计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(9):133−140 [doi: 10.15888/j.cnki.csa.009990] [CSTR: 32024.14.csa.009990]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

FSLW-YOLOv8n: 基于改进 YOLOv8n 的轻量化汽车密封圈缺陷检测①

李文辰, 曾映海, 葛　健, 叶子渊, 秦　琴

(上海第二工业大学 智能制造与控制工程学院, 上海 201209)
通信作者: 曾映海, E-mail: zengyinghai199812@163.com

摘　要: 深度学习算法在汽车密封圈缺陷检测中展示出巨大潜力, 但是依然面临着模型复杂、部署困难的问题, 因此本文提出一种基于改进 YOLOv8n 的轻量化汽车密封圈缺陷检测算法 FSLW-YOLOv8n. 首先, 优化 C2f 模块中的 Bottleneck 结构, 引入 Faster block 提升内存访问效率并增强特征提取能力. 同时, 颈部网络采用 GSConv 与 Slim-neck 的设计理念, 显著减少了参数量, 实现模型轻量化. 此外, 使用轻量级的非对称解耦头 LADH-Head, 在提升检测精度的同时进一步精简模型结构. 然后, 引入 Wise-IoU 损失函数, 通过精细化的小目标定位策略, 提升整体检测性能. 最后将改进的算法经过模型转换部署到海思平台, 并进行模型的实际性能验证. 实验结果表明, 与基线模型相比, *mAP* 提升了 2.1%, 计算量、参数量和模型大小分别下降了 55.6%、42.7% 和 38.3%. 在海思 SD3403 嵌入式平台上, 检测速度达到了 31.3 f/s.

关键词: YOLOv8n; 汽车密封圈; 缺陷检测; 轻量化; Slim-neck; 边缘部署

引用格式: 李文辰,曾映海,葛健,叶子渊,秦琴.FSLW-YOLOv8n: 基于改进 YOLOv8n 的轻量化汽车密封圈缺陷检测.计算机系统应用,2025,34(9): 133–140. http://www.c-s-a.org.cn/1003-3254/9990.html

FSLW-YOLOv8n: Lightweight Automotive Seal Defect Detection Based on Improved YOLOv8n

LI Wen-Chen, ZENG Ying-Hai, GE Jian, YE Zi-Yuan, QIN Qin

(School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai 201209, China)

Abstract: Deep learning algorithms have shown great potential in automotive seal defect detection, but challenges remain, such as model complexity and deployment difficulties. Therefore, FSLW-YOLOv8n, a lightweight algorithm for automotive seal defect detection based on an improved YOLOv8n, is proposed in this paper. First, the Bottleneck structure in the C2f module is optimized by introducing the Faster block, which improves memory access efficiency and feature extraction capabilities. Meanwhile, the neck network adopts the design concepts of GSConv and Slim-neck, significantly reducing the parameter count to achieve model lightweight. Additionally, LADH-Head, a lightweight asymmetric decouple head, is used to further streamline the model structure while improving detection accuracy. Then, the Wise-IoU loss function is introduced, enhancing overall detection performance by a refined small-object localization strategy. Finally, the improved algorithm is converted and deployed on the HiSilicon platform, followed by performance validation. Experimental results show that, compared to the baseline model, *mAP* has increased by 2.1%, while calculation amount, parameter count, and model size have decreased by 55.6%, 42.7%, and 38.3%, respectively. On the HiSilicon SD3403 embedded platform, the detection speed reaches 31.3 f/s.

Key words: YOLOv8n; automotive seal; defect detection; lightweight; Slim-neck; edge deployment

① 收稿时间: 2025-01-10; 修改时间: 2025-03-14, 2025-05-19; 采用时间: 2025-05-23; csa 在线出版时间: 2025-07-25
CNKI 网络首发时间: 2025-07-28

近年来, 随着汽车技术的不断进步, 尤其是智能化和自动驾驶技术的快速发展, 车辆安全面临着前所未有的挑战[1]. 作为保障汽车安全与性能的核心, 电子控制单元 (ECU) 的正常运行依赖于多个关键部件的协同工作, 而其中一个常被忽视但极为重要的组件便是汽车密封圈. 密封圈的质量直接影响 ECU 的寿命与可靠性. 然而, 在制造过程中, 密封圈可能因材料缺陷或工艺偏差出现微小问题, 这些问题会引发严重的安全隐患. 因此, 为了确保汽车制造的高标准, 开发出能够精准检测这些微小缺陷的技术对保障零部件的质量和可靠性至关重要. 在当前生产环境中, 汽车密封圈的缺陷检测主要依赖于传统人工检测, 但是这种检测方法耗时长、效率低、成本高且难于保证检测精度.

随着深度学习技术的快速发展, 该技术在各个行业的缺陷检测中得到广泛应用. 基于深度学习的检测算法大致分为双阶段检测算法和单阶段检测算法. 双阶段检测算法通常检测精度高, 但推理速度较慢, 需要更多的计算资源, 如 R-CNN[2]、Fast R-CNN[3]与 Faster R-CNN[4]等. 而单阶段算法往往具有更快的推理速度, 并且经过不断迭代优化, 其检测精度甚至超过双阶段算法, 如 SSD[5]、YOLO[6]系列算法等. 针对航天密封圈人工检测效率低、传统图像检测通用性差的问题, 陶晓天等人[7]以 RetinaNet 网络为基础, 提出了 MoGaA-RetinaNet 算法, 对航天密封圈表面缺陷具有良好的检测效果, 但是该算法参数量大, 不利于边缘设备部署. 朱文博等人[8]提出一种改进 YOLOv5 的 O 型密封圈缺陷检测方法, 通过引入双向特征金字塔与注意力机制来提升检测精度. 张相胜等人[9]引入 PConv 和全局注意力机制改进 YOLOv7, 提升算法的特征提取与融合能力, 进一步提升了航天密封圈表面缺陷的检测精度.

然而现有的密封圈表面缺陷检测算法普遍存在模型复杂、部署困难的问题, 难以满足汽车密封圈在工业现场的检测需求. 对此, 本文基于 YOLOv8[10]网络进行改进, 提出了一种轻量化的汽车密封圈缺陷检测算法 (FSLW-YOLOv8n), 然后将改进后的 YOLOv8 模型部署到海思 SD3403 嵌入式平台进行实际缺陷检测效果验证. 本文的主要贡献为: 1) 在 YOLOv8 骨干网络中引入 FasterNet[11]中的 Faster block 来替换原始网络 C2f 中的 Bottleneck, 避免模型学习特征冗余, 优化内存访问; 2) 基于 Slim-neck[12]设计范式改进颈部网络, 实现网络轻量化; 3) 使用非对称解耦检测头 (LADH-Head)[13], 利用非对称多级压缩技术提升检测性能, 减少解耦头的数量, 精简模型结构; 4) 采用 Wise-IoU[14]作为边界框回归损失函数, 增强小目标定位能力, 加快网络收敛, 提升检测精度; 5) 将改进后的网络模型移植到海思 SD3403 嵌入式平台进行模型性能验证, 识别准确率提升并达到了 31.3 f/s 的检测速度.

1 YOLOv8 算法

YOLOv8 是一种高效的单阶段目标检测算法, 由 YOLOv5 改进得到, 根据网络深度可分为 n、s、m、l 和 x 这 5 个版本. YOLOv8 主要由输入层 (Input)、骨干网络 (Backbone)、颈部网络 (Neck) 和检测头 (Head) 这 4 部分组成. 其中, 输入层自适应图片缩放, 并结合 Mosaic 数据增强[15]来提升训练效果. 骨干网络主要由 Conv、C2f 和 SPPF 等模块组成, 通过卷积、池化等方式提取图片特征. 颈部网络采用 FPN-PAN 的结构设计, 对浅层特征信息和深层特征信息进行融合, 获取预测特征图. 在检测头网络中, 使用无锚节点 (anchor-free) 替换 YOLOv5 中的有锚节点 (anchor-base), 减少锚框数量, 同时使用解耦头结构, 将分类头和检测头分开, 提升检测精度.

2 FSLW-YOLOv8n 算法

在 YOLOv8 系列算法中, YOLOv8n 是其中最轻量化的模型, 在保持检测精度的同时, 有效控制了模型的参数规模. 但是在汽车密封圈缺陷检测算法部署的过程中, 仍然存在参数多、模型复杂、对小目标不敏感等问题. 为了解决上述问题, 本文基于 YOLOv8n 模型进行了改进, 提出了 FSLW-YOLOv8 算法, 其网络结构图如图 1 所示. 在骨干网络中引入 C2f-Faster 模块, 以减少模型参数量, 优化内存访问. 在颈部网络中, 使用 PConv 模块替换标准卷积, 使用 VoVGSCSPC 模块替换 Neck 层中的 C2f 模块, 在保证检测精度的同时降低模型复杂度. 在检测头网络中, 引入非对称解耦检测头 (LADH-Head), 进一步减少模型计算量. 最后采用 Wise-IoU 损失函数, 提升模型整体检测性能.

2.1 C2f-Faster 特征提取模块

汽车密封圈尺寸小, 只有 1.3×0.5 cm^2, 其表面缺陷更加难以定位. 且由于工业现场环境复杂, 会有许多噪声干扰, 在缺陷提取过程中往往会获取大量无关信息. 随着网络深度的增加, 更多的通道将附带这些冗余信

息, 导致检测精度下降. 对此, Chen 等人[11]提出了一种基于部分卷积 (partial convolution, PConv) 的 FasterNet 轻量网络. 与标准卷积不同, PConv 仅在输入特征图的部分连续通道上执行标准卷积, 然后使用滤波器进行特征提取. 这种方式能够显著减少冗余计算和内存访问, 并有效提升特征提取能力.

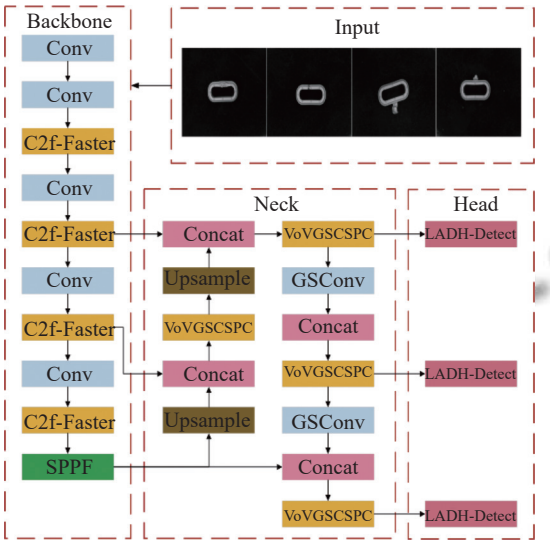


图 1 FSLW-YOLOv8n 网络结构图

标准卷积与 PConv 原理如图 2 所示. PConv 使用了 c_p 数量的通道来进行卷积操作, 其他通道不变, 然后将卷积过的特征与未卷积的特征进行拼接.

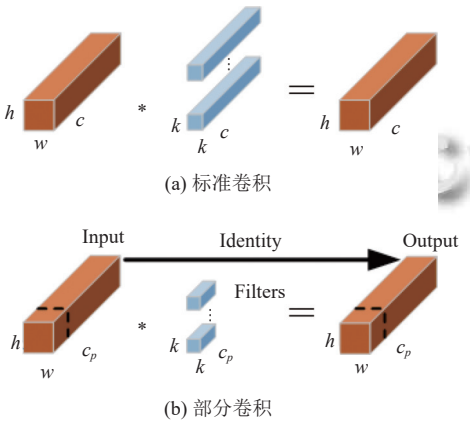


图 2 标准卷积与部分卷积结构图

PConv 的浮点运算量 *FLOPS* 如式 (1) 所示, 内存访问量 *MEM* 如式 (2) 所示:

$$FLOPS = hwk^2c_p^2 \tag{1}$$

$$MEM = 2hwc_p + k^2c_p^2 \approx 2hwc_p \tag{2}$$

其中, h、w 分别为特征图的高和宽; k 为卷积核尺寸; c_p 为 PConv 作用的通道数; *FLOPS* 为浮点运算量; *MEM* 为内存访问量. 部分卷积通道数 c_p 仅为标准卷积通道数 c 的 1/4, 进而 PConv 的运算量是原来的 1/16, 内存访问量为原来的 1/4, 这为在有限资源下的边缘设备部署算法提供了有力支持, 同时也大大提高了推理速度.

为了减少网络中的冗余信息, 减少运算量, 本文使用 PConv, 设计出 Faster block 模块, 其结构如图 3(a) 所示. 在 Bottleneck 的分支上引入一个 3×3 的 PConv, 并在 PConv 与 Conv 之间建立一个残差连接, 在中间 Conv 层加入了 BN (batch normalization) 层和 ReLU 激活函数, 进一步加速推理. 图 3(b) 为本文的新结构 C2f-Faster, 使用 Faster block 模块替换 C2f 中的 Bottleneck, 增强在汽车密封圈缺陷检测过程中的特征提取能力并降低模型复杂度.

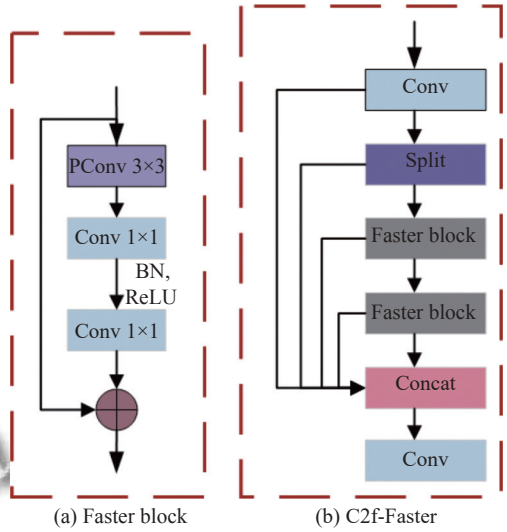


图 3 Faster block 与 C2f-Faster 结构图

2.2 Slim-neck 设计范式

在汽车密封圈缺陷检测的边缘计算中, 移动端算力及存储受硬件资源限制, 因此在算法设计中要充分考虑网络轻量化. 在 YOLOv8n 中, 使用标准卷积进行特征提取, 虽然保留了丰富的特征信息, 但是随着网络层数的增加, 参数量显著增加. 为解决这一问题, 本文引入 GSConv 模块和 Slim-neck 设计范式, 在减少模型计算量的同时保持检测精度. 如图 4(a) 所示, GSConv 将标准卷积与深度可分离卷积 (depthwise separable convolution, DSConv)[16]进行拼接, 通过 Shuffle 混合操作, 将标准卷积得到的信息融合到深度可分离卷积的

输出中. 将 Bottleneck 中的标准卷积替换为 GSConv, 并在其分支上添加一个 DSConv, 与 GSConv 进行拼接, 增强网络特征处理能力, 组成 GSbottleneck 模块, 其结构如图 4(b) 所示. 结合 GSbottleneck, 采用一次性聚合方式构建了 VoVGSCSPC 模块, 用于替换颈部网络中的 C2f, 结构如图 4(c) 所示.

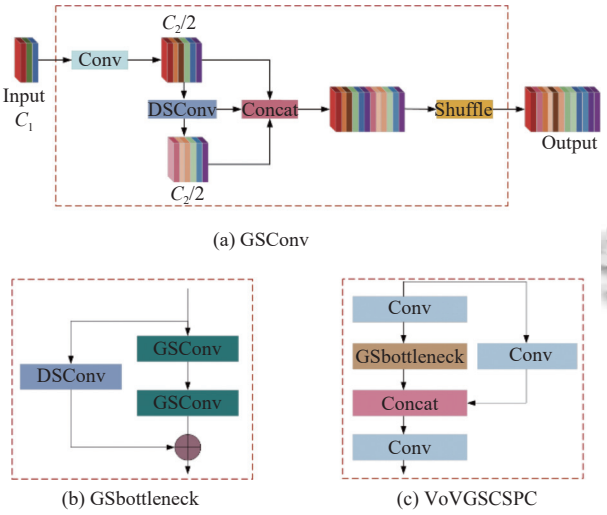


图 4 GSConv、GSbottleneck 和 VoVGSCSPC 的结构图

2.3 轻量级检测头 LADH-Head

YOLOv8n 的解耦头将分类头与检测头分离, 这种设计虽然提升了模型的检测能力, 但是却大大增加了参数量, 降低了推理速度, 不利于边缘设备部署. 为了解决这个问题, 本文引入一种轻量级的非对称检测头 (LADH-Head), 结构如图 5 所示. 在 LADH-Head 中, 根据任务类型划分网络, 使用 3 个独立通道来执行相应的任务. 为了扩大感受野并增加 IoU 分支的任务参数, 在每个分支中, 通过 3 次卷积操作来减少通道维度特征, 并使用 3 个 3×3 深度可分离卷积代替标准 3×3 卷积提升计算效率.

DSConv 由深度卷积和逐点卷积组成, 先在特征维度上对每个通道进行独立的 3×3 深度卷积, 在输出前使用一个 1×1 的点卷积将所有通道的特征聚合起来, 实现通道间的信息融合. 引入 DSConv 来处理分类和边界框任务, 通过精确的特征提取和信息融合, 提升了模型对正样本的预测精度, 避免了任务之间的相互干扰. 将 YOLOv8n 中的原解耦头替换为 LADH-Head, 不仅在保证精度的前提下显著降低了模型的复杂度, 还提升了检测性能.

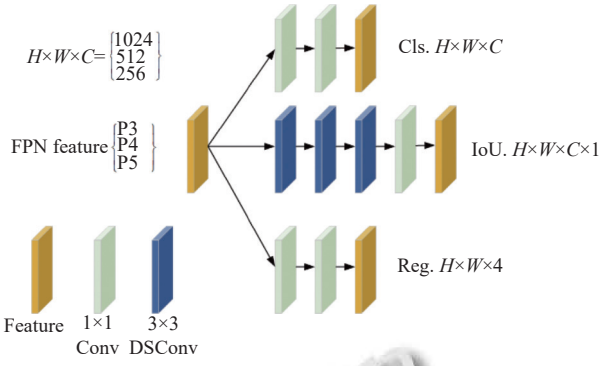


图 5 LADH-Head 结构图

2.4 Wise-IoU 损失函数

在目标检测算法中, 损失函数的选择对模型训练效果有重要的影响. YOLOv8n 采用了 CIoU 作为损失函数, 如式 (3) 所示. CIoU 综合了长宽比、中心距离等几何因素, 对低质量样本的惩罚力度较大, 导致检测精度降低及模型泛化能力减弱. 此外, 与传统的 IoU 相比, CIoU 计算复杂度更高, 网络收敛慢, 训练期间会占用更多的资源.

$$L_{\mathrm{CIoU}}=1-IoU+\frac{\rho^2(b,b_{\mathrm{gt}})}{d^2}+\alpha v \tag{3}$$

其中, IoU 是预测边界框与真实边界框的交并比; b 为预测框的中心点; b_{gt} 为真实框的中心点; ρ 为两个中心点之间的欧氏距离; d 为两个矩形框的最小外接矩形的对角线长度; α 为平衡参数; v 用于测量两个矩形框高宽比的一致性.

为增强模型的泛化能力, 加速网络推理, 减小低质量样本对模型的负面影响, 本文采用 Wise-IoU (WIoU) 作为边界回归损失函数, 其计算方式如式 (4)–式 (6) 所示. WIoU 结合了注意力机制和动态非单调聚焦机制, 通过使用离群值来评估锚框质量. 当离群值较大时, 意味着锚框质量较差, 这时会为其分配较小的梯度增益. 这种梯度增益分配策略不仅减少了高质量锚框之间的竞争, 还降低了低质量样本带来的不利梯度影响, 使得 WIoU 能够更加专注于普通质量的锚框, 提高模型的泛化能力, 增强小目标的定位能力, 提升整体检测性能.

$$L_{\mathrm{WIoU}}=r\cdot\exp\left(\frac{(x-x_{\mathrm{gt}})^2+(y-y_{\mathrm{gt}})^2}{(H_g^2+W_g^2)^*}\right)\cdot(1-IoU) \tag{4}$$

$$r=\frac{\beta}{\delta\alpha^{\beta-\delta}} \tag{5}$$

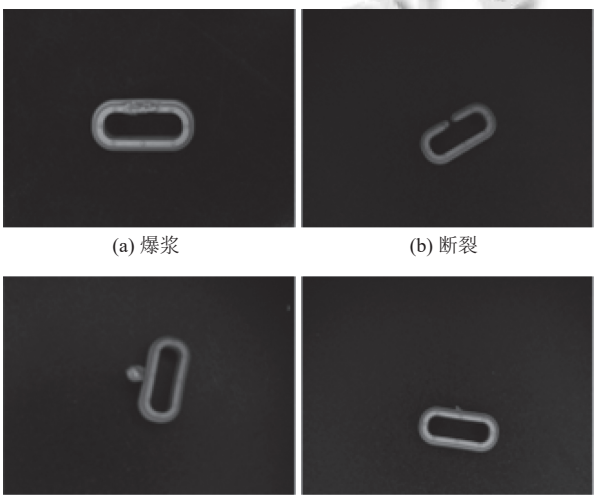
$$\beta=\frac{(1-IoU)^*}{1-\overline{IoU}}\in[0,+\infty] \tag{6}$$

其中, x, y、x_{gt}, y_{gt}分别为预测框与真实框中心点的横纵坐标; H_g和W_g分别为预测框与真实框的最小外接矩形框的高和宽; *为分离操作, r 为非单调聚焦系数; δ为学习参数; β为离群值; $\overline{IoU}$为预测边界框与真实边界框交并比的平均值.

3 实验结果与分析

3.1 数据集准备

本文实验采用的汽车密封圈数据集, 共计 4 958 张图片, 包含 4 种缺陷类型: 爆浆 (1 258 张)、断裂 (1 438 张)、多料 (1 085 张) 与毛刺 (1 177 张), 图 6 展示了各类缺陷的具体样例. 为了保证模型训练的科学性和有效性, 将数据集按照 8:1:1 的比例分为训练集 (4 031 张)、验证集 (446 张) 和测试集 (481 张).



(a) 爆浆 (b) 断裂

(c) 多料 (d) 毛刺

图 6 缺陷样例图

3.2 实验环境及参数配置

本文实验参数配置如下: 训练批次设置 800; 批次大小 32; 初始学习率 0.01; 使用 SGD 优化器; 动量 0.937; 工作线程为 8; 输入图像高宽均为 640; 启动早停机制, 早停等待批次为 50.

实验硬件配置如表 1 所示.

表 1 实验硬件配置

实验环境	配置
操作系统	Ubuntu 22.04.2 LTS
CPU	13th Gen Intel(R) Core(TM) i7-13700KF
GPU	NVIDIA GeForce RTX 3070 Ti
内存	32 GB
PyTorch	2.4.0
Python	3.9.19
CUDA	11.8

3.3 评价指标

汽车密封圈缺陷检测模型常用的评价指标包括准确率 P, 召回率 R, 检测精度 AP, 平均检测精度均值 mAP, 计算公式如式 (7)–式 (10) 所示:

$$P=\frac{TP}{TP+FP} \tag{7}$$

$$R=\frac{TP}{TP+FN} \tag{8}$$

$$AP=\int_0^1 P\cdot R\mathrm{d}R \tag{9}$$

$$mAP=\frac{1}{n}\sum_{i=1}^{n}AP_i \tag{10}$$

其中, TP 为正样例且识别为正例的数量; FP 为负样例且错误识别为正例的数量; FN 为正样例被错误识别为负样例的数量; n 为样本类别数量; AP_i为第 i 个类别的平均精度, 实验中使用的 $mAP@0.5$ 表示 IoU 阈值为 0.5 时的平均精度.

此外, 考虑到算法的边缘部署, 还使用浮点运算数 (*FLOPS*)、参数量 (Parameters) 和每秒帧处理数 (FPS) 这 3 个指标来评价模型的整体性能.

3.4 消融实验

为了验证改进后模型在汽车密封圈缺陷检测上的有效性, 本文以 YOLOv8n 为基线模型, 设计了 5 组消融实验. 其中, F-YOLOv8n 表示在骨干网络添加 C2f-Faster 模块; FS-YOLOv8n 表示同时使用了 C2f-Faster 模块与 Slim-neck 设计范式; FSL-YOLOv8n 表示在 FS-YOLOv8n 基础上采用 LADH-Head 替换原有检测头; FSLW-YOLOv8n 表示本文提出的改进模型.

实验结果如表 2 所示, 每个改进模块都在一定程度上提升了模型的性能. F-YOLOv8n 在骨干网络引入了 C2f-Faster 模块, 虽然在毛刺缺陷的检测中精度下降了 0.2%, 但是其余 3 种缺陷的检测精度提升较为明显, 其他各项指标较基线模型也显著提升; FS-YOLOv8n 在 F-YOLOv8n 的基础上使用 Slim-neck 设计范式改进颈部网络, 大大提升了毛刺缺陷的检测精度, 并以损失较少准确率和速度的代价, 将召回率提升了 2.3%, 整体的 $mAP@0.5$ 提升了 0.6%, 模型计算量、参数量和大小分别降低了 14.3%、11.4% 和 9.3%; FSL-YOLOv8n 使用 LADH-Head 替换原有检测头, 一定程度补偿了 Slim-neck 带来的准确率和速度损失, 并且大幅度降低了模型复杂度, 相比 FS-YOLOv8n, 计算量降低了 40%,

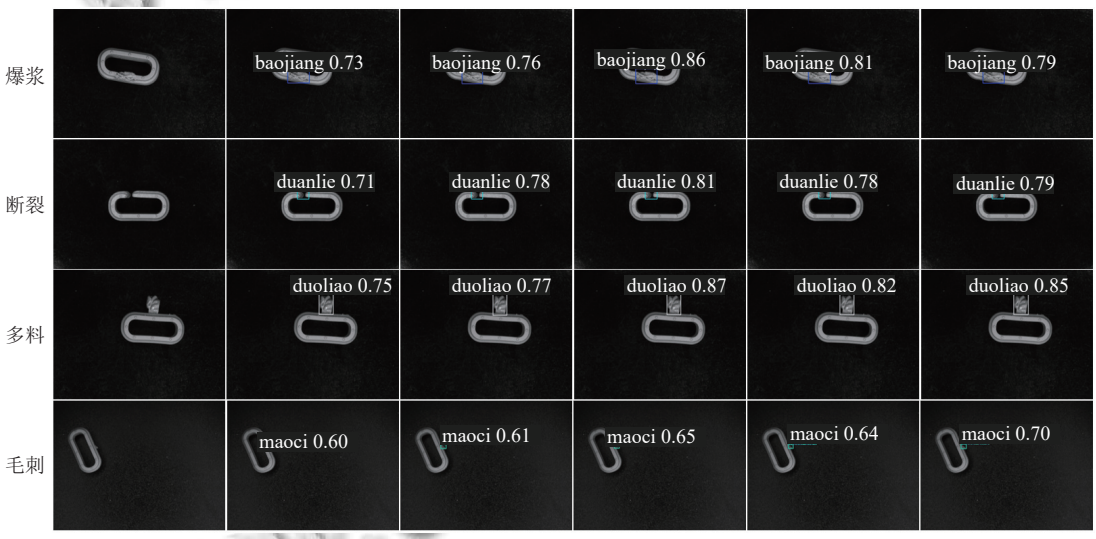
参数量下降了 26.5%, 模型大小减小了 24.5%, 但是多料和毛刺缺陷的检测精度分别下降了 2.5% 和 1.1%; 最后 FSLW-YOLOv8n 使用 Wise-IoU 改进损失函数, 弥补了轻量化检测头带来的部分缺陷检测精度损失, 使得多料和毛刺缺陷 *mAP*@0.5 分别提升了 3.2% 和 0.1%, 整体 *mAP*@0.5 提升了 0.2%, 进一步提高了检测精度.

与 YOLOv8n 相比, 经过改进后的模型对爆浆、断裂、多料和毛刺这 4 种缺陷类型的检测精度分别提升了 3.7%、2.1%、1.3% 和 1.4%, 整体 *mAP*@0.5 提升了 2.1%, 召回率提升了 3.6%, 模型计算量下降了 55.6%, 体积削减了 38.3%, 参数量减少了 42.7%, 有效降低计算和存储需求, 实现了性能与效率的平衡.

图 7 为消融实验中不同算法对 4 种缺陷类型的检测效果对比, 其中每行分别对应爆浆、断裂、多料和毛刺缺陷样本, 每列依次为原图及各个算法检测结果. 可以看到中间阶段改进模型与最终改进模型对 4 种缺陷检测的置信度均高于基线模型.

表 2 消融实验

算法	*P* (%)	*R* (%)	*mAP*@0.5 (%)				*mAP*@0.5 (%)	*FLOPS* (G)	Parameters	Size (MB)	FPS (f/s)
			爆浆	断裂	多料	毛刺					
YOLOv8n	90.0	77.0	80.2	88.0	91.8	83.1	85.8	8.1	3 006 428	6.0	542.7
F-YOLOv8n	90.7	78.3	82.7	89.0	93.5	82.9	87.0	7.0	2 645 228	5.4	559.3
FS-YOLOv8n	88.7	80.6	81.2	91.2	92.4	85.5	87.6	6.0	2 343 988	4.9	493.3
FSL-YOLOv8n	89.6	79.5	84.5	92.0	89.9	84.4	87.7	3.6	1 721 908	3.7	505.2
FSLW-YOLOv8n	89.6	80.6	83.9	90.1	93.1	84.5	87.9	3.6	1 721 908	3.7	506.1



(a) 原图 (b) YOLOv8n (c) F-YOLOv8n (d) FS-YOLOv8n (e) FSL-YOLOv8n (f) FSLW-YOLOv8n

图 7 缺陷检测效果示意图

3.5 对比实验

为进一步验证本文算法的性能, 以 *mAP*@0.5、计算量、模型体积及参数量为评价指标, 将本文算法与常见轻量化检测模型 EfficientDet-D1[17]、NanoDet-m, 以及 YOLO 系列模型 YOLOv3-tiny[18]、YOLOv5n、YOLOv5s、YOLOv7-tiny 和 YOLOv8n 做对比实验. 为得到更精准的比较结果, 所有模型均采用相同的硬件配置和同一个数据集, 实验结果如表 3 所示. 数据表明: FSLW-YOLOv8n 各项指标均优于 EfficientDet-D1; 虽然 NanoDet-m 的体积、计算量和参数量都要小于 FSLW-YOLOv8n, 但是检测精度仅为 78.7%, 远低于本文算法; 与 YOLOv3-tiny、YOLOv5n、YOLOv5s、YOLOv7-tiny 和 YOLOv8n 算法相比, *mAP*@0.5 分别提高了 9.9%、2.5%、2.2%、7.1% 和 2.1%. 可以表明本文算法在精度高于其他算法的同时, 计算量、模型体积和参数量也远低于其他算法.

综上, 本文提出的算法更加适用于汽车密封圈的缺陷检测, 其较低的模型复杂度也更适合在一些计算

能力较弱、硬件资源有限的设备上部署.

表 3 对比实验

算法	*mAP*@0.5 (%)	*FLOPS* (G)	Parameters	Size (MB)
EfficientDet-D1	76.4	6.1	6 557 012	26.9
NanoDet-m	78.7	0.72	937 232	3.8
YOLOv3-tiny	78.0	12.9	8 673 622	17.2
YOLOv5n	85.4	4.1	1 764 577	3.7
YOLOv5s	85.7	15.8	7 020 913	14.2
YOLOv7-tiny	80.8	13	6 015 714	11.7
YOLOv8n (baseline)	85.8	8.1	3 006 428	6.0
FSLW-YOLOv8n (Ours)	87.9	3.6	1 721 908	3.7

3.6 嵌入式平台实验

3.6.1 嵌入式平台

本实验采用的是以海思 SD3403 V100 为主控芯片的嵌入式开发平台. SD3403 V100 是华为推出的一款面向监控市场的专业 ultra-HD Smart IP Camera SOC, 广泛应用于安防监控、智能摄像头和边缘计算等场景. 该芯片配备四核 ARM Cortex-A55 处理器, 内置神经网络加速引擎, 算力最高 4 TOPs INT8, 具备高效的数据处理能力和低功耗的特性. 此嵌入式平台的内核版本为 Linux 4.19.90, 文件系统为 rootfs_glibc_arm64, AI 开发套件 CANN (compute architecture for neural networks) 的版本为 5.13.t5.0.b050.

3.6.2 模型部署

PC 端 PyTorch 框架生成的 pt 模型无法直接部署在海思设备上, 需要进行一系列的转换处理, 其模型部署流程如图 8 所示. 首先在 PC 端将训练好的 pt 模型转换为 onnx 模型, 接着利用昇腾张量编译器 (ascend tensor compiler, ATC) 将 onnx 模型转换为 SD3403 V100 芯片支持的离线 om 模型. 在进行模型推理之前, 需对输入图像进行预处理, 将图像尺寸调整为 640×640, 并将图片从 BGR 格式转换成 RGB 格式传入模型. 模型随后通过 ACL 深度学习框架进行推理, 推理结果中的锚框经过非极大值抑制 (NMS) 进行处理, 最后在图片上绘制检测框并输出.

3.6.3 模型性能验证

模型转换以及嵌入式平台的硬件资源限制会导致算法实际检测精度下降, 为了验证转换后的模型在海思 SD3403 设备上的真实性能, 选取部分可移植到该平台的常用模型进行对比实验, 如表 4 所示. 实验数据表明: 改进后的模型在准确率、模型大小、检测速度方面都明显优于 YOLOv3-tiny、YOLOv5n 和 YOLOv5s 算法. 虽然在检测速度方面略慢于 YOLOv7-tiny 算法, 但是准确率高了 8.3%, 体积小了 54%; 对比基线模型准确率提升了 4.3%, 模型体积减小了 21.9%, 虽然检测速度略微下降, 但仍有 31.3 f/s 的检测速度, 能够满足工业现场检测要求. 总体而言, 改进后的模型在性能和准确率上展现出了明显优势, 为实际应用提供了更为可靠的解决方案.

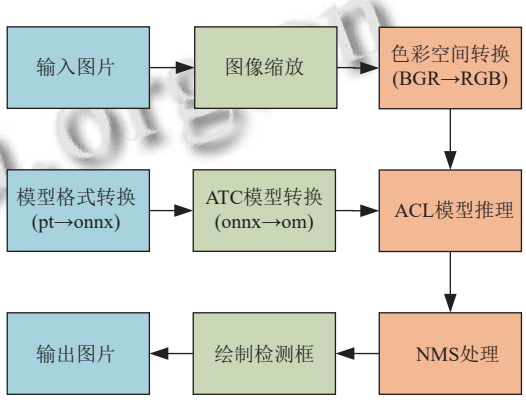


图 8 模型部署流程图

表 4 模型性能对比实验

算法	*P* (%)	om大小 (MB)	FPS (f/s)
YOLOv3-tiny	75.4	17.0	30.3
YOLOv5n	74.4	4.8	25
YOLOv5s	75.6	14.9	19.6
YOLOv7-tiny	80.1	12.4	33.3
YOLOv8n (baseline)	84.1	7.3	35.7
FSLW-YOLOv8n (Ours)	88.4	5.7	31.3

4 结论

本文针对汽车密封圈缺陷检测精度低、算法复杂度高的问题, 提出了基于 YOLOv8n 的改进算法 FSLW-YOLOv8n. 通过对主干网络进行优化, 将 C2f 模块中的 Bottleneck 结构替换为 Faster block, 提升特征提取效率, 接着结合 GSConv 和 Slim-neck 架构进行颈部轻量化处理, 并采用 LADH-Head 优化检测头进一步精简模型结构, 然后用 Wise-IoU 替代原损失函数, 提升小目标的定位能力与检测精度. 最后通过模型转换将本文算法成功部署在海思 SD3403 嵌入式平台上, 实现了 4.3% 的准确率提升以及 31.3 f/s 的检测速度. 本文在汽车密封圈缺陷检测算法的边缘部署上取得了一定进展, 但在检测速度和模型裁剪方面仍有提升空间, 未来计划对模型量化和通道剪枝进行进一步优化.

参考文献

1 《中国公路学报》编辑部. 中国汽车工程学术研究综述·2017. 中国公路学报, 2017, 30(6): 1–197. [doi: 10.3969/j.issn.1001-7372.2017.06.001]

2 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014. 580–587.

3 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015. 1440–1448.

4 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: 10.1109/TPAMI.2016.2577031]

5 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot MultiBox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 21–37.

6 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779–788.

7 陶晓天, 何博侠, 张鹏辉, 等. 基于深度学习的航天密封圈表面缺陷检测. 仪器仪表学报, 2021, 42(1): 199–206.

8 朱文博, 夏林聪, 陈龙, 等. 基于改进 YOLOv5 的 O 型密封圈缺陷检测方法. 上海理工大学学报, 2022, 44(5): 440–448.

9 张相胜, 杨骁. 基于改进 YOLOv7-tiny 的橡胶密封圈缺陷检测方法. 图学学报, 2024, 45(3): 446–453.

10 Ultralytics. Explore Ultralytics YOLOv8. https://docs.ultralytics.com/models/yolov8/#overview. [2025-01-01].

11 Chen JR, Kao SH, He H, *et al*. Run, don't walk: Chasing higher FLOPS for faster neural networks. Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Vancouver: IEEE, 2023. 12021–12031.

12 Li HL, Li J, Wei HB, *et al*. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. arXiv:2206.02424, 2022.

13 Zhang JR, Chen ZH, Yan GX, *et al*. Faster and lightweight: An improved YOLOv5 object detector for remote sensing images. Remote Sensing, 2023, 15(20): 4974. [doi: 10.3390/rs15204974]

14 Tong ZJ, Chen YH, Xu ZW, *et al*. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. arXiv:2301.10051, 2023.

15 Zhang HY, Cissé M, Dauphin YN, *et al*. mixup: Beyond empirical risk minimization. arXiv:1710.09412, 2017.

16 Zhang XY, Zhou XY, Lin MX, *et al*. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 6848–6856.

17 Tan M, Pang R, Le QV. EfficientDet: Scalable and efficient object detection. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 10781–10790.

18 Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv:1804.02767, 2018.

(校对责编: 王欣欣)