

YOLOv5-W桥梁裂纹实时检测算法

史鸣凤¹, 董琴¹, 郭乃瑄^{1,2}, 许铭³

¹盐城工学院信息工程学院, 江苏 盐城

²东南大学计算机网络和信息集成教育部重点实验室, 江苏 南京

³武汉邮电科学研究院, 湖北 武汉

收稿日期: 2023年10月29日; 录用日期: 2023年11月15日; 发布日期: 2024年1月15日

摘要

桥梁的定期检修与维护, 是保证交通安全、保障桥梁使用年限的重要措施。在桥梁的诸多损伤中, 桥梁裂纹是最为普遍的损伤。针对现有裂纹研究中模型计算量大、实时性较差、需要大数据集训练等问题, 提出了一种基于YOLOv5改进的YOLOv5-W模型。对YOLOv5的损失函数和颈部网络的设计进行了优化, 使用Wise IoU, 通过离群度来衡量检测框质量, 做出最合适的梯度增益分配, 提高检测精度。使用轻量化颈部Slim-Neck设计缩小模型的参数量, 提高检测速度。在小数据集Crack400上验证表明, 改进模型准确率为98.7, 均值平均精度(mean average precision, MAP)为98.5, 检测速度为47.958 FPS, 模型参数量减少到14.5 GFLOPs。相较于原始的YOLOv5, 平均精度提升3%, FPS提升12, 模型大小减小1.3 GFLOPs。

关键词

目标检测, 桥梁裂纹, 深度学习, YOLOv5

YOLOv5-W Bridge Crack Real-Time Detection Algorithm

Mingfeng Shi¹, Qin Dong¹, Naixuan Guo^{1,2}, Ming Xu³

¹School of Information Engineering, Yancheng Institute of Technology, Yancheng Jiangsu

²Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing Jiangsu

³FiberHome Technologies Group, Wuhan Hubei

Received: Oct. 29th, 2023; accepted: Nov. 15th, 2023; published: Jan. 15th, 2024

Abstract

Regular maintenance and repair of bridges is an important measure to ensure traffic safety and to

safeguard the service life of bridges. Among the many damages to bridges, bridge cracks are the most common damage. A YOLOv5-W model based on an improved YOLOv5 is proposed to address the problems of large model computation, poor real-time performance and the need for large data sets for training in existing cracking studies. The loss function of YOLOv5 and the design of the neck network are optimized, and Wise IoU is used to measure the quality of the detection frame by the outlier degree to make the most appropriate gradient gain assignment and improve the detection accuracy. A lightweight neck Slim-Neck design was used to reduce the number of parameters in the model and improve detection speed. Validation on the small dataset Crack400 shows that the improved model has an accuracy of 0.987, a mean average precision (mAP) of 0.985, a detection speed of 47.958 FPS and a reduction in model parameters to 14.5 GFLOPs. The model size was reduced by 1.3 GFLOPs.

Keywords

Object Detection, Bridge Crack, Deep Learning, YOLOv5

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

裂纹作为桥梁常见的危害，对桥梁的安全运营而言是重大的安全隐患[1]。为确保桥梁始终处于正常运行状态，需要定期对桥梁进行检查和养护工作，但是我国的桥梁技术并没有完全因为技术高速发展而出现飞跃式进步，科学有效的检测评估还有待完善[2]。原先的桥梁裂纹检测，大部分是依靠人工费时费力。科研工作者们开始考虑用机器代替人工，逐渐将机器视觉应用于桥梁检测领域，以此来降低对人工的依赖性，同时也减少了人工的主观或经验知识导致的错误评估[3] [4]。图像处理技术和传统的机器学习技术慢慢渗透裂纹检测的研究中来[5]。近几年来，神经网络层数不断增加，深度神经网络被应用到各种领域，其中也包含桥梁的裂纹检测[6] [7]。

从深度学习的角度来说，学者们对桥梁裂纹检测的研究主要有两个方向，一是多阶段检测，二是单阶段检测。常见的双阶段检测有，Faster R-CNN [8]、Mask R-CNN [9]等，单阶段检测模型有，YOLO [10] [11]系列、SSD [12]等。单阶段的检测是一种密集预测，而多阶段的检测是一种稀疏预测。相比于双阶段的检测算法，单阶段只需要进行一次回归，虽然精度会下降，但是其速度会大大提升，更加适合于实时的目标检测。Kemal 等人[13]使用基于深度学习的 Faster R-CNN 方法进行混凝土路面裂缝检测，将双阶段的目标检测算法应用于裂纹的检测中。张等人[14]基于改进的 YOLOv3 算法，结合了 MobileNets 和卷积块注意模块(CBAM)。使得模型大小减小了 223.9 M。于等人[15]基于 YOLOv4 算法，使用 Focal loss 损失函数并通过剪枝提高检测精度，满足实时性的要求。改进后的算法平均精度(mAP)提高了 0.064。张等人[16]尝试将目标检测和图像分割结合起来，对 PSPNet 算法进行了改进，将桥梁裂缝与非裂缝区域分割开来，避免了检测算法的视觉干扰，最终得到的分割结果 mAP 达到 0.87，精度高于主流的分割模型。马等人[17]基于 YOLO 框架和图像分割设计出两阶段混凝土结构裂缝检测方法，使得 mAP 达到 0.9502，该文还强调 YOLOv5 模型是快速准确且易于实现的。

上述学者都使用目标检测方法在裂纹研究中取得了良好的结果，但是研究的数据集都是数量庞大的大数据集，且这些数据集的尺寸较小，如 256×256 或 500×350 。此外不少的研究只考虑提升检测的精

度,忽略了模型的大小与实时检测问题。为合理解决上述两个问题,本文提供了两种类型的数据集,分别为只有 400 张且的背景较为复杂 Crack400 数据集,和经过筛选和扩增之后有 2000 张的 Crack2000 数据集,这两个数据集分辨率均为 640×640 。为桥梁裂纹检测、桥梁养护工作的开展,本文设计出了一个对小数据集也拥有较好检测效果,并且模型大小更为轻量化的网络模型 YOLOv5-W。以平衡速度、精度、模型大小三个方面的需求,既满足了精度与速度的提升,又减小了计算量。

2. YOLO 介绍

自 2015 年单阶段的目标检测算法 YOLO v1 发布以来, YOLO 系列算法不断迭代更新,但是整体的设计思路仍然使用的是单独的卷积神经网络模型实现端到端的检测。输入图像经过卷积神经网络即可得到结果,相比于 R-CNN 算法,其速度更快,更适合做实时监测任务。算法的思路是将图片分割为 7×7 的网格,每个网格负责检测中心点落在格子内的目标。每个单元格会预测两个边界框,每一个边界框包含五个参数(x, y, w, h, c)即宽度高度横纵坐标和置信度。由于 V1 版本划分的单元格有限,可以识别的类别也有限,并且每一个单元格只预测一个类别,如果出现重叠就无法解决并且对于小物体检测的效果较为一般。

应运而生的 YOLOv2 就增加了更多的技巧来解决上述问题,引入了批量归一化(Batch Normalization, BN),舍弃原有的 dropout 操作,卷积后加入批量归一化,每一层的输入都做了归一化,收敛也更加容易,如今 BN 操作已成为网络的必备处理。使用分辨率更大的图像,图片可以划分成 13×13 的网格。摒弃全连接层,参照 VGG 网络使用更小的卷积核减小计算开销。引入 Anchor Box 增加更多的 box 数。使用 K-means 聚类[18]的方法,得到 5 种不同但更适用于目标检测的预选框尺寸。直接的位置预测,限制了位置预测,参数化更容易学习,使网络更稳定。

2018 年 YOLO 的作者提出了 YOLOv3,它是前作的改进,最大的改进特点包括使用了残差模型 Darknet-53,以及为了实现多尺度检测采用了特征金字塔网络(Feature Pyramid Networks, FPN) [19]。YOLOv4 最显著的改变是引入了跨阶段局部模块(Cross Stage Partial, CSP) [20],使用 CSPDarknet53 作为骨干网络增强了网络的学习能力,移除计算瓶颈,降低内存的使用。颈部网络部分使用了空间金字塔池化模块(Space Pyramid Pooling, SPP) [21],使用不同尺度的最大池化方式链接不同尺寸的特征图,显著分离上下文特征,一定程度上解决了多尺度的问题。在 V3 的基础上增加了路径聚合网络(Path Aggregation Network, PAN) [22],既有高层语义信息向底层融合,又有低层的语义信息向高层融合,起到了特征聚合的作用。YOLOv5 实际上也是在 YOLOv4 的基础上进行了一定程度的优化。本文所使用的基础网络是 YOLOv5-6.0 版本,骨干网络部分使用尺寸大小为 6×6 ,步长为 2 的卷积核代替了 YOLOv5 的 Focus 模块;使用 C3 模块代替 CSPDarknet53,使用 SPPF 模块来代替 SPP 模块,通过多个小池化核级联代替 SPP 模块中单个大尺寸池化核,保证原始性能的同时融合不同感受野的特征图,不仅丰富了特征图的表达能力还进一步提高了运行速度。颈部网络使用 PANet,在 FPN 的基础上又引入了一个自底向上的路径,主要用于特征融合和尺度变换,可以提高检测效果。本文所使用的 YOLOv5-6.0 版本的网络结构更加精简和高效,可以在保持高精度的同时,提高检测速度和计算性能。

3. YOLOv5 模型优化改进

3.1. 采用动态损失函数 Wise IoU

在目标检测领域中,边界框回归的损失函数对检测结果起至关重要的结果,合适的损失函数将给模型的性能带来显著的提升。YOLO 系列算法中,使用交并比来衡量真实框和预测框的重叠程度。如图 1 所示,绿色部分为真实框,橙色部分为预测框,重叠的部分越多,说明预测的越准确。

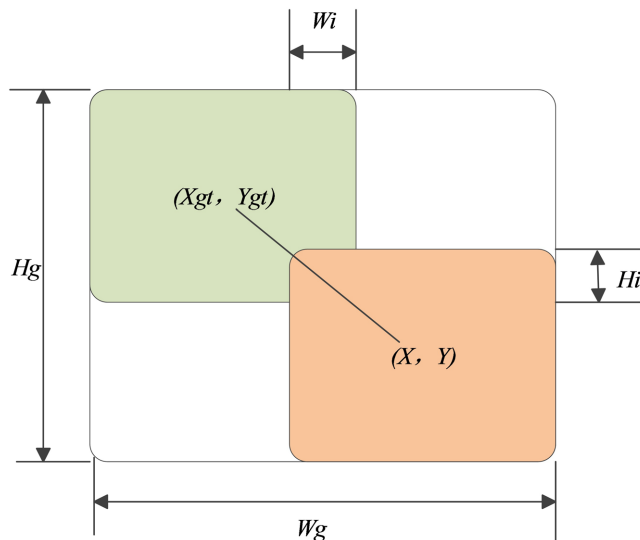


Figure 1. IoU schematic diagram

图 1. IoU 原理图

其中预测框表示为:

$$\vec{B} = [xywh] \quad (1)$$

真实框表示为:

$$\vec{B}_{gt} = [x_{gt} y_{gt} w_{gt} h_{gt}] \quad (2)$$

损失函数可以表示为:

$$S_u = wh + w_{gt} h_{gt} - W_i H_i \quad (3)$$

使用 IoU 作为损失函数能够更好反应重合程度, 且具有尺度不变性, 无论矩形框的大小如何, 重叠程度都不会受到尺度的影响。但是对损失函数求梯度, 不难看出, 当边界框之间没有重叠时, 反向传播的梯度会消失, 导致梯度无法更新。于是在原有的损失函数中添加了惩罚项来解决梯度消失的问题, 此时的惩罚项是与边界框相关的几何因素。紧接着出现的 D IoU, 将惩罚项定义为中心点连线的归一化长度。不仅考虑到 IoU 的值还考虑到了框中心点的距离, 直接最小化两个 boxes 之间的欧式距离, 因此收敛速度更快。而后又出现了 C IoU, 在 D IoU 的基础上增加了纵横比。优秀的回归定位损失函数考虑到了三种几何参数, 包括重叠面积、中心距离、长宽比。

但是在数据的训练过程中, 难免会存在质量比较低的数据, 上述所使用的几何度量会不断加剧对低质量数据的学习, 从而降低模型的泛化性能。现有的数据集中不乏高中质量的数据, 同时也存在低质量的数据, 如果一味强调边界框对低质量数据的回归, 必然会影响模型的检测性能。Focal E IoU v1 被提出以解决这个问题, 但由于其聚焦机制是静态的, 并未充分挖掘非单调聚焦机制的潜能[23]。而后有学者基于这个观点提出 Wise IoU [24], 通过动态非单调聚焦机制使用“离群度”替代 IoU 对锚框进行质量评估, 并提供了更为明智的梯度增益分配策略。该策略在降低高质量锚框的竞争力的同时, 也减小了低质量数据产生的有害梯度。这使得 Wise IoU 可以聚焦于普通质量的锚框, 并提高检测器的整体性能。本文在对 YOLOv5 的优化中, 正是替换掉了原始的 C IoU, 使用 Wise IoU 以专注于普通质量的数据, 提高整体的检测性能。

Wise IoU v1 使用两层注意力机制, 用 \mathcal{R}_{WIoU} 来放大质量好的检测框的 l_{iou} 。

$$\mathcal{L}_{WIoUv1} = \mathcal{R}_{WIoU} \mathcal{L}_{IoU} \quad (4)$$

其中 $\mathcal{R}_{WIoU} \in [1, e]$, $\mathcal{L}_{IoU} \in [0, 1]$, 并且

$$\mathcal{R}_{WIoU} = \exp \left(\frac{(x + x_{gt})^2 + (y + y_{gt})^2}{(W_g^2 + H_g^2)^2} \right) \quad (5)$$

Wise IoU v2 是仿照 focalloss 针对于交叉熵的单调聚集机制, 降低简单数据对损失的影响, 聚焦于困难的数据上这一目标, 而构造的单调聚集函数 \mathcal{L}_{IoU}^* :

$$\mathcal{L}_{WIoUv2} = \mathcal{L}_{IoU}^{\gamma^*} \mathcal{L}_{WIoUv1}, \gamma > 0 \quad (6)$$

为了防止训练后期收敛速度减小, 引入了均值 \mathcal{L}_{IoU} 作为归一化因子:

$$\mathcal{L}_{WIoUv2} = \left(\frac{\mathcal{L}_{IoU}^*}{\mathcal{L}_{IoU}} \right)^{\gamma} \mathcal{L}_{WIoUv1} \quad (7)$$

Wise IoU v3 中引入了离群度 β 来衡量检测框的质量:

$$\beta = \frac{\mathcal{L}_{IoU}^*}{\mathcal{L}_{IoU}} \in [0, +\infty) \quad (8)$$

离群度小意味着检测框质量高, 为离群度较小的框分配较大的梯度增益, 为离群度较大的检测框分配较小的梯度增益, 减小质量较差的数据对模型的影响。并构造非单调聚焦系数 r :

$$\mathcal{L}_{WIoUv3} = r \mathcal{L}_{WIoUv1}, r = \frac{\beta}{\delta \alpha^{\beta - \delta}} \quad (9)$$

其中, 当 $\beta = \delta$ 时, 使得 $r = 1$ 。当锚框的离群程度满足 $\beta = C$ (C 为定值) 时, 检测框将获得最高的梯度增益。 \mathcal{L}_{IoU} 是动态的, 这也使得检测框的质量划分标准也是动态的, 保证了每时每刻都能做出最适合的梯度增益分配。

3.2. 引入轻量化颈部设计 Slim-Neck

3.2.1. GSConv

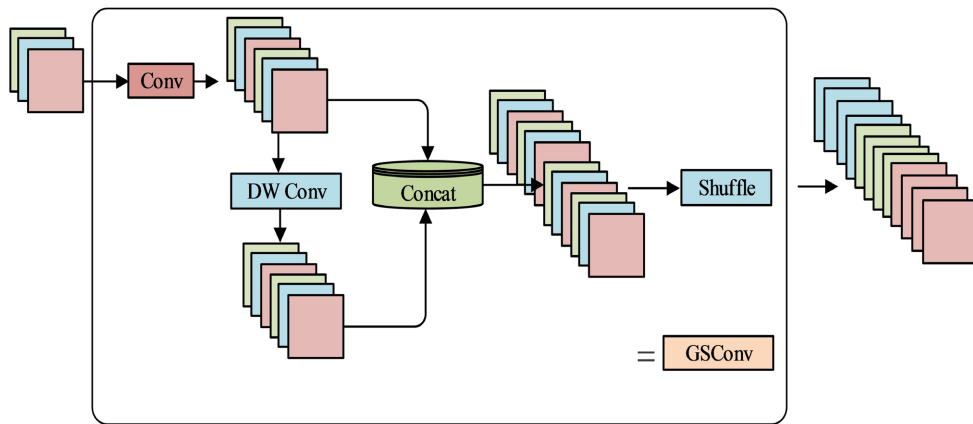


Figure 2. GSConv network structure

图 2. GSConv 网络结构

轻量化网络设计中常常使用深度可分离卷积(Depth-Wise Separable Convolution, DSConv) [25], 使用 DSConv 分别对每一个通道进行操作, 以此来减少参数量和计算量。但是, DSConv 存在着输入图像的通

道在计算过程中是分离的缺点，不可避免会丢失有用的信息，该缺点会在主干网络中被放大。为了使 DSConv 更接近于标准卷积(Standard Convolution, SConv)，引入了一种新的方法混合卷积。混合卷积由 SConv、DSConv 和 shuffle 构成，称为 GSConv。如图 2 所示，Conv 表示标准卷积，DW Conv 表示深度可分离卷积。GSConv 使用 shuffle 操作，将标准卷积生成的信息渗透到深度可分离卷积生成的信息的每一个部分，允许来自标准卷积的信息完全融合进深度可分离卷积的输出中，GSConv 以较低的时间复杂度保留了通道间隐藏的连接。

3.2.2. Slim-Neck

GSConv 能够尽量保留通道之间的连接，但是如果将其应用于整个网络中，必然加深网络模型的层数，增加推理时间。因此仅将 GSConv 用于颈部结构中，更能发挥其优势。原始网络颈部中标准卷积替换为 GSConv，保证计算结果尽可能接近标准卷积的同时，降低计算成本。而后又在 GSConv 的基础上，设计 GS bottleneck，图 3 显示了该模块的网络结构。

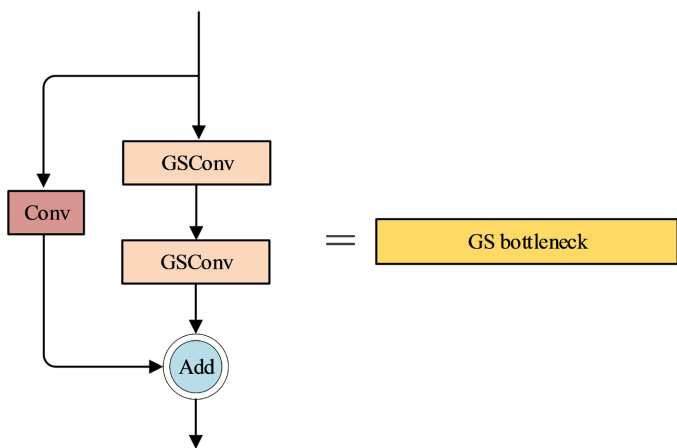


Figure 3. GSbottleneck network structure
图 3. GSbottleneck 网络结构

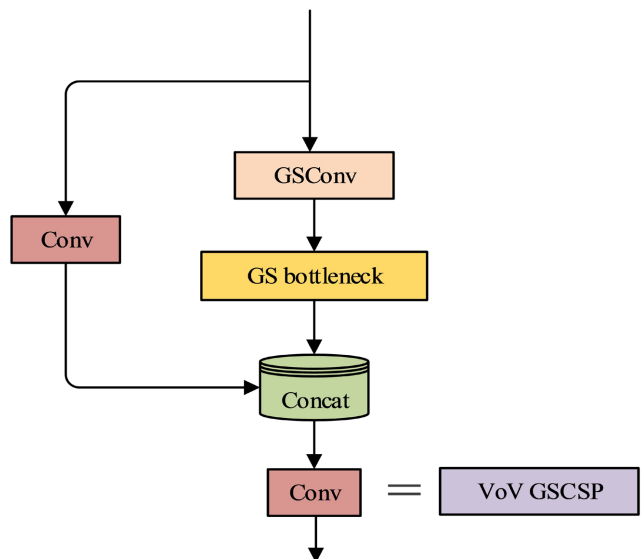


Figure 4. GSVoVGSCSP network structure
图 4. GSVoVGSCSP 网络结构

在卷积神经骨干网络中会经历空间信息逐步向通道传输，特征图压缩和通道拓展都会使得语义信息的一部分丢失。密集卷积计算会保留通道间的链接，稀疏的卷积则会完全切断这些连接。DenseNet [26]使用密集的连接，聚合了不同尺寸感受野的特征，每一层的输出都会聚合之前层的信息。虽然表现出了良好的检测性能，但是过于密集的连接，造成了特征冗余并且增大了计算的开销。有学者提出了一次性聚合操作(One-Shot Aggregation, OSA) [27]，就是减少密集卷积在中间层的聚合，只在最后一层进行一次聚合，既保留了通道间的隐秘连接又减少了冗余的计算开销。因为 OSA 模块的多样化特征表示和效率，所以可以通过单次聚合机制来构建精度高、速度快的 VoVNet 网络[28]。如图 4，VoV-GSCSP 的设计便是参考了 VoVnet 的机制，替换了标准卷积所设计出来的。

3.3. YOLOv5-W 网络结构

本文的优化算法是对 YOLOv5 算法的损失函数替换为 Wise IoU，原始的 Wise IoU 中，有两个超参数 α ， δ 分别设为 2.1 和 3。

将原始网络模型的颈部更换为了一个更轻量化的颈部设计，通过 GSConv 引入了 Slim-Neck，在减少模型复杂度的同时保持精度，提高速度。使用了轻量化的颈部设计得到的一个性能更优越、更轻量化的网络模型，并将其命名为 YOLOv5-W，如图 5 所示。

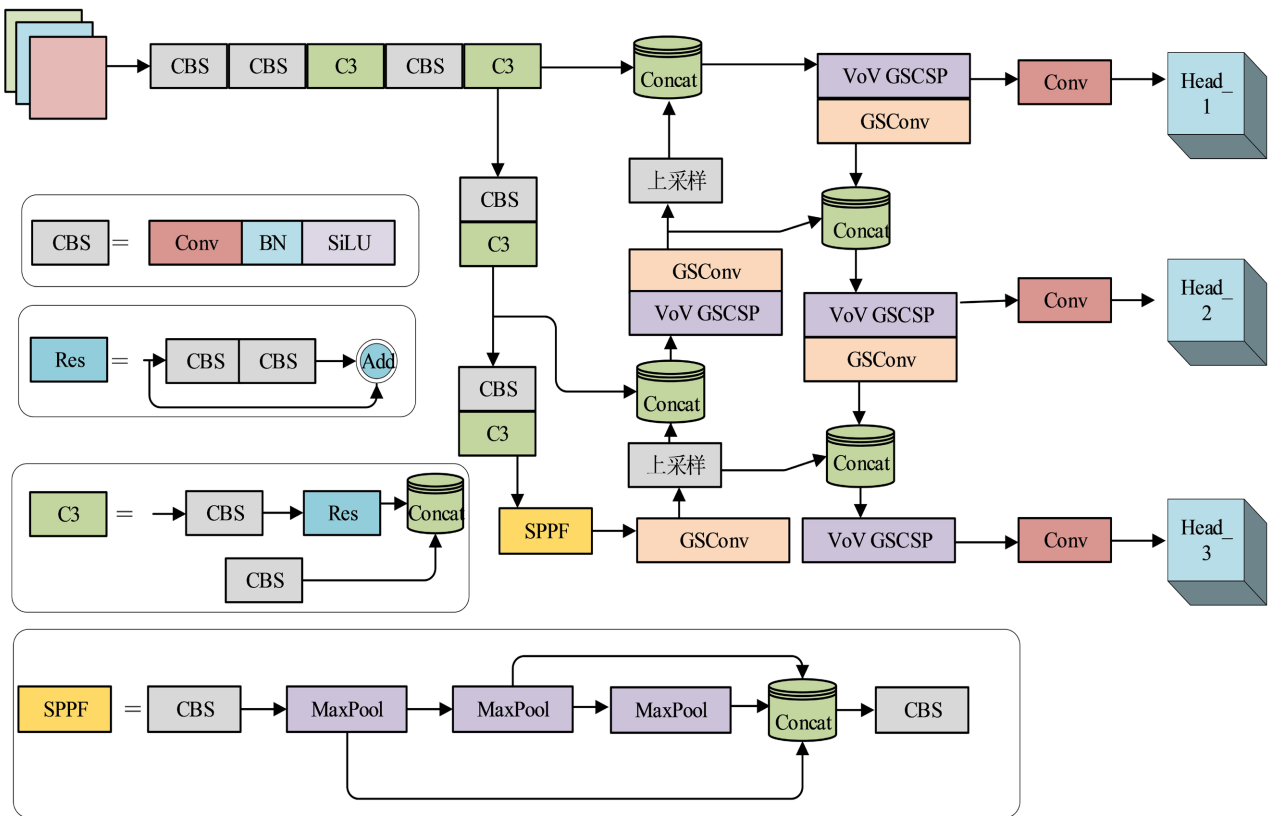


Figure 5. YOLOv5-W network architecture

图 5. YOLOv5-W 网络结构图

4. 数据集

本文的数据集为自制数据集，对有裂纹的箱梁进行实地拍摄两次，共计 400 张。两次设备分别是

iqooZ1 和 iPhone13。

根据现有的桥梁表面伤损规范提出相应的标注范式进行人工标注。通过 labellmg 标注后得到每一张图片所需检测的目标矩形框位置的标注文件。人工采集并标注后背景较为复杂的数据集命名为 Crack400。如图 6 所示，在 Crack400 数据集中，裂纹背景斑驳明显，伴有黑点和白色纹路等天然噪声，在复杂的背景下，对裂纹特征的提取必然会带来一定的干扰。

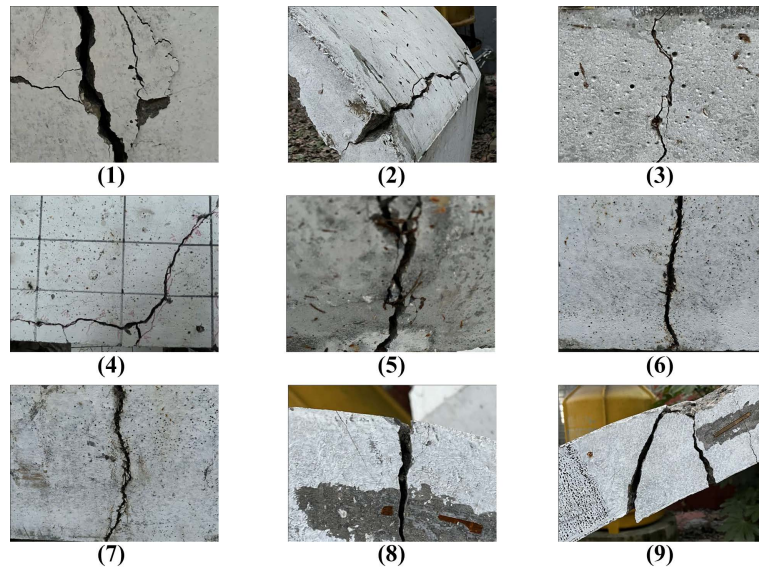


Figure 6. Crack400 dataset
图 6. Crack400 数据集

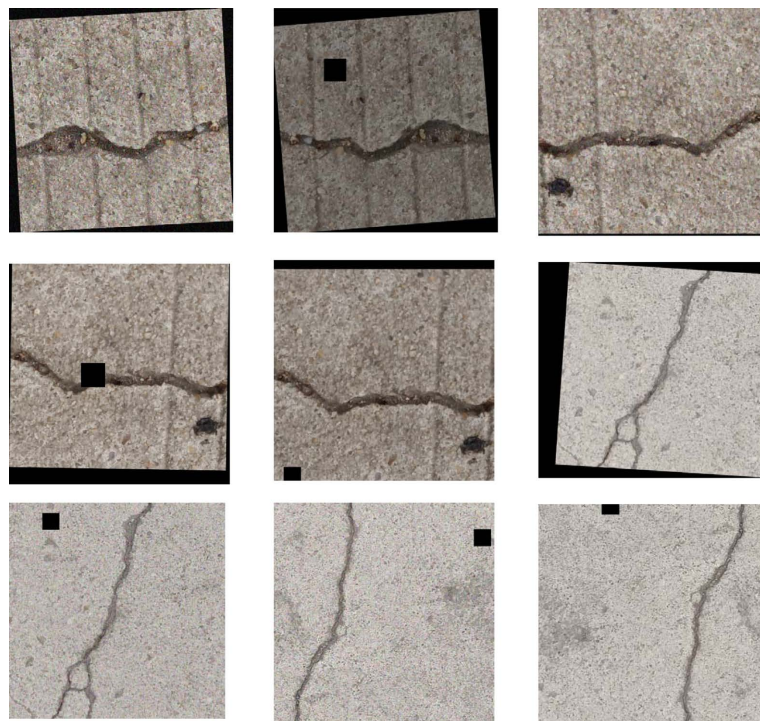


Figure 7. Crack2000 dataset
图 7. Crack2000 数据集

简单且重复的背景,更易于提取所需的裂纹特征。为对比模型对不同背景和数量的数据的检测效果,检测模型的可泛化性。本研究对 Crack400 数据集进行了一定的筛选。选取背景较为单一,没有天然干扰的数据,并对这些数据进行几何尺度的数据增强。具体的增强手段包括模糊,亮度,裁剪,旋转,平移,镜像等。对于数据集中出现的对比度比较低,肉眼不容易标注的图像。使用图像增强技术,使得图像对比度得到提升,桥梁裂纹细节得到增强。筛选与扩增后,背景较为单调的数据集命名为 Crack2000。如图 7 所示,人工扩增的数据裂纹形状更为单一,噪声多由黑色的小方块构成。

5. 实验结果

5.1. 实验环境

本文算法在 window10 系统下 Pytorch1.11.0 框架上实现,编程语言是 Python3.9,并使用 GPU 进行加速。实验平台中 CPU 配置为 Intel (R) xeon (R) Gold 6258R 2.7 GHz, GPU 配置为: NVIDIA Tesla V100 s, 显存大小为 32 GB。

5.2. 评价指标

本文采用的评价指标包括精度(Precision)、召回率(Recall)、均值平均精度(mean Average Precision, mAP)、帧率(Frames Per Second, FPS)、计算量(GFLOPSs)。

对于二分类问题,可以根据样例的真实类别和网络模型预测出的组合划分为真正例(true positive)、假正例(false positive)、真反例(true negative)、假反例(false negative),这四种情况。 TP 、 FP 、 TN 、 FN 分别表示对应的样例数,便可以得到 $TP + FP + TN + FN =$ 总数。精度 P 和召回率 R 的定义为:

$$P = \frac{TP}{TP + FP} \quad (10)$$

$$R = \frac{TP}{TP + FN} \quad (11)$$

查准率和查全率是一对矛盾的量,查准率越高,查全率往往越低;查全率较高时,查准率就会偏低。单独用两个评价指标难以衡量好网络模型,将两者结合起来,以查准率为纵轴查全率为横轴作图,得到的曲线就是 P - R 曲线。平均精度 AP 就是 P - R 曲线所围成的面积,平均精度均值 mAP 就是所有类别的平均精度求平均, mAP50 指的是 IoU 的值取 50%时的平均精度。mAP50: 95 指的是 IoU 的值从 50%取到 95%,步长为 5%。其公式如下:

$$AP = \int_0^1 P(r) dr \quad (12)$$

$$mAP = \frac{\sum_{i=1}^n AP_i}{N} \quad (13)$$

除平均精度外,平衡 F 分数即 F1-Score 也兼顾了模型的精确度和召回率, F1 分数可以看作是模型精确率和召回率的一种加权平均,其公式如下:

$$F_1 = \frac{P * R}{P + R} \quad (14)$$

FPS 是实时检测工作中一个尤为重要的衡量指标,它指的是目标检测网络每秒可以处理多少张图片,一般情况下每秒能够处理 30 张图片,就可以达到人眼能感知到的实时效果。FLOPs 是浮点运算数也就是计算量,该评价指标可以衡量模型的复杂程度。

5.3. 训练策略与实验结果

训练采用的是随机梯度下降算法(SGD), 预热回合数为 3, 初始动量为 0.8, 预热初始偏差为 0.1。在 Crack400 和 Crack2000 的对比实验中, 训练回合数为 100; 在 Crack400 消融实验中, 训练回合数为 300。

5.3.1. Crack400 实验

为验证 YOLOv5 作为基准模型的可行性, 在数据集 Crack400 上与常见的目标检测模型做了对比实验。在所有的对比实验中, 以 8:2 的比例随机分配各类裂纹图片作为模型的训练验证集和测试集, 设定训练轮次为 100 轮, 结果如表 1 所示。

Table 1. Crack400 comparison experiment

表 1. Crack400 对比实验

方法	主干	P/%	R/%	mAP50/%
SSD [29]	Vgg	82.1	65.3	71.0
SSD	Moilenet v2	75.0	55.2	72.7
FasterRcnn [30]	Vgg	54.6	73.5	66.0
FasterRcnn	Resnet50	62.1	73.5	68.0
YOLOv5	-	92.0	89.0	91.2

根据实验结果可知, 黑色加粗部分表示各项指标的最大值。不论是精度、召回率、还是 mAP50 各项指标上, YOLOv5 在 Crack400 数据集中检测的效果最好, 平均精度可达 91%, 相较于 SSD、Faster-RCNN 的平均精度在 65%~73%, 有显著优势。实验结果反映出 YOLOv5 在小数据集上更具有优势, 因此以其作为基准模型并优化更能提升现阶段网络模型对小型桥梁裂纹数据集的检测效果。

为验证优化策略中各模块对基准模型性能提升的效果, 以不同模块构成的配置文件作为实验环境进行消融实验, 仍然在 Crack400 小数据集上进行训练和验证。为更好验证收敛效果, 将训练轮次设为 300, 消融实验结果如表 2 所示。

Table 2. Crack400 ablation experiments

表 2. Crack400 消融实验

方法	mAP50%	mAP50: 95/%	FLOPs/10 ⁹	FPS/ms
YOLOv5	95.5	57.0	15.8	35.6
+Wisev3	97.1	56.8	15.8	40.2
+GS	96.0	55.5	15.8	40.5
+GS+Slim	97.5	61.8	14.5	30.5
Ours	98.5	61.0	14.5	47.9

表 2 从上至下依次添加了本文的优化模块, 通过消融实验可以看出, 本文的模块对模型检测目标的精度均有所提升。加入 Wise IoU 和 GSConv 能有效提升模型检测精度, 加入 GS + Slim-Neck, 虽然该模块降低了检测速度, 但是保证精度提升的同时也减小了模型 1.3GFLOPs 的计算量。本文的网络模型结合二者的优势, 在 mAP50 指标上提升 3%, mAP50: 95 指标上提升 4, 检测速度也提升 12 FPS。

Loss曲线对比

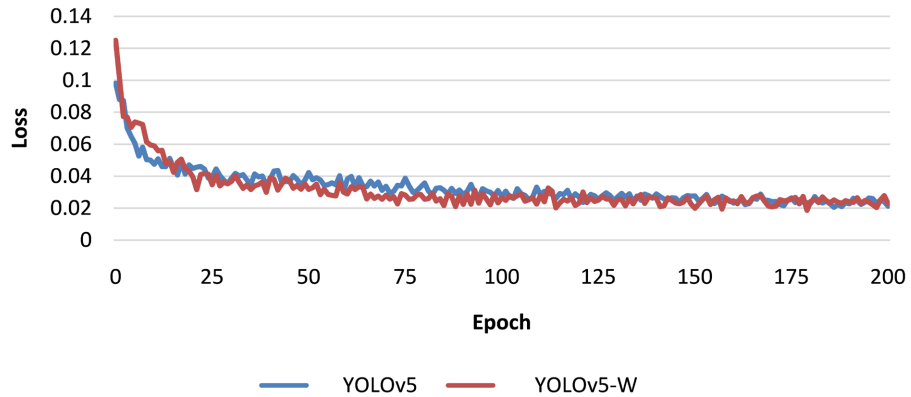


Figure 8. Comparison of loss curves
图 8. Loss 曲线对比

为验证优化后的算法收敛效果，绘制了与基准模型的 Loss 曲线对比图，如图 8 所示。本文的算法相较于原始算法收敛速度更快，更有效减少网络的训练时间。

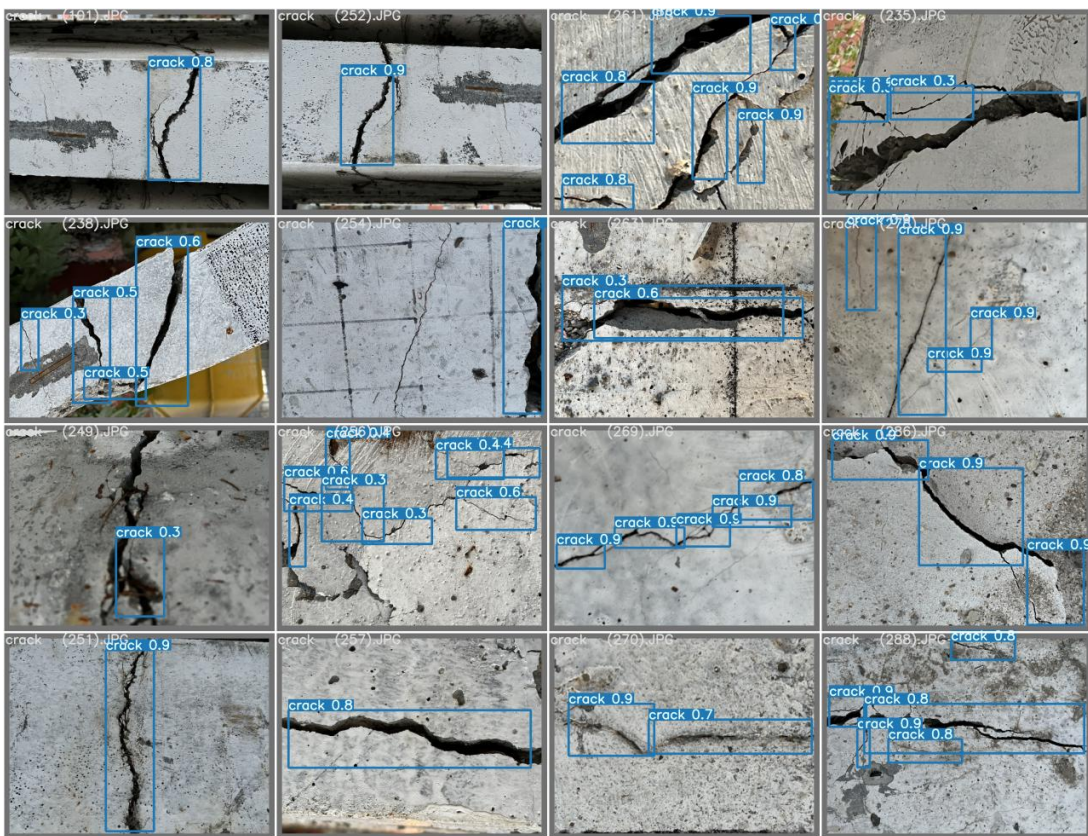


Figure 9. Forecast results
图 9. 预测结果

从图 9 的预测结果图中，可以看出本文的网络模型能够有效检测到样本中的目标，准确率能够保持在 80%~90%。

5.3.2. Crack2000 实验

Table 3. Crack 2000 comparison experiment
表 3. Crack 2000 对比实验

方法	主干	P/%	R/%	mAP50/%
SSD	Vgg	99.5	97.6	99.0
SSD	Mobilenetv2	99.4	83.9	96.6
FasterRcnn	Vgg	81.6	99.0	99.0
FasterRcnn	Resnet50	91.9	99.0	99.0
YOLOv5	-	99.4	99.4	99.5
YOLOv7 [31]	-	96.7	97.5	99.0
YOLOv8 [32]	-	99.9	99.5	99.5
Ours	-	99.9	99.5	99.5

如表 3, 由 Crack2000 对比实验可知, 不论 YOLO 系列、SSD、Faster-RCNN, 在该数据集上都能获得较为优秀的检测结果。效果最差的方法, 其平均精度也能够达到 0.99, 接近 1。其中本文的算法 YOLOv5-W 在 Crack2000 数据集上, 精度、召回率、平均精度上都能达到与 YOLOv8 相同的效果。

6. 结束语

本文自制了背景复杂数量较小的数据集 Crack400 和背景更为单调数据量大的数据集 Crack2000, 用于桥梁裂纹的研究。针对 Crack400 数据集中, 纹检测效率不高、背景复杂难以检测的问题设计出了改进的 YOLOv5-W 算法, 采用了动态化损失函数 Wise IoU, 引入了轻量化颈部网络设计。轻量化的颈部网络设计使得模型更小, 能更好地满足实时检测的要求。实验表明本文的改进方法明显优于原始算法、Faster-RCNN、SSD 等目标检测模型。但是, 本文的 YOLOv5-W 算法在模型大小上还有提升的空间, 后续将会进一步优化网络以获得更好的检测效果。

参考文献

- [1] 赵建华. 断裂力学在桥梁裂纹检测分析中的应用[D]: [硕士学位论文]. 西安: 长安大学, 2008.
- [2] 王恩永. 道路桥梁养护存在的问题及其预防措施探讨[J]. 中国高新技术企业, 2013(8): 98-99.
- [3] Yeum, C.M. (2015) Vision-Based Automated Crack Detection for Bridge. *Computer-Aided Civil and Infrastructure Engineering*, **30**, 759-770. <https://doi.org/10.1111/mice.12141>
- [4] Zhang, C., Wan, L., Wan, R.Q., Yu, J. and Li, R. (2022) Automated Fatigue Crack Detection in Steel Box Girder of Bridges Based on Ensemble Deep Neural Network. *Measurement*, **202**, Article ID: 111805. <https://doi.org/10.1016/j.measurement.2022.111805>
- [5] Dung, C.V., Sekiya, H., Hirano, S., et al. (2019) A Vision-Based Method for Crack Detection in Gusset Plate Welded Joints of Steel Bridges Using Deep Convolutional Neural Networks. *Automation in Construction*, **102**, 217-229. <https://doi.org/10.1016/j.autcon.2019.02.013>
- [6] Jang, K., Jung, H. and An, Y.K. (2022) Automated Bridge Crack Evaluation through Deep Super Resolution Network-Based Hybrid Image Matching. *Automation in Construction*, **137**, Article ID: 104229. <https://doi.org/10.1016/j.autcon.2022.104229>
- [7] 令雅莉, 杨桂芹, 张又元, 王小鹏. 基于改进算法 YOLOv5+的混凝土轨枕裂纹检测[J]. 铁道标准设计, 2023: 1-10. <https://doi.org/10.13238/j.issn.1004-2954.202210030001>
- [8] Ren, S., He, K., Girshick, R. and Sun, J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>

- [9] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2017) Mask R-CNN. 2017 *IEEE International Conference on Computer Vision (ICCV)*, Venice, 22-29 October 2017, 2980-2988. <https://doi.org/10.1109/ICCV.2017.322>
- [10] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
- [11] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 21-26 July 2017, 6517-6525. <https://doi.org/10.1109/CVPR.2017.690>
- [12] Liu, W., Anguelov, D., Erhan, D., et al. (2016) SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *ECCV 2016: Computer Vision—ECCV 2016*, Springer, Cham, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
- [13] Hacıfendioglu, K. and Başağa, H.B. (2022) Concrete Road Crack Detection Using Deep Learning-Based Faster R-CNN Method. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, **46**, 1621-1633. <https://doi.org/10.1007/s40996-021-00671-2>
- [14] Zhang, Y.X., Huang, J. and Cai, F.H. (2020) On Bridge Surface Crack Detection Based on an Improved YOLO v3 Algorithm. *IFAC-PapersOnLine*, **53**, 8205-8210. <https://doi.org/10.1016/j.ifacol.2020.12.1994>
- [15] Yu, Z., Shen, Y. and Shen, C. (2021) A Real-Time Detection Approach for Bridge Cracks Based on YOLOv4-FPM. *Automation in Construction*, **122**, Article ID: 103514. <https://doi.org/10.1016/j.autcon.2020.103514>
- [16] Zhang, J., Qian, S. and Tan, C. (2022) Automated Bridge Surface Crack Detection and Segmentation Using Computer Vision-Based Deep Learning Model. *Engineering Applications of Artificial Intelligence*, **115**, Article ID: 105225. <https://doi.org/10.1016/j.engappai.2022.105225>
- [17] Mishra, M., Jain, V., Singh, S.K. and Maity, D. (2022) Two-Stage Method Based on the You Only Look Once Framework and Image Segmentation for Crack Detection in Concrete Structures. *Architecture, Structures and Construction*, **3**, 429-446. <https://doi.org/10.1007/s44150-022-00060-x>
- [18] Zhang, Y.F., Ren, W.Q., Zhang, Z., Jia, Z., Wang, L. and Tan, T.N. (2022) Focal and Efficient IOU Loss for Accurate Bounding Box Regression. *Neurocomputing*, **506**, 146-157. <https://doi.org/10.1016/j.neucom.2022.07.042>
- [19] Lin, T.Y., Dollár, P., Girshick, R., et al. (2017) Feature Pyramid Networks for Object Detection. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 21-26 July 2017, 936-944. <https://doi.org/10.1109/CVPR.2017.106>
- [20] Wang, C.Y., Bochkovskiy, A. and Liao, H.Y.M. (2021) Scaled-YOLOv4: Scaling cross Stage Partial Network. 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, 20-25 June 2021, 13024-13033. <https://doi.org/10.1109/CVPR46437.2021.01283>
- [21] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934
- [22] Liu, S., Qi, L., Qin, H., et al. (2018) Path Aggregation Network for Instance Segmentation. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 8759-8768. <https://doi.org/10.1109/CVPR.2018.00913>
- [23] Yin, X., Sasaki, Y., Wang, W. and Shimizu, K. (2020) 3D Object Detection Method Based on YOLO and K-Means for Image and Point Clouds. arXiv:2005.02132
- [24] Tong, Z.J., Chen, Y.H., Xu, Z.W. and Rong, Y.U. (2023) Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. arXiv:2301.10051. <https://doi.org/10.48550/arXiv.2301.10051>
- [25] Li, H., Li, J., Wei, H., et al. (2022) Slim-Neck by GSConv: A Better Design Paradigm of Detector Architectures for Autonomous Vehicles. arXiv:2206.02424
- [26] Huang, G., Liu, Z., Van Der Maaten, L., et al. (2017) Densely Connected Convolutional Networks. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 21-26 July 2017, 2261-2269. <https://doi.org/10.1109/CVPR.2017.243>
- [27] Liang, T., Wang, Y., Tang, Z., et al. (2021) OPANAS: One-Shot Path Aggregation Network Architecture Search for Object Detection. 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, 20-25 June 2021, 10190-10198. <https://doi.org/10.1109/CVPR46437.2021.01006>
- [28] Lee, Y., Hwang, J.W., Lee, S., Bae, Y. and Park, J. (2019) An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection. 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, 16-17 June 2019, 752-760. <https://doi.org/10.1109/CVPRW.2019.00103>
- [29] Jia, Z.G., Su, X., Ma, G.D., Dai, T.T. and Sun, J.B. (2023) Crack Identification for Marine Engineering Equipment Based on Improved SSD and YOLOv5. *Ocean Engineering*, **268**, Article 113534. <https://doi.org/10.1016/j.oceaneng.2022.113534>

-
- [30] Ju, H.Y., Wei, L., Tighe, S., Zhai, J.Z., Xu, Z.C. and Chen, Y. (2019) Detection of Sealed and Unsealed Cracks with Complex Backgrounds Using Deep Convolutional Neural Network. *Automation in Construction*, **107**, Article 102946. <https://doi.org/10.1016/j.autcon.2019.102946>
- [31] Kumar, P.S. and Kumar, N.K. (2023) Drone-Based Apple Detection: Finding the Depth of Apples Using YOLOv7 Architecture with Multi-Head Attention Mechanism. *Smart Agricultural Technology*, **5**, Article ID: 100311. <https://doi.org/10.1016/j.atech.2023.100311>
- [32] Wang, G., Chen, Y., An, P., *et al.* (2023) UAV-YOLOv8: A Small-Object-Detection Model Based on Improved YOLOv8 for UAV Aerial Photography Scenarios. *Sensors*, **23**, Article 7190. <https://doi.org/10.3390/s23167190>