DOI：10.13733/j.jcam.issn.2095-5553.2026.01.016
刘鹏扬，亚森江·木沙. 基于剪枝的 YOLOv8 轻量化苹果表面缺陷检测算法[J]. 中国农机化学报，2026，47(1)：108－117
Liu Pengyang，Yasenjiang Musha. Pruned YOLOv8-based lightweight algorithm for detecting apple surface defects [J]. Journal of Chinese Agricultural Mechanization，2026，47(1)：108－117

基于剪枝的 YOLOv8 轻量化苹果表面缺陷检测算法*

刘鹏扬，亚森江·木沙

（新疆大学智能制造现代产业学院，乌鲁木齐市，830000）

摘要：在苹果表面缺陷检测中，快速且高精度的检测技术至关重要。当前的研究在精度上取得进展，但推理速度仍然有待提升，为此，提出一种基于剪枝的轻量化苹果表面缺陷检测算法。采用 YOLOv8n 为基础模型，结合 GhostNetV2 与 YOLOv8 结构中 C2f 的特性，设计一种 C2f—GhostV2 模块，显著减少模型参数量并加快推理速度。为进一步减小计算负荷，模型引入幽灵卷积(GhostConv)代替传统卷积，并采用动态上采样(DySample)机制提升灵活性与信息保留能力。此外，轻量化模型经过基于层自适应幅度的剪枝(LAMP)，进一步减少浮点运算量。结果表明，剪枝后的模型平均精度均值达到 97.3%，与原模型相比，浮点运算次数减少 78.05%，推理速度提高 27.85%。

关键词：苹果；表面缺陷检测；轻量化；剪枝；小目标检测
中图分类号：S651；TP391.4 **文献标识码**：A **文章编号**：2095-5553 (2026) 01-0108-10

Pruned YOLOv8-based lightweight algorithm for detecting apple surface defects

Liu Pengyang，Yasenjiang Musha

(*School of Intelligent Manufacturing and Modern Industry*，*Xinjiang University*，*Urumqi*，830000，*China*)

Abstract：In apple surface defect detection，rapid and high-precision detection technology is crucial. Current research has made progress in accuracy，but detection speed still needs improvement. To address this，this paper proposes a lightweight apple surface defect detection algorithm based on pruning. This algorithm uses YOLOv8n as the base model and incorporates the characteristics of GhostNetV2 and the C2f structure in YOLOv8 to design a C2f—GhostV2 module，significantly reducing the number of model parameters and accelerating inference speed. To further reduce computational load，GhostConv is introduced to replace traditional convolutions，and the DySample sampling mechanism is adopted to improve flexibility and information retention. Additionally，the lightweight model undergoes LAMP pruning，further decreasing the number of floating-point operations. Experimental results show that the pruned model achieves *mAP* of 97.3%，*FLOPs* reduced by 78.05%，and a frame rate increased by 27.85% compared with original model.

Keywords：apple；surface defects detection；lightweight；prune；small object detection

0 引言

在苹果的生长和采摘过程中会出现各种缺陷，影响供应链的不同环节[1]。这些缺陷不仅影响苹果的外观质量，还会缩短其保质期并降低市场价值。因此，苹果缺陷检测已成为一个重要的研究领域[2]。

近年来，深度学习技术被广泛应用于苹果缺陷检测和分级中。Lee 等[3]提出一种多摄像头的苹果分拣系统，该系统带有旋转机制以实现均匀成像，并采用卷积神经网络(CNN)分类器，其准确率达到 93.83%，推理时间为 0.069 s。Han 等[4]利用高光谱成像和深度学习来评估厚皮坚果的质量，准确率达到 93.48%。Hu 等[5]设计 ASDINet 用于苹果缺陷检测，通过 AU—Net[6]和全局决策模块(GDM)实现高精度和快速检测，但不适合嵌入式设备。Ismail 等[7]开发一种基于深度学习的机器视觉系统用于无损苹果检测，采用

收稿日期：2024 年 11 月 26 日 修回日期：2025 年 3 月 12 日
* 基金项目：新疆维吾尔自治区教育厅自然科学基金(2024D01C31)
第一作者：刘鹏扬，男，2000 年生，河南驻马店人，硕士研究生；研究方向为计算机视觉与图像处理。E-mail：17518629177@163.com
通讯作者：亚森江·木沙，男，1972 年生，乌鲁木齐人，博士，副教授；研究方向为计算机视觉与模式识别。E-mail：yasenjiangmusha@163.com

EfficientNet[8]网络架构实现 99.2%的准确率和 96.7%实时测试准确率。Karthikeyan 等[9]提出 YOLOAPPLE 模型，使用增强版的 YOLOv3[10]模型进行多类苹果检测，精度达到 99.13%，但并未解决推理速度问题。Xiao 等[11]研究发现，YOLOv8 的跨阶段部分融合（C2f）模块提升水果分类的准确率至 99.5%，但模型的速度和大小不适合实时检测。Fan 等[12]提出一种基于剪枝 YOLOv4[13]算法和近红外摄像头的实时苹果缺陷检测方法，实现在双通道分拣机上的高效检测。Xu 等[14]改进 YOLOv5 用于苹果分级，准确率达 90.6%，处理速度达 59.63 帧/s。尽管深度学习在苹果缺陷检测中的表现十分出色，但仍面临一些挑战。例如，许多深度学习模型需要高性能硬件支持，难以在资源受限的环境中部署。此外，一些模型在推理速度和模型大小上存在局限性，无法满足实时检测的要求。

为解决这一问题，本文提出一种基于 YOLOv8n 的轻量化改进模型 PGD—YOLOv8。该模型结合轻量化模块与基于动态权重的自适应上采样机制，并通过基于层自适应幅度的剪枝（LAMP）对改进后的模型进行压缩，实现检测效率与精度的有效平衡。首先，模型将 YOLOv8n 主干网络中的 CSPDarkNet53 经典卷积模块替换为更轻量的幽灵卷积（GhonstConv）。通过将 GhostNetV2 网络中的关键组件幽灵瓶颈改进版（GhostBottleneckV2）与 C2f 进行融合，在实现模型参数大幅减少的同时提高精度，加快模型的推理速度。其次，采用动态上采样机制（DySample）取代传统的上采样机制，增强模型检测的灵活性，并提高对边缘和小细节区域的处理能力。最后，采用 LAMP 剪枝方法对改进后的模型进行剪枝，进一步减少模型的浮点运算次数和参数量，使其能够轻松部署在资源受限的设备上。

1　YOLOv8 模型简介

YOLOv8 模型不仅可以提升检测精度，且性能表现出色，更加轻量化。YOLOv8 的主要改进包括引入 C2f 模块以替代传统的 C3 模块，增加并行梯度流分支，从而获取更丰富的梯度信息，使模型在特征提取方面表现更强，特别适合在复杂背景下识别细微特征，如苹果表面的斑点、裂纹等缺陷。此外，YOLOv8 在瓶颈层（Neck）中去除路径聚合网络（PAN）结构中上采样后的卷积运算，简化特征融合流程并提升效率，同时空间金字塔池化模块（SPP）被快速空间金字塔池化模块（SPPF）取代，在保持检测效果的前提下，执行时间减半，大幅提升速度。最后，预测头部采用双检测头的设计，进一步加快收敛速度，使模型在端到端预测中表现优异。

在苹果缺陷检测中，YOLOv8 表现出显著优势，特别是在精确捕捉细微特征和实时响应方面。通过改进的 C2f 模块，YOLOv8 能够在复杂光照条件下准确检测苹果表面的轻微瑕疵；SPPF 结构和双检测头设计提升处理速度和检测精度，确保在苹果分拣线上实现实时反馈。

与 SSD[15]和 RetinaNet[16]等一阶段检测器相比，YOLOv8 在速度和精度上更胜一筹。其 C2f 模块和 SPPF 结构不仅增强特征提取和多尺度检测能力，使其适应不同尺寸的苹果并精确识别表面缺陷，还通过优化内部结构降低误报率，提升识别准确性。此外，YOLOv8 在硬件兼容性和部署灵活性方面表现突出，无论是在高性能 GPU 还是资源受限的边缘设备上都能运行良好，是苹果缺陷检测任务中的理想选择。

2　YOLOv8 的算法改进

鉴于苹果缺陷检测的实时性要求，采用单阶段检测器 YOLOv8，并选择参数最少的 YOLOv8n 模型作为基础。尽管 YOLOv8n 具备较高的检测精度，但其骨干和颈部网络中堆叠的卷积层显著增加计算资源的消耗，影响实时推理速度和在边缘设备上的部署。改进后的 YOLOv8 结构如图 1 所示。

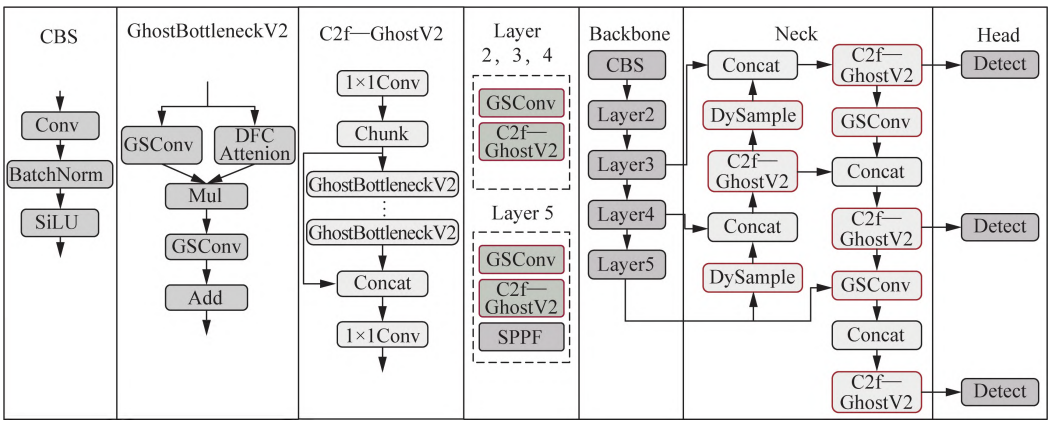


图 1　改进后的 YOLOv8 结构

Fig. 1　Improved YOLOv8 structure

对 YOLOv8n 进行轻量化改进，结合 GhostNetV2 的低计算开销和高推理速度，以及 YOLOv8n 中创新的 C2f 模块，设计一种 C2f—GhostV2 模块，有效提升推理速度并减少模型参数。为进一步减少计算成本，GhostConv 被引入作为传统卷积结构的替代方案，它能够显著降低参数量和内存占用，同时保持丰富的特征表达能力。此外，模型还采用基于 DySample 的动态范围提升机制，提升对图像细节、边缘及多尺度特征的适应性，特别适用于苹果表面缺陷的高密度预测任务。

2.1 GhostConv 模块

GhostConv 是由 Han 等[17]在幽灵网络(GhostNet)中引入的关键模块。GhostConv 是 GhostNet 架构中的核心操作，通过一种成本效益高的方法生成更多的特征图，减少计算和参数开销。如图 2 所示，GhostConv 将传统卷积分为两部分。首先，通过传统卷积使用较少的计算生成少量通道的特征图。然后，基于这些生成的特征图，GhostConv 通过一系列简单的线性操作(例如逐元素操作、乘法等)生成额外的“Ghost 特征图”。最后，将这两组特征图拼接在一起，生成最终的输出特征图。

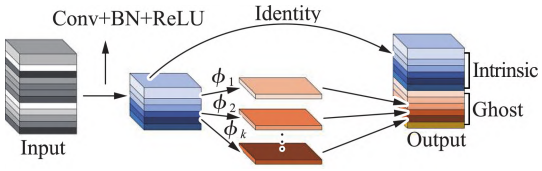


图 2 GhostConv 模型结构

Fig. 2 GhostConv module structure

假设输入特征图尺寸为 $H\times W\times C_{in}$，输出特征图尺寸为 $H\times W\times C_{out}$，卷积核大小为 $k\times k$，复杂度计算如式(1)～式(5)所示。

$$Complexity_{std}=H\cdot W\cdot C_{in}\cdot C_{out}\cdot k^2 \tag{1}$$

$$Complexity_{Conv}=H\cdot W\cdot C_{in}\cdot \frac{C_{out}}{r}\cdot k^2 \tag{2}$$

$$Complexity_{linear}=H\cdot W\cdot C_{out}\cdot \frac{r-1}{r} \tag{3}$$

$$Complexity_{GhostConv}=H\cdot W\cdot C_{in}\frac{C_{out}}{r}\cdot k^2+H\cdot W\cdot C_{out}\cdot \frac{r-1}{r} \tag{4}$$

$$\lambda=\frac{Complexity_{GhostConv}}{Complexity_{std}}=\frac{1}{r}+\frac{k^2\cdot(r-1)}{r\cdot C_{in}} \tag{5}$$

式中：r ——压缩比，$r>1$；

λ ——复杂度对比值；

$Complexity_{std}$ ——初始卷积模块复杂度；

$Complexity_{Conv}$ ——卷积操作模块复杂度；

$Complexity_{linear}$ ——线性操作模块复杂度；

$Complexity_{GhostConv}$ ——幽灵卷积模块复杂度；

H、W ——特征图的高和宽；

C_{in}、C_{out} ——输入通道数和输出通道数。

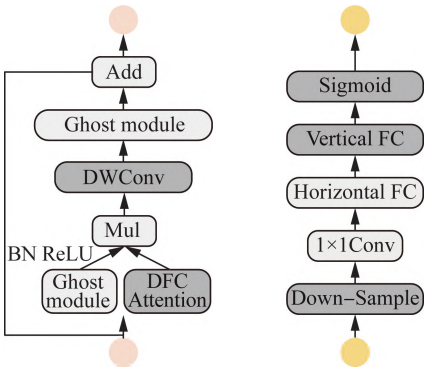
在实际苹果缺陷检测过程中，典型压缩比 $r=2$ 时，复杂度降低约为 50%。这种轻量化特性使 GhostConv 非常适合部署在计算资源有限的设备上，例如用于实时检测任务的嵌入式系统中。苹果缺陷检测是一项具有挑战的任务，要求模型能够高效处理复杂纹理并精准捕捉弱特征(如小斑点、细微裂痕)。GhostConv 在该任务中表现出显著的优势：其低计算复杂度显著加快推理速度，适合资源受限设备，满足实时检测的需求；同时，尽管 GhostConv 压缩部分特征，但消融实验表明，这些影响并未对检测精度产生显著影响，仍能保留足够的全局特征，尤其是在大范围缺陷(如裂纹、压痕)检测中表现出色。此外，降低计算复杂度后，模型能够处理更高分辨率的输入，从而提升对细节特征的解析能力。在理论上 GhostConv 的特征压缩可能导致部分高频特征和弱特征的表达能力略有削弱，但结果表明，其对检测精度的影响可以忽略不计，证明 GhostConv 的高效性和适应性。整体而言，GhostConv 在满足轻量化、实时性需求的同时，兼顾检测精度的稳定性，是苹果缺陷检测任务中一种极具优势的轻量化模块。

2.2 C2f—GhostV2 模块

在边缘设备上运行深度神经网络时，性能与推理速度的平衡至关重要。传统网络通常只能捕捉局部信息，难以显著提升性能，而注意力机制虽能捕捉全局信息，但会大幅增加计算复杂度和推理时间。为此，引入 GhostNetV2[18]，通过动态特征校准注意力机制(DFC Attention)，快速生成注意力图，捕捉像素之间的长距离依赖，不仅增强特征表示能力，还保持较低的计算成本。

C2f—GhostV2 模块通过 GhostBottleneckV2 模块的堆叠和特征融合，结合 Ghost 模块，深度卷积(Depthwise Convolution)[19]、DFC Attention 和残差连接(Shortcut)等技术，以轻量化和高效特征提取为设计核心，显著优化小目标检测能力和整体推理速度。图 3 中 GhostBottleneckV2 模块首先使用 Ghost 模块对输入特征进行轻量化处理，通过标准卷积生成主要特征，并通过线性变换生成轻量化的虚拟特征，显著降低计算复杂度，同时保留关键特征信息，为后续模块提供优化基础。随后，轻量化特征通过深度卷积进一步提取局部特征，逐通道卷积操作有效捕捉裂纹、色斑等小目标特征，避免全局特征对小目标的掩盖。在此基础上，DFC Attention 模块则通过全局平均池化(GAP)提取全局背景信息，并结合动态权重生成机制

对关键区域特征进行增强，使得模型对裂纹和疤痕等小目标的感知能力显著提升。随后，Shortcut 将输入特征直接叠加到增强后的特征中，实现全局信息和增强细节特征的高效融合。有效保留深层和浅层特征的互补性，增强模型对细微特征的表达能力。



(a) GhostBottleneckV2　(b) DFC Attention

图 3　GhostBottleneckV2 和 DFC Attention 模型结构

Fig. 3　GhostBottleneckV2 module and DFC Attention module structure

$$F_{Ghost}=F_{Conv}+\Phi(F_{Conv}) \tag{6}$$

$$F_{depthwise}=F_{Ghost}\times K_{depth} \tag{7}$$

$$F_{att}=\sigma(W_2\cdot \mathrm{ReLU}(W_1\cdot \mathrm{GAP}(F_{depthwise}))) \tag{8}$$

$$F_{Shortcut}=F_{input}+F_{att} \tag{9}$$

式中：F_{Ghost}——幽灵卷积生成的特征；

F_{Conv}——标准卷积生成的主要特征；

$\Phi(F_{Conv})$——线性变换操作；

F_{input}——输入特征；

$F_{depthwise}$——深度卷积生成的主要特征；

$F_{Shortcut}$——残差连接生成的主要特征；

K_{depth}——深度卷积核；

GAP(·)——全局平均池化；

F_{att}——DFC Attention 生成的通道注意力特征；

W_1、W_2——用于生成权重的全连接层；

σ——Sigmoid 函数。

图 4 中 C2f—GhostV2 模块通过堆叠多个 GhostBottleneckV2 模块构建完整的模块结构。

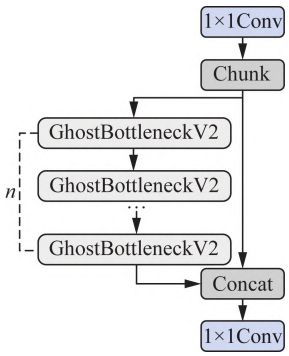


图 4　C2f—GhostV2 模块结构

Fig. 4　C2f—GhostV2 module structure

输入特征被分块(Chunk)为多个特征组，每个特征组分别由 GhostBottleneckV2 进行处理，输出局部增强特征。所有特征通过拼接操作整合为全局特征图，拼接后的特征通过 1×1 卷积进一步融合并输出最终特征。这一设计通过动态特征增强和深度卷积优化小目标检测的适配性，显著提升检测精度。

相比于传统 C2f 模块，C2f—GhostV2 在每个子模块(GhostBottleneckV2)中利用 Ghost 模块减少计算复杂度，使用深度卷积捕捉局部特征，通过 DFC Attention 增强小目标特征，再通过残差连接保留全局信息。这种设计使得 C2f—GhostV2 在小目标检测中的表现显著优于传统 C2f 模块。同时，C2f—GhostV2 的整体复杂度得到优化。传统 C2f 模块的浮点运算次数 *FLOPs* 计算如式(10)～式(13)所示。

$$F_{processed}=\mathrm{Concat}(F_{Ghost},F_{Shortcut}) \tag{10}$$

$$F_{output}=\mathrm{Conv1\times1}(\mathrm{Concat}(\mathrm{GhostBottleneckV2}(F_1),\mathrm{GhostBottleneckV2}(F_2),\cdots,\mathrm{GhostBottleneckV2}(F_n))) \tag{11}$$

$$FLOPs_{C2f}=m\cdot H\cdot W\cdot C\cdot C' \tag{12}$$

$$FLOPs_{C2f-GhostV2}=m\cdot(H\cdot W\cdot C\cdot C_{Ghost}+H\cdot W\cdot K_{depth}+H\cdot W\cdot C_{att}) \tag{13}$$

式中：$F_{processed}$——GhostBottleneckV2 模块的输出特征；

F_{output}——模型的最终输出特征；

Concat——特征连接操作；

F_1、F_2、…、F_n——n 个 GhostBottleneckV2 模块的输出特征；

$FLOPs_{C2f}$——C2f 模块的浮点运算次数；

$FLOPs_{C2f-GhostV2}$——C2f—GhostV2 模块的浮点运算次数；

m——特征图的数量；

C、C'——输入和输出特征图的通道数；

C_{Ghost}——幽灵卷积后的通道数；

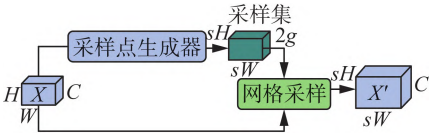
C_{att}——注意力通道数。

因为 $C_{Ghost}<<C$，C2f—GhostV2 模块在降低计算复杂度的同时，能够显著提升对小目标(如裂纹和色斑)的检测性能。

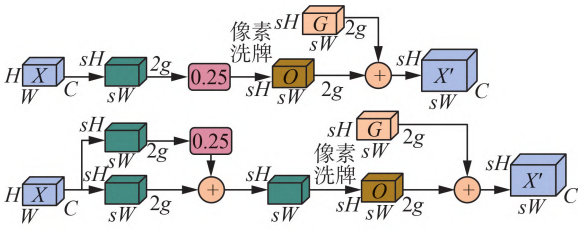
2.3　动态上采样

DySample[20]是一种高效且灵活的动态上采样方法，其通过点采样设计和动态范围因子实现高质量的上采样。与传统的基于卷积核的上采样方法不同，如图 5 所示，DySample 采用点采样方式，通过生成偏移量，将一个采样点分割为多个采样点，并动态调整这些点的位置。其动态范围因子使用 Sigmoid 函数并根据输入特征动态调整偏移范围，确保上采样过程的灵活

性和适应性。



(a) 基于采样的动态上采样



(b) DySample 中的采样点生成器

图 5 DySample 中基于采样的动态上采样和模块设计

Fig. 5 Sampling based dynamic upsampling and module designs in DySample

与传统的固定卷积上采样方法不同，DySample 采用两种模式：线性投影＋像素洗牌(Linear Projection ＋Pixel Shuffle，LP)和像素洗牌＋线性投影(Pixel Shuffle＋Linear Projection，PL)。其中，LP 先进行线性投影，再应用像素洗牌，计算效率高，适合资源受限的场景，能有效提升推理速度和节省内存。PL 先用像素洗牌提升分辨率，再通过线性投影细化特征，能更好地捕捉小目标的细节信息，适合对小物体检测精度要求较高的任务。

通过后续的对比试验结果，LP 与 PL 在提升推理速度和计算资源利用方面表现相近，但 PL 在小目标检测上效果更为显著。由于 PL 更擅长捕捉小目标的细微特征，因此在精度需求较高的场景中表现优异，适合作为检测方法。基于这一优势，最终选择 PL 作为上采样方法。

与 YOLOv8 的上采样结构相比，DySample 不仅提升分类和定位的精度，还显著减少推理时间，简化实现过程，尤其适用于快速苹果缺陷检测等实时任务。通过动态调整上采样点，DySample 能够在保持计算效率的同时，显著提升模型在密集预测任务中的性能。

2.4 LAMP 剪枝

尽管 YOLOv8 模型已经通过通道剪枝进行优化，以满足苹果缺陷检测任务在嵌入式设备上的应用需求，但在提升检测精度的同时，推理速度和浮点运算量仍未达到理想的要求。因此，需在现有优化的基础上，探索更多的模型压缩方法，以进一步降低计算需求和内存占用，从而达到更高的运行效率，使其更适合在嵌入式设备上实时运行。

在苹果缺陷检测任务中，部分通道对特征提取的重要性较低，例如与背景区域或常见颜色无关的通道。通过剪枝可以保留对缺陷检测更重要的特征通道，同时减少冗余计算。相较于传统的层剪枝，通道剪枝在保持模型精度的同时，显著降低硬件的计算压力。对通道剪枝算法进行对比后，选取 LAMP 作为苹果缺陷检测算法的剪枝方法。

LAMP 是由 Lee 等[21]提出的一种层次自适应权重大小剪枝算法，通过在每层选择不同的稀疏度来实现通道剪枝的自适应调节，提供一种通用的模型压缩方法。LAMP 通过重新缩放的权重大小近似剪枝，提出新的全局剪枝重要性评分，最小化模型失真，并自动确定每层的稀疏度，避免手动调整超参数的复杂性。LAMP 简化剪枝过程，同时确保高效的剪枝效果和模型性能。如图 6 所示，LAMP 评分用于衡量整个权重集内每个权重的相对重要性。

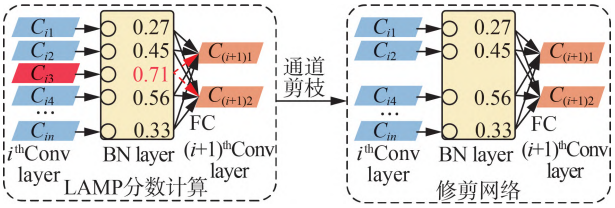


图 6 LAMP 剪枝方法

Fig. 6 LAMP pruning method

通过计算权重的绝对值并对其进行重新缩放，LAMP 评分近似剪枝后模型的失真。基于 LAMP 评分，对具有最高 LAMP 评分的连接进行全局剪枝，直到实现指定的全局稀疏目标。LAMP 算法无需超参数调整，计算效率高，且仅依赖于简单的张量操作即可完成。LAMP 评分计算如式(14)所示。

$$LAMP(i)=|W_i|\times\left(\frac{1}{\sum_j|W_j|}\right) \tag{14}$$

式中：$LAMP(i)$ ——第 i 个连接层的重要性度量；

W_i ——第 i 个连接层的权重参数；

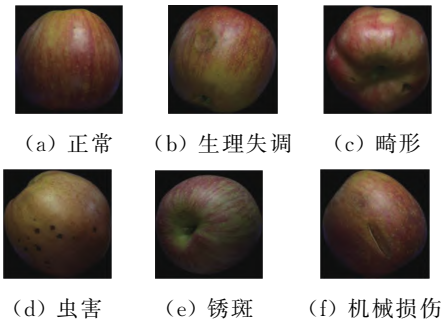
i、j ——索引值，表示不同的连接层。

3 试验结果与分析

3.1 试验数据集

使用的苹果表面缺陷(SDA)数据集[3]来自全南 ICT 融合系统工程实验室，用于苹果表面缺陷分级。苹果图像按是否有表面缺陷分为两类：正常苹果和表面有缺陷的苹果，如图 7 所示，缺陷包括生理失调、虫害、畸形、锈斑和划痕等。选取 1 433 张苹果 RGB 图像，按 8∶2 划分为训练集(1 146 张)和测试集(287 张)，无重叠。如图 8 所示，为提高模型泛化能力并减少过拟合，使用 8 种图像增强方法：最大池化、对比度受限的自适应直方图均衡(Clahe)、色温调整、椒盐噪声、透视变换、边缘增强、运动模糊和随机对比度[22]，

共生成 11 464 张图像。



(a) 正常　(b) 生理失调　(c) 畸形

(d) 虫害　(e) 锈斑　(f) 机械损伤

图 7　苹果缺陷类型

Fig. 7　Types of apple defects

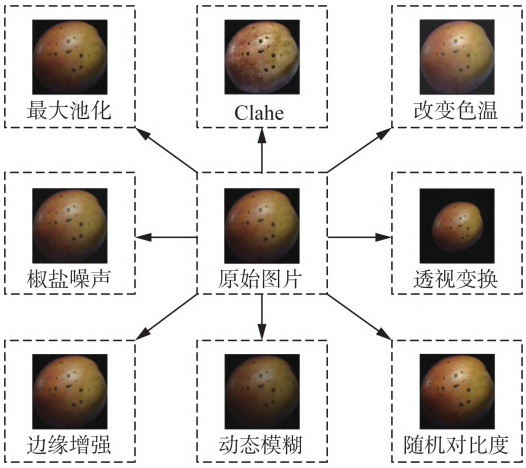


图 8　图像增强方法

Fig. 8　Image enhancement methods

3.2　试验环境与训练策略

试验训练过程中使用的硬件平台和环境参数如表 1 所示，试验中使用的关键参数如表 2 所示。

表 1　训练过程的试验环境

Tab. 1　Experimental environment for the training process

环境	配置
CPU	Intel Core i7－11800H
操作系统	Windows 11
GPU	NVIDIA GeForce RTX 3050 Ti
深度学习框架	PyTorch 1.12

表 2　训练参数设置

Tab. 2　Training parameter setting

参数	数值
轮次	300
批大小	16
输入图片大小/(像素×像素)	640×640
初始学习率	0.01
SGD 动量	0.937
权重衰减	0.000 5
优化器	SGD
学习率调整	余弘退火算法

3.3　评价指标

为评估改进模型的性能，使用的评估指标包括平均精度均值(mAP)、浮点运算次数($FLOPs$)和推理速度(FPS)。这些指标共同评估模型在保持高准确率的同时，是否能实现更小的尺寸、更低的计算复杂度和更高的实时性能。其中 mAP 计算如式(15)所示。

$$mAP=\frac{1}{N}\sum_{n=1}^{N}AP(n) \tag{15}$$

$$AP=\int_0^1 P(R)\mathrm{d}R \tag{16}$$

$$P=\frac{TP}{TP+FP} \tag{17}$$

$$R=\frac{TP}{TP+FN} \tag{18}$$

式中：TP——识别正确的正样本的数量；

FP——错误识别成正样本的负样本数量；

FN——错误识别成负样本的正样本数量；

AP——平均精度；

P——精确率；

R——召回率；

N——检测类别数量。

3.4　消融实验

在动态上采样的方法选择过程中，设计对比试验，以评估 PL 和 LP 两种模式在评估缺陷检测中的表现。分别应用 PL 和 LP 两种上采样方式于相同的 YOLOv8 模型中，保持其他模型参数一致，试验结果如表 3 所示。PL 在小目标检测的精度上优于 LP，上采样过程中对细小目标的特征捕捉更加精细。此外，PL 和 LP 在推理速度上表现大致相同，都符合嵌入式设备资源受限的应用需求。基于以上试验数据，最终选取 PL 作为改进模型的上采样方式，提升精度的同时实现较低的计算和内存占用，从而满足评估缺陷检测在嵌入式场景中的应用需求。

表 3　DySample 的 LP 与 PL 模式对比试验

Tab. 3　Comparison experiment of LP and PL modes in DySample

模型	mAP /%	$FLOPs$ /G	FPS /(帧·ms^{-1})
YOLOv8n	96.2	8.2	69.3
YOLOv8n＋DySample(LP)	96.5	7.9	72.3
YOLOv8n＋DySample(PL)	97.2	7.9	71.3

为验证提出的各个改进模块的有效性，进行一系列消融实验。将原 YOLOv8 作为基线模型，通过向 YOLOv8 模型中逐步加入改进模块进行消融实验，分别取实验中最优的模型在该测试集进行验证，结果如表 4 所示。其中 A 方法表示将 YOLOv8n 的 Backbone 和 Neck 中的 Conv 替换为 GhostConv，B 方

法表示将 YOLOv8n 中的所有 C2f 模块替换为 C2f—GhostV2 模块，C 方法表示将 Neck 部分的上采样替换为 DySample(PL)模块。可以看出，替换 GhostConv 模块后，*mAP* 几乎不变，但 *FLOPs* 与 *FPS* 效果提升明显。在此基础上采用 C2f—GhostV2 模块后，*mAP* 提升 0.6 个百分点，*FLOPs* 降低 2.6 G，*FPS* 提升 7 帧/ms，最后添加 DySample 模块后，*mAP* 提升 0.8 个百分点。另外，单独采用 C2f—GhostV2 模块时，*mAP* 提升 0.7 个百分点，同时 *FLOPs* 和 *FPS* 得到良好的改进；单独使用 DySample 模块时，*mAP* 提升 1 个百分点，推理速度同样有明显提升。可以看出，每个模块在单独应用时均对性能提升具有积极作用，而逐步添加多个模块则进一步优化模型的整体性能。

表 4 添加模块的消融实验

Tab. 4 Ablation experiments of sequentially adding modules

A	B	C	*mAP*/%	*FLOPs*/G	*FPS*/(帧·ms^{-1})
×	×	×	96.2	8.2	69.3
√	×	×	96.1	7.4	72.1
×	√	×	96.9	5.9	74.8
√	√	×	96.8	5.6	76.3
×	×	√	97.2	7.9	71.3
√	√	√	97.6	5.4	78.2

3.5 剪枝试验

剪枝方法与加速比的选取也是依靠试验方法，将不同剪枝速率以及剪枝方法加入改进 YOLOv8n(GD—YOLOv8n)中。试验中设定的剪枝参数如表 5 所示，硬件和环境参数如表 1 所示。

表 5 剪枝试验参数设置

Tab. 5 Pruning experiment parameter settings

参数	数值	参数	数值
全局剪枝	是	优化器	SGD
剪枝速率	2.0	*patience*	50
轮次	300	*close_mosaic*	10
批大小	8		

为验证 LAMP 剪枝方法的有效性和优势，与基于瘦化的剪枝(Slim)[23]、基于 Taylor 展开的组剪枝(Group_taylor)[24]、基于 L1 范数的剪枝(L1)[25]和基于 Hessian 矩阵的组剪枝(Group_Hessian)[26]等方法进行对比试验，结果如表 6 所示。LAMP 方法表现最佳，*mAP* 为 97.3%，*FPS* 为 88.6 帧/ms，*FLOPs* 为 1.8 G，完美地平衡速度、精度和计算复杂度。这表明 LAMP 剪枝在保持高精度的同时，还具备高推理速度和较小模型，适用于高精度和实时任务。

在模型剪枝中，剪枝速率(*speed-up*)指剪枝后模型推理速度相较于原始模型的提升倍数，是评估剪枝效果的重要指标之一。通过剪除冗余的通道、卷积核或网络层，剪枝可以显著减少模型的计算量和参数量，从而提升推理速度。为探究 LAMP 剪枝在不同剪枝速率下的剪枝效果，在剪枝过程中，分别设定剪枝速率为 2.0、2.5、3.0 和 3.5，对苹果缺陷检测模型的通道进行剪枝。由表 7 可知，当剪枝速率为 3.0 时，模型在平均精度均值、浮点运算次数和推理速度之间达到较好的平衡，模型浮点运算次数减少但精度无明显下降。图 9 呈现剪枝速率设置为 3.0 时，模型各层通道数的变化，每个柱状代表一层网络，黄色和红色分别代表剪枝前后该层的通道数。

表 6 不同剪枝方法的对比试验结果

Tab. 6 Comparative experimental results of different pruning methods

剪枝方法	*mAP*/%	*FLOPs*/G	*FPS*/(帧·ms^{-1})
Slim	96.3	1.8	86.3
Group_taylor	92.3	1.8	87.1
L1	94.9	1.9	85.1
Group_Hessian	95.3	1.7	89.1
LAMP	97.3	1.8	88.6

表 7 不同剪枝速率下的剪枝结果

Tab. 7 Pruning results under different *speed_up*

剪枝速率	*mAP*/%	*FLOPs*/G	*FPS*/(帧·ms^{-1})
2.0	97.6	2.5	85.1
2.5	97.5	1.9	86.4
3.0	97.3	1.8	88.6
3.5	94.8	1.5	90.4

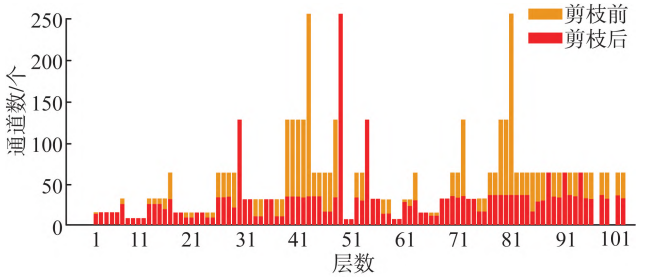


图 9 模型通道数剪枝前后对比

Fig. 9 Comparison of model before and after channel pruning

3.6 对比试验

为验证改进模型的先进性及其在苹果缺陷识别中的迁移能力，选取 Faster R—CNN[27]、SSD[15]、YOLOv5s、YOLOv7[28]、YOLOv9[29]、YOLOv10[30]、OGV—YOLOv5s[31]、VEW—YOLOv8n[32]、改进模型 GD—YOLOv8 以及 PGD—YOLOv8 在苹果缺陷数据集上进行对比试验，结果如表 8 所示。改进 PGD—YOLOv8 模型在苹果缺陷检测任务中表现卓越，在检测精度、浮点运算次数和推理速度 3 个关键指标上均

实现优异的性能。具体而言，PGD—YOLOv8 的检测精度达到 97.3%，相比于 YOLOv7—tiny、OGV—YOLOv5s、YOLOv5s、YOLOv9 和 YOLOv10 分别提升 2.6%、0.7%、1.1%、1.1%和 1.0%，相比于两阶段目标检测算法 Faster R—CNN 和 SSD 分别提升 9.3%和 6.1%，仅次于 GD—YOLOv8(97.6%)。在计算复杂度(*FLOPs*)方面，PGD—YOLOv8 的 *FLOPs* 仅为 1.8 G，远低于两阶段目标检测算法 Faster R—CNN 和 SSD，相比于 YOLOv7—tiny、OGV—YOLOv5s、YOLOv5s、YOLOv9 和 YOLOv10，分别降低 72.3%、84.2%、89.1%、83.18% 和 73.13%，相比于 VEW—YOLOv8 和 GD—YOLOv8 分别降低 69.5%和 66.7%，显示其极低的计算资源需求。在推理速度(*FPS*)方面，PGD—YOLOv8 达到 88.6 帧/ms，相比于 YOLOv7—tiny、OGV—YOLOv5s、YOLOv5s、YOLOv9 和 YOLOv10 分别提升 25.3%、18.0%、35.71%、36.94%和 24.3%，相比于 Faster R—CNN 和 SSD 分别提升 129.84%和 119.05%，是所有对比模型中推理速度最快的。总体而言，PGD—YOLOv8 在检测精度、浮点运算次数和推理速度方面实现最佳平衡，展示其在苹果缺陷实时检测中的广阔应用前景。

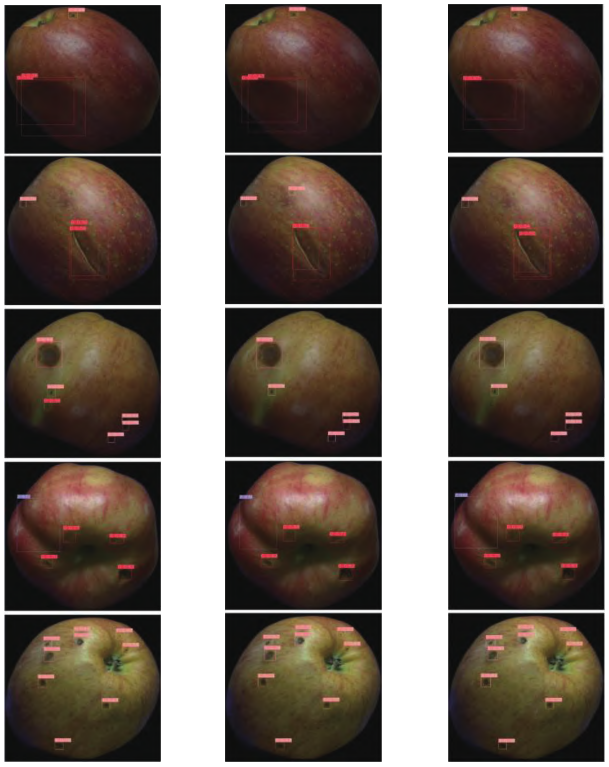
表 8　不同模型对比试验

Tab. 8　Comparison experiments of different models

模型	*mAP*/%	*FLOPs*/G	*FPS*/(帧·ms^{-1})
Faster R—CNN	88.4	251.4	38.6
SSD	91.2	35.4	40.4
YOLOv5s	96.7	16.5	65.3
YOLOv7—tiny	94.7	6.5	70.7
YOLOv8s	96.6	28.6	65.9
YOLOv8n	96.2	8.2	69.3
YOLOv9	96.4	10.7	64.7
YOLOv10n	96.3	6.7	71.3
OGV—YOLOv5s	96.6	11.4	75.1
VEW—YOLOv8n	97.2	5.9	73.8
GD—YOLOv8	97.6	5.4	78.2
PGD—YOLOv8	97.3	1.8	88.6

3.7　可视化检测结果

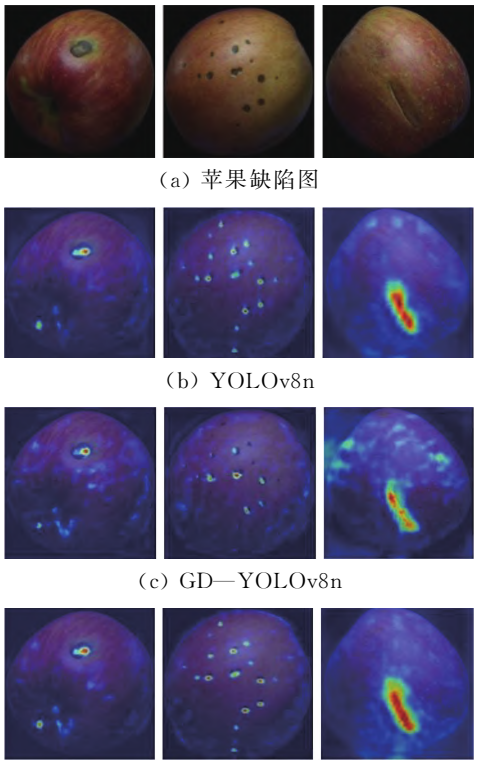
为展示改进模型的检测能力，对比原始模型 YOLOv8n、轻量化模型 GD—YOLOv8n 以及剪枝后的改进模型 PGD—YOLOv8n 的检测图和可视化热力图。如图 10 所示，改进模型的预测置信度明显高于原始 YOLOv8n 模型，表明其对小目标的检测能力更强，例如细小挫伤、虫害以及划痕等。同时在检测颜色区别不大的瘀伤与畸形区域时，检测效果更明显。由图 11 可知，与相同的苹果缺陷相比，改进模型的特征轮廓更清晰，对目标位置的响应更强，表明改进模型的特征提取效果更好，显示出其在检测小苹果缺陷方面的能力更强。



(a) YOLOv8n　(b) GD—YOLOv8n　(c) PGD—VOLOv8n

图 10　改进前后检测效果对比

Fig. 10　Comparison of detection results before and after improvements



(a) 苹果缺陷图

(b) YOLOv8n

(c) GD—YOLOv8n

(d) PGD—YOLOv8n

图 11　改进前后可视化热力图对比

Fig. 11　Improved before and after visualization of heat map comparisons

4 结论

1）消融实验结果表明，GhostConv 模块、C2f—GhostV2 模块和 DySample 模块的联合使用显著提升基于 YOLOv8 的网络性能。所有模块均不同程度地降低模型的计算复杂度。添加 GhostConv 模块后，*mAP* 几乎保持不变；而添加 C2f—GhostV2 模块和 DySample 模块后，模型的检测精度分别提升 0.5% 和 1%。逐个添加这些模块后，相比于原始 YOLOv8 基线模型，*mAP* 提升 1.4 个百分点，*FLOPs* 从 8.2 G 降至 5.4 G，推理速度提升 8.9 帧/ms。

2）剪枝试验结果表明，与 Slim、Group_taylor、L1 和 Group_Hessian 剪枝方法相比，LAMP 剪枝方法在苹果表面缺陷检测任务中表现更为优异。采用 LAMP 剪枝后的改进模型的 *mAP* 达到 97.3%，*FLOPs* 为 1.8 G，*FPS* 为 88.6 帧/ms，不仅显著降低计算复杂度，还在精度与推理速度的平衡方面表现优秀，有利于在计算性能较低的嵌入式设备或边缘设备上运行，从而降低苹果表面缺陷检测的成本。

3）对比试验结果显示，改进模型 PGD—YOLOv8 在苹果缺陷检测中表现卓越，超越 YOLOv7、YOLOv5 等模型，在检测精度上表现更好，且在计算复杂度和推理速度方面显著优化。与传统的两阶段检测算法 Faster R—CNN 和 SSD 相比，PGD—YOLOv8 不仅大幅降低计算资源需求，还提高推理速度，达到 88.6 帧/ms。

PGD—YOLO 能够在短时间内实现快速、准确的苹果缺陷检测，具有较高的准确性和较低的计算资源需求，特别适合在嵌入式设备中进行实时部署，满足农业现场环境的需求，具备显著的实用价值。未来的研究将进一步扩展算法的应用范围，优化数据集，以提高对不同类型和形态的苹果缺陷的检测能力。同时，将通过增强数据多样性和优化算法，提升算法在复杂背景和不同环境条件下的鲁棒性与准确性，推动该技术在智慧农业领域的广泛应用。

参 考 文 献

[1] Musacchi S, Serra S. Apple fruit quality: Overview on pre-harvest factors [J]. Scientia Horticulturae, 2018, 234: 409－430.

[2] Li J, Luo W, Wang Z, et al. Early detection of decay on apples using hyperspectral reflectance imaging combining both principal component analysis and improved watershed segmentation method [J]. Postharvest Biology and Technology, 2019, 149: 235－246.

[3] Lee J H, Vo H T, Kwon G J, et al. Multi-camera-based sorting system for surface defects of apples [J]. Sensors, 2023, 23(8): 3968.

[4] Han Y, Liu Z, Khoshelham K, et al. Quality estimation of nuts using deep learning classification of hyperspectral imagery [J]. Computers and Electronics in Agriculture, 2021, 180: 105868.

[5] Hu Z, Tang J, Zhang P, et al. Deep learning for the identification of bruised apples by fusing 3D deep features for apple grading systems [J]. Mechanical Systems and Signal Processing, 2020, 145: 106922.

[6] Oktay O, Schlemper J, Folgoc L L, et al. Attention U—Net: Learning where to look for the pancreas [J]. arXiv preprint arXiv: 1804.03999, 2018.

[7] Ismail N, Malik O A. Real-time visual inspection system for grading fruits using computer vision and deep learning techniques [J]. Information Processing in Agriculture, 2022, 9(1): 24－37.

[8] Tan M, Le Q. EfficientNet: Rethinking model scaling for convolutional neural networks [C]. International Conference on Machine Learning. PMLR, 2019: 6105－6114.

[9] Karthikeyan M, Subashini T S, Srinivasan R, et al. YOLOAPPLE: Augment YOLOv3 deep learning algorithm for apple fruit quality detection [J]. Signal, Image and Video Processing, 2024, 18(1): 119－128.

[10] Redmon J. YOLOv3: An incremental improvement [J]. arXiv preprint arXiv: 1804.02767, 2018.

[11] Xiao B, Nguyen M, Yan W Q. Fruit ripeness identification using YOLOv8 model [J]. Multimedia Tools and Applications, 2024, 83(9): 28039－28056.

[12] Fan S, Liang X, Huang W, et al. Real-time defects detection for apple sorting using NIR cameras with pruning-based YOLOV4 network [J]. Computers and Electronics in Agriculture, 2022, 193: 106715.

[13] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection [J]. arXiv preprint arXiv: 2004.10934, 2020.

[14] Xu B, Cui X, Ji W, et al. Apple grading method design and implementation for automatic grader based on improved YOLOv5 [J]. Agriculture, 2023, 13(1): 124.

[15] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector [C]. European Conference on Computer Vision. Cham: Springer International Publishing, 2016: 21－37.

[16] 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.

[17] 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.

[18] Tang Y, Han K, Guo J, et al. GhostNetv2:

Enhance cheap operation with long-range attention [J]. Advances in Neural Information Processing Systems, 2022, 35: 9969－9982.

[19] Chollet F. Xception: Deep learning with depthwise separable convolutions [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 1251－1258.

[20] Liu W, Lu H, Fu H, et al. Learning to upsample by learning to sample [C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023: 6027－6037.

[21] Lee J, Park S, Mo S, et al. Layer-adaptive sparsity for the magnitude-based pruning [J]. arXiv preprint arXiv: 2010.07611, 2020.

[22] Yan B, Fan P, Lei X, et al. A real-time apple targets detection method for picking robot based on improved YOLOv5 [J]. Remote Sensing, 2021, 13 (9): 1619.

[23] Liu Z, Li J, Shen Z, et al. Learning efficient convolutional networks through network slimming [C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 2736－2744.

[24] Molchanov P, Mallya A, Tyree S, et al. Importance estimation for neural network pruning [C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 11264－11272.

[25] Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network [J]. Advances in Neural Information Processing Systems, 2015, 28.

[26] Scardapane S, Comminiello D, Hussain A, et al. Group sparse regularization for deep neural networks [J]. Neurocomputing, 2017, 241: 81－89.

[27] Girshick R. Fast R—CNN [C]. Proceedings of the IEEE International Conference on Computer Vision, 2015: 1440－1448.

[28] 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]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023: 7464－7475.

[29] Wang C Y, Yeh I H, Liao H Y M. YOLOv9: Learning what you want to learn using programmable gradient information [C]. European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024: 1－21.

[30] Wang A, Chen H, Liu L, et al. YOLOv10: Real-time end-to-end object detection [J]. arXiv preprint arXiv: 2405.14458, 2024.

[31] Ji W, Wang J, Xu B, et al. Apple grading based on multi-dimensional view processing and deep learning [J]. Foods, 2023, 12(11): 2117.

[32] Han B, Lu Z, Dong L, et al. Lightweight non-destructive detection of diseased apples based on structural re-parameterization technique [J]. Applied Sciences, 2024, 14(5): 1907.

(上接第 99 页)

[15] Gao G, Wang C, Wang J, et al. CNN—Bi—LSTM: A complex environment-oriented cattle behavior classification network based on the fusion of CNN and Bi—LSTM [J]. Sensors, 2023, 23(18): 7714.

[16] Yin X, Wu D, Shang Y, et al. Using an EfficientNet—LSTM for the recognition of single cow's motion behaviors in a complicated environment [J]. Computers and Electronics in Agriculture, 2020, 177: 105707.

[17] 方俊泽，郭正，李歌，等. 基于改进 Swin—Transformer 的柑橘病叶分类模型[J]. 中国农机化学报，2024，45 (1)：252－258.
Fang Junze, Guo Zheng, Li Ge, et al. Classification model of citrus disease leaf based on improved Swin—Transformer [J]. Journal of Chinese Agricultural Mechanization, 2024, 45(1): 252－258.

[18] 刘拥民，刘翰林，石婷婷，等. 一种优化的 Swin Transformer 番茄叶片病害识别方法[J]. 中国农业大学学报，2023，28(4)：80－90.
Liu Yongmin, Liu Hanlin, Shi Tingting, et al. Tomato leaf disease recognition base on an optimized Swin Transformer [J]. Journal of China Agricultural University, 2023, 28(4): 80－90.

[19] 谢光达，李洋，曲洪权，等. 基于改进 Transformer 的小目标车辆精确检测算法[J]. 激光与光电子学进展，2022，59(18)：364－371.
Xie Guangda, Li Yang, Qu Hongquan, et al. Small target accurate vehicles detection algorithm based on improved Transformer [J]. Laser & Optoelectronics Progress, 2022, 59(18): 364－371.

[20] Lin T Y, Dollar 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.