

分类号：TU834

单位代码：10363

密 级：公开

学 号：2210330140

题 目 基于深度学习的砌筑墙裂缝检测技术研究
与实现

题 目 Research and Implementation of Masonry Wall
Crack Detection Technology Based on Deep
Learning

学 生 姓 名：马鸣

校 内 导 师：程凡永（副教授）

校 外 导 师：刘广海（工程师）

专业学位类别：电子信息

研究方向（领域）：人工智能与系统集成

论文答辩日期：2024 年 5 月 19 日

安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名： 

日期： 2024 年 5 月 24 日

安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 ____ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：



日期：2024 年 5 月 24 日

指导教师签名：



日期：2024 年 5 月 24 日

基于深度学习的砌筑墙裂缝检测技术研究与实践

摘 要

在建筑施工领域，砌筑墙的墙体裂缝检测是一项决定建筑质量的关键环节。企业受限于成本一般采取结果管控，而不采取全程管控，因此无法避免的在生产过程中产生残次品，对其销毁重制又导致了资源浪费。此外，墙体裂缝影响建筑物的保温和隔音性能，也可能导致公众对结构稳定性的误判，引起安全担忧。将深度学习技术应用于砌筑墙裂缝检测，可以降低管控成本并提升生产成品的良品率，但仍面临若干挑战，例如：缺乏施工现场采集的数据集；通用模型不适配砌筑墙裂缝缺陷检测；模型参数量和检测精度之间没有进行很好的平衡；部署在移动端的检测模型需满足实时检测需求。为了使深度学习方法在砌筑墙裂缝检测领域应用于工程实际，本文针对上述挑战，开展了砌筑墙裂缝检测技术研究与实践，主要研究工作包括以下几个方面：

（1）完成了数据采集与数据集构建，并设计了针对砌筑墙裂缝的目标检测模型。为确保模型实用性，通过与知名施工企业合作，采集了施工现场砌筑墙裂缝数据，筛选并标记了不符合建筑规范的裂缝位置信息，从而构建了高质量的数据集。以 YOLOX 算法为基准，首先，在骨干网络引入 CBAM 注意力机制以提高模型对砖块区域的关注程度；其次，在 HEAD 部分之后采用线性加权 Soft-NMS 应对砌筑

墙裂缝相互交织重叠的问题。该模型有效提升了裂缝检测的准确性，减少了误检和漏检，有效地适配了砌筑墙裂缝检测问题。

(2) 设计了轻量化墙体裂缝检测模型，优化模型参数，减少参数量，使其适应计算能力较低的边缘设备。首先，引入 GhostBottleneck 模块对骨干网络进行改进，该模块在减少模型参数量和计算复杂度的同时，仍能提取到有效的特征表示，达到参数量与检测精度的平衡；其次，将 NECK 部分的 PaNet 改进为 BIFPN，通过这种机制增强特征融合效果，进一步提升特征表示能力，从而有效提升了模型精度。实验结果表明所提的整体轻量化策略非常有效，经过优化后的网络参数量相比原模型总体参数量减少了 62.43%，同时由于 BIFPN 拥有双向加权策略可以更好地对不同尺寸的特征进行融合，使得模型的检测精度提高了 1.52%。

(3) 开发了基于 web 端的砌筑墙裂缝检测系统，推动模型的产业化应用。为提供一个界面友好且能部署在移动端的可视化系统给工程技术人员使用，来满足过程管控中实时检测的需求，通过 Flask 框架搭配 HTML 进行了模型的 Web 端部署。首先，利用 HTML 设计了包含注册、登录及功能选择的用户界面，通过 PyCharm 中引入 Flask 库，实例化应用并设置路由以实现页面交互，用户可通过浏览器实时获取检测结果；其次，完成了服务器端的数据库设计，在服务器端进行数据保存，实现了数据从获取到检测，最后到保存的完整功能。

关键词：深度学习，目标检测，砌筑墙裂缝，轻量级网络，网页部署

RESEARCH AND IMPLEMENTATION OF MASONRY WALL CRACK DETECTION TECHNOLOGY BASED ON DEEP LEARNING

ABSTRACT

In the field of building construction, the crack detection of masonry wall is a key link that determines the quality of construction. Limited by the cost, enterprises generally adopt the result control rather than the whole process control, so it is inevitable that defective products will be produced in the production process, and the destruction and remaking of them will lead to the waste of resources. In addition, wall cracks affect the thermal insulation and sound insulation performance of buildings, and may also lead to public misjudgment of structural stability, causing safety concerns. The application of deep learning technology to crack detection of masonry walls can reduce control costs and improve the yield of finished products, but there are still several challenges, such as: lack of data set collected at the construction site; The general model is not suitable for the crack defect detection of masonry walls. There is no good

balance between the number of model parameters and the detection accuracy. The detection model deployed on the mobile terminal must meet the real-time detection requirements. In order to make the deep learning method applied in the field of masonry wall crack detection in engineering practice, this paper has carried out the research and implementation of masonry wall crack detection technology in view of the above challenges. The main research work includes the following aspects:

(1) The data collection and data set construction are completed, and the target detection model for masonry wall cracks is designed. In order to ensure the practicality of the model, through cooperation with well-known construction enterprises, the crack data of masonry walls at the construction site was collected, and the crack location information that did not meet the building code was screened and labeled, so as to construct a high-quality data set. Based on YOLOX algorithm, firstly, CBAM attention mechanism is introduced into the backbone network to improve the model's attention to the brick region. Secondly, after the HEAD part, a linear weighted Soft-NMS is used to deal with the problem of overlapping cracks in masonry walls. This model can effectively improve the accuracy of crack detection, reduce false detection and missing detection, and effectively adapt to the problem of crack detection of masonry walls.

(2) A lightweight wall crack detection model is designed to optimize the model parameters and reduce the number of parameters, so that it can adapt to the edge equipment with low computing power. Firstly, it introduces the GhostBottleneck module to improve the backbone network. This module reduces the number of model parameters and computational complexity, and can still extract effective feature representations to achieve a balance between the number of parameters and detection accuracy. Secondly, the NECK part of the PaNet is improved to BIFPN, through this mechanism to achieve feature fusion, further improve the feature representation ability, thus effectively improving the accuracy of the model. The experimental results show that the proposed overall lightweight strategy is very effective, and the number of network parameters after optimization is reduced by 62.43% compared with the total number of parameters in the original model. At the same time, because BIFPN has a bidirectional weighting strategy, it can better integrate features of different sizes, so the detection accuracy of the model is increased by 1.52%.

(3) The crack detection system of masonry wall based on web is developed to promote the industrialization application of the model. In order to provide a user-friendly visual system that can be deployed on mobile terminal for engineers and technicians to meet the requirements of real-time detection in process control, the model is deployed on the Web

side through Flask framework and HTML. Firstly, HTML is used to design the user interface including registration, login and function selection. Flask library is introduced into PyCharm to instantiate the application and set the route to achieve page interaction. Users can obtain the detection results through the browser in real time. Secondly, the database design of the server side is completed, and the data is saved on the server side, which realizes the complete function of data acquisition, detection and finally saving.

KEY WORDS: deep learning, object detection, masonry wall cracks, lightweight networking, web deployment

目 录

第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 通用目标检测研究现状.....	2
1.2.2 砌筑墙裂缝目标检测研究现状.....	4
1.3 本文主要研究内容与章节安排.....	6
1.3.1 本文的主要研究内容.....	6
1.3.2 本文的章节安排.....	8
第 2 章 基本理论与数据介绍.....	10
2.1 神经网络.....	10
2.2 神经网络的训练过程.....	10
2.2.1 数据预处理.....	11
2.2.2 CNN 特征提取.....	15
2.2.3 分类和回归预测以及 NMS.....	17
2.2.4 损失函数.....	18
2.3 目标检测常用框架介绍.....	19
2.3.1 YOLO.....	19
2.3.2 SSD.....	20
2.3.3 Faster R-CNN.....	21
2.4 砌筑墙裂缝检测目标数据集.....	21
2.5 线性加权的 Soft-NMS.....	24
2.6 本章小结.....	26
第 3 章 基于注意力机制的砌筑墙裂缝目标检测算法.....	27
3.1 基准算法简介.....	27
3.2 卷积注意力模块简介.....	28
3.3 算法改进介绍.....	29
3.4 实验结果和分析.....	30
3.4.1 评价指标.....	30
3.4.2 实验环境.....	30
3.4.3 NMS 改进效果分析.....	31
3.4.4 基于注意力机制的改进实验结果分析.....	32
3.4.5 消融实验结果分析.....	34
3.4.6 模型性能评估.....	34
3.5 本章小结.....	36
第 4 章 轻量化的砌筑墙裂缝目标检测算法.....	37
4.1 轻量化模型需求.....	37
4.2 骨干网络轻量化改进原理.....	38

4.3 NECK 模块改进原理.....	39
4.4 轻量化改进后模型结构简介.....	41
4.5 实验结果与分析.....	42
4.5.1 实验环境.....	42
4.5.2 骨干网络模块横向对比实验.....	43
4.5.3 改进前后模型性能对比分析.....	43
4.5.4 消融实验结果分析.....	46
4.5.5 算法鲁棒性分析.....	47
4.6 本章小结.....	48
第 5 章 砌筑墙裂缝检测系统 web 端部署.....	49
5.1 系统需求.....	49
5.2 系统总体设计.....	49
5.2.1 系统开发环境.....	49
5.2.2 系统整体架构设计.....	50
5.3 系统软件实现.....	52
5.3.1 登录和注册界面.....	52
5.3.2 检测功能界面.....	53
5.4 系统功能测试.....	54
5.5 本章小结.....	55
第 6 章 总结与展望.....	56
6.1 全文工作总结.....	56
6.2 后续研究展望.....	57
参考文献.....	59
攻读硕士学位期间所发表的论文及其它研究成果.....	65
致谢.....	66

第 1 章 绪论

1.1 研究背景及意义

双碳目标是指“碳中和”与“碳达峰”两个基本目标^[1]，是党中央作出的重大战略决策，是一场广泛而深刻的经济社会系统性的变革。作为排放“大户”，建筑业在实现“双碳”目标中责任重大，亟待向绿色化、智慧化、工业化和国际化转型。由传统粗放建造方式向信息化建造方式变革已是刻不容缓^[2]。

建筑业作为一门传统行业，建筑业信息化的水平及其应用性等方面都存在不足，特别是应用性问题，一直是制约建筑业信息化的主要瓶颈之一。随着人工智能带来的重大技术变革，建筑业与人工智能结合将是必然趋势^[3]因此人工智能对于减少碳排放以及实现“双碳”目标具有十分重要的意义，同时也能为建筑业的进一步信息化发展提供契机。在建筑施工领域，砌筑墙体的裂缝问题时有发生，这一现象若在施工过程中未能得到妥善管理，便会导致大量不合格产品的产生。这些不合格品的出现，不仅需要返工或销毁，重新制作，造成材料资源的极大浪费，还会对项目的整体进度和成本控制产生负面影响。如果在每一个砌筑墙产品从生产开始到结束的整个过程中，都采取全程旁站管控的措施，将耗费大量的管理资源和人力，其所产生的成本可能会超过因出现大量不合格品而引发的损失。这种过度的质量控制措施，虽然能够在一定程度上确保产品质量，但并不是一种经济高效的管理策略。此外，墙体裂缝的存在，尽管可能并不影响结构的整体稳定性，但却容易给人们造成错觉，误认为建筑结构存在安全隐患，甚至担心有倒塌的风险，这会严重影响居住者的安全感。同时，裂缝还会导致建筑物的保温和隔声性能下降，影响居住舒适性和节能效果^[4]。因此，提升产品质量、降低资源浪费是房屋建筑业一项重要课题^[5]。近年来人工智能中的深度学习方向因其在处理背景复杂、缺陷微弱的工业图像时展现出卓越的性能优势逐渐在该领域占据主导地位^[6]。所以，无论对建筑行业信息化的发展还是从实现“双碳”目标的角度考虑，基于深度学习的砌筑墙裂缝检测技术意义重大。

但基于深度学习的方法在砌筑墙裂缝检测领域却没有大规模应用。这一现象

主要源于几个关键因素：首先，由于工程质量图片涉及企业敏感信息，导致缺乏从实际工程现场收集的数据集，这限制了模型训练过程中的数据多样性和实用性。其次，现有算法并不适配砌筑墙裂缝错综复杂、相互重叠的特点，难以精准捕捉。此外，考虑到砌筑墙裂缝仅出现在红砖表面，这就要求模型必须能够特别关注红砖区域，以提升检测精度。最后，土木工程技术人员往往缺乏编写代码和理解深度学习基础理论的经验，因此即使提供了经过训练的模型，如何使其得到有效利用对他们来说仍然是一个不小的挑战。这些限制条件共同构成了目前基于深度学习的砌筑墙裂缝检测技术应用于工程实际所面临的主要障碍。

本研究课题采用基于 YOLOX^[7]算法来探究砌筑墙裂缝的目标检测技术，YOLO^[8]系列目标检测技术在实时性和准确性等方面具备显著的优势。面对上文提到的挑战，采取了直接从施工现场采集数据集的方法，并设计了一款基于 YOLOX 改进的砌筑墙裂缝检测模型专门适配砌筑墙裂缝检测任务。通过优化网络结构和进行轻量化处理，不仅提高了图像处理的效率，还增强了检测精度。最终，该模型被部署在 Web 端，使工程技术人员能够无障碍使用深度学习模型预测砌筑墙裂缝，完成了基于深度学习的砌筑墙裂缝检测技术研究实现。

1.2 国内外研究现状

1.2.1 通用目标检测研究现状

目标检测的发展历程可以分为两个主要阶段：传统目标检测算法时期和基于深度学习的目标检测算法时期^[9]。在传统目标检测算法时期，目标检测的主要流程包括选取感兴趣区域、特征提取和分类检测^[10]。这一时期的代表性算法有 VJ Det^[11]、HOG Det^[12]、DMP^[13]等。这些算法主要依赖于手工提取的特征，而不是自动学习特征表示。

自 2012 年 Alex Krizhevsky 和 Geoffrey Hinton 提出 AlexNet^[14]卷积神经网络算法大幅提高了图像识别的准确率之后，大量的视觉任务开始应用基于深度学习的方法，这标志着目标检测算法的发展开始迅速演进。随着技术的历史长河不断涌动，许多具有里程碑意义的经典算法相继诞生。进入基于深度学习的目标检测算法时期后，目标检测的发展又可以分为一阶段（One-stage）和两阶段

（Two-stage）模型^[15]。具体如下：

两阶段模型：最早的两阶段检测器可以追溯到 R-CNN^[16]。这一阶段的模型首先从输入图像中抽取感兴趣区域（RoI），然后对每个 RoI 抽取特征，最后使用分类器（如 SVM）来识别每个 RoI 的类别。这一阶段的代表性模型包括 RCNN、SPP-Net^[17]、Fast R-CNN^[18]、Faster R-CNN^[19]、R-FCN^[20]、FPN^[21]和 Mask R-CNN^[22]等。

两阶段模型中经典的 R-CNN 算法，这是一种在 2013 年提出结合区域提案与卷积神经网络的体系结构，在 2014 年还提出了 Fast R-CNN 算法，这一算法通过使用单一尺度的网格和池化层来改善感兴趣区域的池化处理。Fast R-CNN 将 R-CNN 在 VOC^[23]数据集上的训练时间从 84 小时缩短到了 9.5 小时，并且将目标检测时间从 45 秒减少到仅 0.32 秒。2018 年，何凯明等人提出了 Mask R-CNN 算法，目的是提高 R-CNN 系列算法的目标检测速度。Mask R-CNN 放弃了 Faster R-CNN 中 ROI 特征取整的方法，转而使用双线性插值法估算非整数位置的像素值。在 COCO 数据集上，Mask R-CNN 的检测准确率相较于 Faster R-CNN 提高了大约 20%。在 two-stage 算法中，主要是在 RCNN 算法上的改进，例如在 2020 年提出的 G-RCNN^[24]是基于 RCNN 的一种改进。G-RCNN 可以同时预测流程图中的边缘和节点信息，并将它们编译成源代码。同年 Lin Jiao 等人提出 AF-RCNN^[25]可以通过端到端的方式，将无锚框的区域建议网络（Anchor Free-Region Proposal Network, AF-RPN）和 Fast RCNN 结合成一个单一的网络。

一阶段模型：与两阶段模型不同，一阶段模型直接对预测框进行回归和分类预测，无需生成 RoI 的步骤。这一阶段的代表性模型包括 YOLO 和 SSD^[26]等。一阶段模型在速度上具有优势，但早期在准确性上不如两阶段模型。随着研究的深入，一阶段模型的准确性也得到了显著提升^[27]。

在 2015 年，推出了 YOLO 算法（You Only Look Once），该算法继承了 OverFeat^[28]算法的滑动窗口方法来获取候选框，并利用多尺度的滑动窗口来解决因目标检测复杂性导致的特征学习难题，将检测速度提高到了每秒处理 45 张图片。在 2016 年，改进了 YOLO 算法的 SSD 算法被提出，它通过采用不同大小的先验框（Anchor Box）以及多尺度的特征图进行检测的策略，使算法的推理速度以及检测精度获得了显著提高。在 2017 年，RetinaNet 算法被提出^[29]，其精度

优于 SSD 算法，使用焦点损失函数（Focal Loss），避免了正样本和负样本失衡情况的出现。同年，又推出了 YOLOv2 和 YOLO9000 两个版本算法^[30]，它们使用 Darknet19 作为基础特征提取网络，并加入了批归一化（Batch Normalization,BN）层来优化图像预处理，同时提升了图像的输入分辨率。在 2018 年，继续提出了 YOLOv3^[31]算法，这是基于之前版本 YOLOv2 的进一步改进，主要改进点包括将原有的 Darknet19 架构升级到了更加复杂的 Darknet53，并且整合了特征金字塔网络（FPN）以及残差网络（ResNet）^[32]，同时检测速度也达到了每张图像 22.2 毫秒的快速处理效果。2020 年，成功开发了 YOLOv4^[33]算法，作为 YOLOv3 的进阶版本。该算法通过融入空间金字塔池化、跨阶段局部网络以及路径聚合策略网络等先进技术，并采纳了 Mish^[34]激活函数，显著增强了性能。2021 年，Glenn Jocher 及其团队提出了 YOLOX 算法，该算法建立在 YOLOv3 的基本框架上，并融入了若干创新设计，这些新增元素包括 Decoupled Head、数据增强（Data Aug）、无锚点（Anchor Free）技术以及 SimOTA 模块。

目标检测领域的研究现状发展过程表明，这是一个活跃且快速发展的研究领域，不断有新的技术和方法被提出和改进。随着技术的不断进步，未来目标检测的应用将更加广泛，性能也将更加强大。图 1-1 为目标检测发展历程图，包括了早期目标检测算法、一阶段和二阶段算法的发展历程。

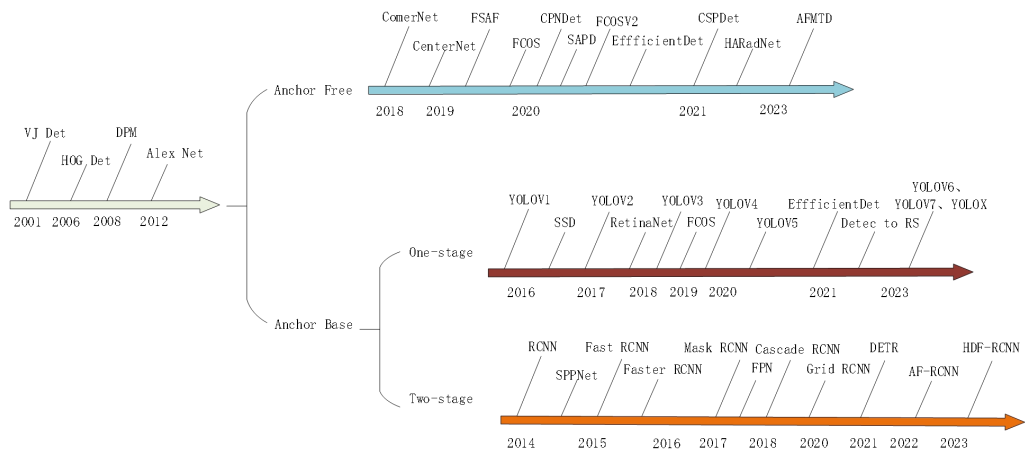


图 1-1 目标检测发展历程图

1.2.2 砌筑墙裂缝目标检测研究现状

砌筑墙作为一种由砖块和水泥砂浆构成的建筑结构，具有砖块的脆性特性，加之完全依赖手工制作的过程，使得在生产过程中出现裂缝成为难以避免的现象

[35]。这些裂缝往往呈现出交错、重叠的特征，当受到外界荷载作用时，可能会导致显著的结构质量问题，并对建筑的保温隔热性能产生不利影响。此外，施工现场环境的复杂多变，工人技术能力的差异，以及材料运输过程中的不可控因素，均增加了砌筑墙体裂缝问题的管控难度。因此，有效识别和低成本监管砌筑墙体裂缝，已经成为了建筑工程领域中的焦点问题和挑战[36]。

随着建筑行业的飞速发展，对建筑物的质量与安全性标准日益攀升，企业亦在不断追求成本效益化。在此背景下，研发出高效且成本较低的砌筑墙裂缝检测技术显得尤为关键。基于深度学习的砌筑墙裂缝检测技术，在当前建筑领域具有显著的工程实践价值，其主要研究内容是对砌筑墙体表面裂缝进行目标检测。在我国，利用深度学习技术进行砌筑墙裂缝检测的研究与应用尚处于初级阶段。在国际层面，Mitchell J. Halle^[37]等学者已经提出了一种采用卷积神经网络（CNN）来执行砌筑墙裂缝检测的方法，为该领域的研究提供了一种新思路。

尽管基于深度学习的砌筑墙裂缝检测技术在理论和实验层面取得了一些进展，但其在实际应用中仍面临一系列挑战。首先，当前方法所依赖的数据集主要采集自网络资源和实验室环境下自制的墙体样本，这些样本与实际工程现场的砌筑墙裂缝存在明显差异。这种数据来源的偏差可能导致模型在工程环境中的识别精度受限，进而影响其在工程应用的可靠性。其次，现行方法通常采用通用的目标检测算法，并未针对砌筑墙裂缝的独特特性进行优化。例如，砌筑墙裂缝的尺寸多变、相互交织或重叠，以及仅出现在砖块区域的特点，都要求算法具备更高的适配性和准确性。然而，通用算法往往无法充分捕捉这些细微特征，从而影响检测效果。此外，考虑到实际工程中的实时检测需求，模型必须能够在计算能力有限的设备上高效运行。目前，基于深度学习的方法通常涉及较大的参数量，对部署平台的计算能力有较高要求，这不仅增加了使用成本，也限制了技术的普及和应用范围。最后，土木工程技术人员普遍缺乏代码编写和深度学习领域的专业知识，这使得他们在采纳和实施该技术时面临较大困难。因此，对于这一专业群体来说，现有方法的复杂性构成了一个实质性的挑战。

综上所述，在推动其向实际工程应用转化的过程中，需对数据来源、算法针对性、模型参数量以及部署系统的用户友好性等方面进行深入研究。这些努力将有助于提升该技术在现场实用性，确保其在工程领域能够发挥更大的价值。

1.3 本文主要研究内容与章节安排

1.3.1 本文的主要研究内容

为了应对当前在建筑行业中，采用深度学习技术进行砌筑墙裂缝检测所遭遇的挑战，本研究对存在的核心问题进行了细致的梳理和分析。这些问题主要包括：数据集的构建、模型的设计、模型的轻量化以及模型在 web 端的部署。本文的技术路线如图 1-2 所示。

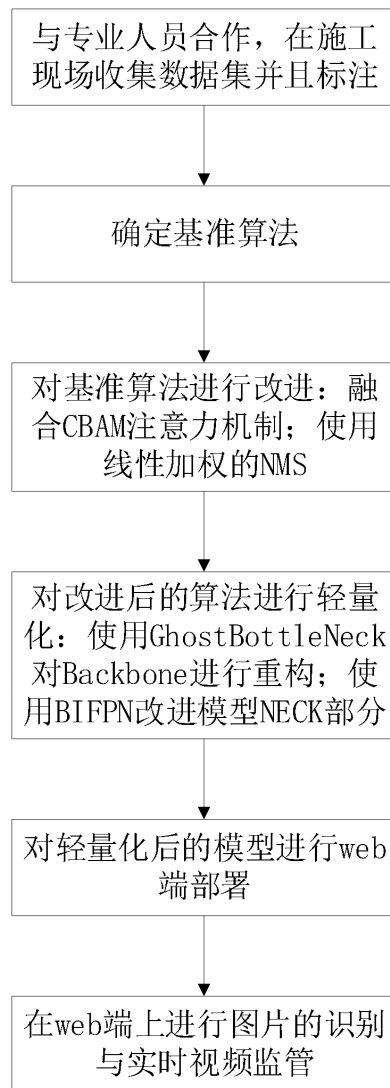


图 1-2 本文技术路线流程图

针对这四个问题，本文深入研究了特征提取及融合、特征选择、数据集构建等方向，涉及了基于 CBAM 注意力模块改进模型^[38]、线性加权 Soft-NMS^[39]、基

于 GhostConv 的骨干网络^[40]、基于 BIFPN 的颈部网络^[41]、基于 web 端的模型部署的方法。本文的主要工作内容如下：

1.构建基于实际工程的数据集，确保模型训练和应用的有效性和可靠性。

在对国内外研究现状进行深入调研后，了解到砌筑墙裂缝图片属于企业敏感信息，涉及质量问题，大量采集基于实际生产的砌筑墙裂缝数据集较为困难。通过寻求与知名企业合作研究，获得了施工现场的砌筑墙图片。使用 Labelimg 软件，与专业的土木工程师合作，根据施工规范标记出不合格的裂缝，构建了基于实际工程的高质量数据集。

2.搭建砌筑墙裂缝目标检测模型，有效适配了砌筑墙裂缝检测问题

本课题旨在寻找一款算法，该算法需要同时具备实时处理能力和高精度的检测性能。通过对三种主流的算法框架、深度神经网络理论和砌筑墙裂缝缺陷特点的详细分析，选择 YOLOX 算法的 L 版本作为研究砌筑墙裂缝检测技术的基准方法。并且根据砌筑墙裂缝相互交织、重叠以及裂缝仅出现在红砖区域的特点，改进模型以适配砌筑墙裂缝特点，达到更有效的特征提取和检测识别。通过改进模型增加通道空间注意力机制来增强网络对重要目标信息也就是红砖区域的关注，最后通过线性加权的 Soft-NMS 算法来减少裂缝目标交织、重叠造成的漏检概率。

3.构建轻量化的砌筑墙裂缝检测模型，检测速度满足工程实时性的需求

研究常用的轻量化网络，选择引入 Ghostnet 中的 Ghost Module 模块对骨干网络改进。使用 GhostConv 来代替标准卷积，降低卷积输出的通道数，通过对卷积得到的通道上的特征，做线性变换来扩充特征通道以此达到减少网络参数，降低计算量的目的，使得砌筑墙裂缝检测模型实现了轻量化。并通过引入拥有双向加权机制的 BIFPN 模块来代替模型原本仅有简单线性叠加策略的 PaNet 模块^[42]，有效提升了模型对不同尺寸裂缝的特征融合能力。

4.构建砌筑墙裂缝检测系统，推动技术的产业化应用

使用 HTML 编写网页端界面模板，在 PyCharm 中安装 Flask 库，通过 Flask 环境与 HTML 对研究出的轻量化的砌筑墙裂缝检测模型进行 web 端的部署。在工程环境中安装 Flask 库后，实例化 Flask 应用并定义一些 Flask 路由，决定用户在网页端的不同操作将触发不同功能。用户通过网页登录，选择需要的检测功能，并在网页上查看网络处理的结果，编写代码来确保处理结果在后端被正确保存，

以便后续分析或记录,使得没有代码与深度学习背景的土木工程技术人员也能使用深度学习技术进行检测。

1.3.2 本文的章节安排

本文的章节安排如下:

第一章: 绪论。本章首先对基于深度学习的砌筑墙裂缝检测技术的研究背景和意义进行了阐述。其次,探讨了通用目标检测和砌筑墙裂缝目标检测的研究现状,同时列举了当前基于深度学习的砌筑墙裂缝检测技术的最新研究成果,并对仍然存在的一些问题进行了分析。最后,介绍了本文的主要研究内容以及文章的章节安排。

第二章: 基本理论与数据介绍。首先,本部分深入阐述深度卷积神经网络的训练过程,以及目标检测领域常用的理论框架和算法基础。其次阐述了三种一阶段目标检测算法的原理并展示结构,同时分析了各个算法的适用场景。然后介绍了自主构建的砌筑墙裂缝数据集和训练时采用的数据增强技术,并且分析和可视化了砌筑墙裂缝数据集的数据分布。最后详细介绍了线性加权的 Soft-NMS 算法原理以及适用场景。

第三章: 基于注意力机制的砌筑墙裂缝目标检测算法。首先,概述了 YOLOX 算法的基础架构和工作原理;其次,详细描述了在 YOLOX 算法中融入注意力机制的改进措施,通过对融入注意力机制的不同位置方案进行实验分析,选择了最优方案进行改进,同时与其他注意力机制进行了实验对比,验证了改进模块的优势地位;之后将原本的 NMS 改进为 Soft-NMS 算法进行对比实验分析,选择更加有小的 Soft-NMS 适配砌筑墙裂缝呈现的相互交织、重叠的特点;接着,说明了实验所用的软件和硬件环境,以及用于评估模型性能的标准指标;最后,通过将改进后的算法在砌筑墙裂缝数据集上进行消融实验并与同类其他算法(包括基准算法)进行目标检测对比实验,验证了所提出改进的有效性,并通过实验结果与识别效果图对比,展示了改进算法在检测精度和效率上的优越性。

第四章: 轻量化的砌筑墙裂缝目标检测算法。在第三章中提出了砌筑墙裂缝检测算法,但是仍然存在算法参数庞大和计算量过高的问题,因此提出了一种解决方案。利用 GhostBottleneck 结构来改进骨干网络,这样做有效地减少了网络

参数并降低了计算量。紧接着，详细介绍了 NECK 模块的改进策略，通过使用拥有双向加权机制的 BIFPN 模块改进 YOLOX 原有颈部简单线性叠加的 PaNet 模块，以应对骨干网络轻量化后导致的精准度下降的问题，随后对整体轻量级模型结构进行了简要介绍。接着，详细阐述了实验设置和训练过程，以确保实验的严谨性和可重复性。最后，通过一系列的消融实验和对比实验，验证了研发的算法在砌筑墙裂缝检测任务中轻量化策略的有效性。

第五章：本章节详细介绍了砌筑墙裂缝目标检测系统的 Web 端部署。首先，介绍了系统 Web 端部署的设计需求，确保系统能够有效地在网络环境中运行并提供用户友好的操作界面。随后，详细描述了在目标检测过程中前端与后端的数据交互机制，以及系统的整体工作流程，包括从用户上传图像到服务器处理，再到将检测结果返回给用户的全过程。接着，本章对 Web 端的用户界面设计进行了阐述，强调了界面布局的直观性和操作的简便性，以使用户能够轻松地上传和查看检测数据。最后，通过一系列的系统运行测试，验证了该系统是否满足设计要求，并确保了其在实际应用中的易用性和可靠性。这些测试结果证明了系统部署的有效性，并展示了系统在实际使用场景中的性能表现，解决了土木工程技术人员使用深度学习模型的难题，推动了技术的产业化应用。

第六章：总结与展望。总结了提出的轻量化的砌筑墙目标检测算法，以及通过 Web 端部署实现的系统设计与实施过程，强调了这些贡献对相关领域的创新和实际应用价值。随后，分析了研究中仍然存在的不足之处，并提出了未来研究的方向。探讨了当前算法和系统可能面临的挑战，如增加检测缺陷种类、优化检测精度、扩大系统的部署范围等。同时，指出了深化研究的可能途径，包括采用更先进的深度学习模型、收集更多的数据集进行其他缺陷种类的标注、以及使用内网穿透等技术实现远程访问等，为后续的研究工作指明了方向。

第 2 章 基本理论与数据介绍

2.1 神经网络

深度神经网络（DNN）是一种由多层神经元组成的模型，其核心思想是通过层层非线性变换学习输入数据的复杂表示^[43]。该网络结构包括输入层、隐藏层和输出层，其中输入层接收原始数据，隐藏层通过学习数据的高层次特征表示将输入映射到更抽象的空间，而输出层生成最终的预测或分类结果^[44]。前向传播是信息从输入层流向输出层的过程^[45]，其中每个隐藏层通过权重和偏置的线性变换，并通过激活函数引入非线性。常见的激活函数包括 Sigmoid、ReLU 和 Tanh^[46]等。反向传播是深度学习中的优化过程，通过计算损失函数对网络参数的梯度，然后使用梯度下降等优化算法来更新参数^[47]。深度学习模型的训练通常涉及大规模数据集、迭代优化算法和模型结构调整^[48]。这一过程使模型逐渐学习输入数据的抽象特征，提高对新数据的泛化能力。

2.2 深度神经网络的训练过程

深度神经网络在深度学习领域扮演着至关重要的角色。DNN 的训练过程是确保模型能够达到高性能的关键步骤，它涉及到多个关键环节，如图 2-1 所示。首先，数据预处理是训练过程的起点。在这个阶段，原始数据被清洗、标准化和增强^[49]，以便更好地适应模型的需求。数据预处理的目的是减少噪声、处理缺失值，并对数据进行归一化，以确保模型能够更好地理解和学习数据^[50]。接下来，卷积神经网络（Convolutional Neural Networks，简称 CNN）特征提取是训练过程中的另一个关键步骤。CNN 是一种专门用于处理图像数据的神经网络结构，它通过卷积层和池化层来提取图像的高级特征^[51]。这些特征对于后续的分类和回归预测任务至关重要，因为它们能够帮助模型识别和理解图像中的关键信息。在特征提取之后，模型需要进行分类和回归预测。分类是指将输入数据分配到预定义的类别中，而回归预测则是预测连续数值的任务^[52]。这些任务通常通过全连接

层和激活函数来实现，以便将提取的特征映射到最终的输出^[53]。最后，损失函数的定义是训练过程的另一个重要环节。损失函数用于衡量模型的预测结果与真实标签之间的差异。常见的损失函数包括交叉熵损失和均方误差损失等。通过最小化损失函数，模型可以不断优化自身的参数，以提高准确性和性能^[54]。

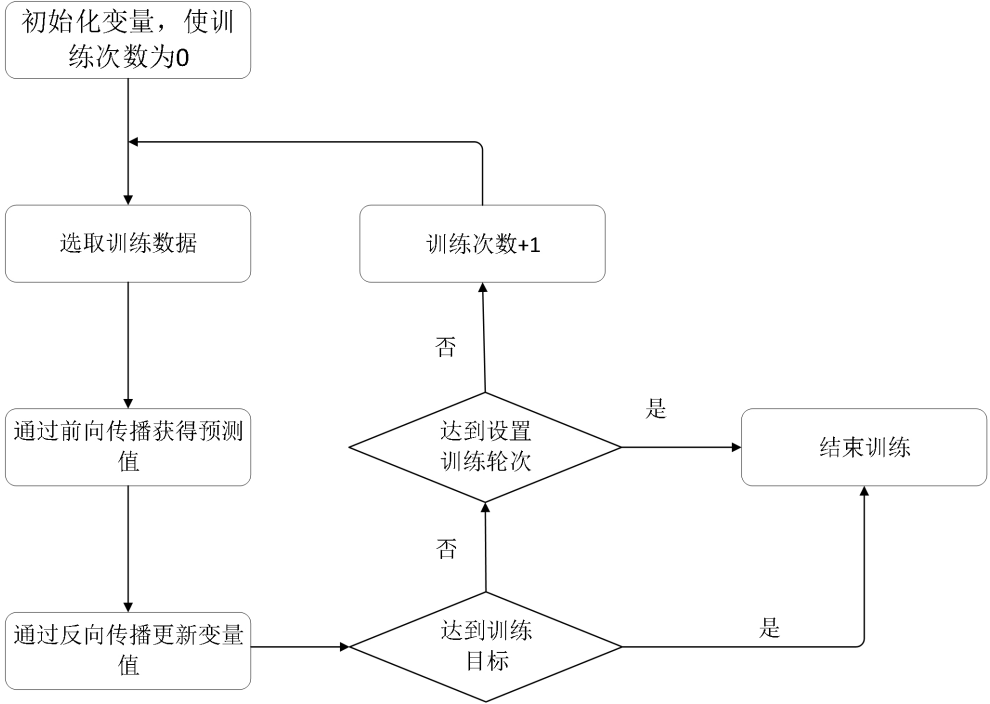


图 2-1 深度学习模型训练流程

在砌筑墙裂缝检测的背景下，改进 YOLOX 模型来实现对砌筑墙裂缝的目标检测是一个具有挑战性的任务。YOLOX 模型是一种基于深度学习的目标检测算法，它可以实时地检测和定位图像中的物体。为了提高模型的准确性，深入了解 DNN 的训练过程是至关重要的。通过研究 YOLOX 模型，可以针对砌筑墙裂缝的特点进行定制化的改进。这可能涉及调整网络结构、增加数据增强技术、优化损失函数等。通过细致的训练过程，模型可以更好地学习砌筑墙裂缝的特征，从而提高目标检测的准确性和鲁棒性。

2.2.1 数据预处理

数据预处理是深度学习中至关重要的一环，直接影响模型的性能。对于砌筑墙裂缝检测，数据预处理包括采集、清洗、标准化和数据增强。采集过程中，收集包含裂缝和少量正常墙体的图像数据，正常墙体图片作为负样本，以确保模型能够学到多样性的特征^[55]。清洗数据是为了去除噪声和异常值，确保输入数据的

质量。标准化操作则有助于将数据转换为模型易于处理的形式，提高训练的稳定性。数据增强通过旋转、翻转、缩放或者更先进的方式扩充数据集，增加模型的泛化能力^[56]。数据增强的核心在于通过对原始图像实施多种变换来产生额外的数据点，以此丰富数据集。在本研究中，训练时实施了以下几种变换和算法来进行数据增强：水平或垂直翻转：此方法通过对图像进行水平和垂直方向的翻转来增加样本的多样性；缩放：通过等比例放大或缩小图像，确保处理后的图片保持原有的长宽比；平移：在水平或垂直方向上移动图像，以确保目标对象位于图片的中心区域；HSV 调整：通过改变图像的色调（Hue）、饱和度（Saturation）和明度（Value），进一步增强数据的多样性。

如图 2-2（a）所示，采用了几何变换，而图 2-2（b）展示了 HSV 变换的效果。这些方法不仅增加了数据集的数量，也提高了其质量，有助于提升模型的泛化能力。

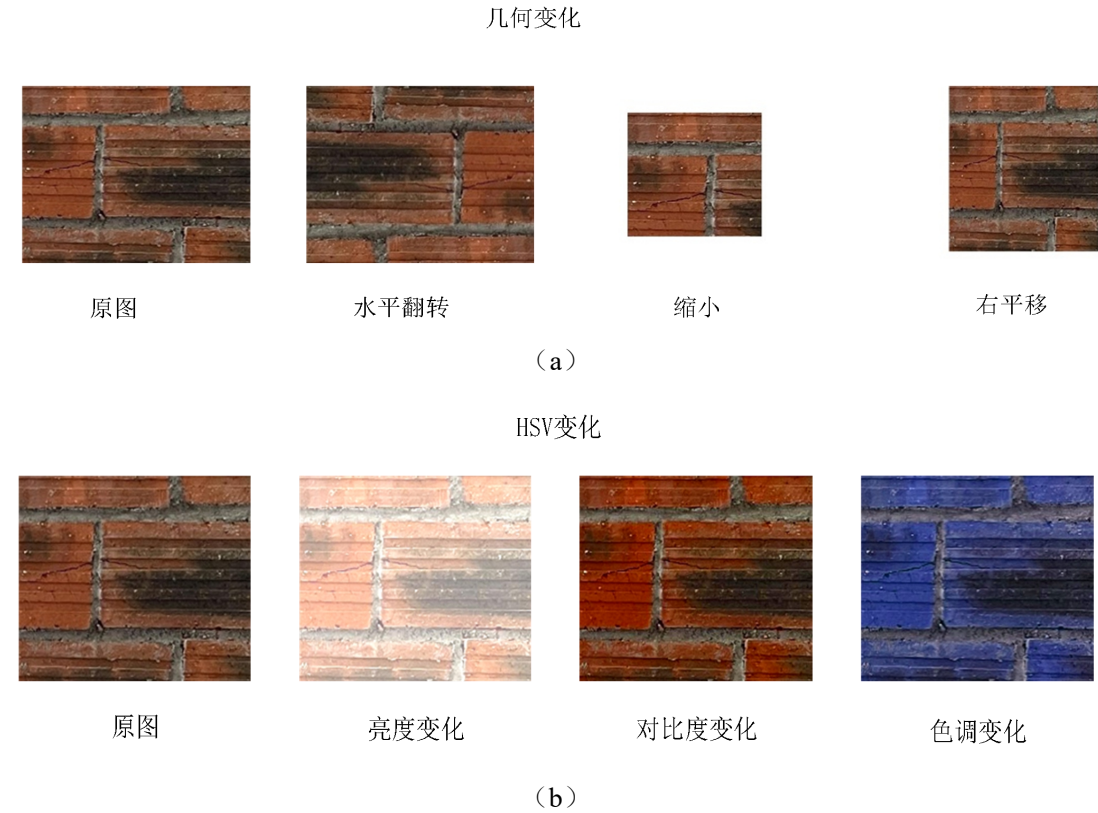


图 2-2 几何和 HSV 变换

马赛克数据增强，在马赛克数据增强方法中，对 CutMix^[57]数据增强的概念进行了拓展。使用四张不同的图片，通过随机选取基准点坐标，将图片进行尺寸调整和比例缩放后，放置在一张大图的四个象限内。这种方法的优点在于能够显

著增加数据的多样性，因为它不仅组合了不同的图像，还通过变换创建了新的背景环境。如图 2-3 中的 A、B、C、D 这四幅图像进行了翻转、缩放，四幅图裁剪后分别形成了图 E、F、G、H。最终图像 I 是通过将处理过的四幅图像组合在一起创建的，同时保留了原始四幅图像的检测框。如图 2-3 所示，Mosaic 使用 4 张不同的图像进行拼接，而 CutMix 只拼接了两张图像，增加了数据量，同时也增强了检测对象的背景。

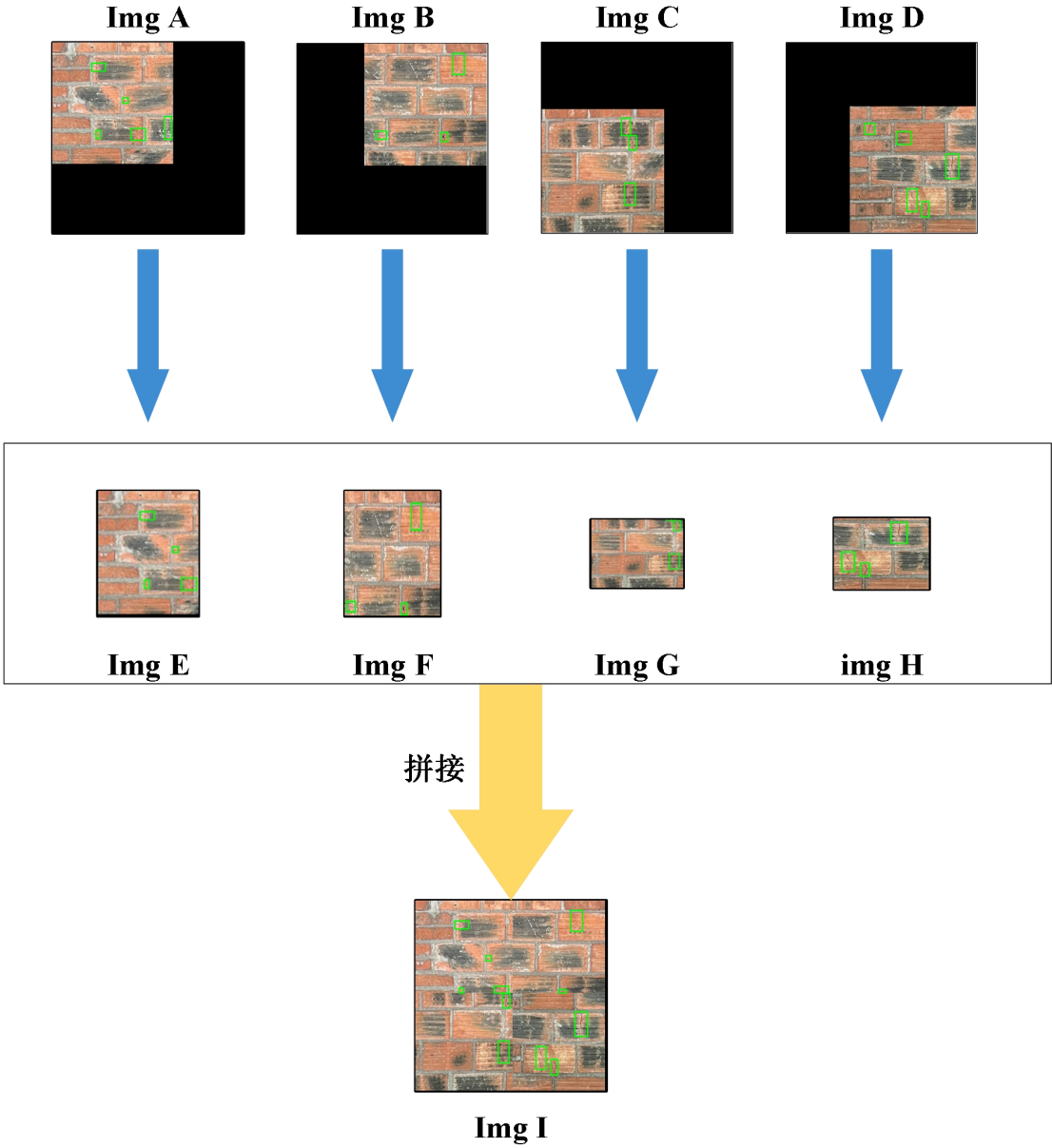


图 2-3 Mosaic 数据增强

在 Mixup 数据增强方法的训练阶段，从每批中随机选择两幅图像，并按特定比例混合生成新图像^[58]，它的核心思想是训练过程中，从每个批次(batch)中随

机选择两张图像，然后按照一定的比例将它们混合起来，生成新的图像用于模型训练。并且还进行图像间的标签混合，与图像混合对应，这两张图像的标签也进行相同比例的混合。例如，如果一张图像的标签是 $[1,0,0]$ （表示属于第一类），另一张是 $[0,1,0]$ （表示属于第二类），那么混合后的标签可能是 $[0.5,0.5,0]$ 。并且混合图像和标签的比例通常是随机选择的，但也可以固定或根据某种策略变化。Mixup 的优势在于其简单性和有效性，它能够有效地提高模型的泛化能力，减少过拟合现象。

Mixup 通过创建新的训练样本来增加数据的多样性，从而帮助模型学习到更加鲁棒的特征表示。模型的训练过程中不使用原始图像;相反，模型只使用混合的新图像进行训练。Mixup 融合前后效果图如图 2-4 所示，图 2-4 上方中的图 A、图 B 为原始图片，最下方为数据增强后的图片。

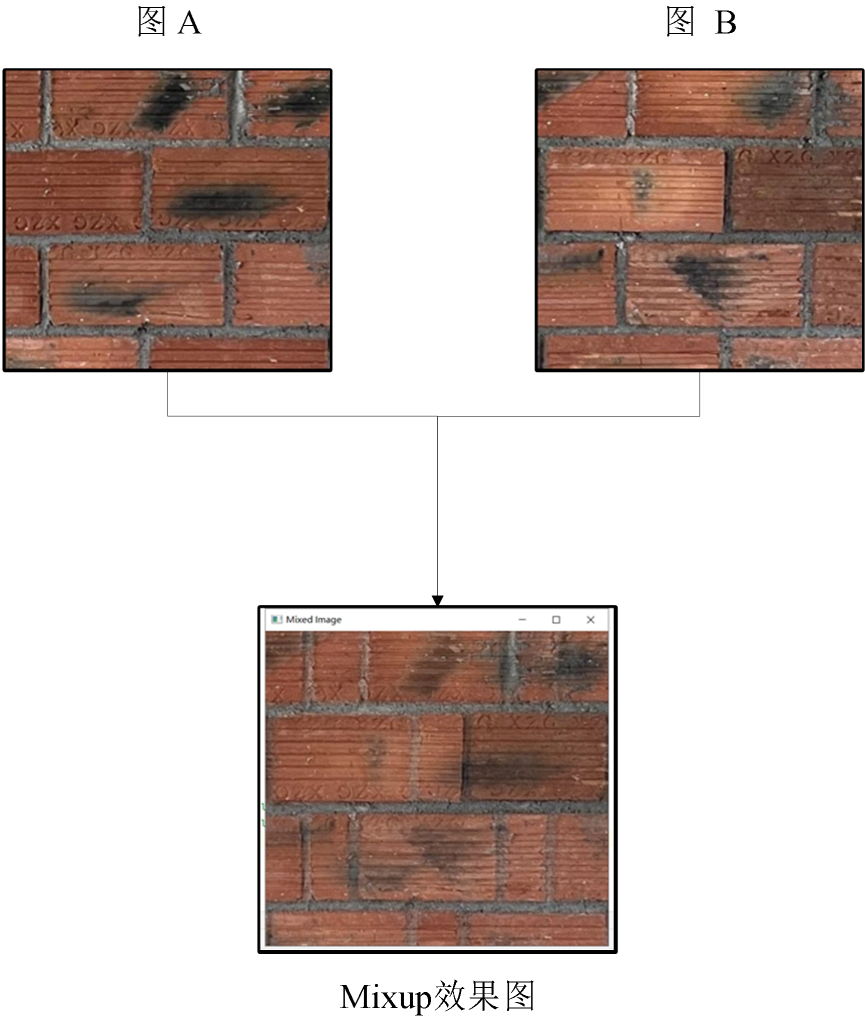


图 2-4 Mixup 融合前后效果图

2.2.2 CNN 特征提取

卷积神经网络（CNN）在深度学习中广泛应用于图像处理任务。在砌筑墙裂缝检测中，CNN 负责提取图像中的关键特征，帮助模型更好地理解和识别图像内容。通过设计适应性强的 CNN 模型，可以捕捉到墙体裂缝等重要特征。这包括卷积层、池化层和全连接层的组合，以逐渐提取和组织图像的高级特征，较为经典的 CNN 网络结构有 LeNet-5^[59]、VGGNet^[60]和 DarkNet 等。

（1）LeNet-5 是一种经典的深度学习模型，是卷积神经网络的早期代表之一，被广泛应用于图像识别和分类任务中，尤其在手写数字识别领域取得了显著的成果。第一层使用 6 个不同的 5×5 的卷积核对输入图像进行滤波。紧跟着第一个卷积层的是一个平均池化层，通过 2×2 的区域进行下采样，步长为 2。第二个卷积层和池化层与第一层类似，但使用了 16 个卷积核，每个卷积核的尺寸为 5×5 。之后再次应用平均池化。将上一层的输出展平后，输入到一个全连接层，该层包含 120 个神经元。紧接着是一个包含 84 个神经元的第二个全连接层。最后，一个全连接层作为输出层，对应于 10 个类别（数字 0 到 9），每个类对应一个神经元，使用 softmax 激活函数进行分类。图 2-5 为 LeNet-5 的结构图包括了输入尺寸和输出尺寸。

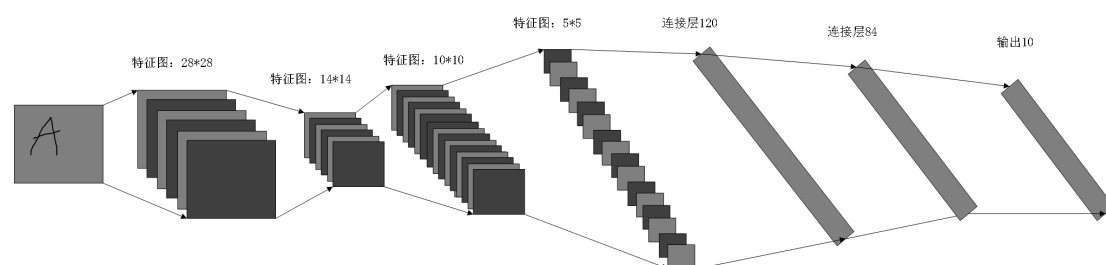


图 2-5 LeNet-5 网络结构图

（2）VGGNet，是一种深度卷积神经网络结构，其特点在于其相对简单的结构设计。该模型使用了 3×3 的小型卷积核和池化核，并且通过增加网络深度来提升模型的性能，常见的 VGG 版本有 VGG-16 和 VGG-19，分别包含 16 和 19 层。在训练阶段，VGGNet 利用多尺度的数据增强技术来提高模型的泛化能力。在测试阶段，VGGNet 将全连接层转换为卷积层，以减少参数数量并提高运算效率。VGGNet 的优势在于其强大的特征提取能力，由于其深层结构，模型能够学习到更加复杂的特征表示。然而，这也带来了参数数量巨大的缺点，导致计算资源消

耗较大，尤其是在没有高性能硬件支持的情况下。图 2-6 为 VGGNet 的结构图包括了输入尺寸和输出尺寸。

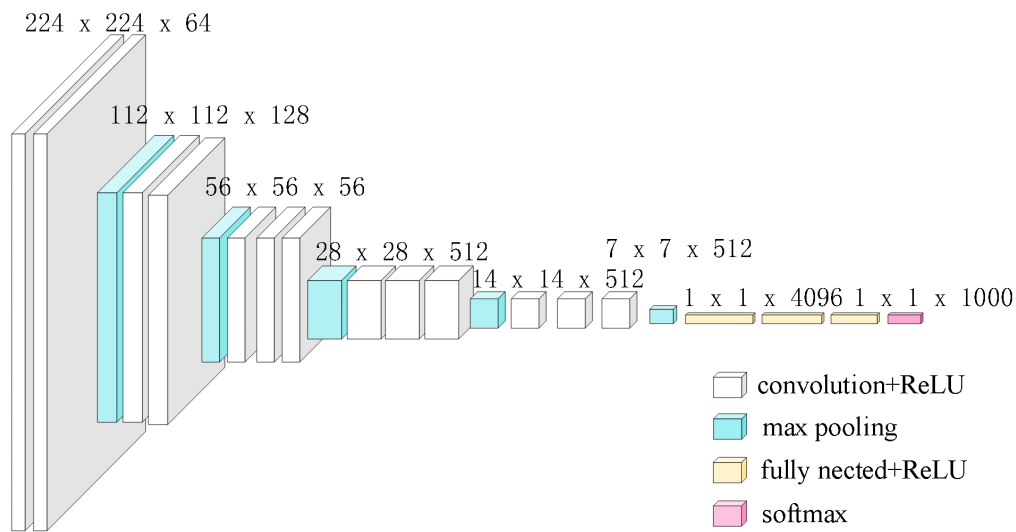


图 2-6 VGGNet 网络结构图

(3) Darknet53 是一种深度卷积神经网络，主要用于目标检测算法 YOLOv3 中的主干网络。Darknet53 是在前一代网络 Darknet19 的基础上发展而来的，它引入了残差结构，并且使用 3x3 的卷积层代替了池化层，这样的设计提高了网络的特征提取能力并加快了运行速度。Darknet53 的网络结构有规律性，基本上是重复堆叠“下采样卷积+n 个残差块”的结构组成，其中 n 代表残差块的数量。这种结构有助于网络学习到更丰富的特征表示。如图 2-7 所示包括了 Darknet53 结构以及输入和输出尺寸。

	Type	Filters	Size	Output
1×	Convolutional	32	3×3	256×256
	Convolutional	64	3×3/2	128×128
	Convolutional	32	1×1	128×128
	Convolutional	64	3×3	
	Residual			
2×	Convolutional	128	3×3/2	64×64
	Convolutional	64	1×1	64×64
	Convolutional	128	3×3	
	Residual			
8×	Convolutional	256	3×3/2	32×32
	Convolutional	128	1×1	32×32
	Convolutional	256	3×3	
	Residual			
8×	Convolutional	512	3×3/2	16×16
	Convolutional	256	1×1	16×16
	Convolutional	512	3×3	
	Residual			
4×	Convolutional	1024	3×3/2	8×8
	Convolutional	512	1×1	8×8
	Convolutional	1024	3×3	
	Residual			
	Avgpool		Global	
	Connected		1000	
	Softmax			

图 2-7 Darknet53 网络参数结构图

2.2.3 分类和回归预测以及 NMS

在训练过程中，模型需要进行两项关键任务：分类和回归预测。分类任务旨在将图像分配到预定义的类别中，涉及将墙体区域分为裂缝和正常部分，而回归任务预测连续值的输出，例如估计图片中某个物体的尺寸或位置，涉及定位和标识裂缝的位置。这两项任务的协同工作使得模型能够准确地检测和定位墙体上的裂缝，为技术人员提供准确的信息。

非极大值抑制效果图如下图 2-8 所示，非极大值抑制（NMS）是一种在目标检测领域中广泛使用的技术，它的主要目的是去除多余的边界框，从而提高检测的精度和效率^[61]。NMS 的基本原理是在所有候选框中找到得分最高的框，然后抑制与其有较大重叠的其他框。这个过程可以迭代进行，直到没有更多的框需要

被抑制。通过这种方式，NMS 能够有效地减少多个检测器预测同一对象时产生的重复边界框。

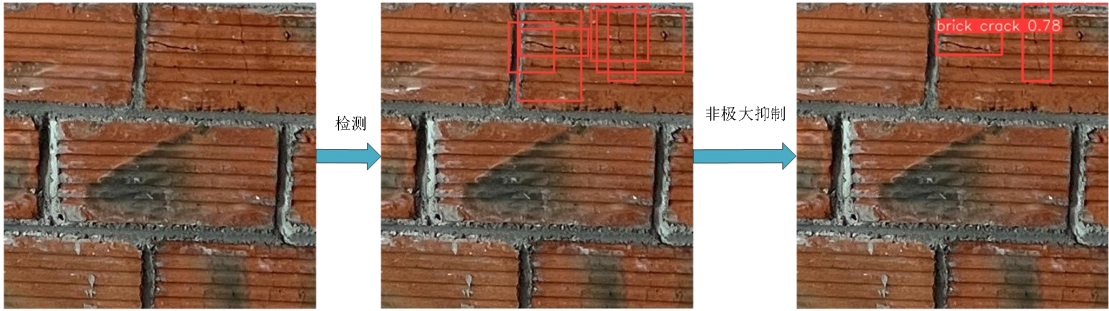


图 2-8 非极大值抑制示意图

2.2.4 损失函数

损失函数在深度学习中起到优化模型参数的关键作用。对于砌筑墙裂缝检测，损失函数需要同时考虑分类和回归的任务。常见的损失函数包括：交叉熵损失用于分类任务，以及均方误差损失用于回归任务。通过合理定义损失函数，模型可以更好地学习和调整参数，提高对裂缝的检测性能。以下是一些常用的损失函数及其特点：

L1 Loss（平均绝对误差）：它衡量预测值与真实值之间距离的平均误差幅度，对离群点或异常值具有较好的鲁棒性。不过，它在 0 点处的导数不连续，可能导致收敛速度慢。

L2 Loss（均方差）：它衡量预测值与真实值之间距离的平方和，也对离群点敏感，但相较于 L1 Loss，它在优化问题上更平滑，有助于网络学习。

DIoU Loss：这是一个针对边界框回归问题的损失函数，它使用了预测框与真实框中心点之间的归一化距离，对于非极大值抑制（NMS）过程中的指标时，可以在有遮挡的情况下提供更为鲁棒的结果。

CIoU Loss（Complete IoU）：它是 IoU 的一个改进版本，通过添加惩罚项来考虑预测框和真实框之间的重叠面积，以及预测框的宽高比和中心点距离。

交叉熵损失（Cross Entropy Loss）：用于衡量实际输出与期望输出的概率分布的接近程度。当实际输出与期望输出相同时，交叉熵取得最小值。

Focal Loss：它是针对类别不平衡问题提出的一种损失函数，通过增加难以分类样本的权重，减少易分类样本的权重，从而提高模型对少数类对象的识别性能。

损失函数的选择取决于具体的任务和数据特性。在实际应用中，可能需要根据模型的性能和数据的特点来选择合适的损失函数。此外，损失函数通常与优化算法一起使用，以最小化损失函数的值，从而优化模型参数。在训练过程中，损失函数的值通常会随着模型参数的更新而变化，反映了模型的学习进度。

2.3 目标检测常用框架介绍

目标检测是计算机视觉领域中的关键任务，涉及在图像或视频中定位和识别多个目标。许多先进的深度学习框架被广泛用于解决目标检测问题，它们通过强大的网络结构和训练技术推动了该领域的发展。

2.3.1 YOLO

YOLO 是一种实时目标检测算法，其核心思想是将目标检测任务转化为一个回归问题，YOLO 网络结构图如图 2-9 所示，展示了完整的输入和输出过程。相比于两阶段目标检测算法，YOLO 采用单一神经网络直接在整个图像上进行推断。它将图像划分为网格，每个网格负责预测图像中的 bounding box 和相应的类别概率。这样的设计使得 YOLO 在速度和准确性之间取得了很好的平衡发展历程。

YOLOv1:YOLO 的第一个版本，提出将目标检测任务转为回归问题，实现了实时性的目标检测。

YOLOv2(YOLO9000):引入更多先进的网络结构和技术，支持多类别目标检测，提升了检测性能。

YOLOv3:进一步改进网络结构，引入了多尺度特征融合，提高了对小目标的检测效果。

YOLOv4:引入更多创新设计，如 CSPDarknet53、SAM 模块等，提高了检测精度和速度。

YOLO 系列以其实时的目标检测能力而著称，适用于需要快速响应的场景，如自动驾驶、实时监控等。通过使用多尺度的特征图，使其对于不同尺寸的目标都能够进行有效的检测。YOLO 被广泛应用于图像识别、视频分析、工业质检等领域。

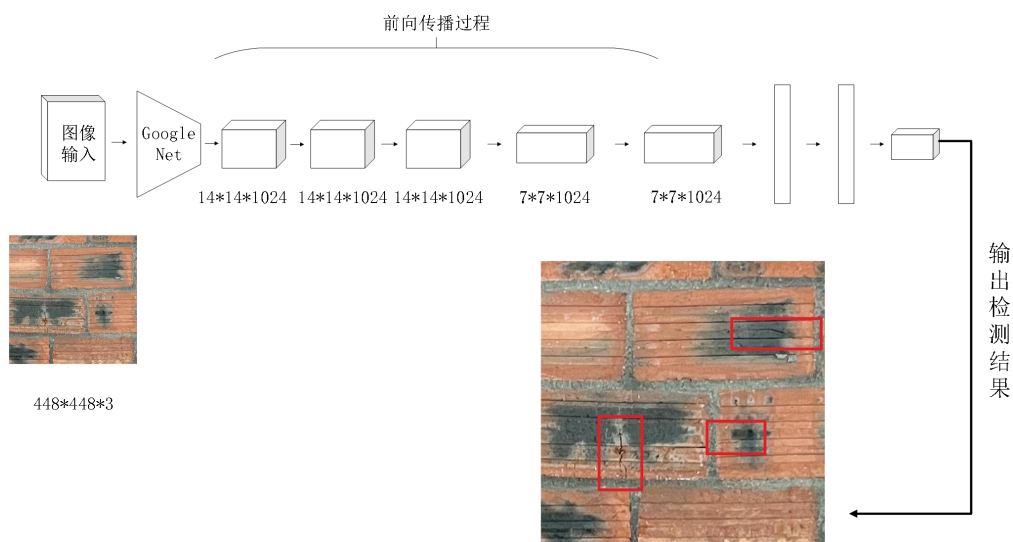


图 2-9 YOLO 网络结构模型

2.3.2 SSD

SSD 是一种采用多尺度特征图的目标检测算法。与 YOLO 不同，SSD 通过在不同层次的特征图上进行预测如图 2-10 所示，实现对不同尺寸目标的有效检测。SSD 引入了先验框（prior box）的概念，通过在不同尺度上设置多个先验框，使得模型能够更好地适应目标的多样性。

最初的 SSD 版本，通过引入多尺度的特征图来实现对不同尺寸目标的检测。SSD300 和 SSD512:不同版本的 SSD，分别采用不同的输入图像尺寸，以平衡速度和检测性能。MobileNet-SSD 将 MobileNet 作为骨干网络，实现了在移动设备上的目标检测。SSD 通过在多个尺度上进行检测，使其能够有效地应对目标的不同尺寸。引入先验框提高了对不同形状目标的适应性，使得 SSD 在复杂场景中表现出色，在静态图像检测以及移动端实时检测等场景中都有着广泛的应用。

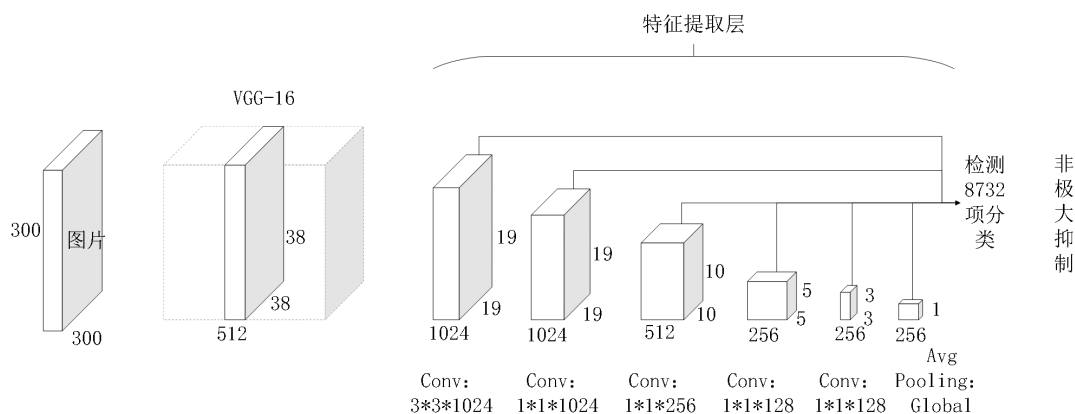


图 2-10 SSD 网络结构模型

2.3.3 Faster R-CNN

Faster R-CNN 是一种经典的两阶段目标检测算法，如图 2-11 展示了完整的 Faster R-CNN 结构图，其关键创新是引入了区域建议网络（RPN）。RPN 负责生成可能包含目标的候选区域，然后这些区域被用于目标的最终检测。这种两阶段的设计使得 Faster R-CNN 在检测精度上有着显著的提升。

最初的版本，引入 RPN 以生成候选区域，然后通过 ROI Pooling 进行目标检测。R-FCN（Region-based Fully Convolutional Networks):使用全卷积网络实现了更高效的目标检测。Faster R-CNN with ResNet:结合 ResNet 等深度网络结构，提高了检测性能。Faster R-CNN 的两阶段设计使其在检测准确性上有着显著优势，适用于要求高精度的应用场景。在复杂场景中，Faster R-CNN 能够更好地处理目标的遮挡和重叠，常被应用于需要高精度目标检测的领域，如医学影像分析和安防监控。

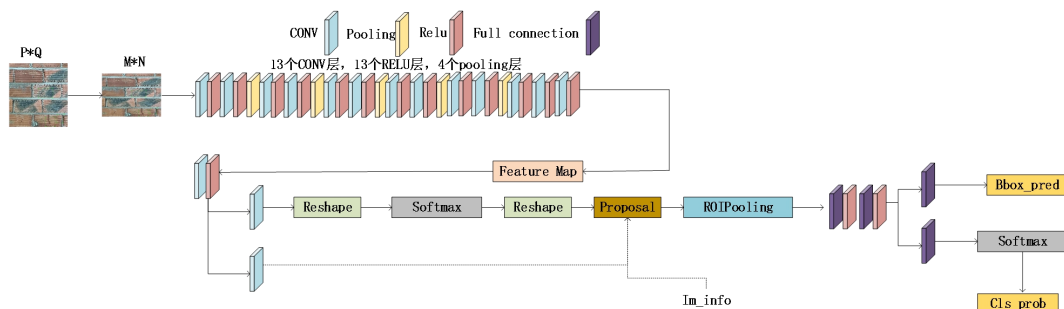


图 2-11 Faster R-CNN 网络结构模型

2.4 砌筑墙裂缝检测目标数据集

由于目前并没有公开基于工程现场采集的砌筑墙裂缝数据集，导致训练出来的模型对工程没有实际意义。数据集的质量直接影响到算法网络训练的成果，包括其准确度和鲁棒性，因此，构建一个高质量的数据集显得尤为关键。

通过与知名企业合作，将采集到的基于施工现场的图片经过处理，分割成 640*640 像素的图像。这些图像将用于模型的训练和验证，以确保其在实际工程中的有效性和适用性。通过与专业技术人员进行沟通，再次筛选了砌筑墙图片，将符合国家规范但带有裂缝的砌筑墙图片剔除，从而使最终的图片数据集中只包

含不符合国家规范的带裂缝砌筑墙图片^[62]。此外，特意添加了一些完好的砌筑墙图片到数据集中，用作负样本，以帮助降低模型的误检率^[63]。图 2-12 为部分数据集原图，展示了部分不同光照环境下采集的数据集图片。

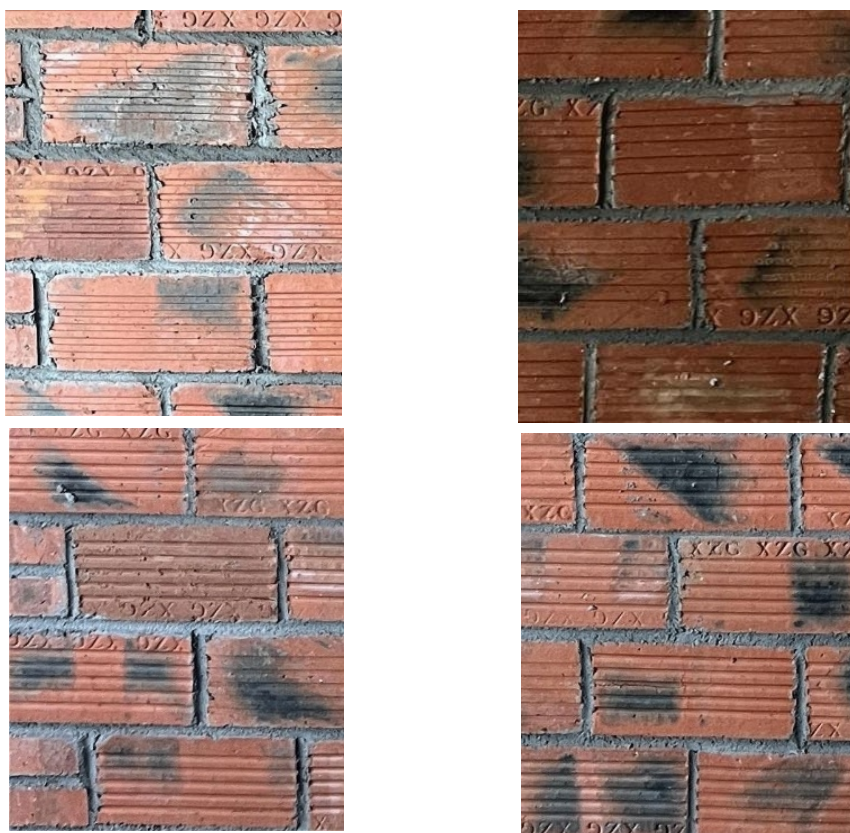
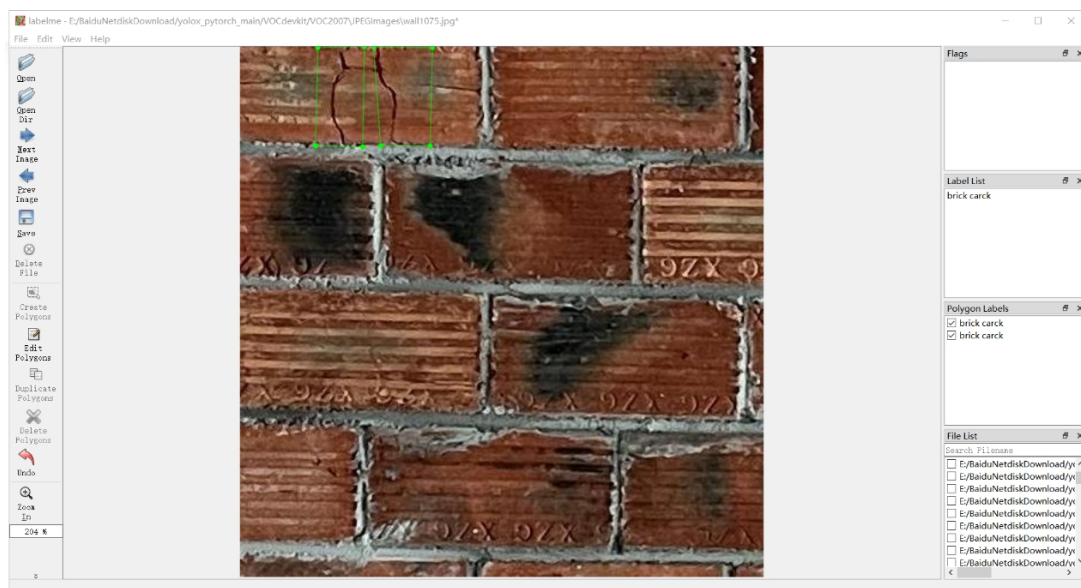


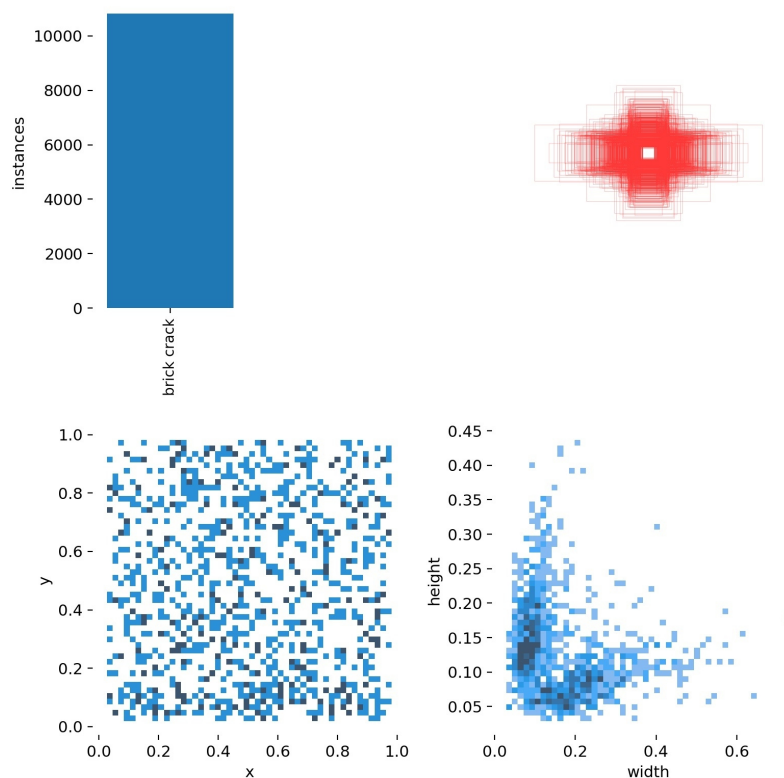
图 2-12 部分数据集原图

最后，使用 Labelimg 工具对采集到的图片中不符合规范的裂缝进行了标注如 2-13 (a)。鉴于裂缝检测不需要区分裂缝产生的原因，因此只定义一个类别，即“裂缝”，所以在本文中平均精确度 (AP) 即为均值平均精确度 (mAP)。制作完成后的 VOC 数据集包括了 2589 张图片以及相应的标签，图片尺寸为 640*640 像素。实验前通过转文件格式代码将 VOC 数据集标签文件转换为适用于 YOLO 网络的 txt 标签文件，然后划分数据集，将数据集按 9:1 划分，则从 2589 幅图片中随机抽取 2230 张作为训练集 259 张作为测试集。图 2-13 (b) 为数据集的可视化结果，其中左上的图为数据集目标种类的统计结果，左下为中心点横纵坐标以及框的高宽间的关系图，右上为 Labelimg 对数据集标注的所有框的尺寸分布图，右下为图中目标相对于整幅图的高宽比例图。图 2-13 (c) 为图 2-13 (b) 中左下图的解析，每一行的最后一幅图代表的是 x, y, 宽和高的分布情况：最上面的第一行图表明中心点横坐标 x 的分布情况，可以看到中心点位置分布比较

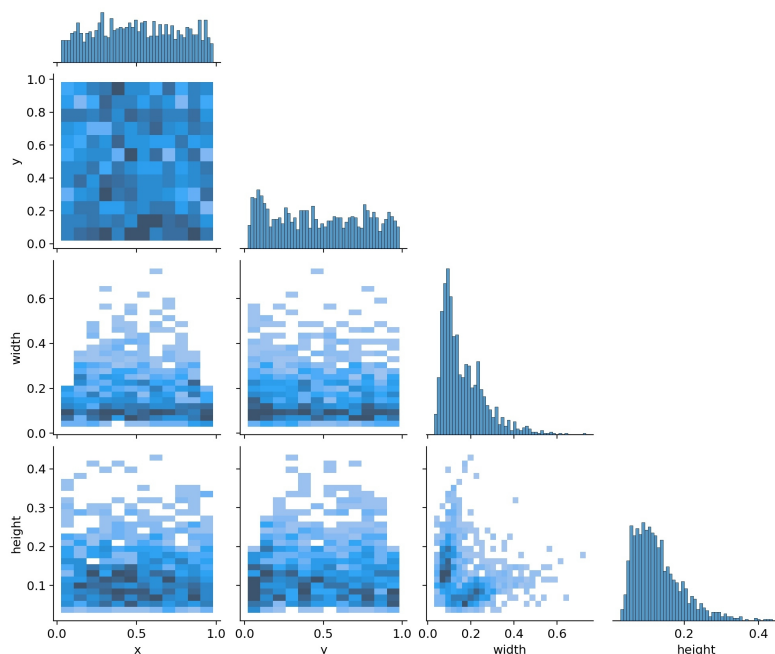
均匀；第二行最后面的图表明中心点纵坐标 y 的分布情况，可以看到中心点位置分布比较均匀；第三行最后面的图表明框的宽的分布情况，可以看到大部分框的宽的大小都小于整幅图的宽的一半；第四行最后面的图表明框的宽的分布情况，可以看到大部分框的高的大小都小于整幅图的高的一半。



(a) 数据集标注图



(b) 数据集可视化



(c) 标注框信息分布图

图 2-13 砌筑墙裂缝数据集可视化图

2.5 线性加权的 Soft-NMS

YOLOX 采用了非极大值抑制算法，非极大值抑制（Non-Maximum Suppression，简称 NMS）是一种用于目标检测和图像处理中的方法，主要用于去除重叠的边界框（bounding boxes）或者峰值。在目标检测任务中，经过一系列操作后，可能会得到多个边界框，这些边界框可能部分或完全重叠。非极大值抑制的目的是保留那些具有最高得分的边界框，同时去除其他与其重叠的边界框。这样可以提高目标检测的准确性和效率。

但是 NMS 存在以下缺陷，当物体重叠时，如图 2-14 所示，会有一个最高分数的框，如果使用 NMS 的话就会只考虑当前选择的区域，忽略了全局最优解的可能性。这意味着它可能会错误地抑制掉一些实际上是正确检测的边界框。

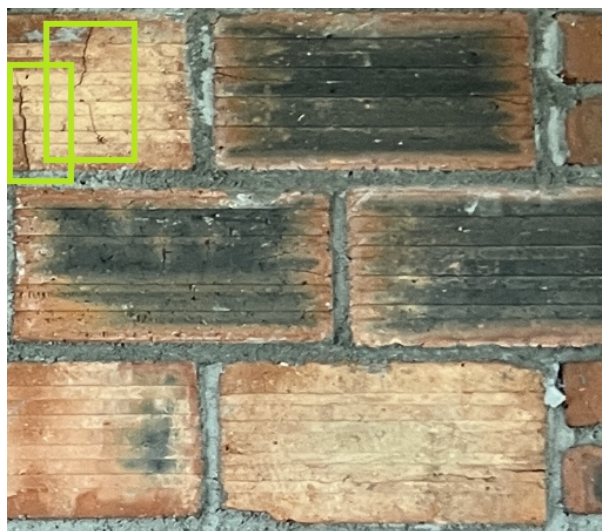


图 2-14 检测目标重叠示意图

物体重叠导致输出多个框中存在某些其实是另一个物体预测框，这种情况另一个物体的预测框容易被 NMS 抑制。为了更精确的删掉预测框，而不是把所有和最高分框删掉，于是采取对于与最高分框重叠且大于阈值的预测框，不是直接删除，而是将其置信度降低的方法，这样便尽可能多的保留预测框，从而一定程度上避免预测框被错误抑制的情况出现。

对于原来周围的多个表示同一个物体的框 Soft-NMS 便发挥作用，同一个物体周围的框有很多，每次选择分数最高的框，抑制其周围的框，与分数最高的框的 IoU 越大，抑制的程度越大，一般来说，表示同一个物体的框（比如都是前面的裂缝）的 IoU 是会比另一个物体的框（比如后面的裂缝）的 IoU 大，因此，这样就会将其他物体的框保留下来，而同一个物体的框被去掉，线性加权的 Soft-NMS 函数公式见公式 2-1。

$$s_i = \begin{cases} s_i, & \text{iou}(M, b_i) < N_t \\ s_i(1 - \text{iou}(M, b_i)), & \text{iou}(M, b_i) \geq N_t \end{cases} \quad (2-1)$$

Soft-NMS 可以提高模型的准确度，特别适合砌筑墙裂缝中有很多裂缝重叠存在的情况，因为它能够更好地处理重叠边界框的情况，减少错误检测的可能性，可以在一定程度上提高模型对目标的定位精度，Soft-NMS 的完整算法如图 2-15 所示。


```

Input :  $B = \{b_1, \dots, b_N\}$ ,  $S = \{s_1, \dots, s_N\}$ ,  $N_t$ 
       $B$  is the list of initial detection boxes
       $S$  contains corresponding detection scores
       $N_t$  is the NMS threshold

begin
     $D \leftarrow \{\}$ 
    while  $B \neq \text{empty}$  do
         $m \leftarrow \text{argmax } S$ 
         $M \leftarrow b_m$   $D \leftarrow D \cup M$  ;  $B \leftarrow B - M$ 
        for  $b_i$  in  $B$  do
             $s_i \leftarrow s_i f(\text{iou}(M, b_i))$ 
        end
    end
    return  $D, S$  end
end

```

图 2-15 Soft-NMS 的完整算法

2.6 本章小结

在本章节的开头，概述了深度学习的基本概念，继而深入讨论了深度神经网络训练的关键步骤，涵盖了数据预处理和特征提取等环节，并对每个步骤进行了详尽的分析与介绍，以便更深入地理解其功能和作用。随后，本章对比了三种主流的目标检测框架，探讨了它们的优势、局限性以及适用的场景。接着，描述了本研究构建数据集的方法以及可视化了数据集的数据分布。最后详细介绍了 NMS 算法，分析了 Soft-NMS 算法对比传统 NMS 具有的优势。为接下来的砌筑墙裂缝检测算法研究奠定了理论与实验的基础。

第 3 章 基于注意力机制的砌筑墙裂缝目标检测算法

分析砌筑墙裂缝特点得出裂缝仅出现在红砖表面，应使算法更加关注红砖区域，并且裂缝还拥有相互交织、重叠等特点，导致检测难度大于普通的目标检测。为此本章将提出一种使用线性加权 Soft-NMS 算法和嵌入卷积块注意力机制的砌筑墙裂缝检测算法，来代替传统的人工观测，专门应对砌筑墙裂缝检测任务。

3.1 基准算法简介

由旷世研究院提出的深度神经网络模型 YOLOX 与早期的 YOLO 模型相比，YOLO-X 模型体积更小、速度更快、精度更高，因此在业界广受欢迎。YOLOX 在多个方面相较之前的 YOLOv3 有了显著的改进和优点，主要体现在以下几个方面：YOLOX 采用了 Focus 结构对输入图像进行切片，这种结构能够在图像下采样的过程中保留更完整的信息；在主干网络中引入了 CSP (Cross Stage Partial) 结构，这种结构分别对残差分量和卷积层数进行了调整，以优化网络的性能和计算效率；对颈部的卷积层数进行了调整，使其数量与残差分量相匹配，这样的设计有助于提高特征提取的效率和准确性；在 Neck 部分，YOLOX 输出小、中、大三种尺度的特征图，这有助于检测不同大小的目标物体；YOLOX 通过解耦检测头，极大地改善了模型的收敛速度，并且相比非解耦的端到端方式，解耦能够带来更高的平均精度 (AP) 提升。如图 3-1 所示，包括了 YOLOX 的完整结构以及输入输出尺寸变化。

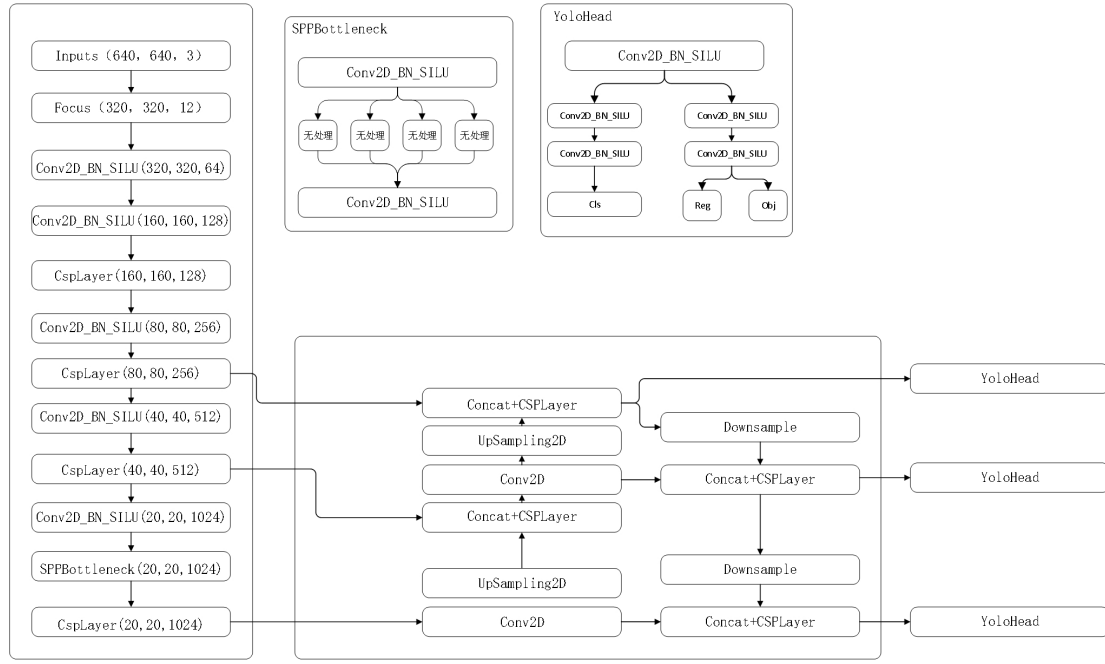


图 3-1 YOLOX 结构图

3.2 卷积注意力模块简介

空间注意机制和通道注意机制共同构成了通道空间注意力机制。通道注意力先于空间注意力，初始输入特征首先进入通道注意力机制。初始输入特征被全局平均池化(GAP)和全局最大化池化(GMP)。经过这种方法之后产生了两种不同的空间上下文信息，然后再将该特征送入到一个共享的多层感知机(MLP)网络中产生最终的通道注意力特征图。综上，通道注意机制总结为下面的等式 3-1。

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(W_1(W_0(F_{avg}^c)) + W_1(W_0(F_{max}^c))) \end{aligned} \quad (3-1)$$

为了获得空间注意力参数，将通道侧的输出结果通过最大池化和平均池化得到两个特征图，然后经过拼接操作对两个特征图进行拼接，通过 7×7 卷积变为 1 通道的特征图，再经过一个 sigmoid 得到空间注意力机制的特征图，最后将输出上采样变为原图大小。空间注意力机制可以总结为下面的等式 3-2。

$$\begin{aligned} M_s(F) &= \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7 \times 7}([F_{avg}^s; F_{max}^s])) \end{aligned} \quad (3-2)$$

该模块使空间和通道两个维度上进行注意力加权计算，增强了特征的通道和空间相关性，使得从目标数据中提取有用的特征变得更加容易。基本结构如图

3-2 所示。图 3-2 中金色立方体部分为通道注意力模块，蓝色立方体部分为空间注意力模块。

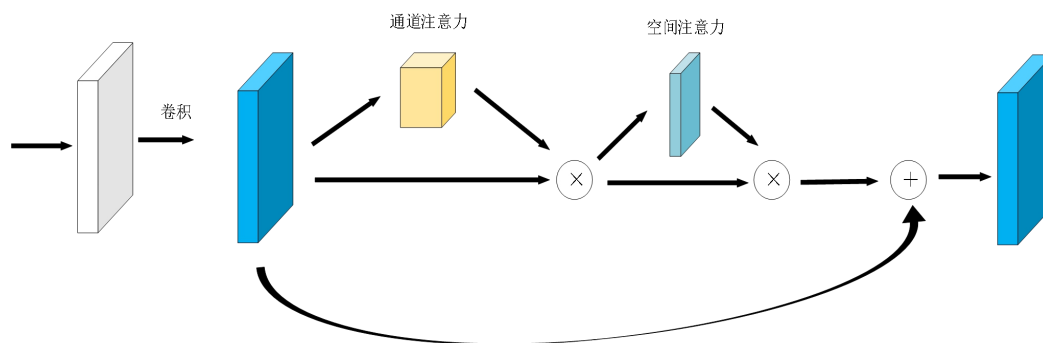


图 3-2 通道空间注意力机制结构

3.3 算法改进介绍

研究 YOLOX 模型发现，虽然 YOLOX 的骨干网络中的空间金字塔池化层（SPP）能够解决输入图像特征尺寸不一致的问题，但特征映射在通道和空间之间没有加权融合，同时残差结构会导致冗余信息，因为残差结构所发送的信息一定存在偏差，降低了学习效率和识别精度^[64]。为此，本文新建立的砌筑墙裂缝检测模型以 YOLOX 为基准，通过对通道方向产生权值来学习不同通道之间的权重分配，强调对分类或检测任务更有用的通道，随后对特征图获得空间方向权值，在空间层对特征图添加权值，关注于特征图的空间信息，即特征图内部不同位置的重要性，以便模型能够聚焦于图像中的关键区域，突出有用的特征，使砌筑墙裂缝检测模型更加关注红砖区域，从而提高识别的精度。

并且经过深入研究砌筑墙裂缝缺陷的特点发现，这些待检测裂缝经常重叠、交织在一起，并非孤立存在，而是往往呈现出一种复杂且相互重叠的模式。在这样的背景下，传统的非极大值抑制策略在处理重叠区域时，往往会简单地选择其中一个较强的检测框，而忽略或剔除其他可能同样重要的检测框。这种方法在面对单一、独立的目标时效果尚可，但在处理高度重叠和交织的裂缝时，其表现较差。为了更有效地处理这种复杂的重叠情况，采用线性加权的 Soft-NMS。这种策略通过为每个检测框分配一个权重，而不是简单地选择最强的检测框，从而更加精细地处理重叠区域。权重的计算基于检测框的置信度和其他相关参数，使得即使在多个裂缝重叠的情况下，也能更准确地识别和保留关键的检测框，能在一

定程度上提高整体的检测精度和鲁棒性。

3.4 实验结果和分析

3.4.1 评价指标

评价指标用 Precision(P)为精确率，表示在所有被检测为正样本的结果中，真正为正样本的比例。P 值越高，表示模型对正样本的预测准确性越高。Recall(R)表示召回率，表示在所有正样本中，被正确检测到的比例。R 值越高，表示模型对正样本的检测能力越强。Average Precision(AP)表示平均精度，它表示在预测结果中，对于每个类别的平均精度。AP 值越高，表示模型在该类别上的检测效果越好。Mean Average Precision(mAP)表示全类平均精度，是多个类别 AP 值的平均值，mAP 值可以全面地评估模型在各个类别上的性能。其计算公式如下：

$$P = \frac{TP}{TP + FP} \quad (3-3)$$

$$R = \frac{TP}{TP + FN} \quad (3-4)$$

$$AP = \int_0^1 P(R) dR \quad (3-5)$$

$$mAP = \frac{1}{C} \sum_{k=1}^C (AP)_k \quad (3-6)$$

真阳性(TP)是模型预测为正类，实际标签也为正类的数量。假阳性(FP)，即模型预测为正类，但实际标签为负类的数量。漏检阳性样本数(FN)，是指即模型预测为负类，但实际标签为正类的数量。

3.4.2 实验环境

模型训练环境为 Windows 10 64 位操作系统, NVIDIA GTX1080ti, Python3.8 和 Pytorch1.10.0 深度学习框架, Pycharm 版本 2021.3.3,CPU 型号为 i7 10700,OpenCV 库 4.6.0.66,CUDA11.4。

在砌筑墙裂缝检测模型的整个训练过程中, Batchsize 设为 8, 初始学习率设为 0.001, 每一轮 epoch 都进行 YOLOX 模型自带的数据增强算法, 每 20 个 epoch 保存一次权重, 后续实验使用其中最优权值。训练 1700 个 epoch 收敛后的损失

函数变化情况如图 3-3 所示，包括了验证集和训练集的 loss 曲线。

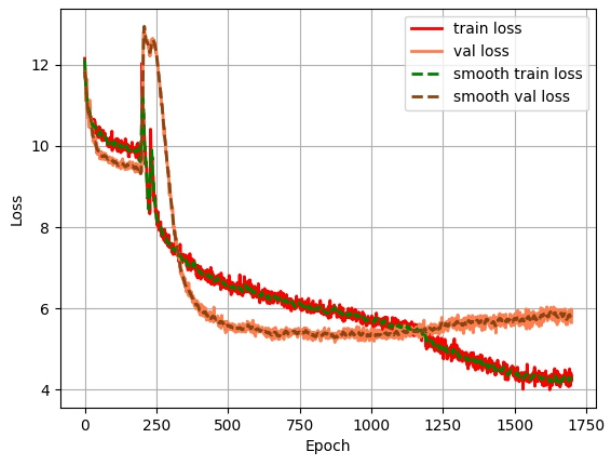


图 3-3 砌筑墙裂缝检测模型损失函数变化曲线图

3.4.3 NMS 改进效果分析

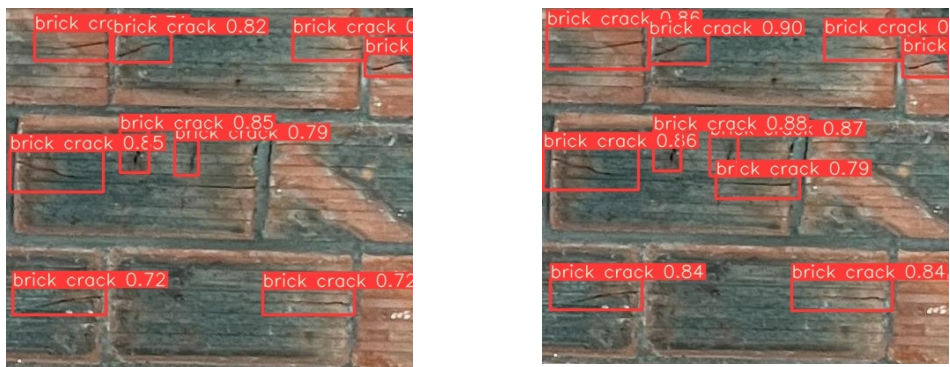
通过对砌筑墙裂缝特点的深入研究发现，砌筑墙裂缝存在大量相互交织、重叠的情况。采用传统 NMS 算法会抑制掉一些实际上应该被保留的目标，导致某些目标被漏检。Soft-NMS 通过降低重叠区域的置信度来避免直接丢弃框，这样可以保留更多的有效检测信息，考虑了所有可能的检测框，而不是简单地抑制掉一部分。本文采用 Soft-NMS 来适配砌筑墙裂缝的特点，改善模型的性能。由表 3-1 所示，通过与 NMS 的对比实验发现，使用 Soft-NMS 之后模型在砌筑墙裂缝检测任务上的准确度获得了可观的提升。

表 3-1 NMS 策略对比实验

策略名称	mAP 值	mAP 提升值
YOLOX-NMS	71.55%	/
YOLOX-Soft-NMS	72.76%	+1.21%

图 3-4 为改进 NMS 前后的裂缝识别对比，通过对比改进前后的同一张图像识别效果，发现使用传统 NMS 的模型漏检的裂缝成功被使用 Soft-NMS 后的模型检测到，图 3-4(a)为使用 NMS 的模型识别效果图，图 3-4(b)为使用 Soft-NMS 的模型识别效果图。可以看出，改进后的模型对相互重叠、交织的裂缝具有较好的检测效果，这是由于 Soft-NMS 算法采用线性加权的方式，可以更加灵活地处理重叠的边界框。它允许部分重叠的边界框也获得一定的分数，而不是像传统

NMS 那样直接丢弃。

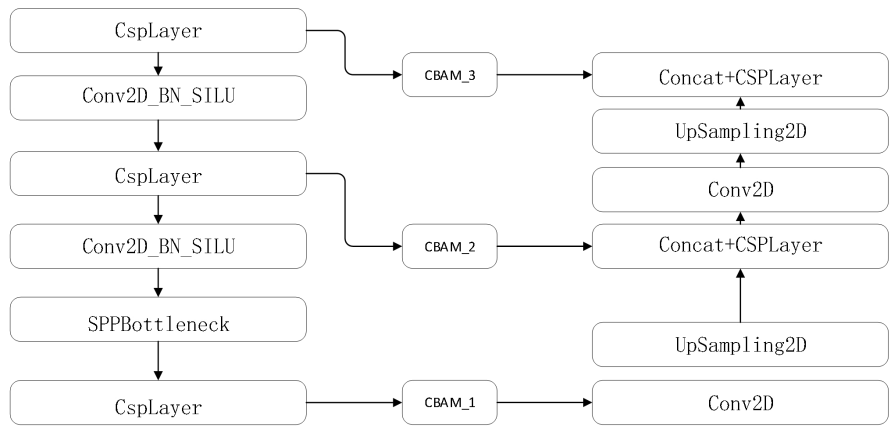


(a) 使用 NMS 的模型识别效果图 (b) 使用 Soft-NMS 的模型识别效果图

图 3-4 效果对比图

3.4.4 基于注意力机制的改进实验结果分析

针对砌筑墙裂缝只出现在红砖表面的特点，模型应当关注红砖区域，而注意力机制可以很好地加强模型对关键区域的关注度，本文采用卷积注意力机制对通用模型改进。CBAM 模块可以嵌入 YOLOX 骨干网络和颈部之间（方案一）如图 3-5（a）所示，也可以嵌入颈部与 Head 部分之间（方案二）如图 3-5（b）所示，由于注意力机制嵌入的位置也会影响到模型的准确率，所以加入不嵌入 CBAM 的模型作为方案三，通过对比实验分析两种策略的有效性。



(a) 方案一

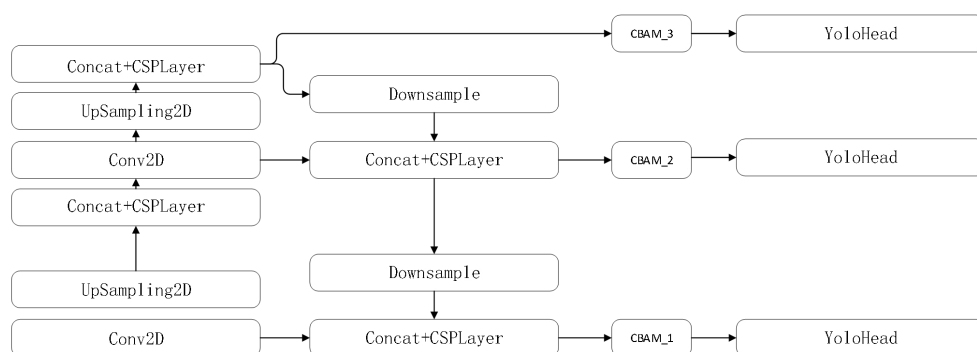


图 3-5 CBAM 模块嵌入位置方案展示图

表 3-2 嵌入注意力机制的不同方案对比实验

为了验证本文注意力机制的优势，横向选取了同类注意力机制，在砌筑墙裂缝检测任务上以 YOLOX 为基准模型进行改进，进行对比实验。选取 EAM、SE、ECA 注意力机制，通过嵌入与 CBAM 同样的位置进行对比。如表 3-3 所示，无论是检测速度和检测精度，CBAM 注意力机制都较其他注意力机制拥有更好的表现，验证了 CBAM 注意力机制相较于其他同类注意力机制在砌筑墙裂缝检测任务中更具有优势。

注意力机制改进策略	mAP	FPS f/s
CBAM	73.59%	33.78
SE	71.22%	34.36
ECA	68.53%	31.92
EAM	73.02%	33.25

3.4.5 消融实验结果分析

XY 轴和 P-R 曲线围成的区域为识别精度值，召回率(R)和精度(P)分别是横坐标和纵坐标。图 3-6 (a) 为 YOLOX 通用模型的 P-R 曲线图，图 3-6 (b) 为本文提出砌筑墙裂缝检测模型的 P-R 曲线图，可以看出通过对通用模型进行针对砌筑墙裂缝数据集特点的改进后，改进后的模型精度高于通用模型 3.55%。

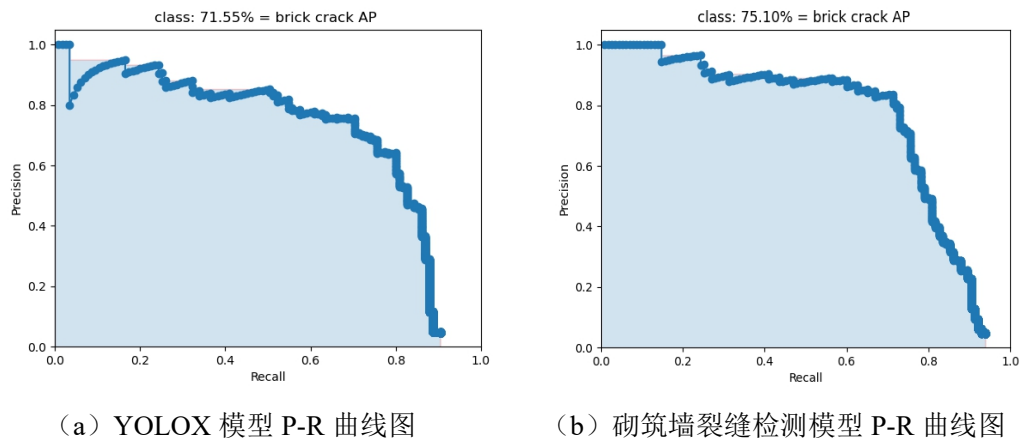


图 3-6 P-R 曲线图对比

表 3-4 展示了消融实验的数据，可以看出两个针对砌筑墙裂缝数据集的改进策略对于模型精度的提升都有着不同程度的增加，可以证明针对砌筑墙裂缝检测任务的改进策略对模型的精度提升是递进且有效的。

表 3-4 消融实验

改进策略	CBAM	线性加权 Soft-NMS	mAP	参数量	FPS f/s
1			71.55%	54.2M	34.26
2	✓		73.59%	54.8M	33.59
3		✓	72.76%	54.2M	33.89
4	✓	✓	75.10%	54.8M	32.41

3.4.6 模型性能评估

YOLOX 骨干部分有多个残差网络，CSP 结构将初始残差块堆栈分成两部分。有效地缓解了与深度增加相关的梯度消失问题，并允许增加深度以提高精度，但它也将噪声传递到更深的网络，同时砌筑墙裂缝仅存在于红砖表面，模型对红砖区域的关注度还有待提高。CBAM 模块的加入使网络能够更多地集中在关键特征上，抑制非必要特征，增强了模型在对红砖区域的关注度，并减轻噪声对网络

传输的影响。同时，线性加权的 Soft-NMS 可以更好地处理重叠的目标识别任务，减少漏检的情况。表 3-5 将本研究使用的砌筑墙裂缝检测模型与 YOLOX、YOLOV5 和 YOLOV3 在砌筑墙裂缝检测任务上的准确率进行了比较。由表 3-5 可知，砌体墙体裂缝检测模型的 mAP 值分别较 YOLOX、YOLOV5 和 YOLOV3 在砌筑墙裂缝数据集检测任务提高了 3.55、4.34 和 5.29 个百分点。

表 3-5 准确度对比实验

模型版本	准确度和差值	
	准确度	和改进后模型的准确度差值
YOLOv3	69.81	-5.29
YOLOv5	70.76	-4.34
YOLOX	71.55	-3.55
砌筑墙裂缝检测模型	75.10	/

相同的数据集和测试环境下，该表比较了各种算法的精度，并计算了砌筑墙裂缝检测模型与各模型相比在砌筑墙裂缝数据集上的精确度提升。图 3-7 为不同模型检测效果对比图，从左至右依次为 YOLOV3、YOLOV5、YOLOX、砌筑墙裂缝检测模型，从图中可以看出只有砌筑墙裂缝检测模型可以做到没有漏检，并且检测精度也领先于其他模型，由此验证了砌筑墙裂缝监测模型在砌筑墙裂缝检测任务较其他模型的优势地位。

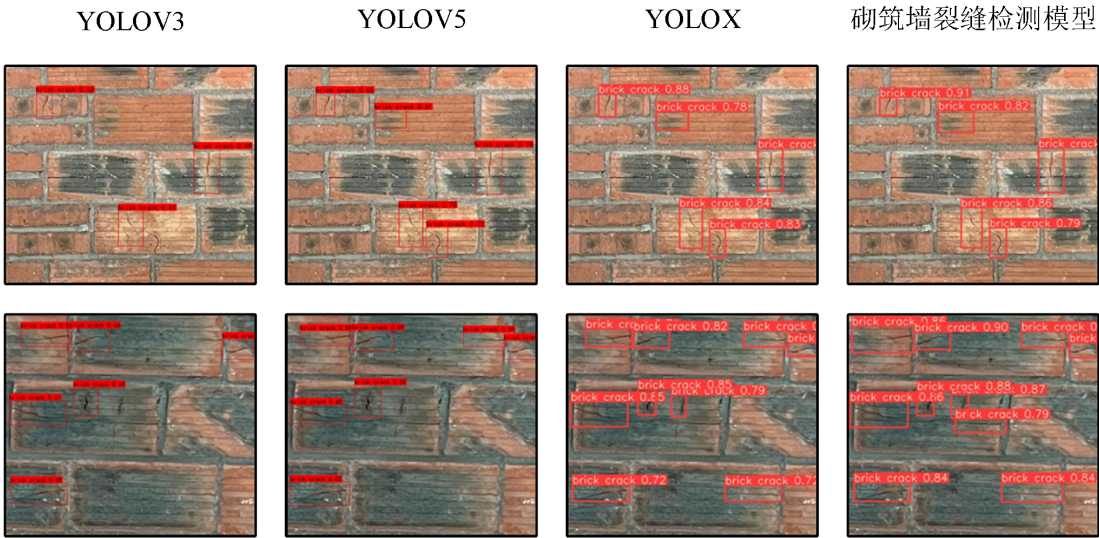


图 3-7 不同模型检测效果对比图

3.5 本章小结

本章提出了一种砌筑墙裂缝检测模型，以取代传统人工检测的方法，降低了企业的管控成本，使砌筑墙裂缝检测由结果管控过渡到过程管控成为了可能。通过改进模型结构，通过不同嵌入位置的对比实验，选择在主干网络与颈部嵌入 CBAM 卷积注意力机制模块，将注意力权重信息与图像的通道维数和空间维数相结合，引入数据中，突出重要的特征，抑制不重要的信息，使模型更加关注红砖表面。同时采用了线性加权的 Soft-NMS 来应对砌筑墙裂缝相互交织、重叠的问题，更有效的减少了漏检的情况，提升了模型的准确性。因此，提高了该模型对砌体墙体表面裂缝的检测能力，但是模型参数量较大，对硬件有一定要求，限制了实际使用场景。

第 4 章 轻量化的砌筑墙裂缝目标检测算法

第三章所提的砌筑墙裂缝检测算法取得了初步成果，对砌筑墙裂缝的检测达到较高的精度，但是由于参数量较大，导致部署到性能较低的设备上进行实时检测较为困难。为了降低部署对硬件的要求，扩大模型在实际中的应用场景，本章将在砌筑墙裂缝目标检测算法基础之上研究轻量化的砌筑墙裂缝目标检测算法。

4.1 轻量化模型需求

基于深度学习的目标检测算法在砌筑墙裂缝检测领域已取得初步成果。然而，这些算法往往参数庞大、计算过程复杂，给模型的部署与应用带来了一定的困难，特别是在资源受限的环境中。由于这些模型难以在移动设备或低性能硬件上运行，使其在实际工程中的应用受到了显著限制。针对这一挑战，对砌筑墙裂缝检测模型进行轻量化处理成为一项迫切的任务。轻量化模型不仅能够提高检测速度和效率，还能有效减少计算资源的消耗，降低模型部署成本。通过优化，轻量化后的模型可以满足在资源受限环境下运行的需求，例如在无人机巡检、移动机器人等应用场景中^[65]，确保算法能够在更广泛的工程实践领域中发挥关键作用。

综上，砌筑墙裂缝目标检测模型的轻量化是出于对实时性、经济性和实用性的考虑，以及对现有技术优化的必然趋势。通过轻量化，可以确保该技术更好地服务于工程质量控制和企业降本增效。通过 Ghost 卷积改进模型的 Backbone 部分实现轻量化的目的，降低了模型的参数量；Neck 部分改进为 BIFPN 模块，舍弃了原本的 PaNet 模块，使用了更为有效的双向加权传播机制，以确保模型的检测精度。这个模型相较于原始模型更为轻量化，减少了对硬件平台的计算需求，使其可以部署在移动端和边缘设备。同时，它在精度和检测速度方面表现也更加出色，适用于工程领域，可完成实际工作任务。

4.2 骨干网络轻量化改进原理

由于传统的 CNN 中存在许多相似的特征图，可以通过更高效的方法生成这些相似的特征图以实现轻量化操作。例如，原本的 128 通道到 256 通道的卷积操作可以分为两步：首先将其分为 128 通道到 64 通道，然后对 64 个特征图进行简单的线性变换，每个特征图生成 3 个与之相似的特征图，从而实现 64 通道到 256 通道的扩展。如图 4-1 所示上述操作本质上将传统的卷积操作分解为两步骤：首先使用传统卷积生成通道较少的特征图，以降低计算量，然后在这些特征图的基础上通过线性变换进一步减少计算量，生成新的特征图，最后，将这两组特征图拼接在一起，得到最终的输出。这种方法有助于有效地减小模型的参数量，提高计算效率，同时保持了良好的性能。

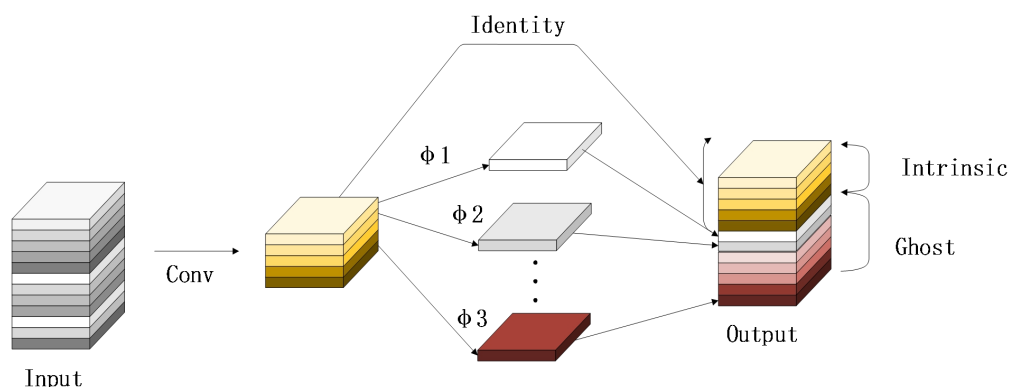


图 4-1 Ghost 卷积

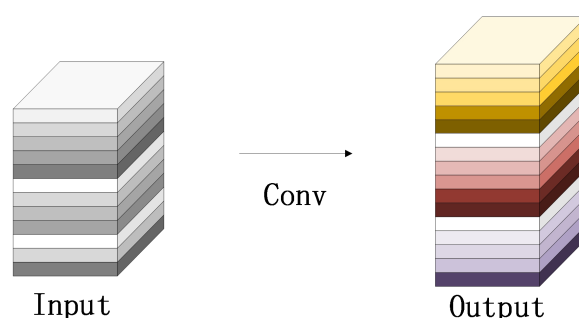


图 4-2 普通卷积

如图 4-2 展示了传统卷积输出的特征图完全由传统卷积操作得到，而 Ghost 卷积的输出特征图则小部分来自传统卷积，其余大部分则是通过线性变换生成的。

Ghost 卷积使用线性变换来替代传统卷积，是为了达到降低计算量的目的。

这种方法有效地减小了计算复杂度，同时保留了特征信息。**GhostBottleneck** 模块主要由先前提到的 **Ghost** 卷积模块堆叠构成，如图 4-3 所示。首个 **Ghost** 卷积单元起到扩展通道数量的作用。紧接着的第二个 **Ghost** 卷积单元通过降低通道数以适应网络中的快捷连接（shortcuts）。**GhostBottleneck** 的具体实施方案会根据步幅的大小而有所不同。当步幅设定为 1 时，如图 4-3（a）所示，**GhostBottleneck** 由两个承担不同功能的 **Ghost** 单元顺序连接形成。而在步幅为 2 时，如图 4-3（b）所示，在这两个连贯的 **Ghost** 单元之间会插入一个深度分离卷积层，这样做是为了与上下层的 **Ghost** 卷积单元协同完成宽高维度的压缩工作。

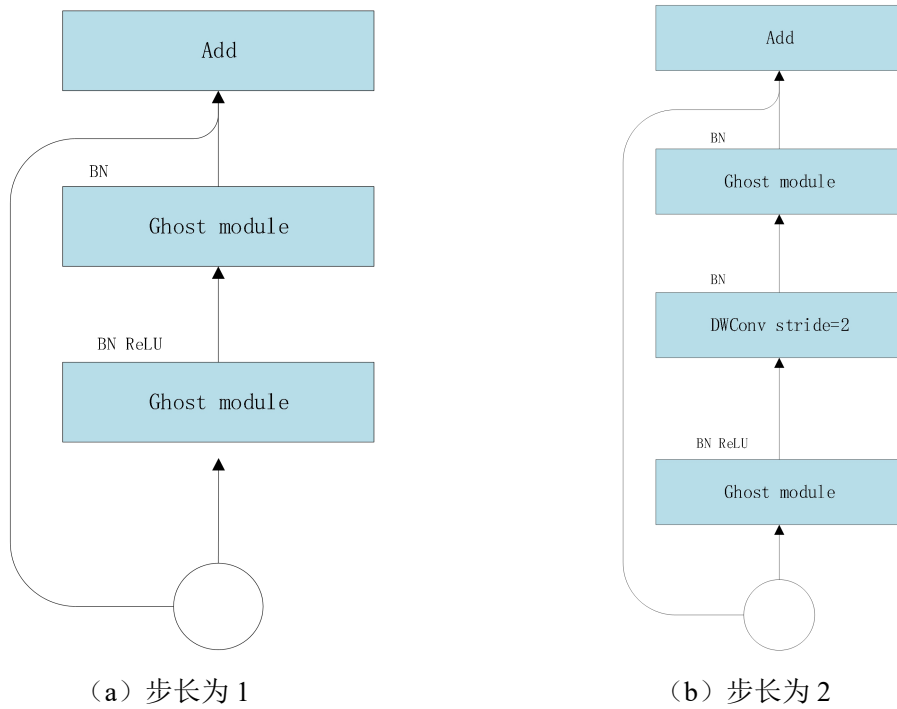


图 4-3 Ghost bottleneck 结构图

4.3 NECK 模块改进原理

经过采用更轻量化的卷积操作后，模型参数量大幅减小，便于移动端部署。然而，此举可能会导致检测精度下降，这对工程实际应用的意义有所减弱。为了解决这一挑战，本文引入了 **BiFPN** 作为模型的 **Neck** 部分，使用双向加权特征传播机制，舍弃只有简单线性叠加机制的 **PaNet** 模块，提高模型特征融合的效率，应对 **Ghost** 模块造成的精度下降问题。

图 4-4 展示了双向特征金字塔网络（**BiFPN**），通过采用金字塔结构、双向

信息传递、加权特征融合，高效地整合多尺度特征图。**BiFPN** 通过自上而下、自下而上的双向信息流来同时增强高级语义信息和低级位置信息，采用加权特征融合和有效的特征融合策略，**BiFPN** 能够提高模型的表示能力。

采用金字塔形状的结构，每一层包含不同尺度的特征图，以捕捉不同尺度的目标和细节，提高模型的表现力。双向信息传递通过自上而下和自下而上的路径，增强高级语义信息和低级位置信息，从而提高特征融合质量，提高模型性能。加权特征融合采用可学习的加权求和策略，以自适应分配权重，实现更有效的特征整合。有效的特征融合策略包括跨层连接和跳跃连接，有助于提高特征传播速度和效率。

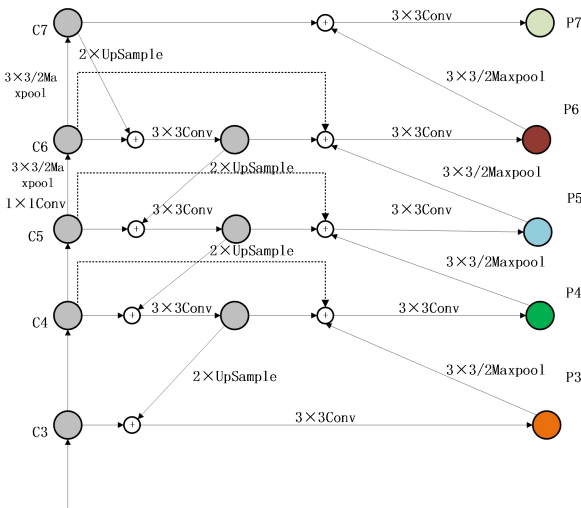


图 4-4 BiFPN 结构图

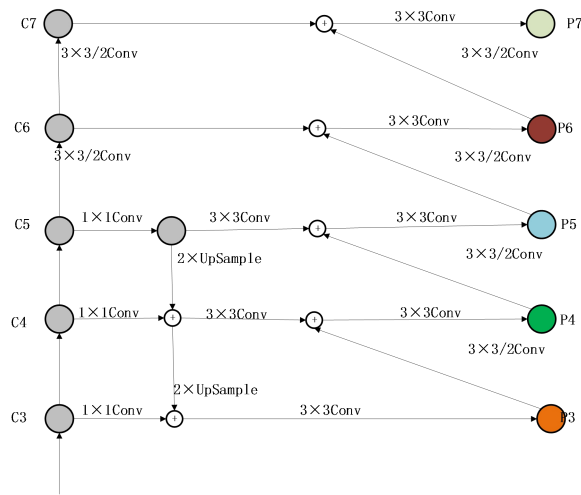


图 4-5 PaNet 结构图

PaNet 结构如图 4-5 所示，图 4-5 展示了 PaNet 的特征融合策略。BiFPN 采用了双向特征传播机制，并且采用了加权融合策略，其中每个特征图都有可学习

的权重，用 BIFPN 来代替采取简单线性叠加和仅有一条自底向上和一条自顶向下路径的 PaNet，能够更好地针对不同尺寸的裂缝特征对融合输出的特征产生不平等权重的问题，使 BiFPN 更有效的整合和调整不同尺寸的特征图，提高了整体模型的检测精度。

4.4 轻量化改进后模型结构简介

模型的改进包括两个主要方面：首先，在原始模型的 **Backbone** 中，使用 **GhostBottleneck** 对原始骨干网络进行改进，以使用计算量更少的线性变换来提取特征图，这一替换大幅减少了模型的参数量，从而减少了对部署平台的计算需求。这种改变有助于使模型更容易在资源受限的硬件上部署，但也带来了精确度下降的问题。

为了弥补这一精确度下降，模型的 **Neck** 部分引入了拥有双向特征传播机制和加权融合策略的 **BiFPN**，以取代原先的 **PaNet**。**BiFPN** 的特性使其在不同尺寸的特征图整合方面具有更好的效果。通过 **BiFPN** 的使用，模型相较于改进前拥有更好的检测精度，很好的解决了引入 **Ghost** 卷积导致的精度下降问题。

这些改进的结合使模型在轻量化和准确性之间取得了平衡，使其成为一个更适用于实际工程需求的解决方案，改进后的模型结构图如图 4-6 所示。通过减少参数量，降低计算成本，以及引入更有效的特征传播机制，模型实现了轻量化，并且在检测精度上面获得了提升，达到了部署在边缘设备的目的。

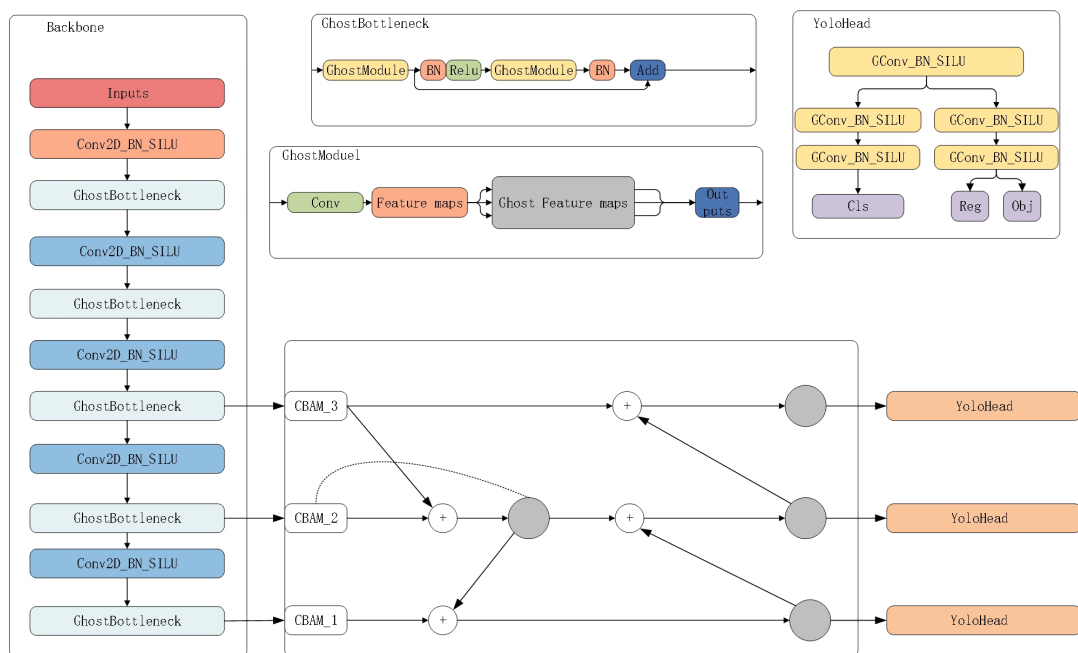


图 4-6 轻量化的砌筑墙裂缝检测模型结构图

4.5 实验结果与分析

4.5.1 实验环境

模型的训练的砌筑墙裂缝数据集是基于 VOC 格式构建的，这个数据集用于训练和评估模型的性能。在训练过程中，使用了 Windows10 64 位操作系统和一块 NVIDIA GTX 1080ti 显卡，为模型提供了足够的计算能力。编程语言选用了 Python 3.8，并采用了深度学习框架 PyTorch 1.10.0 以及 OpenCV 库的 4.6.0.66 版本。此外，为了加速训练过程，CUDA 11.4 和 CUDNN v8.2.1 也被应用。

训练策略包括将训练总轮数设置为 1700 轮，其中前 100 轮进行了冻结训练，即主干网络参数没有进行训练。而在接下来的 1600 轮中，进行解冻训练，训练了整个网络的所有参数。为了提高数据的多样性，训练时采用了 mosaic 和 mixup 两种数据增强方式，从而增强了模型的鲁棒性和泛化能力。优化器选择了 Adam，并初始学习率设置为 0.001。此外，使用了余弦退火算法，以根据训练进展动态调整学习率，从而提高训练的效率 and 性能，训练 1700 轮次后损失函数收敛，如图 4-7 是其曲线变化图，包括了训练集和测试集曲线。

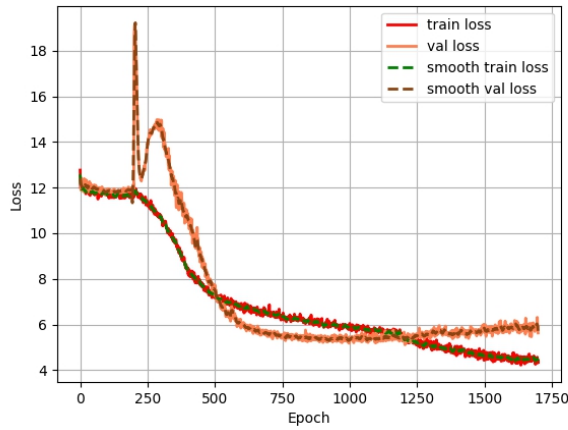


图 4-7 轻量化的砌筑墙裂缝目标检测模型 loss 曲线图

4.5.2 骨干网络模块横向对比实验

为了验证本文轻量化改进模块的优势，横向选取了同类别轻量化的模块，在砌筑墙裂缝检测任务上以 YOLOX 为基准模型进行改进，进行对比实验。MobileNetv2^[66]、MobileNetv3_small^[67]、MobileNetv3_large、ShuffleNetv2^[68]改进基准模型后的不同效果如表 4-1 所示。基于 MobileNetv3_large 拥有最好的准确率但是检测速度下降巨大；MobileNetv3_small 拥有最快的检测速度，但是检测精度与其他模型相比拥有较大劣势，综合得出基于 GhostBottleneck 模块的方法相较于其他同类改进在砌筑墙裂缝检测任务中更具有优势。

表 4-1 轻量化网络模块对比

骨干网络改进策略	mAP	FPS f/s
MobileNetv2	66.53%	30.71
MobileNetv3_small	64.79%	72.63
MobileNetv3_large	70.17%	29.74
ShuffleNetv2	65.42%	44.27
Ghost	67.54%	43.24

4.5.3 改进前后模型性能对比分析

通过对砌筑墙裂缝目标检测模型和经过轻量化的砌筑墙裂缝目标检测模型（包括了 Ghost 模块和 BiFPN 的融合）的实际识别效果、参数量和 P-R 曲线图进行了对比，验证了整体改进策略的有效性。

图 4-8 展示了在相同环境和初始训练参数下的砌筑墙裂缝检测模型的 PR 曲

线，而图 4-9 则呈现了轻量化后砌筑墙裂缝检测模型的 PR 曲线。通过对比这两条曲线，可以明显观察到改进后的模型在召回率和精准率之间取得了更好的平衡。具体而言，改进后的模型的 mAP 值提高了 1.52 个百分点，这意味着在检测任务中，它在不同召回率水平下能够更好地保持高精准率。这对于实际应用非常重要，尤其在需要高度可靠性的工程任务中，更有效地满足了实际工程需求。

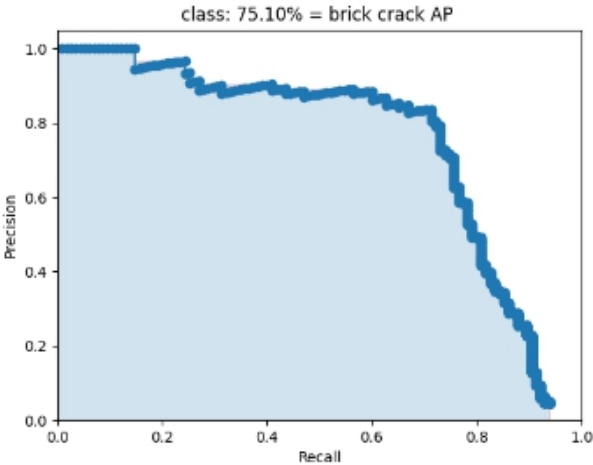


图 4-8 砌筑墙裂缝目标检测模型的 P-R 曲线图

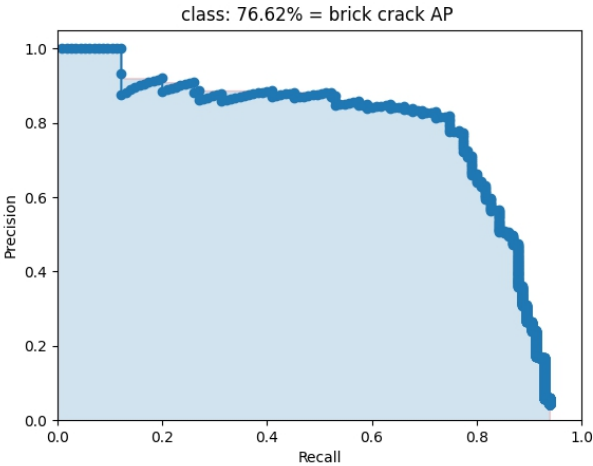


图 4-9 轻量化的砌筑墙裂缝目标检测模型的 P-R 曲线图

通过表 4-2 中的数据，可以得出轻量化后的模型相对于以前的模型参数量减少了 62.43%。参数减少不仅意味着模型可以更迅速地进行砌筑墙裂缝的实时检测，还反映了对部署平台成本的需求降低，从而有助于降低企业在部署和维护方面的开支。这一结果对于工程应用非常重要，因为在实际工程场景中，部署在边缘设备放置在施工现场代替技术人员旁站检测或者部署在移动端随身携带进行

检测是关键需求。而轻量化的模型不仅满足了这些需求，还减少了硬件和计算资源的成本，使其更具经济效益和可持续性。这样的改进将有助于企业降低运营成本，提高产品质量，从而在房屋建筑领域中取得更大的竞争优势。

表 4-2 参数量对比

模型名称	参数量	mAP 值	参数量减小量
轻量化前的模型	54.8M	75.10%	/
轻量化后的模型	20.6M	76.62%	-62.43%

图 4-10(a)、图 4-10(c)展示了砌筑墙裂缝目标检测的实际识别效果图，而图 4-10(b)、图 4-10(d)展示了轻量化后的砌筑墙裂缝目标检测模型的实际识别效果图。从比较这两个模型的实际识别效果图中，可以明显观察到轻量化的砌筑墙裂缝目标检测模型相较于砌筑墙裂缝检测模型在裂缝检测方面表现更出色。

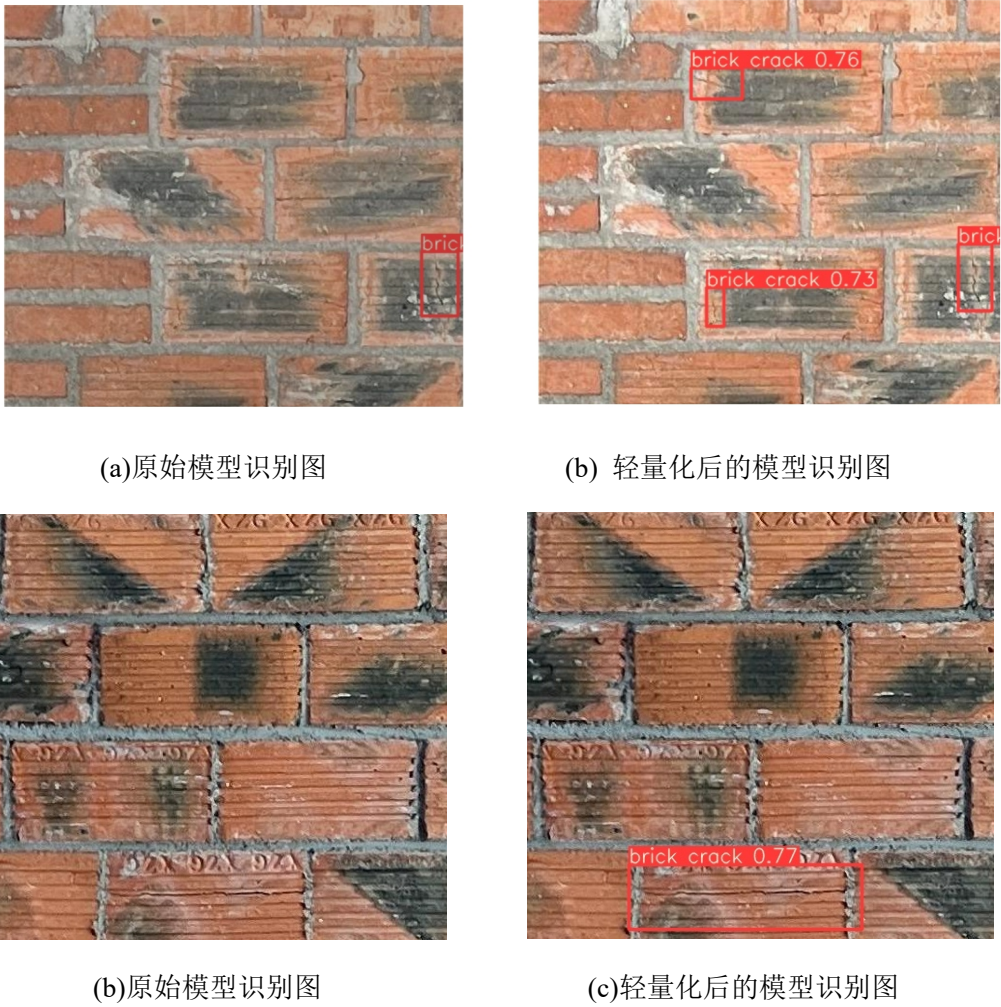


图 4-10 实际识别效果前后对比图

图 4-10（a）上部原始的砌筑墙裂缝检测模型漏检了两个裂缝，未能正常识

别，图 4-10（b）的上部可以得出轻量化后的砌筑墙裂缝检测模型对裂缝都成功进行了识别，图 4-10（a）的下部改进前的模型由于砖块的阴影漏检了一个裂缝，图 4-10（b）下部改进后的砌筑墙裂缝检测模型没有漏检。通过对比实验可以看出轻量化后的模型检测效果稳定和准确率更高，由于参数量较少具有更好的易部署性和实时性，且对于裂缝目标和阴影遮挡等复杂的情况具有更优秀的检测能力。

4.5.4 消融实验结果分析

本文采用砌筑墙裂缝数据集对采用的多种改进策略进行消融实验，以了解这些策略对原始模型的影响。表 4-3 中列出了每种改进策略对模型的精度（mAP）、参数量以及检测速度的影响结果。

表 4-3 消融实验表

	BiFPN	Ghost 模块	参数量	mAP	FPS f/s
1			54.8M	75.10%	32.41
2	√		40.5M	77.18%	24.56
3		√	36.4M	72.49%	41.47
4	√	√	20.6M	76.62%	38.82

表 4-3 展示了各项改进策略对本章节使用的基准模型的影响。实验 2 则采用 BiFPN 来替代改进前模型中的 PANet，结果表明这一改进提高了 mAP 值 2.08 个百分点。虽然检测精度获得提升，但检测速度大幅下降，这是因为 BiFPN 相较于 PANet 有着更复杂的双向加权机制。实验 3 采用 Ghost 卷积替代传统卷积改进主干网络，从表中可以看出，这一变化显著地减少了参数量，但是相较于原砌筑墙裂缝检测模型的准确率下降了 2.61%。这意味着 Ghost 卷积在轻量化模型的同时，并没有很好地保持模型的精度。

最终，采用了本文改进策略的实验 4，相较于本章节采用的基准模型，参数量减少了 34.2M，mAP 值提高了 1.52%个百分点。并且 FPS 提升了 6.41f/s。这表明改进后的模型在降低参数量的同时，也提高了砌筑墙裂缝检测的检测精度和速度。验证了本章对砌筑墙裂缝模型进行轻量化改进策略的成功。

4.5.5 算法鲁棒性分析

为了进行更深入的对比实验，将本文提出的算法与一系列目标检测算法进行比较。这些算法包括一阶段目标检测算法，如 SSD、YOLOv5 以及二阶段目标检测算法，如 Fast R-CNN 算法。

这些算法在本文提出的砌筑墙裂缝数据集上进行了训练和测试。以平均精度均值（mAP）和检测速度作为两项评价指标，以评估各算法的性能。表 4-4 展示了不同算法在砌筑墙裂缝数据集目标检测方面的实验结果。

表 4-4 各算法对比表

算法名称	mAP	FPS f/s
SSD	67.23%	9.73
YOLOv5	70.76%	33.26

续表 4-4 各算法对比表

算法名称	mAP	FPS f/s
Fast R-CNN	76.09%	5.03
YOLOV3	69.81%	17.33
砌筑墙裂缝检测算法	75.10%	32.41
轻量化砌筑墙裂缝检测算法	76.62%	38.82

从表 4-4 可以看出，轻量化砌筑墙裂缝检测算法在砌筑墙裂缝数据集上的平均精度均值方面达到了 76.62%，这相比表中的一阶段检测、二阶段检测和之前设计的砌筑墙裂缝检测算法都有显著提升，这是由于 BiFPN 相较于 PaNet 拥有更为有效的双向加权机制，可以更有效地融合不同尺寸的特征图，使得 Head 部分能够更好的完成回归与分类任务，有效提高模型的准确性。

如图 4-11 展示了各算法在砌筑墙裂缝检测任务上的识别效果图。通过与各算法识别效果图进行比较与上述结论，可以得出，本文提出的算法在保持较好检测速度的同时，在砌筑墙裂缝数据集上实现了更高的平均精度均值，进一步提高了砌筑墙裂缝目标检测的准确性相较于其他算法具有与优势地位。

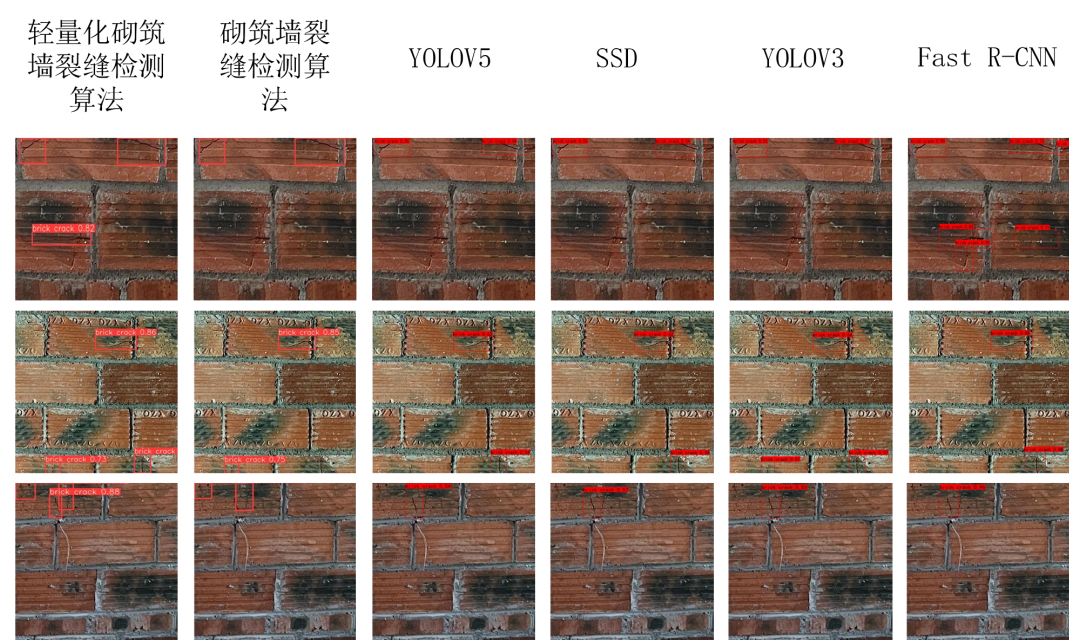


图 4-11 与其他算法识别效果对比图

4.6 本章小结

针对砌筑墙裂缝检测的准确性、实时性和便携性等需求，提出了经过轻量化的砌筑墙裂缝检测模型，相较于第三章提出的砌筑墙裂缝检测模型减少了 62.43% 的参数量，降低了部署的门槛，使其更容易部署在移动端；FPS（每秒帧数）提升了 6.41；mAP 值也提升了 1.52%，满足了砌筑墙裂缝检测的准确性、实时性和便携性等需求。通过与其他目标检测算法对比，表明模型在裂缝检测方面具有更高的精确度和检测速度，在砌筑墙裂缝检测任务中较其他算法更优秀。虽然完成了轻量化改进，但是对于没有代码经验和深度学习背景的土木工程技术人员，使用该模型仍然是一种挑战。

第 5 章 砌筑墙裂缝检测系统 web 端部署

第四章对第三章成果进行轻量化改进,使模型参数量下降并且检测精度在砌筑墙裂缝检测任务上获得了提升,但是对没有代码经验以及深度学习背景的土木工程技术人員使用该模型进行检测仍然是一种挑战。本章将在轻量化的砌筑墙裂缝检测算法基础上实现 web 端的部署,提供一套对用户友好的检测系统解决实际应用问题。

5.1 系统需求

针对用户群体没有代码以及深度学习理论背景的问题,设计并实现了一个以用户友好为宗旨的 Web 端砌筑墙裂缝检测系统。此系统的最终目标是将轻量化的砌筑墙裂缝检测算法应用于实际工程中,便于土木工程技术人員通过局域网上上传图片或使用实时视频流代替旁站管控进行砌筑墙裂缝检测。系统的设计强调了简便性和易用性,用户无需专业的技术背景,仅需通过浏览器访问指定的 IP 地址,即可快速启动实时检测功能。这种设计省去了复杂的安装和配置步骤,也无需长时间的学习,使得无代码经验以及深度学习背景的土木工程技术人員也可快速上手使用。通过这种方式,确保了检测模型的实用性和便捷性,使其能够真正服务于工程实践。

5.2 系统总体设计

5.2.1 系统开发环境

本系统基于本地 PC 服务器开发,用户可以通过 Web 浏览器远程访问并执行砌筑墙裂缝目标检测任务。硬件平台为一台配备 Microsoft Windows 10 x64 操作系统、i7 10700 处理器和 GTX 1080ti 显卡的 PC 端电脑,内存容量为 64.0GB RAM。在开发过程中,使用了 PyCharm 专业版 2021.3.3 作为集成开发环境,编程语言为 Python 3.8。通过 HTML 和 Flask 框架,创建了一个 Web 端界面,实现了网页登录与注册功能、目标检测结果显示和后端保存等功能。

5.2.2 系统整体架构设计

本系统的砌筑墙裂缝目标检测功能是通过前后端的数据交互来实现的。用户首先通过访问本地工程生成的 IP 地址来登录 Web 页面。在浏览器上，用户点击相应的按钮以发送功能请求。Flask 服务器在接收到这些请求后，将其转发至本地部署的模型进行预测处理。如图 5-1 所示，展示了程序设计小段。

```
from importlib import import_module
import os
import cv2
from flask import Flask, render_template, Response, request, redirect, url_for

if os.environ.get('CAMERA'):
    Camera = import_module('camera_' + os.environ['CAMERA']).Camera
else:
    from camera import Camera

app = Flask(__name__)

NAME = ""
FILE_FLAG = False
CAMERA_FLAG=False

@app.route('/')
def index():
    """Video streaming home page."""
    return render_template('index.html')

def video_gen(camera):
    """Video streaming generator function."""
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

def camera_gen():
    vid = cv2.VideoCapture(1)
    while True:
        return_value, frame = vid.read()
        image = cv2.imencode('.jpg', frame)[1].tobytes()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + image + b'\r\n')

@app.route('/video_start')
def video_feed():
```

图 5-1 程序设计小段

完成预测后，服务器将结果数据回传给浏览器端，以便进行展示。如图 5-2 所示，展示了整个系统的前后端数据交互方法。

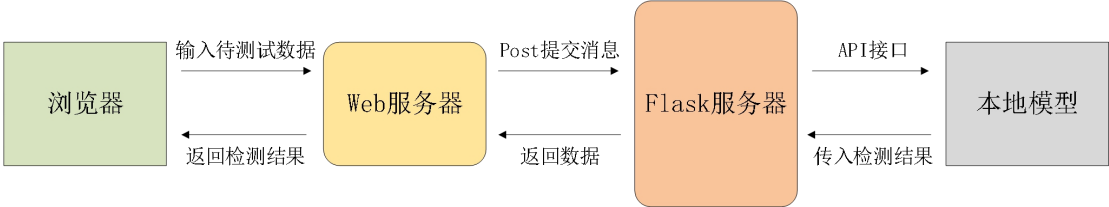


图 5-2 前后端数据交互

该系统的工作流程图如图 5-3 所示，包括了模型从输入到输出的所有步骤。用户需要通过网页登录并注册账户，才能上传砌筑墙图片或使用摄像头进行实时检测。系统的后端采用轻量化后的模型来识别图片中的裂缝目标。用户不仅可以

在网页上直接查看检测结果，还可以通过后端服务将处理后的结果以 jpg 格式下载到本地，以便根据图片对比分析生产质量情况，如图 5-4 所示，展示了存储位置以及文件格式。

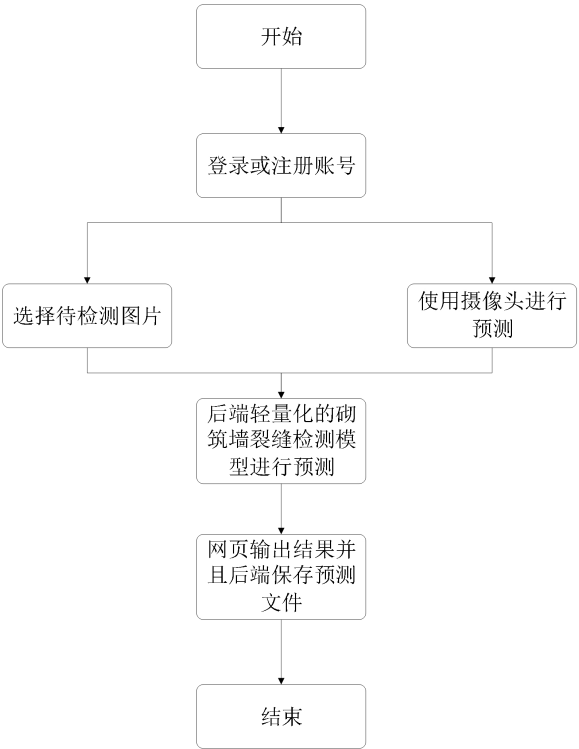


图 5-3 系统工作流程图

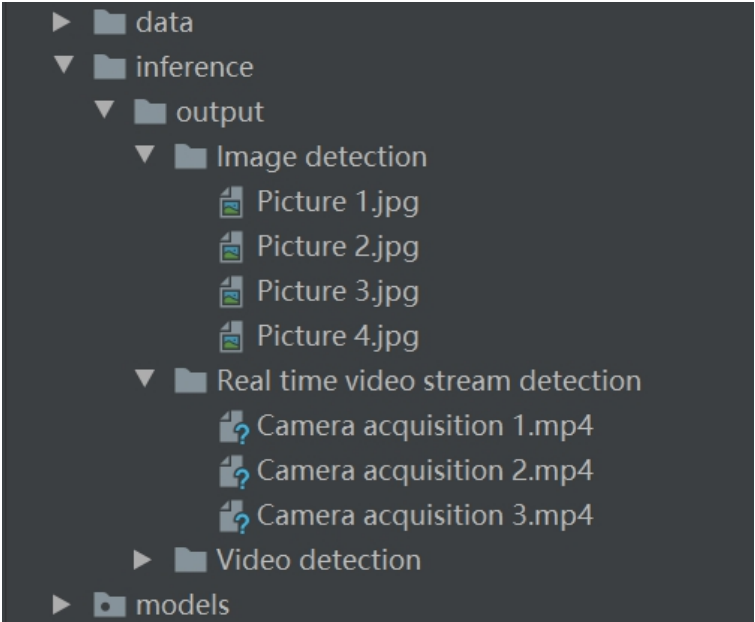


图 5-4 后端保存效果图

5.3 系统软件实现

在本节中，使用 PyCharm 作为开发环境直接编写 HTML 代码，以构建针对砌筑墙裂缝目标检测系统的网页客户端。该系统将利用 Flask 框架实现简洁的登录和注册路由，从而完成用户身份验证等功能。界面设计上，系统将包含三个主要部分：用于用户认证的登录界面；允许新用户创建账户的注册界面；以及功能选择界面，让用户在成功登录后能够进行检测功能的选取。

5.3.1 登录和注册界面

HTML，即超文本标记语言，提供了一种构建网页结构的机制。它利用一系列标签定义内容在浏览器中展示。根据设计需求，用 HTML 编写相应的注册和登录界面代码。在设计前端表单时，只需构建基础的输入字段布局。接着，在 HTML 的基础框架中，设定页面头部信息、主题内容等，涉及背景、按钮样式及颜色方案等视觉元素。通过编码，得以实现了适用于 web 端的登录与注册界面，如示例图 5-5 所展现。项目启动后，用户将被引导至登录页面如图 5-5；若尚未拥有账户，需先行注册如图 5-5（b），完成后将自动返回登录页面以便进行账号登录操作如图 5-5（a）。



(a) 登陆界面展示图



(b) 注册界面展示图

图 5-5 web 端的注册与登录界面

5.3.2 检测功能界面

在完成登录后，用户将被引导至检测功能界面如图 5-6 所示。这个界面同样使用 HTML 进行开发，为用户提供了上传按钮，用于切换不同的检测文件。通过使用超文本标记语言（HTML）设计了这个按钮，以实现不同图片检测文件的上传。用户可以点击选择文件，确认上传的文件，然后进行图片的检测。此外还有摄像头检测按钮，用户可以通过点击该按钮，进行实时视频流的检测。

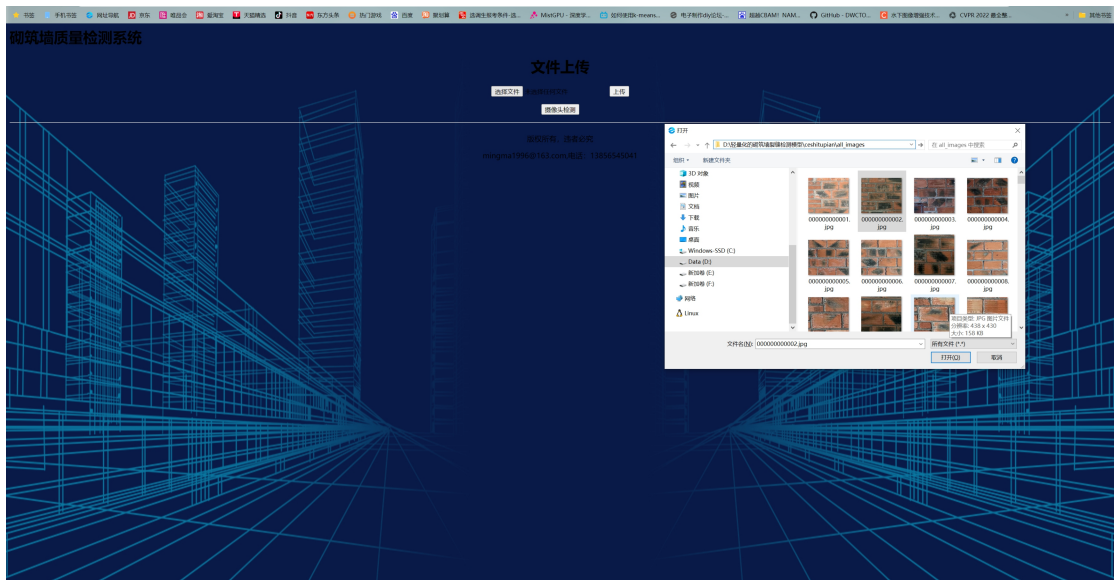
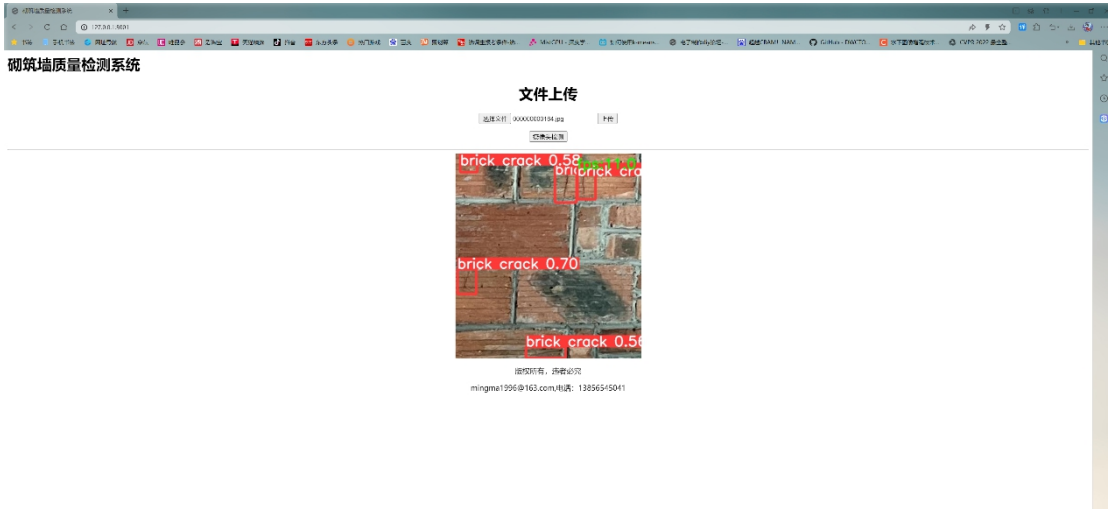


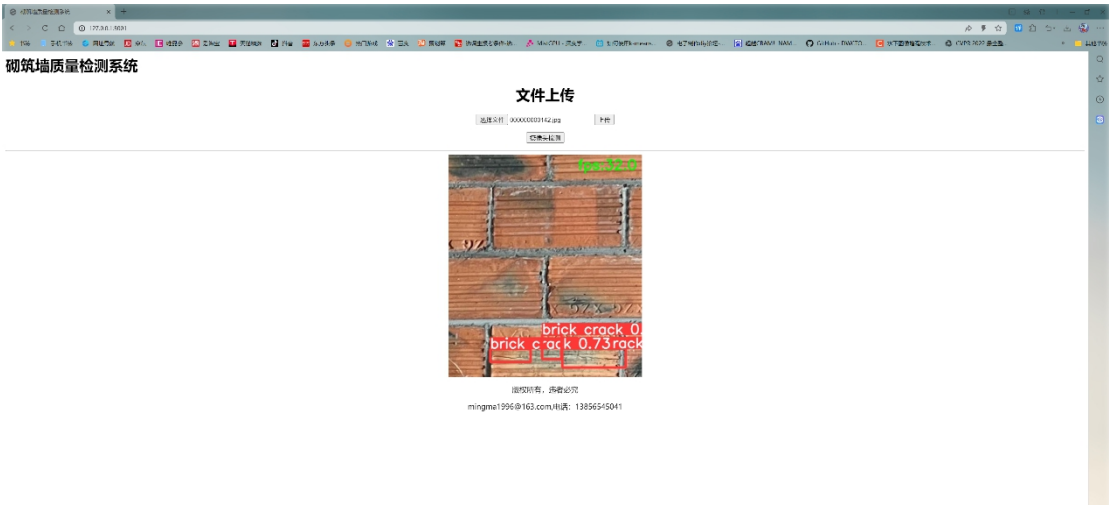
图 5-6 检测功能界面展示

5.4 系统功能测试

为了避免背景干扰用户查看检测效果图，在使用后背景转换为纯白，方便用户直观的查看检测效果图。图片检测的效果如图 5-7（a）和（b）所示。图 5-7（a）（b）展示了图片检测下的检测效果，同时在图片右上角展示了帧率速度。图 5-7（a）和（b）展示了使用摄像头功能检测砌筑墙裂缝的效果图，从右上角的速率可以看出，检测速度可以满足实际工程需求。该系统能够完全实现实时的目标检测，本系统的检测效果已经达到了实际应用的水平。



(a)

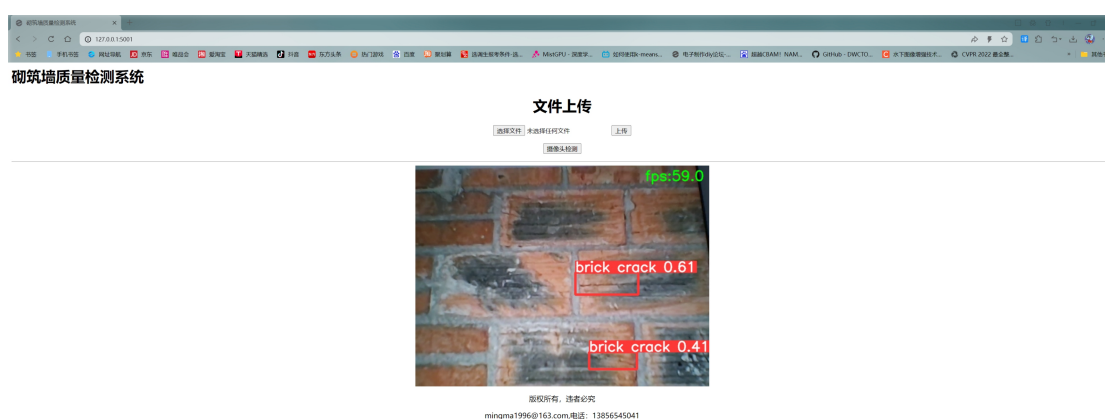


(b)

图 5-7 图片检测的效果



(a)



(b)

图 5-8 摄像头实时检测的效果

5.5 本章小结

在本章节中，详细阐述了基于 Python 语言和 Flask 框架开发的针对砌筑墙裂缝检测系统的设计与实现过程，并对系统进行了全面的运行测试和效果展示。该系统提供了用户注册与登录、图片上传、使用实时视频流检测以及后端文件保存等多项功能，极大地方便了建筑工程领域的研究人员，降低了企业管控砌筑墙质量的成本。通过简单的局域网连接，用户无需专业背景即可通过 Web 页面轻松地对砌筑墙裂缝进行目标检测。经过一系列的验证，系统的功能、性能和可靠性均符合设计时设定的目标。

第 6 章 总结与展望

6.1 全文工作总结

深度学习技术在砌筑墙裂缝检测领域的应用具有显著意义，然而，图像采集过程中受到施工现场保密因素干扰，增加了获取砌筑墙裂缝数据集的难度。此外，通用检测算法对砌筑墙裂缝适配性较差，检测效果不理想，同时现有的砌筑墙裂缝检测算法对计算资源的需求较高，极大地限制了工程应用场景。针对易部署性、实用性和准确性的需求，本研究专注于将深度学习应用在民用建筑中广泛生产的红砖砌筑墙裂缝检测，并实现一套无专业背景即可使用的系统，推动了基于深度学的技术在砌筑工程领域的应用。

研究首先探讨了基于深度学习的砌筑墙裂缝检测的研究背景及其重要性，并概述了当前主流的目标检测算法。特别地，本文深入讲解了 YOLO 系列以及 YOLOX 算法的基础理论，为本研究提供了算法支撑。根据模型应用于实际工程的观点出发，构建了基于真实施工现场采集的数据集，并和专业的土木工程人员协助，标注了不符合国家规范的砌筑墙裂缝，构建了砌筑墙裂缝数据集。为了提高检测精度，对 YOLOX 算法进行了有关砌筑墙裂缝数据集的针对性修改，具体方案是基于注意力机制的改进和 NMS 算法的优化，最终提出了砌筑墙裂缝检测模型，以此专门适配砌筑墙裂缝的目标检测场景。尽管此算法拥有出色的检测精度，但参数量和计算量较大，在实际部署与应用过程中仍面临一些挑战。具体来说，模型对部署平台的计算条件要求较高，并且检测速率不尽如人意，这两个因素限制了模型在工程应用场景中的广泛运用。因此，进一步研究对模型的轻量化处理具有重要意义。通过使用 GhostBottleneck 的对模型进行轻量化，同时改进了模型的 Neck 部分，用 BIFPN 代替 PaNet 模块保持了参数量减少后的模型准确度，最终开发了相较于原模型更加轻量化并且精度更高的的砌筑墙裂缝检测模型，完成了基于深度学习的砌筑墙裂缝检测技术研究。为了实现砌筑墙裂缝检测技术产业化应用，利用 Flask、HTML 和 Python 技术栈，实现了一个能够通过局域网访问的 web 端系统，用于部署轻量化的砌筑墙裂缝检测模型，使建筑工程

专业人员在无代码和深度学习的背景下无障碍使用该系统,实现了使用实时视频流检测代替人工旁站管控砌筑墙裂缝的传统方法,并降低了管控成本。

研究的核心内容与成果概述如下:

(1) 数据集的构建:通过与知名企业合作研究,在施工现场直接采集砌筑墙数据集图片,之后使用 Labelimg 软件,在专业土木技术人员的协助下,根据国家规范挑选出不合格的裂缝对其进行标记,使构建的数据集具有实际工程意义,适配于实际工程检测,完成了砌筑墙裂缝数据集的构建。

(2) 提出砌筑墙裂缝检测模型:通过在不位置融合 CBAM 注意力机制进行对比实验,发现在 YOLOX 主干网络的最后融合卷积块注意力模块(CBAM),可以显著提升模型精度以及对红砖区域的关注度,提升模型检测精度,并且通过与同类注意力机制进行对比验证了 CBAM 在砌筑墙裂缝检测任务上的优势地位。同时,采用了线性加权的 Soft-NMS 对模型改进,可以更有效的应对砌筑墙裂缝相互交织、重叠,改善模型的目标定位和分类性能。实验结果显示,砌筑墙裂缝检测模型相较于原版 YOLOX 模型,在特定的砌筑墙裂缝数据集上,模型的 mAP@0.5 指标提升了 3.55%。成功构建了专门针对砌筑墙裂缝检测的模型。

(3) 模型轻量化改进:为了满足模型能够部署在边缘设备的需求,使用 GhostBottleneck 改进砌筑墙裂缝检测模型的骨干网络部分,大幅减少了网络参数量。同时,将 Neck 部分结构替换为更高效的 BIFPN,以保持检测准确性。实验结果表明,新开发的轻量化网络平均精度达到了 76.62%,且平均检测时间仅为 0.025 秒。表明研发的模型适合部署在资源受限的设备上进行实时砌筑墙裂缝检测。

(4) Web 端系统设计与部署:利用 Python 结合 Flask 和 HTML 技术,设计并实现了一个 Web 端系统用于实时砌筑墙裂缝检测。该系统允许用户通过局域网访问网页,调用摄像头来代替技术人员旁站进行裂缝检测,从而实现跨设备的实时监测功能和降低了企业管控砌筑墙质量的成本。使土木工程技术人员在无深度学习专业背景的条件下,也可使用模型进行检测。

6.2 后续研究展望

在本研究中,基于深度学习的砌筑墙裂缝检测技术研究实现,尽管已取得

一定成果，但存在局限性，如只针对红砖类型的砌筑墙；仅检测裂缝这一种类；以及算法精确度仍有待提高。目前的工作仍有改进空间，未来研究可朝以下方向发展：

（1）检测更多缺陷种类：虽然现在可以有效检测砌筑墙裂缝，但是并不可以检测例如砌筑墙狭缝、灰缝过大以及灰缝过小等问题。未来可以采集更多的数据集图片对其他缺陷进行标注实现检测更多的缺陷种类。

（2）砌筑墙裂缝检测精确度优化：目前对砌筑墙裂缝的精确度已经达到一定水平，但是百分百依靠模型进行检测还是有待提高。可以考虑使用更先进的目标检测模型进行改进以进一步提升检测精度。

（3）扩大 Web 端部署范围：当前系统仅在局域网内部署，限制了使用范围。后续可利用内网穿透等技术对其改良，实现远程访问，拓展系统的实用性。

参考文献

- [1] 徐国良, 胡国标, 张泽勇. 双碳目标背景下我国绿色建筑高质量发展措施分析[J]. 陶瓷, 2022(9):152-153, 196
- [2] 毛志兵. “双碳”目标下中国建造的实现路径[J]. 建筑, 2022(14):59-60.
- [3] 马智亮. 解读《中国建筑业信息化发展报告(2021)智能建造应用与发展》[J]. 中国建设信息化, 2021(24):12-15.
- [4] 刘德平. 建筑物常见裂缝原因分析及防治措施[D]. 同济大学, 2007.
- [5] 肖绪文, 冯大阔. 建筑工程绿色施工现状分析及推进建议[J]. 施工技术, 2013, 42(1): 12-15.
- [6] 罗东亮, 蔡雨萱, 杨子豪等. 工业缺陷检测深度学习方法综述[J]. 中国科学: 信息科学, 2022, 52(6):1002-1039.
- [7] Ge Z, Liu S, Wang F, et al. YOLOX: Exceeding YOLO series in 2021[J]. arXiv preprint 2021:arXiv:2107.08430.
- [8] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, 2016: 779-788.
- [9] 赵永强, 饶元, 董世鹏等. 深度学习目标检测方法综述[J]. 中国图象图形学报, 2020, 25(4):629-654.
- [10] Zou Z, Chen K, Shi Z, et al. Object detection in 20 years: A survey[J]. Proceedings of the IEEE, 2023, 111(3): 257-276.
- [11] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]. Proceedings the 2001 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2001, 1: I-I.
- [12] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C], Computer Vision and Pattern Recognition. IEEE, 2005, 1: 886-893.

- [13]Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models[J]. IEEE transactions on pattern analysis and machine intelligence. IEEE, 2009, 32(9): 1627-1645.
- [14]Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [15]周晓彦, 王珂, 李凌燕. 基于深度学习的目标检测算法综述[J]. 电子测量技术, 2017, 40(11):89-93.
- [16]Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, 2014: 580-587.
- [17]HE Kai ming, Zhang Xiang yu, REN Shao qing, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Trans Pattern Anal Mach Intell, 2015, 37(9):1904-16.
- [18]Girshick R. Fast r-cnn[C]. Proceedings of the IEEE international conference on computer vision. IEEE, 2015: 1440-1448.
- [19]Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
- [20]Dai J, Li Y, He K, et al. R-fcn: Object detection via region-based fully convolutional networks[J]. Advances in neural information processing systems, 2016, 29.
- [21]Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
- [22]He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]. Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [23]Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge[J]. International journal of computer vision, 2010, 88(2): 303-338.

- [24]Lu X, Li B, Yue Y, et al. Grid r-cnn[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 7363-7372.
- [25]Jiao L, Dong S, Zhang S, et al. AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection[J]. Computers and Electronics in Agriculture, 2020, 174: 105522.
- [26]Wei Liu, Dragomir Anguelov, Dumitru Erhan, et al. SSD: Single Shot Multi Box Detector[C]. European Conference on Computer Vision. Springer International Publishing, 2016, 9905:21-37.
- [27]Zhang A, Liao Y, Liu S, et al. Mining the benefits of two-stage and one-stage hoi detection[J]. Advances in Neural Information Processing Systems, 2021, 34: 17209-17220.
- [28]Sermanet P, Eigen D, Zhang X, et al. Overfeat: Integrated recognition, localization and detection using convolutional networks[J]. arXiv preprint arXiv:1312.6229, 2013.
- [29]Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]. Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
- [30]Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271..
- [31]Oseph Redmon, Ali Farhadi. YOLOv3: An incremental improvement[J].arXiv preprint ar Xiv, 2018:1804.02767
- [32]He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [33]Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: optimal speed and accuracy of object detection[J]. arXiv:2004, 2020,10934.
- [34]Misra D. Mish: A self regularized non-monotonic activation function[J]. arXiv preprint arXiv:1908.08681, 2019.

- [35]唐声飞, 胡小芳. 加气混凝土砌块墙体裂缝产生的机理及防治[J]. 南华大学学报(理工版), 2004, 18(2):67-69.
- [36]周后明. 常见砌块建筑墙体裂缝通病的防治[J]. 建筑砌块与砌块建筑, 2004(2): 43-44.
- [37]Hallee M J, Napolitano R K, Reinhart W F, et al. Crack detection in images of masonry using cnns[J]. Sensors, 2021, 21(14):4929-4929.
- [38]Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]. Proceedings of the European conference on computer vision. 2018: 3-19.
- [39]Bodla N, Singh B, Chellappa R, et al. Soft-NMS--improving object detection with one line of code[C]. Proceedings of the IEEE international conference on computer vision. 2017: 5561-5569.
- [40]Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580-1589.
- [41]Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790.
- [42]Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.
- [43]卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述[J]. 数据采集与处理, 2016, 31(1): 1-17.
- [44]Lipton Z C, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning[J]. arXiv preprint arXiv:1506.00019, 2015.
- [45]陈超, 齐峰. 卷积神经网络的发展及其在计算机视觉领域中的应用综述[J]. 计算机科学, 2019, 46(3):63-73.
- [46]张焕, 张庆, 于纪言. 激活函数的发展综述及其性质分析[J]. 西华大学学报(自然科学版), 2021, 40(4):1-10.
- [47]杨丽, 吴雨茜, 王俊丽等. 循环神经网络研究综述[J]. 计算机应用, 2018, 38(S2):1-6+26.

- [48]Zhu X, Cheng D, Zhang Z, et al. An empirical study of spatial attention mechanisms in deep networks[C]. Proceedings of the IEEE/CVF international conference on computer vision. 2019: 6688-6697.
- [49]高友文, 周本君, 胡晓飞. 基于数据增强的卷积神经网络图像识别研究[J]. 计算机技术与发展, 2018, 28(08):62-65.
- [50]Wu Y, He K. Group normalization[C]. Proceedings of the European conference on computer vision. 2018: 3-19.
- [51]Bhanu B, Ratha N. Proceeding of the iee computer scoeity conference on computer vision and pattern recognition: preface[J]. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, 4400:11563-11572.
- [52]Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
- [53]Lin Tsung-Yi, Goyal Priya, Girshick Ross, et al. Focal loss for dense object detection[J]. IEEE Trans Pattern Anal Mach Intell, 2020, 42(2):318-327.
- [54]Lu Y, Lin Y, Wu H, et al. Sirst-5k: Exploring massive negatives synthesis with self-supervised learning for robust infrared small target detection[J]. arXiv preprint arXiv:2403.05416, 2024.
- [55]孙书魁, 范菁, 孙中强, 等. 基于深度学习的图像数据增强研究综述[J]. 计算机科学, 2024, 51(1):150-167.
- [56]郑远攀, 李广阳, 李晔. 深度学习在图像识别中的应用研究综述[J]. 计算机工程与应用, 2019, 55(12):20-36.
- [57]Yun S, Han D, Oh S J, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features[C], Proceedings of the IEEE/CVF international conference on computer vision. IEEE, 2019: 6023-6032.
- [58]Zhang H, Cisse M, Dauphin Y N, et al. Mixup: Beyond empirical risk minimization[J]. arXiv preprint arXiv:1710.09412, 2017.
- [59]Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

- [60]Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [61]Neubeck A, Van Gool L. Efficient non-maximum suppression[C]. 18th international conference on pattern recognition. IEEE, 2006, 3: 850-855.
- [62]GB 50203-2011. 砌体结构工程施工质量验收规范[S]. 2011.
- [63]陈超, 齐峰. 卷积神经网络的发展及其在计算机视觉领域中的应用综述[J]. 计算机科学, 2019, 46(3):63-73.
- [64]张珂, 冯晓晗, 郭玉荣, 等. 图像分类的深度卷积神经网络模型综述[J]. 中国图象图形学报, 2021, 26(10):2305-2325.
- [65]廖广平. 基于改进 SSD 的无人机航拍图像小目标检测算法研究[D]. 长沙:湘潭大学, 2022.
- [66]Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018: 4510-4520.
- [67]Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]. Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314-1324..
- [68]Ma, Ningning, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design. Proceedings of the European conference on computer vision, 2018.

攻读硕士学位期间所发表的论文及其它研究成果

学术论文：

[1] 以第一作者在 EIECS2023 发表论文一篇，EI Compendex 索引

致谢

感谢老师三年的指导，感谢父母养育之恩，感谢同门师兄的互帮互助，感谢一路走来所有帮助过我的人。

特别感谢朱子燕小姐的包容与陪伴，希望能更幸福的走接下来的路。

想要什么样的生活，取决于我们自己。