智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

面向边缘设备的目标检测模型研究

徐伟峰, 雷耀, 王洪涛, 张旭

引用本文:
徐伟峰, 雷耀, 王洪涛, 等. 面向边缘设备的目标检测模型研究[J]. 智能系统学报, 2025, 20(4): 871-881.
XU Weifeng, LEI Yao, WANG Hongtao, et al. Research on object detection models for edge devices[J]. *CAAI Transactions on Intelligent Systems*, 2025, 20(4): 871-881.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202406015

DOI: 10.11992/tis.202406015
网络出版地址: https://link.cnki.net/urlid/23.1538.TP.20241211.1852.004

面向边缘设备的目标检测模型研究

徐伟峰[1,2], 雷耀[1], 王洪涛[1,2], 张旭[1]
(1. 华北电力大学(保定)计算机系, 河北 保定 071003; 2. 河北省能源电力知识计算重点实验室, 河北 保定 071003)

摘　要: 现有目标检测模型在边缘设备上部署时, 其检测性能和推理速度的平衡有较大提升空间。针对此问题, 本文基于 YOLO (you can only look once) v8 提出一种可部署到多类边缘设备上的目标检测模型。在模型的骨干网络部分, 设计了 EC2f (extended coarse-to-fine) 结构, 在降低参数量和计算复杂度的同时降低数据读写量; 在颈部网络部分, 将颈部网络替换为 YOLOv6-3.0 版本的颈部网络, 加速了模型推理, 并将推理精度维持在较好水平; 预测头网络部分设计了多尺度卷积检测头, 进一步降低了模型的计算复杂度和参数度。设计了两个版本 (n/s 尺度) 以适应不同的边缘设备。在 X 光数据集的实验表明, 模型在推理精度上比同尺度的基准模型分别提升 0.5/1.7 百分点, 推理速度上分别提升 11.6%/11.2%。在其他数据集上的泛化性能测试表明, 模型的推理速度提升了 10% 以上, 精度降低控制在 1.3% 以内。实验证明, 模型在推理精度和速度之间实现了良好的平衡。

关键词: 目标检测; YOLO; 边缘设备; 推理精度; 推理速度; 数据读写量; 计算复杂度; 模型部署
中图分类号: TP391.4　　**文献标志码**: A　　**文章编号**: 1673−4785(2025)04−0871−11

中文引用格式: 徐伟峰, 雷耀, 王洪涛, 等. 面向边缘设备的目标检测模型研究 [J]. 智能系统学报, 2025, 20(4): 871–881.
英文引用格式: XU Weifeng, LEI Yao, WANG Hongtao, et al. Research on object detection models for edge devices[J]. CAAI transactions on intelligent systems, 2025, 20(4): 871–881.

Research on object detection models for edge devices

XU Weifeng[1,2], LEI Yao[1], WANG Hongtao[1,2], ZHANG Xu[1]
(1. Department of Computer, North China Electric Power University(Baoding), Baoding 071003, China; 2. Hebei Key Laboratory of Knowledge Computing for Energy & Power, Baoding 071003, China)

Abstract: Existing object detection models can be improved in terms of balancing detection performance and inference speed on edge devices. Hence, a YOLO (you can only look once) v8-based model optimized for various edge devices is proposed. In the Backbone, an EC2f (extended coarse-to-fine) structure is designed to reduce parameters, computation, and data read/write volume. In the Neck, the YOLOv6-3.0 version is used to accelerate inference while maintaining accuracy. In the Head, a multiscale convolutional detection head, which further reduces computational load and complexity, is featured. Two versions (n/s scales) are designed to suit different edge devices. Experiments on an X-ray dataset demonstrate that the proposed model improves inference accuracy by 0.5%/1.7% and speed by 11.6%/11.2% compared with baseline models of the same scale. Generalization tests on other datasets present an increase in inference speed of over 10% and an accuracy reduction controlled within 1.3%. Overall, the model achieves a satisfactory balance between inference accuracy and speed.

Keywords: object detection; YOLO; edge devices; inference accuracy; inference speed; data read/write volume; computational load; model deployment

近年来, 随着深度学习理论的深入研究和计算机算力的显著提升, 基于深度学习的目标检测技术在多个领域(如安全检测、智慧农业等)得到了广泛应用。研究人员对基于卷积神经网络的目标检测模型已经从两阶段 (例如 Faster RCNN[1] (fast region-based convolutional neural network)、掩码 RCNN[2] (mask fast region-based convolutional neural network) 和 SPP-Net[3](spatial pyramid pooling network)) 发展到单阶段 (例如 SSD[4] (single shot multibox detector)、RetinaNet[5]、YOLO(you can

收稿日期: 2024−06−11.　**网络出版日期**: 2024−12−12.
基金项目: 国家自然科学基金项目(61802124); 中央高校基本科研业务费专项(2023MS137); 中国高校产学研创新基金项目(2023DT6).
通信作者: 王洪涛. E-mail: wanght@ncepu.edu.cn.

only look once)[6]), 并从基于锚的方法 (例如 YOLOv3[7]、YOLOv4[8]) 进一步演化到无锚的方法, 如 CenterNet[9]、FCOS[10] (fully convolutional one-stage object detection)、YOLOX[11]。特别是 YOLO 系列模型, 相比其他模型速度更快, 在边缘设备智能化中受到广泛欢迎。YOLOv8[12] 在性能上相对其前几代模型表现更卓越, 但由于其复杂的结构, 部署在边缘设备上后, 在推理精度以及推理速度上很难达到一个较好的平衡点。因此, 迫切而重要的问题是在边缘设备上实现 YOLOv8 模型的轻量化, 同时保持其高性能表现, 这对于推动边缘设备上目标检测模型的研究和应用至关重要。轻量化的目标检测模型可以有效降低在边缘设备上的计算负担, 提高实时性能。本研究旨在为实际边缘设备的目标检测模型应用提供更可行、高效的解决方案, 推动目标检测在边缘设备智能化领域的深入发展。

目前, 目标检测模型的轻量化主要集中在降低模型的参数量以及计算复杂度上, 如 MobileNet[13]、ShuffleNet[14] 和 GhostNet[15] 等, 利用深度卷积或组卷积提取空间特征, 通过减少浮点运算数 (floating point operations, FLOPs) 来降低模型的复杂度, 但对性能的影响较大。其他的工作对特征融合交互模块进行了改进, 例如 YOLOv5[16] 中的 C3 模块被替换成了 C2f 模块, 通过结构的精炼实现了进一步的轻量化。本文考虑其他改进工作未重视实际推理速度以及边缘设备特性的问题, 基于 YOLOv8 进行了以下工作:

1) 在 Backbone 部分设计 EC2f(efficient CSP-Darknet53 to 2-stage FPN) 模块。EC2f 使用基于部分卷积设计的 PartialBlock 替换 C2f 中的 Bottleneck, 在保证计算复杂度 (参数量) 降低的同时, 不增加数据读写负担。有效解决卷积计算冗余以及替换结构后数据读写量大的问题, 使得模型推理更加迅速。

2) 替换颈部网络。参考 YOLOv6-3.0[17] 版本中的可重参化双向融合 PAN(RepBi-PAN) 颈部 (Neck) 网络, 在新 Neck 中设计 BiC 模块, 引入自底向上的信息流, 使浅层特征能更高效地参与多尺度特征融合。同时结构中的 RepBlock 在推理时将训练时的多分支结构变为单分支结构, 能在硬件上达到高效推理。全新的 Neck 进一步增强了模型融合特征的能力, 也提升了推理速度。

3) 设计全新的多尺度高效检测头。YOLOv8 使用了解耦头以提高性能, 但这也导致了计算复杂度的大幅增加。针对边缘设备以及处理任务的特性, 设计了全新的多尺度高效检测头提升模型的特征提取能力, 合并了原先的分类和回归分支, 有效降低了模型的计算复杂度。

1　相关工作

1.1　YOLOv8

YOLOv8 是由 Ultralytics 公司于 2023 年 1 月 10 日开源的, 目前支持图像分类、目标检测和实例分割任务。作为目前在工业场景中应用最广泛的单阶段检测模型之一, YOLOv8 提供了 5 个不同尺度的模型 (YOLOv8-n、YOLOv8-s、YOLOv8-m、YOLOv8-l、YOLOv8-x), 以满足不同场景的需求。由于 YOLOv8-n 和 YOLOv8-s 相较于其他尺度在推理精度和推理速度上具有更好的平衡, 本文将基于这两个尺度改进模型。

与之前的 YOLO 系列相似, YOLOv8 主要由骨干网络 (Backbone)、颈部网络 (Neck) 和预测头网络 (Head)3 部分组成。在骨干网络的构建方面, 该模型延续了 CSP(cross-stage partial connections) 的核心理念, 参考了 YOLOv7 ELAN[18] (efficient layer aggregation networks) 的设计思想, 并采用 C2f 模块替代了 YOLOv5 中的 C3(csp bottleneck with 3 convolutions) 模块; 同时, 针对不同尺度模型进行了通道和模块数量的调整, 以实现更为轻量化的设计。此外, YOLOv8 还沿用了 YOLOv5 中的 SPPF(spatial pyramid pooling fast) 模块, 以维持模型的性能优势。在颈部网络的设计中, YOLOv8 对 YOLOv5 的 PAN-FPN(path aggregation network-feature pyramid network) 结构进行了优化, 移除了自顶而下的上采样阶段中的卷积操作, 并将 C3 模块替换为 C2f 模块, 进一步提升了网络的效率。在预测头网络部分, YOLOv8 采用了当前流行的解耦头结构, 将分类与检测头分离, 并且从 Anchor-Based 方式转换为 Anchor-Free, 使得模型在预测时更加灵活和准确。

1.2　主流的改进方向

为了在工业场景上应用, 先前的研究人员致力于设计快速的神经网络模型, 许多改进工作都着重于减少计算复杂度 (用 FLOPs 衡量)。例如 MobileNet、ShuffleNet 和 GhostNet 等轻量化神经网络模型。MobileNet 由 Google 团队提出, 深度可分离卷积作为其核心操作, 将常规卷积层的滤波和线性组合解耦为逐通道卷积和逐点卷积两个步骤, 从而大幅降低了计算复杂度。而 ShuffleNet 则由旷视科技提出, 在分组卷积的基础上引入通道混洗机制, 通道混洗是对不同组的输出进行均匀打乱, 实现了不同组之间的信息交互, 提高模型的特征提取能力, 并有效降低了计算复杂度。

GhostNet 是华为诺亚方舟实验室提出的，其改进关键在于引入了幻影卷积，通过常规卷积生成部分本质特征图，然后利用低代价操作生成额外的幻影特征图。幻影卷积通过减少常规卷积生成的特征图数量，采用低代价操作如简单线性变换等，生成更多特征图，有效减少了模型的计算复杂度。

在针对 YOLO 系列模型的改进上。李源鑫等[19]将 YOLOv5s 模型的骨干网络替换为 MobileNetV3，并对特征融合结构进行优化以减少模型参数和计算复杂度，从而提高推理速度。曲英伟等[20]在做如上颈部网络替换工作的同时提出了一种新的非极大值抑制算法来提高重叠目标的识别精度。何宇豪等[21]在 YOLOv5 的主干网络上使用 GhostConv 替换 Conv，GhostBottleneckC3 模块替换 C3 模块以提取丰富特征并优化模型效率，引入加权双向特征金字塔网络（BiFPN[22]）结构进一步提高了对小目标的推理精度。胡丹丹等[23]利用深度可分离卷积替换部分普通卷积，减少模型参数量以提升推理速度，在特征融合网络中引入改进的基于感受野模块模仿人类视觉感知，增加特征图的有效感受野区域。Gupta 等[24]使用 YOLOv6 作为基线模型提出了一种修剪微调算法以及迁移学习算法，以提高模型的推理精度和推理速度。Chen 等[25]提出的改进 YOLOv7 网络模型，通过引入小目标检测层、轻量级卷积和 CBAM（convolutional block attention module）[26]注意力机制，实现多尺度特征提取和融合，减少模型参数量以实现模型推理的提速。邹珺渼[27]等提出了 LW-YOLOv7SAR 模型，通过重参数化、Shuffle 技巧结合 GhostConv 模块去除冗余信息的思想轻量化了模型，同时增强了模型对多尺度信息的提取能力。高德勇等[28]提出了一种针对道路检测的改进 YOLOv8 模型，首先引入多样化分支块结合 C2f 模块构建 C2fDBB 模块，增强特征提取能力。结合路径聚合网络（path aggregation network, PANet）[29]以及渐进特征金字塔网络（asymptotic feature pyramid network, AFPN）[30]思想，提出 PA-AFPN 特征融合方式，提升网络对多尺度特征的融合能力，最后采用新边界回归损失函数加速算法收敛，加速模型推理。

可以看到上述研究工作中的重点在于降低模型的参数量和计算复杂度，目前对于参数量以及计算复杂度的降低已经达到一个瓶颈，因此本文在降低模型参数量及计算复杂度的同时，也重点关注了部分模块的数据读写量以及在实际设备上部署后的推理速度。

2　本文方法

本文设计模型基于 YOLOv8 实现，对于模型的 3 个部分都进行了改进，模型整体结构如图 1 所示。Backbone 部分使用了更为轻量化的 EC2f 结构，Neck 整个替换为 YOLOv6-3.0 版本中的 Neck，Head 部分合并了原先的双支路结构，并设计了全新的多尺度卷积替换了原先的普通卷积，本部分将对各个改进进行详细叙述。

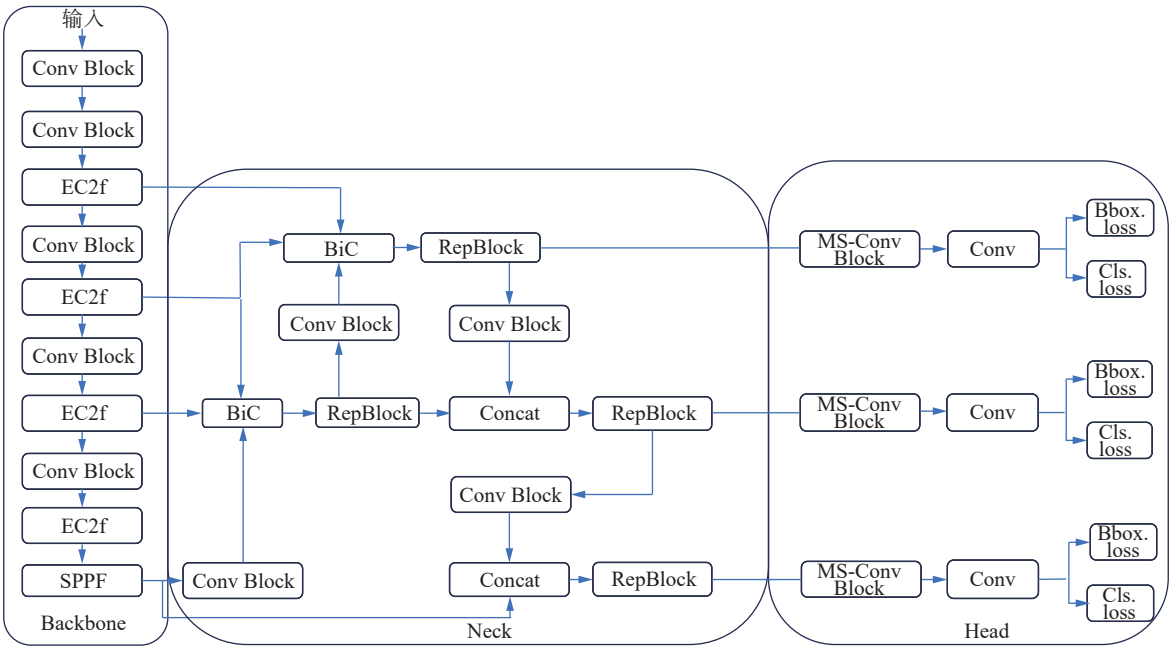


图 1　模型整体结构

Fig. 1　Overall structure of the model

2.1　EC2f 结构设计

在 YOLOv8 模型中，C2f 模块被广泛运用于 Backbone 和 Neck 中，占据了模型相当大的计算复杂度，其模型结构如图 2 所示。其中 Bottleneck 是 C2f 的主要组成部分，完成特征提取融合，由两个卷积块 (Conv Block) 构成，一个卷积块是由卷积，批归一化和激活函数组成的。Bottleneck 结构使用如图 3 所示两个 3×3 卷积块进行特征提取，同时具有残差连接，以确保特征信息不会丢失。

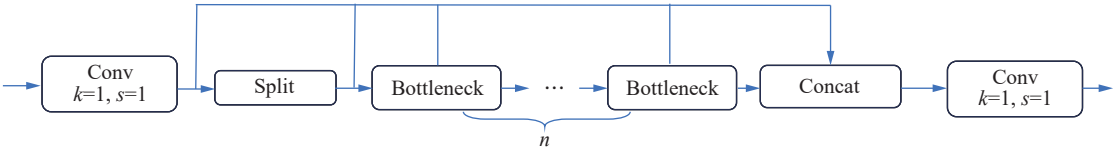


图 2　C2f 结构

Fig. 2　C2f structure

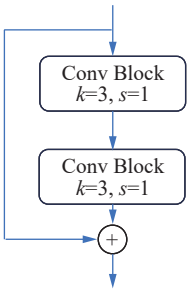


图 3　Bottleneck 结构

Fig. 3　Bottleneck structure

将目标检测模型部署到边缘设备时，模型的推理时长不仅取决于 FLOPs，还受到数据读写量的影响。EfficientNet 等网络的显著特点在于广泛使用低 FLOPs 但数据读写量较大的操作，尤其是 Depthwise 卷积操作。这些高数据读写量的操作，结合 GPU 访存带宽的限制，导致模型在从显存读写数据的过程中浪费了大量时间，导致 GPU 算力得不到充分利用，模型推理速度增加。这里以普通卷积，深度可分离卷积为例，来看一下两个卷积之间计算复杂度和数据读写量的差异。假设读取或写入一个数值便为一次数据读写，一次乘法和一次加法定义为一个浮点运算。将一个 $h\times w\times c_1$ 的特征图进行卷积得到 $h\times w\times c_2$ 的特征图 (h 是特征图的高，w 是特征图的宽，c_1、c_2 是通道数)。

对于卷积核大小为 $k\times k$ 的普通卷积，数据读写次数和总计算复杂度分别为 $h\times w\times c_1+h\times w\times c_2+k^2\times c_1\times c_2$ 和 $h\times w\times c_1\times c_2\times k^2$。

而对于深度可分离卷积，数据读写次数和总计算复杂度分别为 $h\times w\times c_1\times 3+h\times w\times c_2+k^2\times c_1+c_1\times c_2$ 和 $c_1\times k^2\times h\times w+c_1\times c_2\times h\times w$。

在模型中，网络层数越深，通道数越多，特征图的宽高越小，网络越深，相比普通卷积，深度可分离卷积的 FLOPs 更低，但数据读写量明显增加。这也是之前许多改进工作中的问题，FLOPs 的减少不一定会带来类似程度的延迟减少。同时作者发现在 Ghostnet 和 Fasternet[31]等工作中，普通卷积得到的特征图存在很多信息冗余，所以设计的部分卷积如图 4 所示，选取原先通道数的 1/4 进行常规卷积操作，其余的 3/4 通道保持不变，在对特征进一步提取的同时也不丢弃其余 3/4 通道，后续可以通过 1×1 的卷积进行未卷积通道特征信息的提取融合，将部分卷积 FLOPs 降低到常规卷积的 1/16，数据读写量降到常规卷积的 1/4。

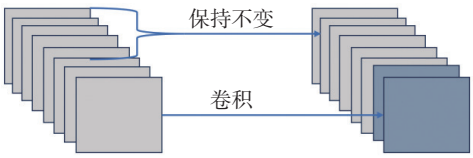


图 4　部分卷积示意

Fig. 4　Schematic diagram of partial convolution

基于部分卷积，本文设计了如图 5 所示的 Partial Block，先使用一个 3×3 的部分卷积对特征进行提取，后跟 1 个 1×1 的卷积块和 1×1 的卷积，可以整合不同通道的特征信息。使用基于部分卷积的 Partial Block 替换了 Bottleneck，得到了全新的 EC2f 模块，在具有较低的 FLOPs 情况下也减少了数据读写量，因此本文的改进在很大程度上加速了模型的推理。

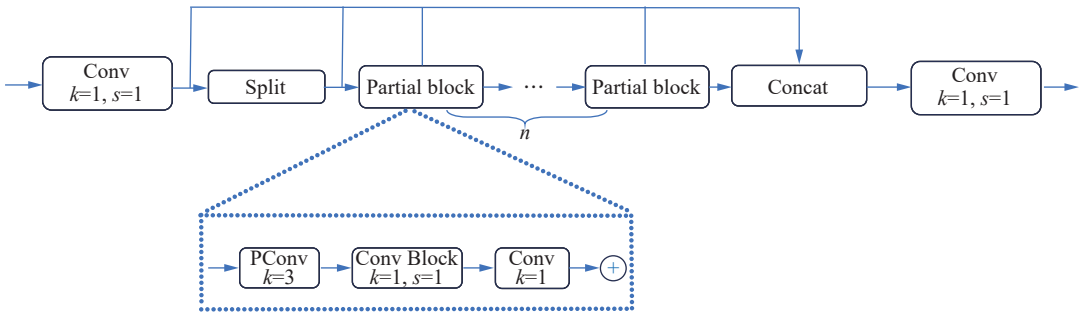


图 5　EC2f 结构

Fig. 5　EC2f structure

2.2　更适合边缘设备的 Neck 替换

YOLOv8 原先采用的 PAFPN 结构中, 使用 C2f 结构作为基础算子, 推理速度上较慢, 为了提升模型推理速度本文选择了 YOLOv6-3.0 版本中的全新 Neck 来进行替换, 结构如图 1 中 Neck 部分所示。新的 Neck 结构在特征提取方面更为高效, 同时所使用的基础算子 RepBlock 能够得到更好的硬件支持, 因此在边缘设备上进行部署后推理能更快。

RepBlock 是基于 RepVGG[32] 风格设计的基础算子, 其在硬件上的推理速度更快。RepBlock 实现了训练与推断过程的解耦, 在训练时保持多分支结构以保证检测模型的性能。而在推理时, 利用结构重参化思想将其等价转换为单分支结构。多分支结构需要保存中间结果, 导致数据读写量增加, 而单分支结构则不存在这个问题。同时, 经过重参化的 RepBlock 由 3×3 卷积构成, 而现有的加速库如英伟达的 cudNN 和相关硬件对 3×3 卷积核进行了良好的性能优化。在 RepVGG 的工作中, 研究人员在相同条件下, 对不同卷积核的计算密度 (理论运算量除以所用时间) 进行了比较, 在 GPU 上, 3×3 卷积的计算密度可达 1×1 和 5×5 卷积的 4 倍。因此推理时, 更换 Neck 的本文模型推理速度更快。

全新的 Neck 中还使用了双向联结 BiC(birectional concatenate) 模块, 如图 6 所示。

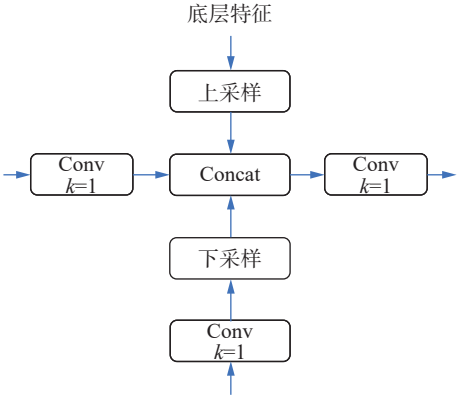


图 6　Bic 结构

Fig. 6　Bicic structure

在自顶而下的特征融合路径中引入自底向上的信息, 使用一个消耗极小的下采样将底层的特征和高层的信息拼接在一起, 再经过一个 1×1 卷积进行特征提取融合, 以更高效的方式参与多尺度特征融合, 进一步增强模型融合特征的能力, 提升模型的推理精度。

2.3　多尺度高效检测头

YOLOv8 为了提升检测性能, 在 Head 部分使用的解耦头, 这导致了模型的计算复杂度大幅上升。因为 Head 中的回归以及分类都是独立分支且参数独立, 在多类别检测的情况下比耦合头有更强的特征学习能力, 但也导致了参数更多, 计算复杂度更大。本文是针对边缘设备的模型设计, 而边缘设备上处理的检测任务有一个特性就是检测类别较为单一, 通常为单类检测或者少类别检测, 因此在 Head 部分使用解耦头中的两条独立支路是有冗余的, 所以本文将 Head 部分的两条分支合并。

如图 7 所示, YOLOv8 的 Head 是双支路结构。一条支路有 1 个 3×3 的卷积块加上 1 个 1×1 的卷积, 这里我们设计了多尺度卷积来提高 Head 部分对不同尺度特征的提取能力, 不同大小的卷积核有不同的感受野, 基于此我们设计了多尺度卷积 MS-Conv(multi-scale convolution), 如图 8 所示。

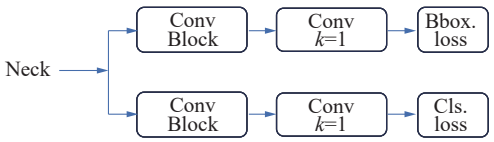


图 7　YOLOv8 Head 结构

Fig. 7　Structure of YOLOv8 head

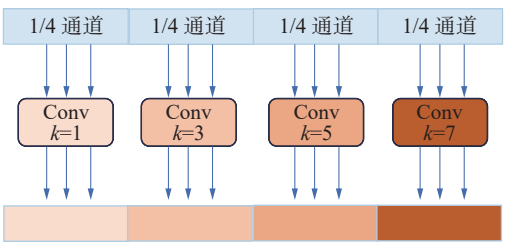


图 8　多尺度卷积示意

Fig. 8　Schematic diagram of multi-scale convolution

在 1/4 通道上使用 1×1 的卷积，1/4 通道上使用 3×3 的卷积，1/4 通道上使用 5×5 的卷积，1/4 通道上使用 7×7 的卷积对不同尺度的特征进行提取，再拼接在一起。

这个全新的卷积数据读写量和计算复杂度分别为 $h\times w\times c+h\times w\times c'+84\times c\times c'$ 和 $5.25\times h\times w\times c\times c'$。其中输入特征图大小为 $h\times w\times c$，输出特征图大小为 $h\times w\times c'$。

一个普通的 3×3 卷积对应的数据读写量和计算复杂度分别为 $h\times w\times c+h\times w\times c'+9\times c\times c'$ 和 $9\times h\times w\times c\times c'$。

可以看出计算复杂度降低了 41%，数据读写量小幅增加。YOLOv8 的检测头是两条支路，本文将检测头中卷积块的 3×3 卷积替换为新的卷积后得到新的多尺度卷积块 MS-Conv Block，再经过 1 个 1×1 的卷积将分通道卷积得到的多尺度特征进行进一步的交互，如图 9 所示。

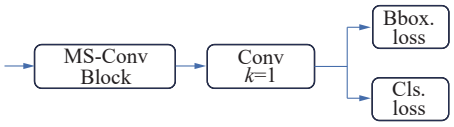


图 9　多尺度检测头示意

Fig. 9　Schematic diagram of multi-scale detection head

该分支替代了原解耦头的两个分支，实现高效多尺度检测头，不必再去重新训练针对分类和回归的两条路径，相比原先的两条支路减少了计算复杂度，加速了模型的推理。

3　实验仿真及结果分析

3.1　数据集

本文为验证模型的泛化性，使用 3 个较有代表性的数据集，均为边缘设备上常见的任务类型，包括特殊类图片任务检测——X 光数据集，密集型目标任务检测——小麦数据集，常规图片检测——安全帽数据集。X 光数据集来源于北航软件开发环境国家重点实验室，已得到授权，其余两个数据集均为开源数据集。数据集划分为按照 7:2:1 划分为训练集、测试集、验证集，具体图片数量如表 1 所示，数据集示例如图 10~12 所示，依次为 X 光数据集、小麦数据集、安全帽数据集。

表 1　数据集划分

Table 1　Dataset partitioning　　张

数据集	训练集	测试集	验证集	总数量
X光数据集	3 317	948	474	4 739
安全帽数据集	900	257	128	1 285
小麦数据集	980	280	140	1 400

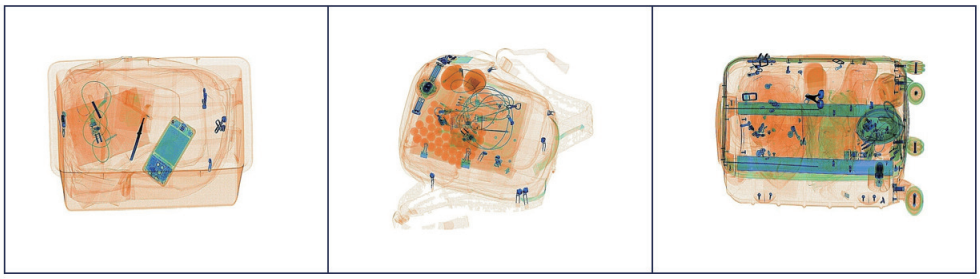


图 10　X 光数据集示意

Fig. 10　Schematic diagram of the X-ray dataset

图 11　小麦数据集示意

Fig. 11　Schematic diagram of the wheat dataset

图 12　安全帽数据集示意

Fig. 12　Schematic diagram of the helmet dataset

3.2　实验环境与参数设置

本实验的训练环境: 操作系统 Ubuntu20.04, PyTorch 版本 1.11.0, CUDA 版本 11.3, 使用 GPU 为 Nvidia 2080Ti。鉴于边缘设备的多样性和异构性, 本研究选择了一个算力较低但扩展性兼容性较好的边缘设备 Jetson Orin Nano (4 GB 版本) 作为实验平台。如图 13 所示, 纵坐标为算力, 用每秒执行的万亿次浮点运算数(tera floating point operations per second, TFLOPS) 来衡量, 横坐标为不同主板型号, 本文使用设备算力仅高于同系列中的两个型号设备。选用该设备作为实验平台的理由在于其适度的计算能力对模型性能提出了更高的挑战, 同时其良好的扩展性和兼容性能够展示本文模型的广泛适用性。该设备足以验证模型在边缘计算场景下的有效性和实用性。

训练过程中, 输入图像分辨率为 640×640, 这个分辨率既保证了模型能够捕捉到足够的图像细节, 又不过度增加计算复杂度。Batch size 为 16, 在不过度消耗计算资源的前提下, 提高训练效率。初始学习率为 0.01, 训练轮次为 300, 为基准模型的默认参数, 未做修改, 便于后续的实验对比。在设备上进行实际推理时, 默认输入图像分辨率为 640×640, 保证图像质量在训练以及推理时的一致性。因为推理设备的性能限制, Batch size 调整为 4。

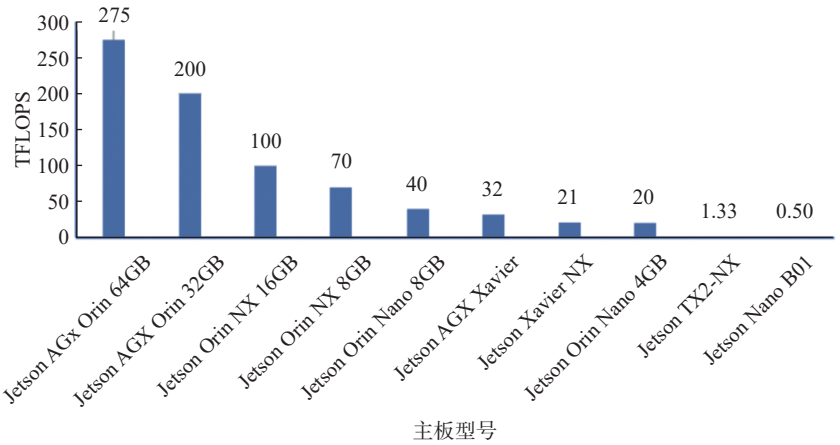


图 13　不同主板型号算力对比

Fig. 13　Comparison chart of computing power across different motherboard models

3.3　实验结果与分析

本研究设计了两种不同尺度的模型: n 尺度和 s 尺度。n 尺度模型旨在实现较低的参数量和计算复杂度, 以便部署于边缘设备; 而 s 尺度模型虽然具有更高的参数量和计算复杂度, 但提供了更优的性能表现。

3.3.1　评价指标

本文的重点在于评估部署后模型的性能, 评价指标定为

1) 模型参数量: 模型参数量是衡量模型复杂度的直接指标, 它反映了模型在训练过程中需要学习的权重和偏置的总数。较多的参数量意味着较大的模型, 部署在边缘设备上会更困难。

2) 模型的计算复杂度: 模型在推理过程中所执行的浮点运算总数。衡量模型对计算资源的需求。

3) 精度: 分为训练精度和推理精度 (采用 mAP50_90 表示交并比阈值在 [0.5, 0.9] 的平均 mAP)。

4) 推理时长: 在边缘设备上推理单张图片所需时间, 这一指标直接体现了模型在实际应用中的响应速度。

5) 帧率: 算法在 1 s 内能够完成目标检测并输出结果的图像数量, 使用 f/s(每秒图像数) 作为单位。

3.3.2　对比实验

在维持实验条件不变的基础上，本文选取了年初最新发布的 YOLOv9 算法以及其他 YOLO 系列作为先进单阶段检测器的代表，同时加入了经典的双阶段检测模型 Faster RCNN 以及另一高效的单阶段检测模型 SSD，以全面对比不同模型在边缘设备上部署后的性能差异。Faster RCNN 的骨干网络使用了 VGG 和 Resnet 两种，SSD 的骨干网络使用了 VGG 和 Mobilenet 两种。对比结果如表 2 所示。

表 2　与其他检测模型的对比结果

Table 2　Comparison results with other detection models

模型	模型计算复杂度/10^9	模型参数/10^6	训练精度/%	推理精度/%	推理时长/ms	帧率/(f/s)
YOLOv5-n	4.5	1.9	80.97	77.08	26.6	37.6
YOLOv5-s	16.5	7.2	85.30	80.90	50.5	19.8
YOLOv6-n	11.4	4.7	87.03	83.40	32.8	30.5
YOLOv6-s	45.3	18.5	88.66	84.50	66.3	15.1
YOLOv7-tiny	13.7	6.2	82.90	79.80	34.5	28.9
YOLOv8-n	8.7	3.2	88.40	84.10	30.8	32.5
YOLOv8-s	28.7	11.1	90.97	85.60	61.6	16.2
YOLOv9-t	7.7	2.0	89.40	88.00	63.4	15.8
YOLOv9-s	26.4	7.1	92.50	88.20	116.6	8.6
Faster-RCNN(VGG)	370.2	137.9	81.17	80.62	442.1	2.3
Faster-RCNN(Resnet)	941.2	28.5	79.24	—	—	—
SSD(VGG)	62.7	26.3	76.86	74.26	96.9	10.3
SSD(Moblienet)	1.8	6.2	73.07	73.18	33.0	30.3
本文模型-n	5.3	2.2	89.10	84.60	27.2	36.8
本文模型-s	20.7	9.0	91.29	87.30	54.7	18.3

从表 2 可以可看出，本文提出的 n 版本模型与 YOLO 系列中同尺度的 v6 至 v8 模型相比，展现出了显著的优势。在推理精度方面，本文模型-n 相较于 YOLOv6-n、YOLOv7-tiny 及 YOLOv8-n 分别实现了 1.2、4.8 及 0.5 百分点提升。同时，在推理时长上相较于上述 3 个模型分别降低了 17.0%、21.2% 及 11.6%，有效缩短了推理时间。尽管 YOLOv9-t 在推理精度上与本文 n 版本模型相比，提升了 3.4 百分点，但这一优势是以牺牲推理时长为代价的，推理时长增加了一倍多。相比之下，本文模型的两个尺度在推理精度与推理时长之间实现了更优的平衡。本文 n 版本模型在模型计算复杂度上仅比 YOLOv5-n 多出 0.8×10^9，低于其他同尺度模型，在模型参数量上也只比 YOLO 系列中的 v5 和 v9 同尺度模型高出 0.3×10^6 和 0.2×10^6，有效控制了模型规模，有利于在资源受限的环境下部署。在帧率表现上，本文 n 版本模型达到了 36.8 f/s，较于 v6、v7、v8、v9 同尺度模型提升了 20.6%、27.3%、13.2%、132.7%。本文 s 版本模型与 YOLO 系列中同尺度模型对比，同样有显著的优势，在推理精度上比 YOLOv5-s、YOLOv6-s、YOLOv8-s 提升了 7.9、3.3、1.9 百分点，推理时长上比 YOLOv5-s 增加了 8.3%，在其他两个模型上降低了 17.4%、11.2%。YOLOv9-s 和本文模型-s 相比推理精度提升了 1 百分点，但是推理时间增加了 113.1%。在帧率方面，相较于 YOLOv6-s、YOLOv8-s 以及 YOLOv9-s，本文模型分别实现了 21.1%、12.9% 以及显著提升的 112.7% 的增幅。可以看出本文模型在推理精度与推理速度之间找到了更佳的平衡点，既保证了高准确率，又维持了高效的推理速度。

在和双阶段模型 Faster-RCNN 对比中，本文模型计算复杂度和参数量远低于 Faster-RCNN，同时 s 版本在推理时长上比 Faster-RCNN(VGG) 降低了 87.6%。推理精度提升了 12.4 百分点。Faster-RCNN(Resnet) 由于过大的参数量和模型复杂度无法在设备上部署进行推理。与另一流行的单阶段检测模型 SSD 相比，本文 s 版本模型相较于 SSD (VGG) 在推理速度上降低了 43.5%，推理精度上提升了 17.5 百分点。本文 n 版本模型较于 SSD(Moblienet)，在推理速度上降低了 21.3%，推理精度上提升了 15.6 百分点。实验结果表明，本文提出的模型在边缘设备上的表现显著优于其他主流目标检测模型，有着更优异的推理精度，更短的推理时长。

3.3.3　模块改进及消融实验

我们单独对 s 版本改进前和改进后的模块进

行分析, 如表 3 所示, 设计的 EC2f 相较于 C2f 减少了 52.54% 的计算复杂度和 75.31% 的参数量。全新的多尺度高效检测头相比原检测头减少了 83.5% 的计算复杂度和 67.8% 的参数量。

表 3　模块改进前后数据对比

Table 3　Comparison of data before and after module improvement

模块	改进前计算复杂度/10^9	改进前参数量/10^6	改进后计算复杂度/10^9	改进后参数量/10^6
C2f	8.05	5.71	3.82	1.41
Head	8.32	2.15	1.37	0.69

为了对比在改进中使用的不同卷积的性能差异, 假设输入图像大小为$h\times w\times c_1$(依次为特征图的高、宽、通道数), 经过卷积的输出图像大小为$h\times w\times c_2$, 计算得到了如表 4 所示的数据, 可以看出在计算复杂度上, 本文使用部分卷积的计算复杂度是普通卷积的 1/16, 多尺度卷积的计算复杂度高于普通卷积。普通卷积的数据读写量是部分卷积的 4 倍, 多尺度卷积的 1.71 倍。

表 4　不同卷积对比

Table 4　Comparison of different convolution types

模块	计算复杂度	数据读写量
普通卷积	$h\times w\times c_1+h\times w\times c_2+9\times c_1\times c_2$	$9\times h\times w\times c_1\times c_2$
部分卷积	$1/16\ (h\times w\times c_1+h\times w\times c_2+9\times c_1\times c_2)$	$2.25\times h\times w\times c_1\times c_2$
多尺度卷积	$h\times w\times c_1+h\times w\times c_2+84\times c_1\times c_2$	$5.25\times h\times w\times c_1\times c_2$

为了验证模型各个改进的有效性, 实验侧重于考察在降低参数量与计算复杂度的同时, 推理时长是否也得到了相应的减少做消融实验, 实验对比如表 5 所示。从整体来看模型的计算复杂度降低了 27.8%, 参数量降低了 18.73%, 推理时长降低了 11.2%。具体于 3 个改进的模块来说, 增加之后, 推理时长均会降低, 效果最为显著的即为 EC2f, 同时可以看到替换 Neck 之后模型计算复杂度以及参数量小幅上升, 但是推理时长并未增加, 这也验证 2.1 节所说的模型的推理时长不仅体现在计算复杂度上, 也受到数据读写量的影响。

表 5　消融实验数据对比

Table 5　Result after the addition of each module

模型变化	模型计算复杂度/10^9	模型参数量/10^6	训练精度/%	推理精度/%	推理时长/ms
Baseline	28.7	11.14	90.9	85.6	61.6
+EC2f	21.6	8.32	88.5	84.1	56.1
+ Neck + EC2f	22.5	9.15	90.4	85.4	55.3
+ New head+ Neck + EC2f	20.7	9.05	91.2	87.3	54.7

3.3.4　泛化性能实验

为评估模型的泛化能力, 将优化后的 n 版本和 s 版本模型在 3 个不同数据集上与基准模型 YOLOv8 进行比较, 重点分析推理精度与推理时长, 结果如表 6、表 7 所示。n 版本在 X 光、小麦、安全帽数据集上的推理时长分别减少了 11.7%、14.3% 和 9.8%, 而 s 版本则分别减少了 11.2%、18.6% 和 13.4%。在推理精度方面, X 光数据集略有提升, 而在小麦和安全帽数据集中基本保持稳定, 精度最大降幅为 1.3 百分点(n 版本在安全帽数据集上)。鉴于推理时长的显著缩短, 笔者认为这种程度的精度降低是可以接受的。

表 6　n 版本模型下不同数据集结果

Table 6　n version of the model results of different datasets

数据集	基准模型-n版本		本文模型-n版本	
	推理精度/%	推理时长/ms	推理精度/%	推理时长/ms
X光数据集	84.1	30.8	84.6	27.2
小麦数据集	92.0	39.2	91.5	33.6
安全帽数据集	88.9	38.6	87.6	34.8

表 7　s 版本模型下不同数据集结果

Table 7　s version of the model results of different datasets

数据集	基准模型-s版本		本文模型-s版本模型	
	推理精度/%	推理时长/ms	推理精度/%	推理时长/ms
X光数据集	85.6	61.6	87.3	54.7
小麦数据集	92.6	72.1	92.7	58.7
安全帽数据集	90.5	69.4	89.3	60.1

4　结束语

本文考虑了边缘设备的任务特性以及边缘设备的特点，针对之前目标检测的模型主流改进方向存在的问题，基于 YOLOv8 引入了一系列改进措施。首先将 C2f 中的 Bottleneck 替换为新设计的 Partial Block，在保证参数量计算复杂度降低的同时降低数据读写量。接着，替换了更受硬件支持且提取特征能力更强的 YOLOv6-3.0 的 Neck，保证模型的检测性能，也提升了模型在边缘设备上的推理速度。最后，使用设计的全新多尺度高效检测头替换原解耦头，进一步减低了模型的推理时长。相较于目前其他 YOLO 系列模型，本文模型在实际设备上的推理速度以及推理精度的平衡更好，为目标检测模型在边缘设备上应用提供了切实可行的高效解决方案。后续工作重点在于通过模型蒸馏，进一步轻量化模型，不再局限于对模块的结构改进，通过模型蒸馏，可以将大型教师模型的知识转移到一个更小的学生模型中，实现在较低资源下的高效检测。

参考文献：

[1] 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.

[2] HE Kaiming, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]//2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017: 2980−2988.

[3] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904−1916.

[4] LIU Wei, ANGUELOV D, ERHAN D, et al. SSD: single shot MultiBox detector[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016: 21−37.

[5] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[C]//2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017: 2999−3007.

[6] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016: 779−788.

[7] REDMON J, FARHADI A. YOLOv3: an incremental improvement[EB/OL]. (2018−04−08)[2024−06−11]. https://arxiv.org/abs/1804.02767v1.

[8] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: optimal speed and accuracy of object detection[EB/OL]. (2020−04−23)[2024−06−11]. https://arxiv.org/abs/2004.10934v1.

[9] DUAN Kaiwen, BAI Song, XIE Lingxi, et al. CenterNet: keypoint triplets for object detection[C]//2019 IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019: 6568−6577.

[10] TIAN Zhi, CHU Xiangxiang, WANG Xiaoming, et al. Fully convolutional one-stage 3d object detection on lidar range images[J]. Advances in neural information processing systems, 2022, 35: 34899−34911.

[11] GE Zheng, LIU Songtao, WANG Feng, et al. Yolox: exceeding yolo series in 2021[EB/OL]. (2021−07−18)[2024−06−11]. https://arxiv.org/abs/2107.08430.

[12] JOCHER G, CHAURASIA A, QIU J. Ultralytics YOLO (Version 8.0. 0). (2023−01−15)[2024−06−11]. http://github.com/ultralytics/ultralytics.

[13] HOWARD A G, ZHU Menglong, CHEN Bo, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017−04−17)[2024−06−11]. https://arxiv.org/abs/1704.04861v1.

[14] ZHANG Xiangyu, ZHOU Xinyu, LIN Mengxiao, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 6848−6856.

[15] HAN Kai, WANG Yunhe, TIAN Qi, et al. GhostNet: more features from cheap operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020: 1577−1586.

[16] JOCHER G. YOLOv5 by Ultralytics (Version 7.0). [2024−06−11]. https://doi.org/10.5281/zenodo.3908559.

[17] LI Chuyi, LI Lulu, GENG Yifei, et al. YOLOv6 v3.0: a

full-scale reloading[EB/OL]. (2023−01−13)[2024−06−11]. https://arxiv.org/abs/2301.05586v1.

[18] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver: IEEE, 2023: 7464−7475.

[19] 李源鑫, 郭忠峰, 杨钧麟. 基于轻量化 YOLOv5s 的集装箱锁孔识别算法[J]. 计算机科学, 2024, 51(S1): 524−529.
LI Yuanxin, GUO Zhongfeng, YANG Junlin. Container keyhole identification algorithm based on lightweight YOLOv5s[J]. Computer science, 2024, 51(S1): 524−529.

[20] 曲英伟, 刘锐. 基于 YOLOv5-MobileNetV3 算法的目标检测[J]. 计算机系统应用, 2024, 33(7): 213−221.
QU Yingwei, LIU Rui. Object detection based on YOLOv5-MobileNetV3 algorithm[J]. Computer systems and applications, 2024, 33(7): 213−221.

[21] 何宇豪, 易明发, 周先存, 等. 基于改进的 Yolov5 的无人机图像小目标检测[J]. 智能系统学报, 2024, 19(3): 635−645.
HE Yuhao, YI Mingfa, ZHOU Xiancun, et al. UAV image small-target detection based on improved Yolov5[J]. CAAI transactions on intelligent systems, 2024, 19(3): 635−645.

[22] TAN Mingxing, PANG Ruoming, LE Q V. EfficientDet: scalable and efficient object detection[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020: 10778−10787.

[23] 胡丹丹, 张忠婷. 基于改进 YOLOv5s 的面向自动驾驶场景的道路目标检测算法[J]. 智能系统学报, 2024, 19(3): 653−660.
HU Dandan, ZHANG Zhongting. Road target detection algorithm for autonomous driving scenarios based on improved YOLOv5s[J]. CAAI transactions on intelligent systems, 2024, 19(3): 653−660.

[24] GUPTA C, GILL N S, GULIA P, et al. A novel fine-tuned YOLOv6 transfer learning model for real-time object detection[J]. Journal of real-time image processing, 2023, 20(3): 42.

[25] CHEN Junyang, LIU Hui, ZHANG Yating, et al. A multiscale lightweight and efficient model based on YOLOv7: applied to citrus orchard[J]. Plants, 2022, 11(23): 3260.

[26] WOO S, PARK J, LEE J Y, et al. CBAM: convolutional block attention module[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 3−19.

[27] 邹珺渼, 任酉贵, 冷芳玲, 等. LW-YOLOv7SAR: 轻量 SAR 图像目标检测方法[J/OL]. 小型微型计算机系统, 1−9. [2024−06−15]. https://www.cnki.com.cn/Article/CJFDTotal-XXWX20231103009.htm.
ZOU Junhao, REN Yougui, LENG Fangling, et al. LW-YOLOv7SAR: Lightweight SAR image object detection method [J/OL]. Journal of Small Computer Systems, 1−9. [2024−06−15]. https://www.cnki.com.cn/Article/CJFDTotal-XXWX20231103009.htm.

[28] 高德勇, 陈泰达, 缪兰. 改进 YOLOv8n 的道路目标检测算法[J]. 计算机工程与应用, 2024, 60(16): 186−197.
GAO Deyong, CHEN Taida, MIAO Lan. Improved road object detection algorithm for YOLOv8n[J]. Computer engineering and applications, 2024, 60(16): 186−197.

[29] LIU Shu, QI Lu, QIN Haifang, et al. Path aggregation network for instance segmentation[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 8759−8768.

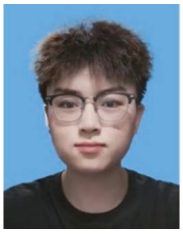
[30] YANG Guoyu, LEI Jie, ZHU Zhikuan, et al. AFPN: asymptotic feature pyramid network for object detection[C]//2023 IEEE International Conference on Systems, Man, and Cybernetics. Honolulu: IEEE, 2023: 2184−2189.

[31] CHEN Jierun, KAO S H, HE Hao, et al. Run, don't walk: chasing higher FLOPS for faster neural networks[C]//2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver: IEEE, 2023: 12021−12031.

[32] DING Xiaohan, ZHANG Xiangyu, MA Ningning, et al. RepVGG: making VGG-style ConvNets great again[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021: 13728−13737.

作者简介：

徐伟峰，讲师，博士，主要研究方向为图像识别技术、形式化验证方法和低空空管系统，承担科研项目 10 项。E-mail: weifengxu@163.com。



雷耀，硕士研究生，主要研究方向为深度学习和目标检测，发表学术论文 1 篇。E-mail: 2260140046@qq.com。

王洪涛，副教授，博士，中国计算机学会会员，主要研究方向为人工智能安全、自然语言处理、隐私计算和知识计算。主持国家自然科学基金项目 1 项、中央高校基本科研业务费专项 2 项。发表学术论文 28 篇。E-mail: wanght@ncepu.edu.cn。