激光与光电子学进展 先进成像

基于YOLOv5的瓶盖封装缺陷轻量化检测算法

赵磊[1,2,3*]，矫立宽[1,2,3**]，翟冉[1,2,3]，李彬[1,2,3]，许美叶[1,2,3]

[1]天津理工大学天津市先进机电系统设计与智能控制重点实验室，天津 300384；

[2]机电工程国家级实验教学示范中心（天津理工大学），天津 300384；

[3]天津理工大学机械工程学院，天津 300384

摘要 为解决白酒瓶盖封装表面质量检测和算法参数庞大难部署的问题，对YOLOv5s进行改进并提出了更轻量化和高精度的SEGC-YOLO算法。首先，采用ShuffleNet V2替换原始骨干网络，有效简化参数，引入高效通道注意力机制增强骨干网络。再使用基于GhostNet改进的GhostConv和C3-Ghost模块增强颈部网络，减少颈部参数量。另外，使用CARAFE算子替代最近邻插值上采样算子，利用自适应内容感知的上采样预测核提升颈部网络的信息表达能力，进而提升检测精度。最后，训练应用Adam梯度优化器来提高检测精度。实验结果表明：所提SEGC-YOLO算法在不同交并比（IoU）阈值下的平均精度均值mAP@0.5为84.1%和mAP@0.5∶0.95为49.0%，分别优于原始YOLOv5s算法1.2个百分点和0.5个百分点，并且浮点运算数（FLOPs）比原始算法减少了69.94%、参数量减少了71.15%和模型文件大小减小了69.66%，更加精准和轻量化。所提SEGC-YOLO可以快速、精准地检测瓶盖表面缺陷，为相关领域快速缺陷检测和设备部署提供了数据和算法支持。

关键词 缺陷检测；轻量化算法；YOLOv5；ShuffleNet V2；GhostNet；CARAFE算子

中图分类号 TP183 **文献标志码** A **DOI**：10.3788/LOP231231

YOLOv5-Based Lightweight Algorithm for Detecting Bottle-Cap Packaging Defects

Zhao Lei[1,2,3*], Jiao Likuan[1,2,3**], Zhai Ran[1,2,3], Li Bin[1,2,3], Xu Meiye[1,2,3]

[1]*Tianjin Key Laboratory for Advanced Mechatronic System Design and Intelligent Control, Tianjin University of Technology, Tianjin* 300384, *China*;

[2]*National Demonstration Center for Experimental Mechanical and Electrical Engineering Education (Tianjin University of Technology), Tianjin* 300384, *China*;

[3]*School of Mechanical Engineering, Tianjin University of Technology, Tianjin* 300384, *China*

Abstract With the aim to solve surface-quality detection of liquor bottle-cap packaging and the difficulty of deploying algorithms owing to large parameters, this study proposes a more lightweight and high-precision detection algorithm, named SEGC-YOLO, which is based on YOLOv5s. First, the ShuffleNet V2 is used to replace the original backbone network to effectively simplify the parameters, and the backbone network is enhanced using efficient channel attention mechanism. Next, the improved GhostConv and C3-Ghost modules, based on GhostNet, are used to improve the neck network and reduce the neck parameters. In addition, the CARAFE operator is introduced to replace the nearest neighbor interpolation upsampling operator. The upsampling prediction kernel with adaptive content awareness can improve the information-expression ability of the neck network and thereby the detection accuracy. The Adam gradient optimizer is used for training. Experimental results show that the proposed SEGC-YOLO algorithm achieves the mean accuracy precision mAP @0.5 of 84.1% and mAP@0.5∶0.95 of 49.0% at different intersection over union (IoU) thresholds, which are 1.2 and 0.5 percentage points higher than the original YOLOv5s algorithm, respectively. The overall floating-point operations (FLOPs), parameter volume, and model file size are also reduced by 69.94%, 71.15%, and 69.66%, respectively, indicating higher accuracy and lighter weight compared with that of the original algorithm. Therefore, SEGC-YOLO can quickly and accurately identify the surface defects of bottle caps, providing data and algorithm support for rapid detection and equipment deployment in related fields.

Key words defect detection; lightweight algorithm; YOLOv5; ShuffleNet V2; GhostNet; CARAFE operator

收稿日期：2023-05-06；**修回日期**：2023-06-03；**录用日期**：2023-06-25；**网络首发日期**：2023-07-10

基金项目：国家自然科学基金（51975412）

通信作者：*leizhaotjut@163.com；**1913194980@qq.com

1 引 言

人工智能技术结合智能机器、机器人或传感器等工业设备后能够模拟人类行为，替代人类完成任务[1]。白酒作为中国传统酒类，是重要的深加工农业商品，不同种类的白酒由不同种类粮食酿造而成，白酒生产结合自动化、智能化技术将带动现代农业种植、农产品加工等产业快速发展。然而，针对液体类型农业产品生产的封装检测，人工方式已无法满足高效、高节拍生产的要求且存在误检和漏检问题，亟待一种智能化技术实现对封装表面快速且高效的检测。

目前，缺陷检测任务主要通过机器视觉或者深度学习技术来实现。针对瓶盖缺陷，Zhou 等[2]利用基于投影直方图和稀疏表示的机器视觉技术实现了对瓶盖缺陷的快速检测，但识别强烈受光线和印刷密度影响。此外，还可通过滤波、特征提取、模板匹配、色彩空间变换和阈值调整等依托颜色特征来实现相关瓶盖缺陷检测，但它们应用范围窄且易受环境因素影响，均存在一定局限性[3-4]。深度学习快速发展，广泛应用在缺陷检测领域中，检测效率和精确性得到大幅提升。现阶段主要包含单阶段和双阶段两大派别的检测算法，如 Faster R-CNN 双阶段算法具有更高的检测精度，但检测效率不如单阶段算法。随着算法不断进步，像 SSD、YOLO 等单阶段检测算法的检测精度不断提高，同时还保持很好的检测速度，更加适应现代化生产节奏，更加适合应用于缺陷检测。Kou 等[5]在 YOLOv3 骨干中加入密集结构，并采用基于 anchor-free 的方法来预测目标，所提出的方法对 NEU-DET 带钢数据集的检测平均精度均值（mAP）为 72.2%，优于 Faster R-CNN 算法的检测精度。针对 NEU-DET 数据集的检测，Guo 等[6]将基于 Transformer 设计的 TRANS 模块添加到 YOLOv5 的骨干和颈部网络中，并采用 BiFPN 对颈部网络进行改造，最后采用多步训练方法，得到了 75.2% 的 mAP，比原始算法提高了 7.5%。此外，Wang 等[7]将 YOLOv7 的骨干引入 ConvNeXt 模块，在 MPConv 中加入卷积块注意力模块（CBAM），颈部网络采用改进的 C3C2 模块，对 NEU-DET 数据集实现了 82.9% 的 mAP。算法不断更新，改进方式不断升级，精度不断提高，算法模型结构越发复杂，参数逐渐臃肿，导致模型难以部署到移动和嵌入式设备上，故轻量化网络逐渐得到关注和研究。

Wu 等[8]采用 GhostNet 对原始 YOLOv5 进行改进，并采用 SE 和 CBAM 双注意力机制增强骨干网络，实现了对印刷电路板的缺陷检测，改进后算法的参数量减少了 50.38%，mAP@0.5 和 mAP@0.5∶0.95 提高了 0.9 个百分点和 1.5 个百分点，其中，mAP@0.5 为交并比（IoU）为 0.5 时的 mAP，mAP@0.5∶0.95 则是 IoU 为 0.5～0.95、步长为 0.05 时的 mAP。Zeng 等[9]针对番茄目标检测，通过去除 YOLOv5s 的 Focus 层，采用 MobileNet V3 的 bneck 结构作为骨干网络，然后剪枝颈部网络以减少参数量，使用遗传算法优化超参数以提高精度，相比原始网络，所设计的网络减少了 78% 的参数量和 84.15% 的浮点运算数（FLOPs），最终模型大小只有 3.01 MB，mAP 为 96.9%，mAP 仅减少了 0.5%。算法轻量化改进主要通过改进更加轻量化的网络结构、剪枝操作修剪、去除冗余参数等方式实现；但轻量化后的算法性能通常会弱于原始算法，所以通过加入注意力机制、更换激活函数、更换损失函数等策略来提高精度。因此，轻量化深度学习网络设计的难点主要是简化网络模型和降低计算成本的同时仍保持或优于原有网络精度。而现在并未有轻量化深度学习检测算法应用于白酒瓶盖缺陷检测，所以采用合理的优化和改进策略实现轻量化设计的同时保证高效高精度检测，已成为农业白酒生产线封装缺陷的自动化检测领域亟待解决的难题。本文结合现有主流检测算法 YOLOv5 提出一种更加轻量化、高效的 SEGC-YOLO 检测算法，以实现对该类农业产品曲面多缺陷特征的快速、高精度检测。

2 检测方法

2.1 封装缺陷及数据集制备

生产线上白酒瓶封装设备采用多封装头绕柱旋转冲压方式完成瓶口封装。在高速冲压下，封装头进行快速旋转、挤压和切割，实现瓶口密封。由于在高速下完成挤压和切割工作，存在的封装缺陷主要包括断点、坏边、变形、切割损伤、日期错误、打旋，如图 1 所示；也存在多种缺陷并存的情况，如图 2 所示。缺陷情况非常复杂，现有方法已无法有效检测，所以亟待一种深度

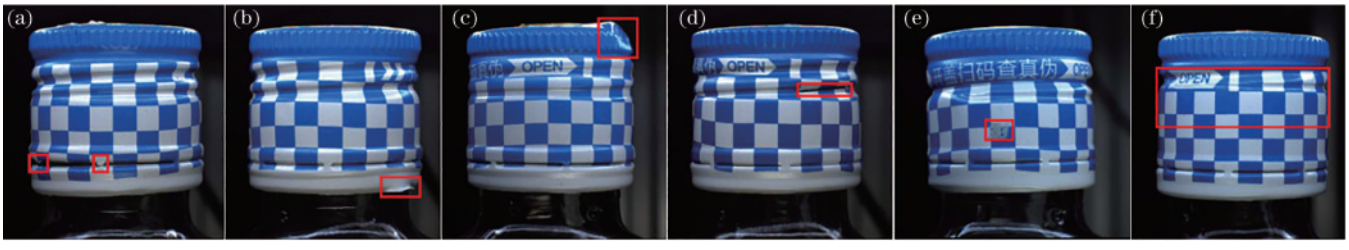

图 1 瓶盖缺陷图片。(a)断点；(b)坏边；(c)变形；(d)切割损伤；(e)日期错误；(f)打旋

Fig. 1 Pictures of bottle cap defects. (a) Break point; (b) broken edge; (c) deformation; (d) damage; (e) data error; (f) spinning

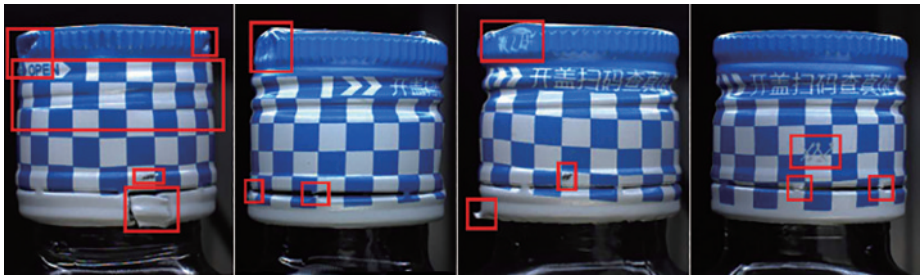

图 2　情况复杂的缺陷图片

Fig. 2　Pictures of defects in complex situation

学习检测算法用于实现对缺陷快速、准确的检测，满足生产的需求。训练采用阿里云天池大赛白酒瓶品数据集中白酒瓶盖部分的图片数据；使用 labelimg 进行缺陷标注，最终将 2320 张图片按照 8∶2 划分出训练集和验证集。

2.2　原始 YOLOv5 算法

原始的 YOLOv5 算法包含输入（Input）、骨干（Backbone）、颈部（Neck）和检测头（Head）4 部分，根据组成又分为 n、s、m、l、x 等众多版本。其中，因 YOLOv5s 版本的检测精度、检测速度及轻量性，最适合用来完成缺陷检测和设备部署任务，结构如图 3 所示。

原始 YOLOv5 算法中的骨干网络由 Focus、Conv、C3 以及 SPP 模块构成，拥有超强的特征信息提取能力。Focus 模块能够有效缩减采样，增加通道尺寸；C3 和 Conv 模块是 YOLOv5 网络最主要的特征提取手段，如图 3 所示，可以有效提取图像的特征信息；SPP 模块分别经过 kernel-size 为 5、9、13 的最大池化层。4 部分特征融合很好地提高了感受野，在不降低检测速度的情况下提取更多的特征信息[10]。颈部网络由特征金字塔网络（FPN）与路径聚合网络（PAN）构成，从而实现对骨干网络提取到的图像特征的融合，然后将融合结果送至检测端进行识别。通过 FPN 结构对顶层的语义特征自顶向下地进行上采样并将结果侧向连接骨干网络中具有相同空间尺寸的特征层[11]。最后，PAN 对 FPN 中的信息进一步从下到上地进行提取和融合，并输出至 Head 端，从而增强了 YOLOv5 网络的特征聚合能力[12-13]。

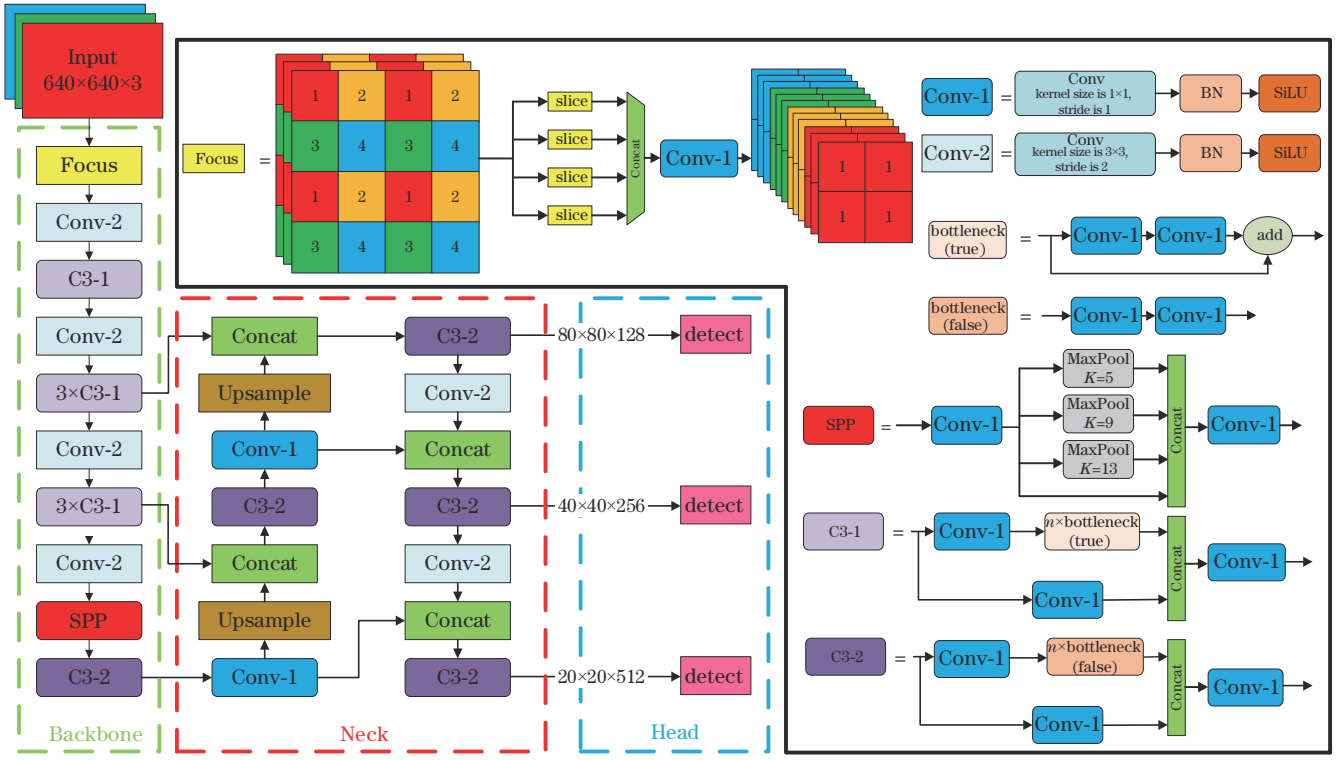

图 3　原始 YOLOv5s 模型的结构

Fig. 3　Structure of the original YOLOv5s model

YOLOv5 算法损失函数主要由分类损失、置信度损失、预测框与目标框的位置损失组成，可表达为

$$L = L_{obj} + L_{cls} + L_{box}, \tag{1}$$

式中：L 为模型的总损失；L_{cls} 为分类损失；L_{obj} 为置信度损失；L_{box} 为预测框和真实框的位置损失。L_{cls} 和 L_{obj} 主要是通过 BCEWithLogitsLoss 函数来计算的，公式分别为

$$L_{\text{obj}}=-\lambda_{\text{obj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{\text{obj}}\left[\hat{C}_i^j\log C_i^j+(1-\hat{C}_i^j)\log(1-C_i^j)\right]-\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{\text{noobj}}\left[\hat{C}_i^j\log C_i^j+(1-\hat{C}_i^j)\log(1-C_i^j)\right],\tag{2}$$

$$L_{\text{cls}}=-\sum_{i=0}^{S^2}I_{ij}^{\text{obj}}\sum_{c\in N_{\text{classes}}}\left\{\hat{P}_i(c)\log\left[P_i(c)\right]+\left[1-\hat{P}_i(c)\right]\log\left[1-P_i(c)\right]\right\},\tag{3}$$

式中：S^2为特征图中划分的单元格数量；B为预测边界框的数量；I_{ij}^{obj}和I_{ij}^{noobj}为第i个单元格中第j个边界框是否参与该物体的检测，若是，则为1，反之为0；λ_{obj}和λ_{noobj}为网格的权重因子；C_i^j和$\hat{C}^j$为第i个单元格的第j个边界框的预测和实际目标的置信度值；$P_i(c)$和$\hat{P}_i(c)$为第i个单元格的第j个边界框的预测和实际目标的概率值。

考虑到检测精度和训练效果，采用高精度的完全交并比（CIoU）损失函数[14]计算位置损失，充分考虑重叠区域、中心点距离和纵横比等因素的影响，CIoU损失的计算公式为

$$L_{\text{box}}=L_{\text{CIoU}}=1-R_{\text{IoU}}+\mathfrak{R}_{\text{DIoU}}+\alpha v,\tag{4}$$

$$R_{\text{IoU}}=\frac{\left|B\cap B^{\text{gt}}\right|}{\left|B\cup B^{\text{gt}}\right|},\tag{5}$$

$$\mathfrak{R}_{\text{DIoU}}=\frac{\rho^2(b,b^{\text{gt}})}{c^2},\tag{6}$$

$$\alpha=\frac{v}{(1-R_{\text{IoU}})+v},\tag{7}$$

$$v=\frac{4}{\pi^2}\left(\arctan\frac{w^{\text{gt}}}{h^{\text{gt}}}-\arctan\frac{w}{h}\right)^2,\tag{8}$$

式中：L_{CIoU}为CIoU损失函数；R_{IoU}为预测框与实际标签位置的交并比；B、B^{gt}为真实框和预测框的中心点坐标（b、b^{gt}）以及框的宽、高尺寸（w、h、w^{gt}、h^{gt}）等参数，如图4所示；$\mathfrak{R}_{\text{DIoU}}$为距离交并比（DIoU）损失的惩罚项；$\rho(\cdot)$为欧氏距离；$c$为覆盖$B$和$B^{\text{gt}}$最小外接矩形对角线的长度。

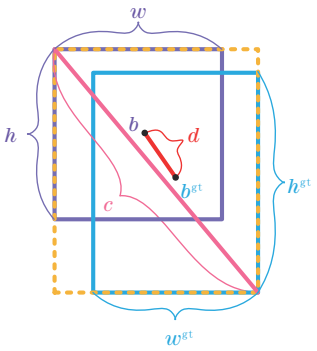

图4　CIoU loss

Fig. 4　CIoU loss

YOLOv5算法对不同损失注重度又设置了相应的权重（α_{obj}、α_{cls}、α_{box}）来平衡各个损失，最终公式为

$$L=\alpha_{\text{obj}}\times L_{\text{obj}}+\alpha_{\text{cls}}\times L_{\text{cls}}+\alpha_{\text{box}}\times L_{\text{box}}。\tag{9}$$

2.3　改进的SEGC-YOLO算法

YOLOv5s算法具有很强的特征提取能力、很高的识别精准度及较少的参数量，故在其基础上提出一种更加轻量化的深度学习算法SEGC-YOLO，如图5所示。改进后的算法有效减少了参数量，实现了比YOLOv5s算法更高的检测精度，弥补了轻量化结构精度的不足。

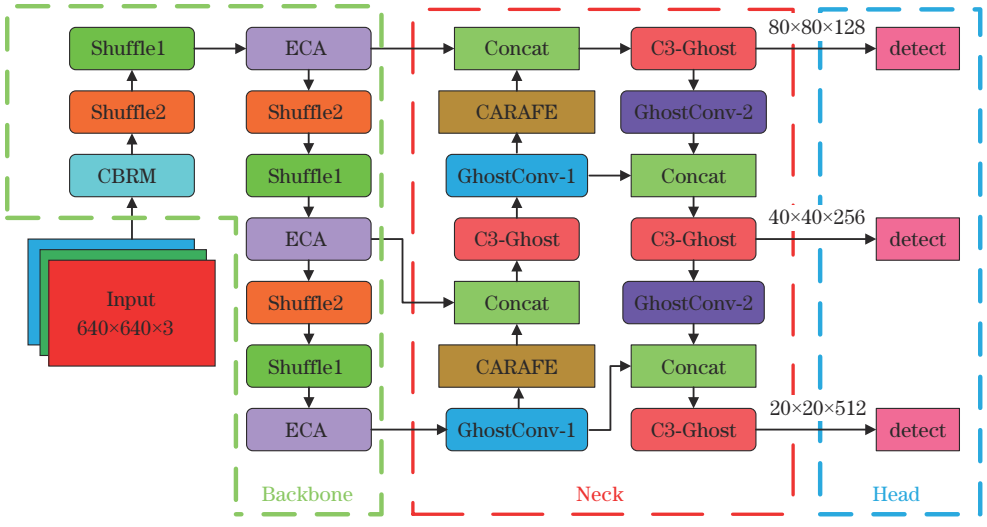

图5　改进的深度学习模型SEGC-YOLO

Fig. 5　Improved deep learning model SEGC-YOLO

1）ShuffleNet V2轻量化改进Backbone

首先，用CBRM层代替Fcous层，并采用ShuffleNet V2[15]的基础模块和下采样模块替换掉原来的Backbone，实现骨干网络的参数量减少，达到轻量

化设计的目的。ShuffleNet V2存在4条设计准则[15]：G1. 等通道宽度最小化内存访问成本(MAC)；G2. 过多的群卷积增加MAC；G3. 网络碎片化降低并行度；G4. 元素操作影响不可忽略。根据采样时DWConv模块采用的不同步长，ShuffleNet V2的基本模块和下采样模块分别定义为Shuffle1和Shuffle2两种模块，如图6所示。Shuffle1结构首先采用Channel split将原来通道一分为二，实现了分组卷积的效果，这样同时满足了G1和G2设计准则，而Shuffle2模块直接同时处理输入。两个模块均采用1×1的Conv，摒弃了ShuffleNet V1[16]中的分组卷积，满足G2设计准则，避免了速度下降。Shuffle1有一个分支不采用任何操作，减少了碎片化操作，满足了G3准则。

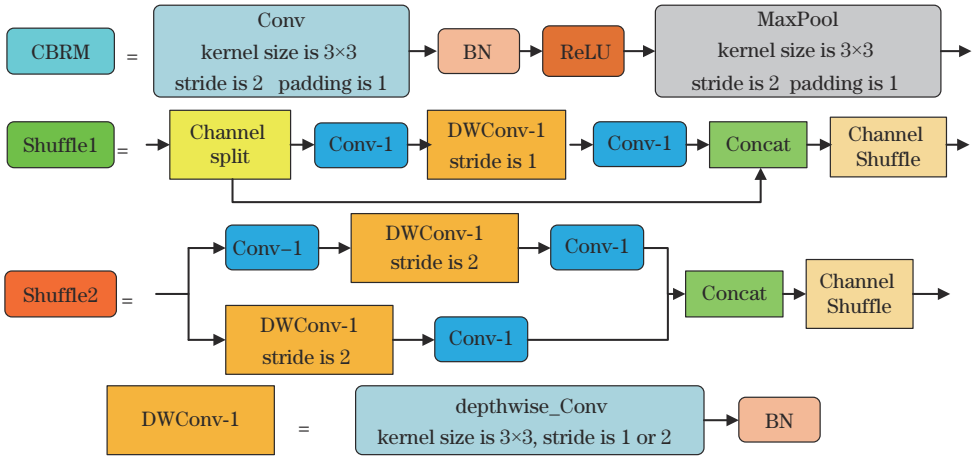

图6　轻量化骨干模块

Fig. 6　Lightweight Backbone module

Shuffle1和Shuffle2模块都采用Concat进行特征融合，并不再使用ShuffleNet V1中Add+ReLU的元素处理操作，符合G4准则，减少了元素操作。两个模块在最后使用了Channel Shuffle结构，增强了通道与通道之间的信息交流。Channel Shuffle结构如图7所示。

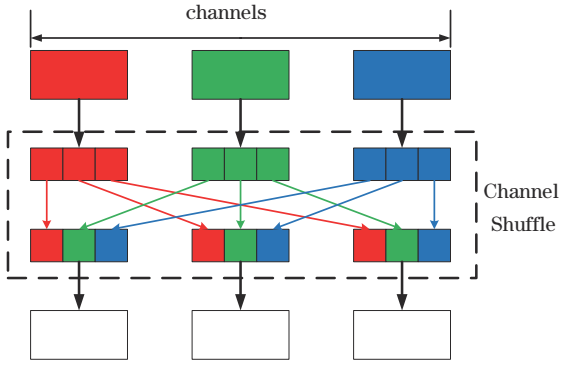

图7　通道洗牌

Fig. 7　Channel Shuffle

2) 高效通道注意力(ECA)增强特征信息

在骨干网络中，每一个Shuffle1模块后加入ECA机制增强骨干网络提取的特征信息，并将已增强特征输至颈部网络的FPN结构进行特征信息融合，具体结构如图5所示。目前存在多种不同注意力机制，而ECA机制具有优秀的跨通道信息交互能力[17]。如图8所示，ECA机制先对输入的特征图进行全局平均池化，再使用一维卷积对已获得的平均特征图中各K个通道之间的特征关系进行提取，采用的一维卷积的卷积核为K，最后经过Sigmoid函数得到特征关系的权重。ECA机制就是对得到的权重和一开始输入进来的特征图进行相乘进而增强特征图的有效特征，抑制无效特征的。

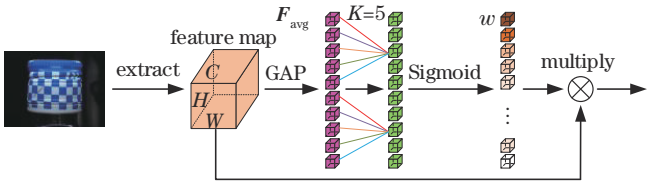

图8　ECA机制结构

Fig. 8　Structure of the ECA mechanism

3) GhostNet优化Neck

颈部网络(Neck)采用GhostNet[18]对原始颈部中的Conv-1、Conv-2和C3-2模块进行改进，得到GhostConv-1、GhostConv-2和C3-Ghost，如图9所示，进一步简化了颈部网络参数和提高了检测精度。

GhostConv-1和GhostConv-2先通过Conv模块生成feature maps，再通过DWConv生成Ghost feature maps，然后两者结合，如图10所示，实现了特征信息与内在特征信息的融合，从而提升了特征信息的表达能力，提高了检测精度，也减少了网络参数量。C3-Ghost通过更换新的GhostBottleneck(false)进一步减少了自身参数量，更好地利用了内在特征信息，提高了原始C3结构的性能，从而提升了检测精度。

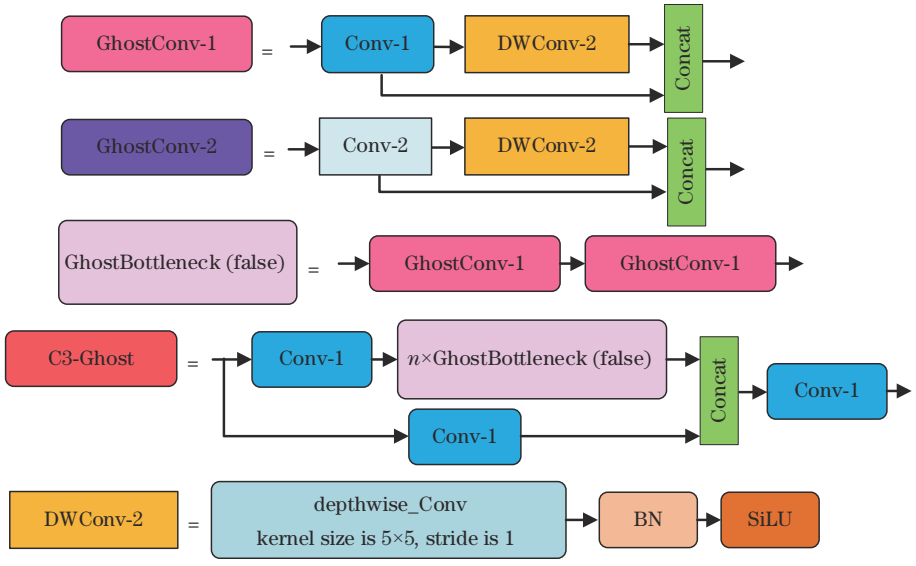

图 9　GhostConv-1、GhostConv-2 和 C3-Ghost 结构

Fig. 9　Structures of GhostConv-1, GhostConv-2, and C3-Ghost

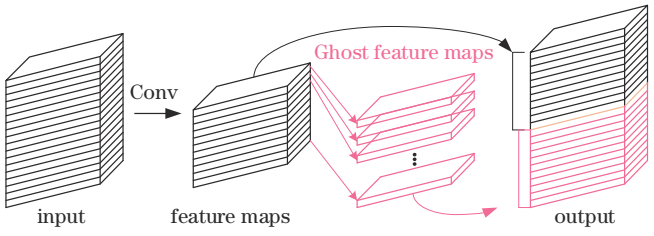

图 10　Ghost 操作图

Fig. 10　Diagram of Ghost operation

4) CARAFE 上采样算子提升 Neck

颈部网络除了采用更加轻量、高效的 C3-Ghost 等模块，还引入 CARAFE 上采样算子[19]替代原始的最近邻插值上采样算子。CARAFE 可以在一个大的感受野内聚集上下文信息并对特定实例进行内容感知处理，即时生成自适应内核以更好地实现上采样操作[19]。同时，CARAFE 也具有轻量化和快速计算的特性。

CARAFE 包含内核预测模块（kernel prediction module）和内容感知重组模块（content-aware reassembly module）两部分，如图 11 所示。第一部分先将输入的尺寸为 $H\times W\times C$ 的特征图 $\boldsymbol{X}$ 压缩通道至 C_{m}，再经内容编码器得到 $H\times W\times\sigma^2\times k_{\mathrm{up}}^2$ 的特征图，然后将其展开成 $\sigma H\times\sigma W\times k_{\mathrm{up}}^2$ 特征图，最后进行 Softmax 归一化得到上采样预测核；第二部分会将上采样预测核按照位置展开成 $k_{\mathrm{up}}^2\times k_{\mathrm{up}}^2$ 的小特征层，然后对其与相同位置的输入特征层进行点乘，最终得到 $\sigma H\times\sigma W\times C$ 的输出特征图 $\boldsymbol{X}'$，实现上采样。

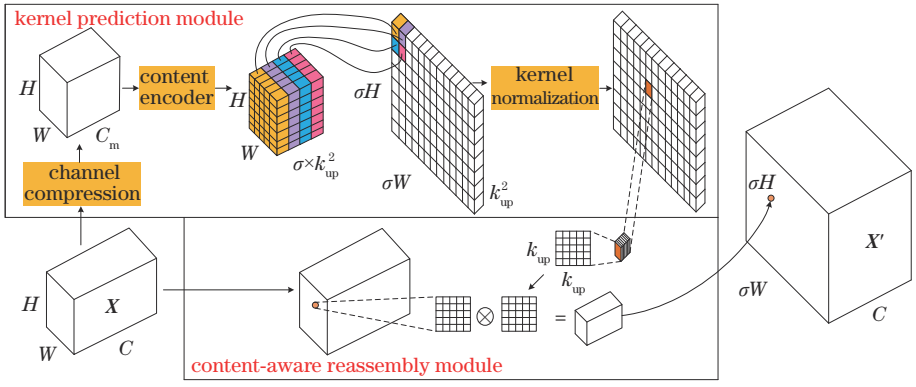

图 11　CARAFE 上采样算子的结构

Fig. 11　Structure of CARAFE upsampling operator

2.4　算法训练策略

算法整体实验过程分为两部分：一是训练和验证，利用数据集来训练算法，寻找算法的最优权重，然后经消融实验分析改进后的算法；二是对比和测试，进行多算法对比实验和实际检测效果的对比，验证改进算法的可靠性。进行 250 轮训练，批量大小为 20，初始学习率为 0.01，momentum 为 0.937，weight decay 为 0.0005。训练设备 CPU 为 Intel(R) Core(TM) i7-7700@3.60 GHz，GPU 为 Quadro RTX 5000，操作系统为 Windows 10 专业版，采用的深度学习框架为

PyTorch 1.10，GPU 加速为 CUDA 11.3+cuDNN 8.2.1，编程语言为 Python 3.8。模拟检测设备如图 12 所示，使用维视智造公司的 MV-EM510C 工业相机，在拍摄时镜头与瓶盖表面垂直，给到环形光圈合适亮度和位置以照亮整个瓶盖，使缺陷清晰。

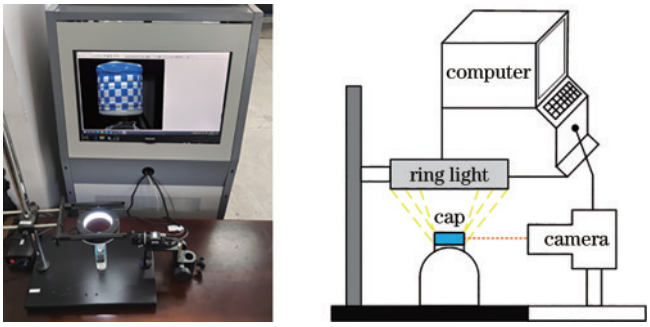

图 12　模拟检测设备

Fig. 12　Analog inspection equipment

2.5　评价指标

采用 mAP@0.5、mAP@0.5∶0.95、FLOPs、参数量（Parameters）、模型文件大小（Model file）和推理时间（Inference time）作为评价指标进行数据对比。其中，FLOPs、Parameters、Model file 和 Inference time 分别代表算法的计算复杂度、整体算法参数量、训练结束后最优算法模型参数文件所占用内存大小、算法的推理检测时间。mAP 作为多种缺陷平均精度（AP）的平均值，更能反映模型的整体检测效果。AP 是由以 Precision（P）、Recall（R）值作为横、纵坐标轴数据绘制出的 P-R 曲线与坐标轴包围起来的面积来表示的。具体公式分别为

$$P_{\mathrm{AP}}=\int_0^1 P(R)\mathrm{d}R, \tag{10}$$

$$P_{\mathrm{mAP}}=\frac{\sum_{i=1}^{N}P_{\mathrm{AP}i}}{N}, \tag{11}$$

式中：N 为缺陷的种类数；i 为某一种类缺陷；$P(R)$ 表示 P-R 曲线。计入 IoU 阈值对精度的影响，故采用更加均衡的评价指标 mAP@0.5 和 mAP@0.5∶0.95。

3　分析与讨论

3.1　训练实验分析

经验证集检验，得到原始 YOLOv5s 算法和 SEGC-YOLO 算法检测精度和损失值的变化曲线，如图 13 所示。经过 250 轮的训练后，改进算法和原始算法均逐渐平稳，达到了最佳的检测精度和最小的损失。改进后的算法的检测精度曲线均超过原始 YOLOv5s 算法，损失曲线也低于原始算法，表现出比原始算法更优异的检测性能。

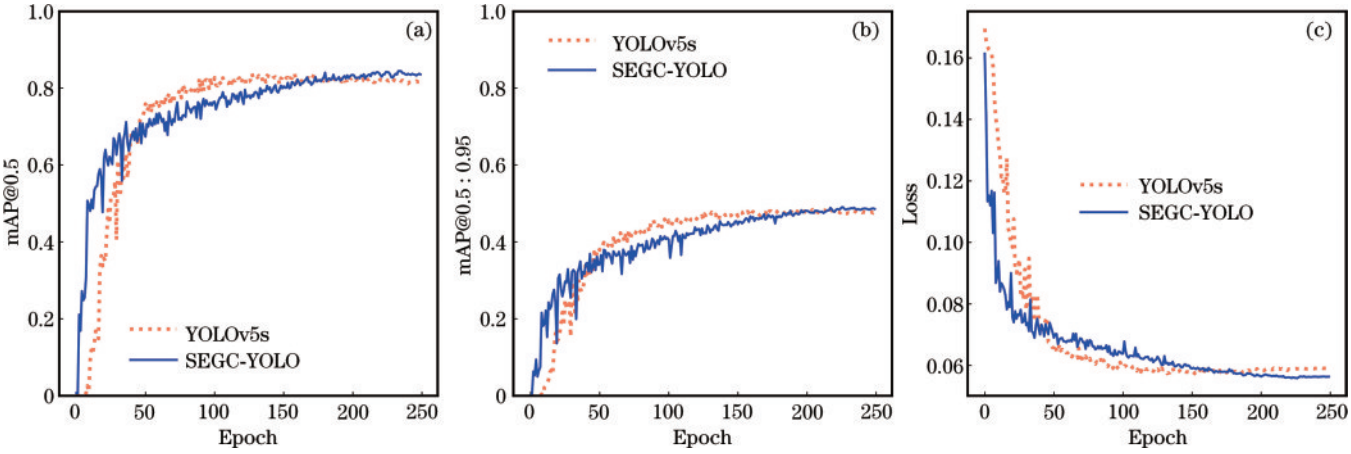

图 13　YOLOv5s 与 SEGC-YOLO 的 mAP 和损失值曲线。（a）mAP@0.5；（b）mAP@0.5∶0.95；（c）损失值

Fig. 13　mAP and loss curves of YOLOv5s and SEGC-YOLO. (a) mAP@0.5; (b) mAP@0.5∶0.95; (c) loss

3.2　消融实验分析

SEGC-YOLO 是对 YOLOv5s 版本进行轻量化改进得到的，故以 YOLOv5s 搭配 SGD 优化器作为改进基础进行消融实验对比，实验数据如表 1 所示。更换完 ShuffleNet V2 骨干之后，模型的 FLOPs、参数量和模型文件大小相比原始算法均实现了 50% 以上的减少，说明该骨干具有足够轻量性，但 mAP@0.5 和 mAP@0.5∶0.95 下降了 2.3 个百分点和 2.0 个百分点，骨干网络的精度随着骨干参数量的减少而减小，这也是轻量化算法的通病。后续通过 Adam 梯度优化器进行优化，精度得到一定提升。注意力机制可以增加算法的检测精度，但引入 ECA 机制后，相比实验 3，实验 4 的 mAP@0.5 下降了 0.9 个百分点，mAP@0.5∶0.95 提升了 0.3 个百分点。相比 mAP@0.5，mAP@0.5∶0.95 是更加均衡的检测精度指标，它的提升意味着该算法在多 IoU 阈值情况下的检测精度都得到提高，总体上说 ECA 的加入提高了算法的性能，同时也未引入过多参数量，可以保持很好的轻量化特性。

采用 Ghost 改进后，相比实验 4，算法分别减少了 2.5×10^9 的 FLOPs、1.4×10^6 的参数量和 2.8 MB 的模型文件，但是 mAP@0.5 得到提升，充分利用了冗余的特征信息。更换 CARAFE 上采样算子后，其自适应内

表 1 消融实验数据

Table 1 Data of ablation experiments

No.	Backbone	Adam	ECA	Ghost	CARAFE	FLOPs /10^9	Parameters /10^6	Model file /MB	mAP@0.5 /%	mAP@0.5:0.95 /%
1						16.3	7.07	14.5	82.9	48.5
2	√					6.9	3.31	6.9	80.6	46.5
3	√	√				6.9	3.31	6.9	82.7	48.3
4	√	√	√			6.9	3.31	7.0	81.8	48.6
5	√	√	√	√		4.4	1.91	4.2	82.7	48.6
6	√	√	√	√	√	4.9	2.04	4.4	84.1	49.0
7	√		√	√	√	4.9	2.04	4.4	83.4	48.3

核更好地实现了上采样操作，进一步提升了算法的检测精度，mAP@0.5 和 mAP@0.5:0.95 为 84.1% 和 49.0%。最终，相比原始算法，改进后的算法的 mAP@0.5 和 mAP@0.5:0.95 提高了 1.2 个百分点和 0.5 个百分点，整体减少了 69.94% 的 FLOPs、71.15% 的参数量和 69.66% 的模型文件大小，更加精准和轻量化。

针对添加 ECA 后精度下降的现象，又添加多种不同注意力机制[20-24]进行对比实验，实验数据如表 2 所示。在引入 ECA 后，改进算法相比引入其他注意力机制的算法实现了最少的 FLOPs 和参数量、最小的模型文件大小和最高的 mAP@0.5、mAP@0.5:0.95 检测精度。通过消融实验和多注意力机制对比实验，有效验证了所提 SEGC-YOLO 算法的准确性和轻量化。

表 2 多种注意力机制的对比数据

Table 2 Comparative data on multiple attention mechanisms

Attention mechanism	FLOPs /10^9	Parameters /10^6	Model file /MB	mAP@0.5 /%	mAP@0.5:0.95 /%
ECA[17]	4.9	2.04	4.4	84.1	49.0
SE[20]	5.4	2.78	5.8	83.9	49.0
CBAM[21]	6.0	2.73	5.8	82.1	47.7
CA[22]	6.4	3.08	6.5	83.6	48.7
ShuffleAttention[23]	4.9	2.04	4.4	81.7	48.6
NAM[24]	4.9	2.04	4.5	82.9	48.0

3.3 不同模型对比实验分析

为探究其他不同模型对封装缺陷的检测性能，验证所提 SEGC-YOLO 算法的精准性和轻量性，选取 YOLOv3[25]、YOLOv7[26]等算法进行对比，实验数据如表 3 所示。

表 3 多算法的对比实验数据

Table 3 Comparative experimental data of multiple algorithms

Algorithm	FLOPs /10^9	Parameters /10^6	Model file /MB	mAP@0.5 /%	mAP@0.5:0.95 /%	Inference time /ms
YOLOv5s	16.3	7.07	14.5	82.9	48.5	16.3
SEGC-YOLO	4.9	2.04	4.4	84.1	49.0	15.8
YOLOv3-tiny	13.0	8.68	17.5	82.3	46.1	8.3
YOLOv3	155.3	61.55	123.6	83.5	48.4	34.3
YOLOv7	105.2	37.22	74.8	84.1	49.0	41.4

改进后的算法实现了最少的 FLOPs 和参数量，最小的参数文件大小，达到了与 YOLOv7 相同的检测精度，同时比 YOLOv3-tiny 和 YOLOv3 的检测精度更高。SEGC-YOLO 算法推理时间为 15.8 ms，只因改进算法中含有较多的深度可分离卷积，其能有效减少参数量、保证精度但运行速度不及普通卷积。因此 SEGC-YOLO 算法的推理时间稍劣于 YOLOv3-tiny 算法，但是优于大型算法 YOLOv3 和 YOLOv7，与 YOLOv5s 速度持平，具有良好的检测速度，可适用于产线的快速生产。最后，改进后的 SEGC-YOLO 算法在削减了相对于 YOLOv3 和 YOLOv7 算法 96.69% 和 94.52% 参数量的情况下，实现了更佳的检测精度和检测速度，进一步验证了 SEGC-YOLO 算法结构的有效性、轻量性和检测的精准性、快速性。

3.4 缺陷特征检测实验

运用模拟检测设备，对 6 种模拟缺陷进行多算法的实际检测效果对比，结果如图 14 所示。第 1 组缺陷，SEGC-YOLO 可识别出比 YOLOv5s 和 YOLOv3 更多的缺陷，与 YOLOv7 检测效果近似；第 2 组中，仅 YOLOv5s 未识别出变形缺陷，此情况也出现在第 4 组

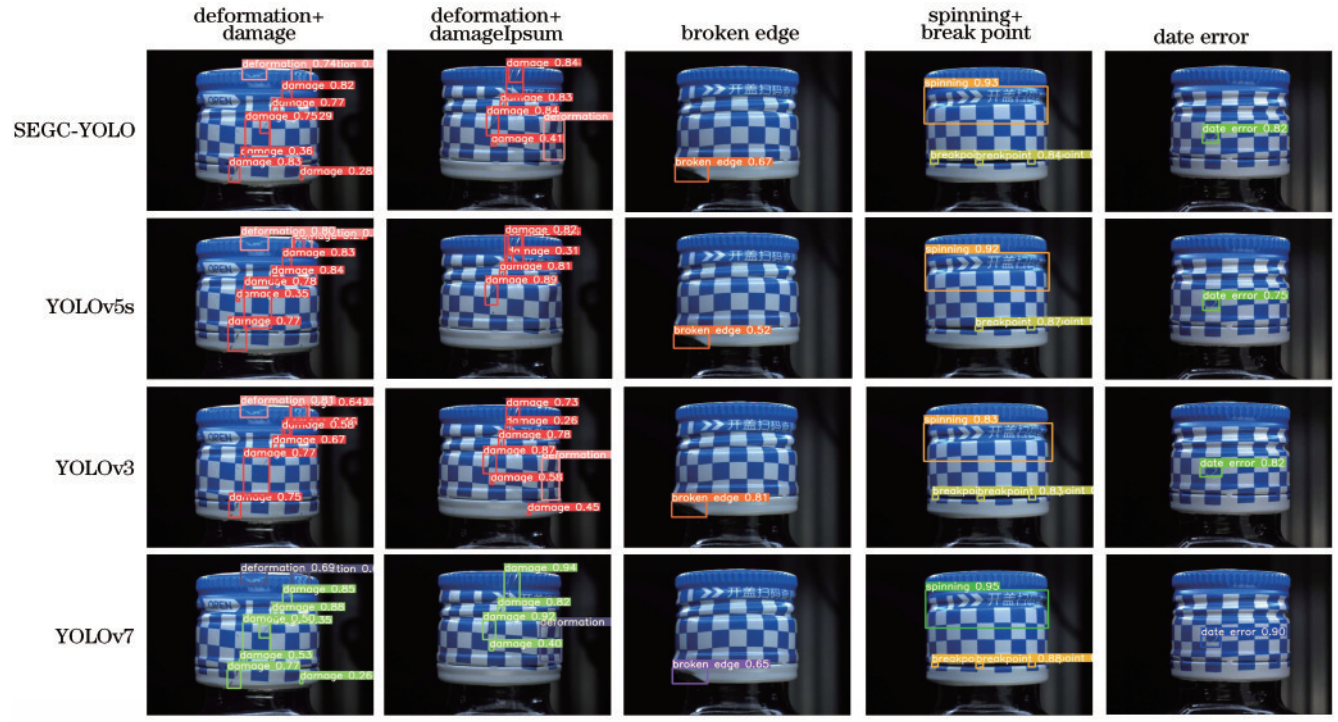

图 14　不同算法的缺陷检测对比

Fig. 14　Comparison of defect detection using different algorithms

中，YOLOv5s 未识别出一个 break point 缺陷；第 3 组和第 5 组缺陷种类单一且明显，所有算法均可检测出缺陷，但 YOLOv5s 效果最差。综上所述，在同时可识别多种缺陷情况下，改进后的 SEGC-YOLO 算法取得了不错的检测效果，大大降低了算法本身参数量，能够有效保证检测精度，确保完成瓶盖封装质量的检测。

4　结　　论

对白酒农业产品在瓶盖封装时的缺陷进行检测，提出了具有更加轻量化特性的深度学习缺陷检测算法 SEGC-YOLO，解决了高精度模型参数量庞大的弊端，实现了对白酒封装缺陷的快速、高精度的检测。SEGC-YOLO 算法的 mAP@0.5 为 84.1%，mAP@0.5:0.95 为 49.0%，优于原始 YOLOv5s 算法，并减少了 69.94% 的 FLOPs 和 71.15% 的参数量，减小了 69.66% 的模型文件大小，更加轻量，易部署，通过多种实验验证了改进后 SEGC-YOLO 算法的准确性和轻量化。改进后的算法极大地减少了算法的参数量，弥补了轻量化设计精度的不足，提高了算法移植的可行性，降低了部署需要的条件，为实时在线识别检测奠定了技术基础，有效地促进了相关类产品生产的发展。

参　考　文　献

[1] Elbasi E, Mostafa N, AlArnaout Z, et al. Artificial intelligence technology in the agricultural sector: a systematic literature review[J]. IEEE Access, 2022, 11: 171-202.

[2] Zhou W J, Fei M R, Zhou H Y, et al. A sparse representation based fast detection method for surface defect detection of bottle caps[J]. Neurocomputing, 2014, 123: 406-414.

[3] Xu M, Ma Y, Chen S. Research on real-time quality inspection of PET bottle caps[C]//2017 IEEE International Conference on Information and Automation (ICIA), July 18-20, 2017, Macao, China. New York: IEEE Press, 2017: 1023-1026.

[4] Kumchoo W, Chiracharit W. Detection of loose cap and safety ring for pharmaceutical glass bottles[C]//2018 International ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI-NCON), February 25-28, 2018, Chiang Rai, Thailand. New York: IEEE Press, 2018: 125-130.

[5] Kou X P, Liu S J, Cheng K Q, et al. Development of a YOLO-V3-based model for detecting defects on steel strip surface[J]. Measurement, 2021, 182: 109454.

[6] Guo Z X, Wang C S, Yang G, et al. MSFT-YOLO: improved YOLOv5 based on transformer for detecting defects of steel surface[J]. Sensors, 2022, 22(9): 3467.

[7] Wang R J, Liang F L, Mou X W, et al. Development of an improved YOLOv7-based model for detecting defects on strip steel surfaces[J]. Coatings, 2023, 13(3): 536.

[8] Wu L G, Zhang L, Zhou Q. Printed circuit board quality detection method integrating lightweight network and dual attention mechanism[J]. IEEE Access, 2022, 10: 87617-87629.

[9] Zeng T H, Li S Y, Song Q M, et al. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment[J]. Computers and

Electronics in Agriculture, 2023, 205: 107625.
[10] Ying Z P, Lin Z T, Wu Z Y, et al. A modified-YOLOv5s model for detection of wire braided hose defects[J]. Measurement, 2022, 190: 110683.
[11] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 936-944.
[12] Dong X, Yan S, Duan C. A lightweight vehicles detection network model based on YOLOv5[J]. Engineering Applications of Artificial Intelligence, 2022, 113: 104914.
[13] Liu S, Qi L, Qin H F, et al. Path aggregation network for instance segmentation[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 8759-8768.
[14] Zheng Z H, Wang P, Liu W, et al. Distance-IoU loss: faster and better learning for bounding box regression [EB/OL]. (2019-11-19)[2023-02-03]. https://arxiv.org/abs/1911.08287.
[15] Ma N N, Zhang X Y, Zheng H T, et al. ShuffleNet V2: practical guidelines for efficient CNN architecture design [M]//Ferrari V, Hebert M, Sminchisescu C, et al. Computer vision-ECCV 2018. Lecture notes in computer science. Cham: Springer, 2018, 11218: 122-138.
[16] Zhang X Y, Zhou X Y, Lin M X, et al. ShuffleNet: an extremely efficient convolutional neural network for mobile devices[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 6848-6856.
[17] Wang Q L, Wu B G, Zhu P F, et al. ECA-net: efficient channel attention for deep convolutional neural networks [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 11531-11539.
[18] Han K, Wang Y H, Tian Q, et al. GhostNet: more features from cheap operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 13-19, 2020, Seattle, WA, USA. New York: IEEE Press, 2020: 1577-1586.
[19] Wang J Q, Chen K, Xu R, et al. CARAFE: content-aware ReAssembly of FEatures[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV), October 27-November 2, 2019, Seoul, Korea (South). New York: IEEE Press, 2020: 3007-3016.
[20] Hu J, Shen L, Sun G. Squeeze-and-excitation networks [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE Press, 2018: 7132-7141.
[21] Woo S, Park J, Lee J Y, et al. CBAM: convolutional block attention module[M]//Ferrari V, Hebert M, Sminchisescu C, et al. Computer vision-ECCV 2018. Lecture notes in computer science. Cham: Springer, 2018, 11211: 3-19.
[22] Hou Q B, Zhou D Q, Feng J S. Coordinate attention for efficient mobile network design[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 20-25, 2021, Nashville, TN, USA. New York: IEEE Press, 2021: 13708-13717.
[23] Zhang Q L, Yang Y B. SA-net: shuffle attention for deep convolutional neural networks[C]//2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), June 6-11, 2021, Toronto, ON, Canada. New York: IEEE Press, 2021: 2235-2239.
[24] Liu Y C, Shao Z R, Teng Y Y, et al. NAM: normalization-based attention module[EB/OL]. (2021-11-24)[2023-02-03]. https://arxiv.org/abs/2111.12419.
[25] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL]. (2018-04-08)[2023-02-03]. https://arxiv.org/abs/1804.02767.
[26] 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[EB/OL]. (2022-07-06)[2023-02-03]. https://arxiv.org/abs/2207.02696.