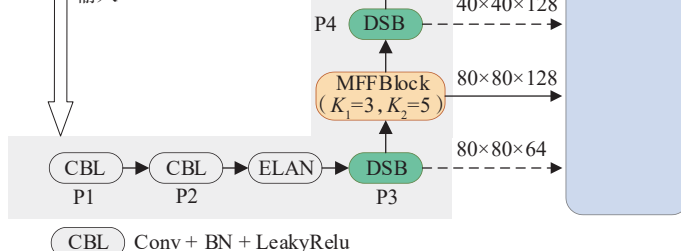


图 4 MFENet 结构

Fig. 4 Structure of MFENet

CBL Conv + BN + LeakyRelu

之前(即 P1 和 P2 层),仍采用原 YOLOv7-tiny 骨干特征提取网络设计原则,在特征层 P3 之后,最大池化与 ELAN 模块分别被 DSB 和 MFFBlock 替代,以实现多尺度特征的提取。

在 MFENet 的不同阶段, MFFBlock 采用不同大小的深度卷积核,并保持逐渐递增,使其与特征分辨率的增加趋势一致。这是因为如果在每个阶段都使用相同的小核卷积,那么在深层阶段进行特征提取时感受野会受到限制,从而影响目标检测的性能。反之,如果在网络的每个阶段都使用相同的大核卷积,虽可以使算法在特征提取时有充足的感受野,但会不可避免地导致运算量骤增,从而降低算法的推理速度。因此,仅在特征提取网络的低分辨率特征层 P3、P4、P5 所对应的 3 个阶段使用 MFFBlock 进行特征提取,并逐渐增大卷积核的大小,即增大  $K_1$  和  $K_2$ 。这样既增大了特征提取时的感受野,增强了算法的多尺度特征表征能力,同时最大限度地规避了大核卷积所带来的运算量骤增问题。

## 1.2 颈部特征融合网络

### 1.2.1 Slim-Neck 范式的引用

深度卷积因低计算复杂度的特点被应用于许多轻量级检测网络,但其弊端也很明显,即输入特征的通道信息在计算过程中被分离,导致丢失许多有效特征信息。GSConv 的提出很好地解决了此问题,它利用洗牌(shuffle)操作将标准卷积生成的信息渗透至深度卷积生成信息的每个部分。图 5 为 GSConv 和 VoV-GSCSP 结构。经验证,GSConv 的计算成本约为标准卷积的一半,并且在推理时间更短的前提下,具有与标准卷积相当的特征学习能力<sup>[24]</sup>(图 5(a))。在 GSConv 的基础上,跨阶段部分网络模块 VoV-GSCSP 被提出(图 5(b))。通过组合 GSConv 与 VoV-GSCSP 模块构建颈部特征融合网络,为轻量级网络的设计提供了 Slim-Neck 范式。本文引用此范式,在颈部特征融合网络部分,用 GSConv 替代标准卷积,用 VoV-GSCSP 替代 ELAN 模块,在低运算量的前提下完成特征聚合。

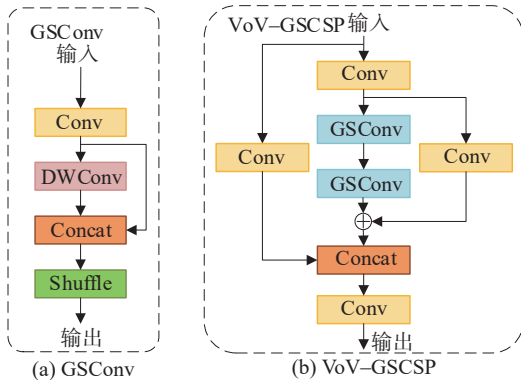


图 5 GSConv 与 VoV-GSCSP 结构

Fig. 5 Structure of GSConv and VoV-GSCSP

### 1.2.2 FSF-PAFPN 结构

YOLOv7-tiny 使用路径聚合特征金字塔网络(PAFP)结构进行颈部特征融合,图 6 为 PAFP 和 FSF-PAFP 结构。在进行特征融合时,PAFP 通过自顶向下和自底向上双向聚合特征,虽然增强了特征融合的效果,但对于浅层网络中的位置信息特征与局部信息特征而言,长路径的上采样与下采样过程会造成特征丢失。尤其是对于一些模糊裂纹与细小裂纹目标,采用 PAFP 结构的特征融合网络会产生漏检的现象(图 6(a))。

为了更加充分地利用浅层网络中的局部特征信息与深层网络中的全局特征信息,并提高检测算法对道路缺陷的定位和分类能力,本文提出 FSF-PAFP 结构(图 6(b))。其中,  $N_j$  为来自自底向上路径中的深层特征,  $N_j \in \mathbb{R}^{H \times W \times C}$ , 下标  $j$  为低分辨率特征层的编号,  $j \in \{3, 4, 5\}$ 。

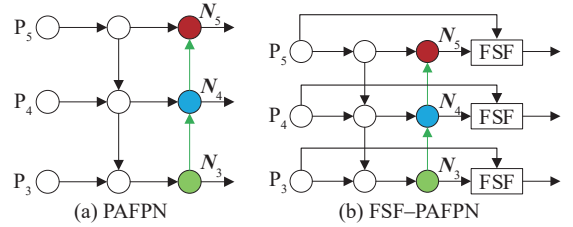


图 6 PAFP 与 FSF-PAFP 结构

Fig. 6 Structure of PAFP and FSF-PAFP

在 PAFP 的基础上, FSF-PAFP 引入特征选择性融合机制(FSF),其利用通道注意力(CA)模块<sup>[28]</sup>将同一分辨率层级的深浅层特征进行跨层级的选择性融合,使得特征融合简单高效,其中, AvgPooling 为平均池化。图 7 为 FSF 结构,融合过程表达式如下。

$$\mathbf{O}_j = \mathbf{Q}_j \cdot \text{CA}(N_j) + N_j \quad (4)$$

式中:  $\mathbf{Q}_j$  为来自骨干特征提取网络的浅层特征,  $\mathbf{Q}_j \in \mathbb{R}^{H \times W \times C}$ ;  $\mathbf{O}_j$  为融合后的输出特征,  $\mathbf{O}_j \in \mathbb{R}^{H \times W \times C}$ 。  $N_j$  被 CA 模块转换为注意力权重以过滤  $\mathbf{Q}_j$ , 使  $\mathbf{Q}_j$  中的关键局部信息被重点关注,加强对小目标特征信息的捕捉。将过滤后的浅层特征  $\mathbf{Q}_j$  与深层特征  $N_j$  融合,得到最终的输出特征  $\mathbf{O}_j$ 。

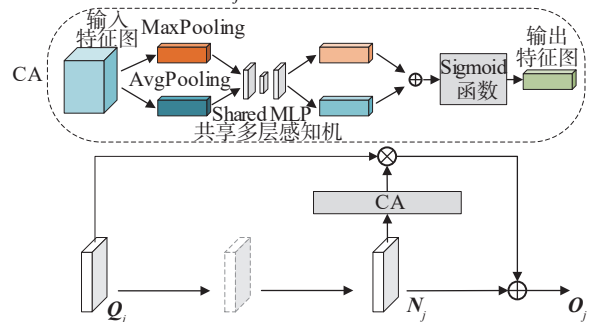


图 7 FSF 结构

Fig. 7 Structure of FSF

## 2 实验准备与验证

### 2.1 数据集与实验环境

选取道路缺陷数据集 RDD2022<sup>[29]</sup>进行算法的训练与验证,数据集 RDD2022 涵盖 6 个国家的道路缺陷数据,共计 47 420 张图像,道路缺陷实例超过 55 000 个。数据集中的图像包含多种视图,如车载摄像头与无人机视图,这有利于增强道路缺陷检测算法的泛化能力。本文重点关注 4 种典型的道路缺陷,分别是纵向裂纹(D00)、横向裂纹(D10)、龟裂(D20)及坑洞(D40),而数据集中的道路修补目标及背景图像并不作为本文的研究对象,因此,将对应的图像及标签进行清理,清理后数据集剩余 23 767 张可用于训练的缺陷图像,并按照 8:1:1 的比例重新划分为训练集、验证集和测试集。图 8 为清理后数据集道路缺陷示例与标签数量。

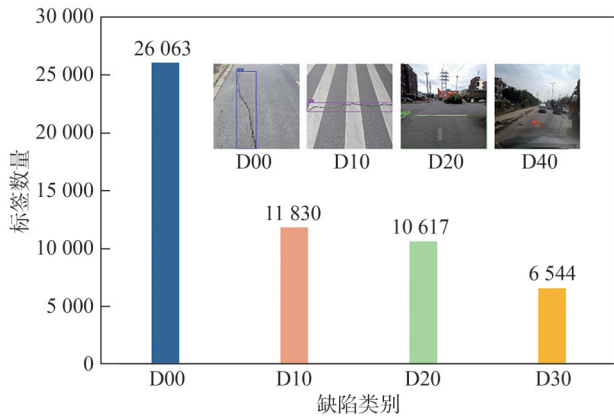


图8 清理后数据集道路缺陷示例与标签数量

Fig. 8 Examples and label quantity of road damage in preprocessed dataset

本文所有算法的训练均基于 Windows11 操作系统,计算机硬件配置为 Intel<sup>®</sup>Core™ i5-12400F@2.50 GHz 处理器,16 G 运行内存和 12G 显存的 NVIDIA GeForce RTX 3060。在实验虚拟环境中利用 Python3.9 语言编写相关程序,利用 PyTorch1.11 搭建算法框架,并使用 CUDA11.2 对网络训练与测试加速。实验的训练轮次设置为 300,训练批次(batch size)设置为 32,数据装载的最大线程数(workers)设置为 8,初始学习率设置为 0.01。

### 2.2 评估指标

算法的评估主要考虑两个部分,即计算复杂度与准确性。计算复杂度由参数量(params)与每秒浮点运算量(FLOPs)来表征,二者的值越小,表示算法的计算成本越小以及对硬件的要求越低,在移动终端设备中部署就越容易。准确性由精确率(precision,记为  $P$ )、召回率(recall,记为  $R$ )、平均精确率(average precision,记为  $C_{AP}$ )、平均精确率均值(mean aver-

age precision,记为  $M_{mAP}$ )来表征。此外,再选择检测帧率(frames per second, FPS,记为  $f_{FPS}$ )以反映算法的实时性能。 $P$ 、 $R$ 、 $C_{AP}$ 、 $M_{mAP}$ 及  $f_{FPS}$ 的计算公式如下。

$$P = \frac{T_p}{T_p + F_p} \quad (5)$$

$$R = \frac{T_p}{T_p + F_n} \quad (6)$$

$$C_{AP} = \int_0^1 P(R) dR \quad (7)$$

$$M_{mAP} = \frac{1}{n} \sum_{s=1}^n C_{AP}(s) \quad (8)$$

$$f_{FPS} = 1/T \quad (9)$$

式(5)~(9)中, $T_p$ 为被正确预测为正样本的数量, $T_n$ 为被错误预测为正样本的数量, $F_p$ 为被错误预测为负样本的数量, $P(R)$ 为  $P$ 关于  $R$ 的函数, $n$ 为检测目标的类别数量, $s$ 为检测目标的类别序号, $T$ 为推理总时间(单位为 ms)。

mAP@0.5 用于反映算法整体准确性,其具体含义为交并比(IoU)阈值取 0.5 时所有检测目标类别的平均精确率。

### 2.3 K-Means 重聚类先验框

先验框是根据待检测目标的宽、高尺寸比例得到的矩形框,设置合理尺寸的先验框有助于降低算法训练难度。YOLOv7-tiny 的先验框是利用 K-Means 聚类数据集 COCO 得到的,虽然其具有较高的泛化性,但并不完全符合道路缺陷目标的分布特点。为提高网络收敛速度与检测精度,本文利用 K-Means 算法对数据集 RDD2022 进行聚类,表 1 为重聚类先验框前后对比。表 1 中,第 2、3 列中括号中的数字为 3 个先验框的尺寸(高度×宽度),如[10,13,16,30,33,23]表示 3 个先验框的尺寸为 10×13、16×30、33×23。

表 1 重聚类前后先验框对比

Tab. 1 Comparison of prior boxes before and after re-clustering

检测层像素	重聚类前	重聚类后
80×80	[10,13,16,30,33,23]	[12,12,26,49,53,24]
40×40	[30,61,62,45,59,119]	[30,61,62,45,59,119]
20×20	[116,90,156,198,373,326]	[133,93,148,204,342,187]

## 2.4 实验结果与分析

### 2.4.1 MFENet 验证

为验证构建 MFENet 时 MFFBlock 深度卷积核大小选择的合理性,设计了 7 组对比实验。以 YOLOv7-tiny 为基线,组别 1~7 将 YOLOv7-tiny 骨干网络均替换为 MFENet,且每一组在 P3、P4、P5 特征层对应的 MFFBlock 中  $K_1$  与  $K_2$  取不同的值。采用清理后的数据集 RDD2022 进行训练,表 2 为 MFFBlock 构建骨干网

络时卷积核选择验证结果。

由表2可见:在组别1~4中,骨干网络的3个特征层均分别采用相同大小深度卷积核的MFFBlock,随着卷积核的增大,mAP@0.5的值也随之增大,这证明了目标检测器的性能与特征提取的感受野关系密切,大的感受野有助于提升检测精度;在组别5~6中,3个特征层的卷积核大小保持递增,且每个特征层的 $K_1$ 与 $K_2$ 取值相同,组别6在每个阶段的卷积核取值均大于

表2 MFFBlock构建骨干网络时卷积核选择验证结果

Tab. 2 Convolutional kernel selection verification results in MFFBlock backbone network construction

组别	P3-MFFBlock	P4-MFFBlock	P5-MFFBlock	mAP@0.5/%	Params/ $10^6$	FLOPs/ $10^9$
基线				57.8	6.02	13.2
1	$K_1=3, K_2=3$	$K_1=3, K_2=3$	$K_1=3, K_2=3$	57.6	5.52	12.0
2	$K_1=5, K_2=5$	$K_1=5, K_2=5$	$K_1=5, K_2=5$	58.1	5.57	12.2
3	$K_1=7, K_2=7$	$K_1=7, K_2=7$	$K_1=7, K_2=7$	58.3	5.62	12.5
4	$K_1=9, K_2=9$	$K_1=9, K_2=9$	$K_1=9, K_2=9$	58.5	5.67	12.7
5	$K_1=3, K_2=3$	$K_1=5, K_2=5$	$K_1=7, K_2=7$	58.3	5.55	12.2
6	$K_1=5, K_2=5$	$K_1=7, K_2=7$	$K_1=9, K_2=9$	58.6	5.63	12.4
7	$K_1=3, K_2=5$	$K_1=5, K_2=7$	$K_1=7, K_2=9$	58.6	5.59	12.2

#### 2.4.2 FSF-PAFPN 结构验证

为验证 FSF-PAFPN 结构的有效性,本文将 YOLOv7-tiny 算法中的特征融合结构 PAFPN 替换为 FSF-PAFPN,并进行了处理效果对比实验。表3为 FSF-PAFPN 与 PAFPN 的处理效果对比。由表3可见, YOLOv7-tiny 在采用 FSF-PAFPN 结构进行特征融合时,参数量与运算量略微增加,但 mAP@0.5 的值提升 0.8 个百分点,表明 FSF-PAFPN 实现了更理想的特征融合效果,检测精度得到显著提升。

表3 FSF-PAFPN 与 PAFPN 的处理效果对比

Tab. 3 Comparison of processing effects between FSF-PAFPN and PAFPN

特征融合结构	mAP@0.5/%	Params/ $10^6$	FLOPs/ $10^9$
PAFPN	57.8	6.02	13.2
FSF-PAFPN	58.6	6.04	13.3

为了更加直观地体现 FSF-PAFPN 对特征融合效果的提升,采用 Grad-CAM<sup>[30]</sup> 技术分别绘制了 PAFPN 与 FSF-PAFPN 在检测时的热力图,以进行可视化分析。图9为 PAFPN 与 FSF-PAFPN 热力图对比。热力图的亮度变化分布情况可以直观地反映算法对不同区域的关注程度。由图9可见,相较于 PAFPN, FSF-PAFPN 在检测道路缺陷时定位更加准确,且具有更高的热力值,再次证实了 FSF-PAFPN 结构对多尺度特征的融合更加高效。

组别5,结果表明组别6取得了更高的检测精度,这证明了 MFFBlock 在骨干网络的不同特征层使用大核卷积,且卷积核大小与特征分辨率的增加趋势保持一致,对提升精度作用显著。组别7即本文策略,与组别6同样取得了最高的检测精度,但参数量与运算量更少,且相较于 YOLOv7-tiny 下降明显,证明本文提出的 MFENet 中大核卷积的选择能更好地平衡检测精度与参数量,其合理性得到了有效证明。

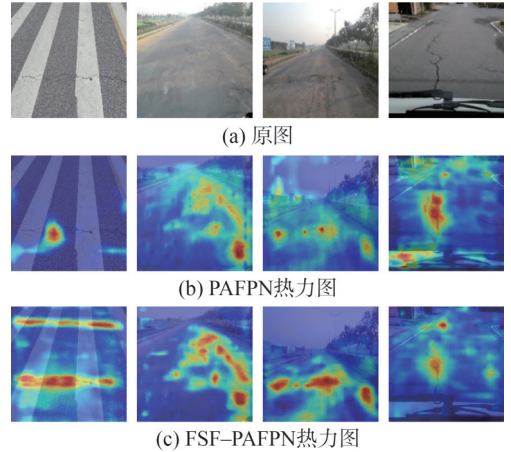


图9 PAFPN 与 FSF-PAFPN 热力图对比

Fig. 9 Comparison of heatmaps between PAFPN and FSF-PAFPN

#### 2.4.3 MFF-YOLO 消融实验

为了评估 MFF-YOLO 相对于 YOLOv7-tiny 算法的优化效果,设计了逐步的消融实验。实验共分为 6 组,均在 RDD2022 训练集上训练,验证集上验证。为确保实验的准确性,每一组实验训练过程均采用相同参数。表4为消融实验结果对比,其中组别1为 YOLOv7-tiny 算法,组别6为本文提出的 MFF-YOLO 算法。

由表4可见:组别2采用 K-Means 重聚类数据集 RDD2022 得到的先验框,检测精度明显提升,说明采用更符合待检测目标形状特点的先验框有助于准确预测目标;组别3进一步将 YOLOv7-tiny 骨干网络中特征层

P3、P4、P5对应的ELAN模块替换为MFFBlock, mAP@0.5的值进一步提升了0.4个百分点,参数量与运算量明显下降,证明MFFBlock在降低计算复杂度的同时,有效增强了网络多尺度特征提取能力;在组别3的基础上,组别4的MFENet中,用DSB替代最大池化,实现下采样,虽然略微增加了参数量和运算量,但mAP@0.5的值提高了0.2个百分点,这表明DSB在下采样过程中有效减少了特征丢失,进而提升了检测精度;组别5进一步采用Slim-Neck范式,利用GSConv与VoV-

GSCSP模块实现颈部特征的聚合,显著降低了参数量与运算量,同时,mAP@0.5的值提升0.5个百分点,这有力地证明了本文采用Slim-Neck设计范式的有效性;组别6在组别5的基础上,采用FSF-PAFPN结构进行颈部特征的融合,进一步将mAP@0.5的值提升了0.7个百分点, mAP@0.5:0.95(表示所有mAP@0.5值的均值)的值提高了0.4个百分点,证明FSF-PAFPN结构实现了更为高效的多尺度特征融合,提升了算法对道路缺陷特征的表征能力。

表 4 消融实验结果对比

Tab. 4 Comparison of ablation experiment results

组别	K-Means	MFFBlock	DSB	Slim-Neck	FSF-PAFPN	mAP@0.5/%	mAP@0.5:0.95/%	Params/10 <sup>6</sup>	FLOPs/10 <sup>9</sup>	f <sub>FPS</sub> /(帧·s <sup>-1</sup> )
1						57.8	27.1	6.02	13.2	101
2	√					58.3	27.4	6.02	13.2	99
3	√	√				58.7	27.6	5.39	11.9	93
4	√	√	√			58.9	27.6	5.59	12.3	89
5	√	√	√	√		59.4	27.9	4.47	9.7	85
6	√	√	√	√	√	60.1	28.3	4.51	9.8	81

注:“√”表示采用了相应的改进方法。

MFF-YOLO的检测精度显著提升,参数量与运算量大幅下降,但其FPS下降了20帧/s。这主要是因为MFF-YOLO在计算图中的依赖性变多,如MFFBlock的3分支特征提取、DSB的双路径下采样、FSF-PAFPN结构的跨层级特征融合方式等。这些操作有效提高了算法的表征能力,但不可避免地增加了数据在网络中流动的复杂性。某些操作必须等待其他操作完成,进而降低了计算并行度,导致算法推理速度变慢、FPS降低。尽管如此,MFF-YOLO的FPS值被合理控制在了81帧/s,这

在道路缺陷检测中仍能维持较好的实时性能。

2.4.4 算法性能对比实验

为验证本文算法的检测性能,将MFF-YOLO与当前主流的轻量级目标检测算法进行对比实验,对比算法包括YOLOv3-tiny、YOLOv4-tiny、YOLOv5s、YOLOv6n、YOLOv7-tiny与YOLOv8n,此外,再选择轻量级道路缺陷检测算法YOLO-LWNet-s与LE-YOLOv5进行性能对比,充分验证MFF-YOLO的检测性能。表5为数据集RDD2022下的算法性能对比。

表 5 数据集RDD2022下的算法性能对比

Tab. 5 Comparison of algorithm performance using RDD2022 Dataset

算法	输入尺寸	C <sub>AP</sub> /%				P/%	R/%	mAP@0.5/%	Params/10 <sup>6</sup>	FLOPs/10 <sup>9</sup>	f <sub>FPS</sub> /(帧·s <sup>-1</sup> )
		D00	D10	D20	D40						
YOLOv3-tiny <sup>[12]</sup>	416×416	47.9	42.4	51.8	39.2	52.4	47.7	45.3	8.76	12.9	119
YOLOv4-tiny <sup>[13]</sup>	416×416	51.2	44.8	53.3	38.4	51.8	45.6	46.9	6.06	8.2	110
YOLOv5s	640×640	60.1	55.7	67.1	52.4	63.4	55.9	58.8	7.03	15.8	99
YOLOv6n <sup>[14]</sup>	640×640	57.4	55.2	66.1	46.9	64.4	52.0	56.4	4.63	11.3	68
YOLOv7-tiny <sup>[15]</sup>	640×640	60.2	54.9	67.2	49.3	63.0	54.1	57.8	6.02	13.2	101
YOLOv8n	640×640	57.1	56.1	68.1	49.0	62.5	53.4	57.6	3.01	8.1	104
YOLO-LWNet-s <sup>[22]</sup>	640×640	59.8	58.3	68.3	50.4	62.5	57.0	59.2	3.68	11.3	66
LE-YOLOv5 <sup>[23]</sup>	640×640	60.5	56.6	67.5	52.6	63.9	57.1	59.3	3.44	7.1	63
MFF-YOLO	640×640	60.8	58.9	68.8	51.9	64.7	55.9	60.1	4.51	9.8	81

由表5可见:在准确性方面,除R与D40类别的AP外,MFF-YOLO在其余评估指标上均表现最佳,其中mAP@0.5为60.1%,比YOLOv7-tiny算法提高了2.3个百分点;在计算复杂度方面,MFF-YOLO表现同样优异,参数量和运算量分别为4.51×10<sup>6</sup>和9.8×10<sup>9</sup>,仅高于YOLOv8n、YOLO-LWNet-s(仅参数量)、LE-

YOLOv5,相较于YOLOv7-tiny,二者分别降低了约25.1%和25.8%;在检测速度方面,MFF-YOLO的FPS达到了81帧/s,仍维持着较好的实时检测性能。对比同样应用于道路缺陷领域的YOLO-LWNet-s和LE-YOLOv5,MFF-YOLO的mAP@0.5值分别高出前二者0.9个百分点和0.8个百分点。虽然MFF-YOLO的

参数量为三者中的最大值,但其检测速度显著快于 YOLO-LWNet-s 与 LE-YOLOv5,展现出高检测精度、高检测速度的优势。综合考虑准确性、计算复杂度与检测速度,MFF-YOLO 实现了更优秀的平衡效果,使其在移动终端设备上道路缺陷检测工作时适用性更强。

#### 2.4.5 算法检测效果对比

为了准确反映 MFF-YOLO 算法在实际检测中的效果,本文在 RDD2022 测试集上随机选取部分图像,并分别使用 YOLOv7-tiny、YOLO-LWNet-s、LE-YOLOv5 及 MFF-YOLO 算法进行检测。图 10 为基于数据集 RDD2022 的算法检测效果对比。

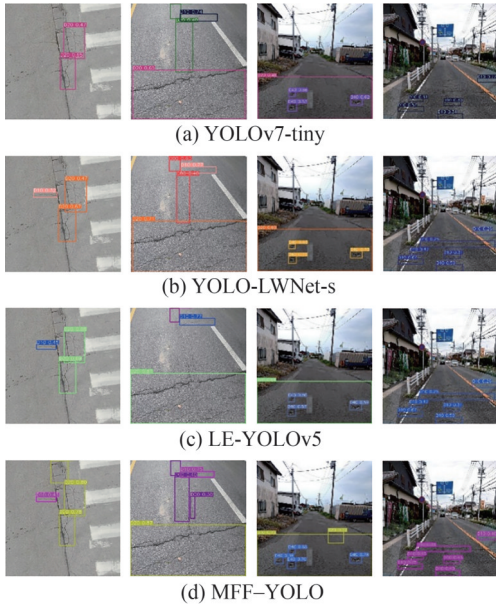


图 10 基于数据集 RDD2022 的算法检测效果对比

Fig. 10 Comparison of algorithm detection effects based on RDD2022 Dataset

由图 10 可见,MFF-YOLO 的检测效果优于其他算法,其能够准确地定位道路缺陷目标,并具有较高的置信度,即使在特征不明显或背景复杂的情况下也能表现出良好的性能。特别是在检测模糊裂纹和小坑洞等目标时,YOLOv7-tiny、YOLO-LWNet-s 和 LE-YOLOv5 算法均存在漏检,而 MFF-YOLO 算法能够全面检测到这些缺陷,这充分证明了 MFF-YOLO 算法在实际检测任务中的优势。

### 3 结论

本文针对现有道路缺陷检测算法未能有效平衡检测精度与计算复杂度的问题,开展了基于多尺度特征融合的轻量级道路缺陷算法研究。通过改进 YOLOv7-tiny 算法,构建了 MFF-YOLO 算法。研究结果表明,MFF-YOLO 算法在道路缺陷检测任务中表现出良好的综

合性能。相较于 YOLOv7-tiny 算法,MFF-YOLO 算法在保持较高检测精度的同时,显著降低了模型复杂度:mAP@0.5 的值提升 2.3 个百分点,达到 60.1%;参数量与计算量分别降低至  $4.51 \times 10^6$  与  $9.8 \times 10^9$ ,降低了约 25.1% 和 25.8%。这种精度稳升、计算锐减的优势,使 MFF-YOLO 更适用于移动终端设备,能够实现高效、准确的道路缺陷实时检测。未来研究将进一步扩展检测对象范围,覆盖更多路面缺陷类型(如松散、沉降等),并致力于构建精细化量化评估体系,实现裂缝长、宽,坑洞面积与深度等关键指标的精准提取与分析。

#### 参考文献:

- [1] Hou Yue, Li Qiuhan, Zhang Chen, et al. The state-of-the-art review on applications of intrusive sensing, image processing techniques, and machine learning methods in pavement monitoring and analysis[J]. *Engineering*, 2021, 7(6): 845–856.
- [2] Ma Jian, Zhao Xiangmo, He Shuanhai, et al. Summary of pavement detection technology[J]. *Journal of Traffic and Transportation Engineering*, 2017, 17(5): 121–137. [马建, 赵祥模, 贺拴海, 等. 路面检测技术综述[J]. *交通运输工程学报*, 2017, 17(5): 121–137.]
- [3] Du Zhenyu, Yuan Jie, Xiao Feipeng, et al. Application of image technology on pavement distress detection: A review[J]. *Measurement*, 2021, 184: 109900.
- [4] Yang Guidong, Liu Kangcheng, Zhang Jihan, et al. Datasets and processing methods for boosting visual inspection of civil infrastructure: A comprehensive review and algorithm comparison for crack classification, segmentation, and detection[J]. *Construction and Building Materials*, 2022, 356: 129226.
- [5] Ren Shaoqing, He Kaiming, 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] He Kaiming, Gkioxari G, Dollár P, et al. Mask R-CNN[C]// *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, 2017: 2980–2988.
- [7] Cai Zhaowei, Vasconcelos N. Cascade R-CNN: Delving into high quality object detection[C]// *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 6154–6162.
- [8] Cao M T, Tran Q V, Nguyen N M, et al. Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources[J]. *Advanced Engineering Informatics*, 2020, 46: 101182.
- [9] Liu Zhen, Yeoh J K W, Gu Xingyu, et al. Automatic pixel-

- level detection of vertical cracks in asphalt pavement based on GPR investigation and improved mask R-CNN[J]. *Automation in Construction*,2023,146:104689.
- [10] Pei Zixiang, Lin Rongheng, Zhang Xiubao, et al. CFM: A consistency filtering mechanism for road damage detection[C]// *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*. Atlanta: IEEE, 2020: 5584–5591.
- [11] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger[C]// *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu: IEEE, 2017: 6517–6525.
- [12] Redmon J, Farhadi A. YOLOv3: An incremental improvement[EB/OL]. (2018–04–08)[2024–05–06]. <https://doi.org/10.48550/arXiv.1804.02767>.
- [13] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection[EB/OL]. (2020–04–23)[2024–05–06]. <https://doi.org/10.48550/arXiv.2004.10934>.
- [14] Li Chuyi, Li Lulu, Jiang Hongliang, et al. YOLOv6: A single-stage object detection framework for industrial applications[EB/OL]. (2022–09–07)[2024–05–06]. <https://doi.org/10.48550/arXiv.2209.02976>.
- [15] Wang C Y, Bochkovskiy A, Liao H M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]// *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver: IEEE, 2023: 7464–7475.
- [16] Liu Wei, Anguelov D, Erhan D, et al. SSD: Single shot Multi-Box detector[M]// *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016: 21–37.
- [17] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]// *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, 2017: 2999–3007.
- [18] Yan Kun, Zhang Zhihua. Automated asphalt highway pavement crack detection based on deformable single shot multi-box detector under a complex environment[J]. *IEEE Access*, 2021, 9: 150925–150938.
- [19] Al Duhayyim M, Malibari A A, Alharbi A, et al. Road damage detection using the hunger games search with Elman neural network on high-resolution remote sensing images [J]. *Remote Sensing*, 2022, 14(24): 6222.
- [20] Ren Miao, Zhang Xianfeng, Chen Xiao, et al. YOLOv5s-M: A deep learning network model for road pavement damage detection from urban street-view imagery[J]. *International Journal of Applied Earth Observation and Geoinformation*, 2023, 120: 103335.
- [21] Du Fujun, Jiao Shuangjian. Improvement of lightweight convolutional neural network model based on YOLO algorithm and its research in pavement defect detection[J]. *Sensors*, 2022, 22(9): 3537.
- [22] Wu Chenguang, Ye Min, Zhang Jiale, et al. YOLO-LWNet: A lightweight road damage object detection network for mobile terminal devices[J]. *Sensors*, 2023, 23(6): 3268.
- [23] Diao Zhuo, Huang Xianfu, Liu Han, et al. LE-YOLOv5: A lightweight and efficient road damage detection algorithm based on improved YOLOv5[J]. *International Journal of Intelligent Systems*, 2023, 2023: 8879622.
- [24] Li Hulin, Li Jun, Wei Hanbing, et al. Slim-neck by GSConv: A lightweight-design for real-time detector architectures[J]. *Journal of Real-Time Image Processing*, 2024, 21(3): 62.
- [25] Chen Yuming, Yuan Xinbin, Wang Jiabao, et al. YOLO-MS: Rethinking multi-scale representation learning for real-time object detection[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025, 47(6): 4240–4252.
- [26] Sandler M, Howard A, Zhu Menglong, et al. MobileNetV2: Inverted residuals and linear bottlenecks[C]// *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 4510–4520.
- [27] Hu Jie, Shen Li, Sun Gang. Squeeze-and-excitation networks[C]// *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 7132–7141.
- [28] Woo S, Park J, Lee J Y, et al. CBAM: Convolutional block attention module[M]// *Computer Vision – ECCV 2018*. Cham: Springer International Publishing, 2018: 3–19.
- [29] 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.
- [30] Selvaraju R R, Cogswell M, Das A, et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization[J]. *International Journal of Computer Vision*, 2020, 128(2): 336–359.

## MFF-YOLO: A Lightweight Road Damage Detection Algorithm Based on Multiscale Feature Fusion

HOU Tao, ZHANG Tianming\*, NIU Hongxia

(School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

### Abstract:

**Objective:** Road damage detection serves as the premise and foundation of road maintenance and is essential for road safety and timely repair. In

real-world scenarios, mobile terminal devices are more practical for detection tasks due to constraints in working environments. However, the limited computational power of these devices makes it challenging to apply detection algorithms with high computational complexity. This study proposes MFF-YOLO, a lightweight road damage detection algorithm that features multiscale feature fusion based on YOLOv7-tiny to address the issue of existing road damage detection algorithms struggling to balance detection accuracy and computational efficiency, which hinders their deployment on mobile devices.

**Methods** Firstly, road damage often occurred in complex backgrounds with high levels of noise, so it was crucial to extract road damage features accurately. The multiscale feature extraction block (MFFBlock) and the downsampling block (DSB) were designed to improve the feature extraction capability of the algorithm. MFFBlock employed a multiscale feature extraction and fusion strategy that integrated a multi-branch structure with various types of convolution to produce outputs with different receptive fields, extracting both global and local feature information. DSB combined max pooling with convolution at a stride of 2 to maximize the retention of effective feature information and to ensure computational stability during downsampling. Based on MFFBlock and DSB, an efficient multiscale feature extraction backbone network (MFEnet) was constructed, utilizing MFFBlock of different sizes at various stages to enhance the algorithm's multiscale feature representation ability. Secondly, after extracting road damage features, it was equally crucial to efficiently fuse these features in the neck network. The Slim-Neck design paradigm was used in the neck feature fusion network, utilizing GSConv and VoV-GSCSPC modules to aggregate features while making the network more lightweight without losing important information. In addition, a novel feature selective fusion structure (FSF-PAFPN) was proposed. Building on PAFPN, FSF-PAFPN introduced a feature selective fusion mechanism (FSF), which used a channel attention (CA) module to selectively fuse shallow and deep features across layers at the same resolution level, achieving simple yet effective multiscale feature fusion. Finally, the K-Means algorithm was utilized to cluster the RDD2022 dataset, obtaining anchors that were more consistent with the shape characteristics of road damage objects, reducing the training difficulty, and improving detection accuracy. The RDD2022 road damage dataset was used for algorithm training and verification and consisted of a total of 23 767 damage images. These images, which included four typical road damage types: longitudinal cracks (D00), transverse cracks (D10), alligator cracks (D20), and potholes (D40), were divided into training, validation, and test sets in a ratio of 8:1:1.

**Results and Discussions** The results from the ablation experiment showed that using K-Means to re-cluster the RDD2022 dataset for generating anchors improved mAP@0.5 by 0.5%. In addition, implementing MFEnet as the backbone feature extraction network increased mAP@0.5 by 0.6%, while its parameters and FLOPs were  $5.59 \times 10^6$  and  $12.3 \times 10^9$ , reduced by 7.1% and 6.8%, respectively. Adopting the Slim-Neck design paradigm enhanced mAP@0.5 by 0.5%, while reducing parameters and FLOPs by 20.0% and 21.1%, to  $4.47 \times 10^6$  and  $9.7 \times 10^9$ , respectively. On this basis, adopting the FSF-PAFPN feature fusion structure further improved mAP@0.5 by 0.7%, with parameters and FLOPs increased to  $4.51 \times 10^6$  and  $9.8 \times 10^9$ , respectively. The results of the algorithm performance comparison based on RDD2022 showed that, in terms of accuracy, MFF-YOLO outperformed all other evaluation metrics except for recall (R) and average precision (AP) in the D40 category. The AP for the D00, D10, and D20 categories reached 60.8%, 58.9%, and 68.6%, respectively, with a precision (P) of 64.7%. It achieved the highest mAP@0.5 of 60.1%, an improvement of 2.3 percentage points compared to YOLOv7-tiny. In terms of computational complexity, MFF-YOLO also excelled, with parameters and FLOPs of  $4.51 \times 10^6$  and  $9.8 \times 10^9$ , respectively, values that were only slightly higher than YOLOv8n, YOLO-LWNet-s, and LE-YOLOv5. Compared to YOLOv7-tiny, these metrics were reduced by 25.1% and 25.8%, respectively. In addition, MFF-YOLO reached a detection speed of 81 frames per second, maintaining an impressive real-time detection capability. Comparing the actual detection effects of YOLOv7-tiny, YOLO-LWNet-s, LE-YOLOv5, and MFF-YOLO algorithms indicates that the detection performance of MFF-YOLO exceeded that of the other algorithms. MFF-YOLO accurately located road damage objects with high confidence and demonstrated strong performance even when the features were not obvious or when the background was complex.

**Conclusions** The results demonstrated that the MFEnet proposed in this study can effectively enhance the network's multiscale feature extraction capability while reducing computational complexity. The Slim-Neck design paradigm ensured the aggregation of features while maintaining a lightweight network structure. The FSF-PAFPN structure achieved more efficient multiscale feature fusion and improved the algorithm's ability to characterize road damage features. Accordingly, MFF-YOLO significantly improved detection accuracy while reducing computational complexity. Considering accuracy, computational complexity, and detection speed, MFF-YOLO achieved a balanced performance and is more suitable for road damage detection on mobile devices, providing a valuable reference for mobile terminal road damage detection.

**Key words:** road damage; object detection; YOLOv7-tiny; lightweight

(编辑 吴芝明)

引用格式: Hou Tao,Zhang Tianming,Niu Hongxia.MFF-YOLO: A lightweight road damage detection algorithm based on multiscale feature fusion[J].Advanced Engineering Sciences,2026,58(1):303-312.[侯涛,张田明,牛宏侠.MFF-YOLO:多尺度特征融合的轻量级道路缺陷检测算法[J].工程科学与技术,2026,58(1):303-312.]