

黄贻望,王国帅,毛 志. KMeans++与注意力机制融合的苹果叶片病害识别方法[J]. 江苏农业科学,2024,52(20):190-198.  
doi:10.15889/j.issn.1002-1302.2024.20.023

# KMeans++与注意力机制融合的苹果叶片病害识别方法

黄贻望<sup>1,2,3</sup>, 王国帅<sup>1</sup>, 毛 志<sup>1,2</sup>, 刘 声<sup>1</sup>

(1. 铜仁学院大数据学院/经济管理学院, 贵州铜仁 554300; 2. 民族教育信息化教育部重点实验室/云南师范大学, 云南昆明 650000;  
3. 铜仁学院乡村振兴研究中心, 贵州铜仁 554300)

**摘要:**为解决复杂环境下小尺度苹果叶片病害识别精度不高、鲁棒性不强的问题,在 YOLO v5s 的基础上提出一种新的改进方法。该方法首先在模型训练之前使用 KMeans++ 聚类算法生成更接近真实框的锚框;其次在骨干网络中加入卷积块注意力模块(convolutional block attention module,CBAM),来提升复杂环境下小目标特征的提取能力;再次为了增强颈部网络对不同大小病害多尺度特征的有效识别,选择 ConvNeXtBlock 模块替换 C3(CSP bottleneck with 3 convolutions)模块,并在颈部网络中融入坐标注意力模块(coordinate attention,CA),来加强模型对关键空间位置的响应,使得不同尺度的特征都能被更有效地利用;最后使用 ECIoU 损失函数替换原始的 CIoU 损失函数,来提高模型的收敛速度和精度。与 Faster R-CNN、SSD、YOLO v5s、YOLO v7、YOLO v8 目标检测模型相比,改进后模型的平均精度均值(mean average precision, $mAP_{0.5}$ )值分别提升 0.6、4.6、6.3、1.7、1.3 百分点,同时在强光照、模糊、暗光的复杂场景下具有较强的鲁棒性。该模型可以为复杂环境下苹果叶片病害的识别提供行之有效的方案。

**关键词:**苹果;叶片病害;病害检测;注意力机制;ConvNeXtBlock;卷积块注意力模块(CBAM);CA

**中图分类号:**S126;TP391.41 **文献标志码:**A **文章编号:**1002-1302(2024)20-0190-09

中国苹果产业协会发布的《2022 年度中国苹果产业报告》中显示,中国苹果产量仍然位居世界第一<sup>[1]</sup>。中国作为苹果种植业大国,在苹果产业发展中依旧存在一些问题。例如,在种植过程中,苹果的叶片病害严重影响着苹果的产量,包括对苹果危害较为严重的腐烂病、褐斑病、斑点落叶病等。那么,如何快速精准地检测和识别苹果叶部病害就是解决这一问题的关键,对于未来苹果产业高质量发展至关重要。

随着科技的进步,越来越多的技术运用到农业生产领域,深度学习凭借其强大的模式识别和自适应学习能力被广泛应用于番茄<sup>[2-4]</sup>、玉米<sup>[5-7]</sup>、小

麦<sup>[8-9]</sup>以及茶叶<sup>[10]</sup>等农作物病害的检测识别任务中,在苹果叶部病害检测中也进行了很多研究,例如,曾晏林等在 YOLO v5 中引入自注意力机制和 Transformer 模块,对于苹果叶部病害进行有效识别<sup>[11]</sup>;Mathew 等使用 YOLO v3 算法来检测苹果叶部病害<sup>[12-13]</sup>;Wang 等通过改进 YOLO v5 并轻量化网络,对多种苹果叶部病害达到较高的检测速度<sup>[14]</sup>;赵嘉威等将 YOLO v4 网络结合注意力机制模块来检测苹果多种叶部病害<sup>[15]</sup>;黄炜等基于 VGG-16 和 Swin Transformer 网络提出了一种复杂特征的细粒度特征分离方法,有效提升了检测精准率<sup>[16]</sup>;李鑫然等通过改进 Faster R-CNN 模型来提高对于苹果叶片病害的检测精度<sup>[17]</sup>;公徐路等使用 ShuffleNet v2 轻量化网络对 YOLO v5 进行改进,来达到对苹果叶片病害检测的高精度及轻量化的需求<sup>[18]</sup>;王瑞鹏等利用 EfficientNet-B0 网络和 DenseNet121 网络作为特征提取网络,对 5 类苹果叶部病害进行检测<sup>[19]</sup>;孙长兰等采用非极大值抑制算法将 YOLO v5 与 EfficientDet 模型进行集成来提升苹果叶部病害的检测精度<sup>[20]</sup>。这些研究采用不同的方式改进模型来提升检测精度,但由于苹果树生长的环境复杂,树叶繁茂,枝杈交错,容易造成病害

收稿日期:2024-07-12

基金项目:国家自然科学基金(编号:62066040);民族教育信息化教育部重点实验室开放课题(编号:EIN2024B003);智能计算与信息处理教育部重点实验室开放课题(编号:2023ICIP05);国家留学基金委西部地区人才培养特别项目(编号:202108525007);铜仁市大数据智能计算与应用重点实验室项目(编号:铜仁市科研[2022]5号);贵州省科技基础研究计划(编号:[2022]557)。

作者简介:黄贻望(1978—),男,湖南溆浦人,博士,教授,硕士生导师,主要从事人工智能、服务计算、软件形式化方法等研究。  
E-mail:yjsyhyw@gztrc.edu.cn。

特征的混淆,所以大多数模型在病害检测与识别任务中,鲁棒性往往较差。除此之外,苹果叶片病害中锈病、灰斑病、蛙眼叶斑病等病斑由于在整个叶片背景中尺寸较小,缺乏相应的纹理、颜色、形状等外观信息<sup>[21]</sup>,多次下采样后,特征信息减少,导致检测难度变大。

因此,本研究在 YOLO v5s 模型的基础上提出了一种新的改进算法。该算法主要包括以下 4 点改进:(1) 使用 KMeans ++ 聚类算法生成新的锚框,使锚框更接近数据集标注的大小,从而提高模型定位的准确性;(2) 在骨干网络引入卷积块注意力模块(convolutional block attention module, CBAM),增强骨干网络对复杂环境下小尺度苹果叶片病害特征的提取能力;(3) 在颈部网络使用 ConvNeXtBlock 模块替换原有的 C3 (CSP bottleneck with 3 convolutions) 模块,增强模型对特征的表达能力,并在颈部网络中加入坐标注意力(coordinate attention, CA) 模块,进一步强化模型对于关键空间位置的响应速度;(4) 为了使模型收敛速度更快,用 ECIoU 损失函数替换原始的 CIoU 损失函数。通过以上改进以满足对于复杂环境下小尺度苹果叶片病害识别任务的可用性。

## 1 材料与方法

### 1.1 试验数据集与处理

苹果叶片病害试验数据集来自于公开数据集 AppleLeaf9<sup>[22]</sup>,此数据集包含 PlantVillage、ATLDS、PPCD2020、PPCD2021 数据集中的 8 种苹果叶部病害<sup>[23-26]</sup>,本试验选择对苹果产量影响较为严重的链格孢叶斑病、蛙眼叶斑病、灰斑病及锈病 4 种病害图片,共计 1 514 张。为提升模型的鲁棒性和泛化性能,试验数据集选择的病害图片包括复杂场景以及小目标病害图片,4 种病害图片如图 1 所示。

试验使用 Labelme 标注软件对图片上的每一处病害进行标注,标注类别和标注框的坐标信息存储在对应的 JSON 文件中。试验将数据集按照 8:2 的比例划分训练集和验证集,具体划分如表 1。

### 1.2 YOLO v5 整体改进

首先,本改进模型使用 KMeans ++ 聚类算法生成新的锚框,其目的在于使生成的锚框更接近数据集中目标的真实大小。数据集标注的大小是指数据集中所有标注目标框的宽度和高度的统计分布。通过聚类算法,可以有效地生成 1 组锚框,这些锚

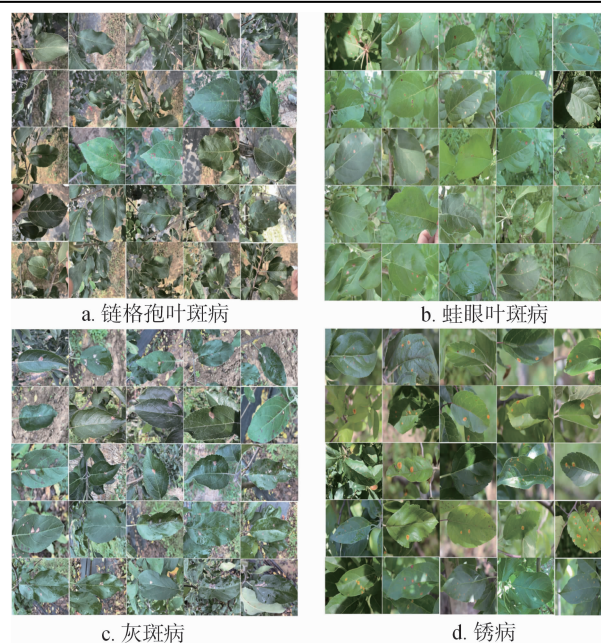


图1 4 种苹果叶片病害图片数据集

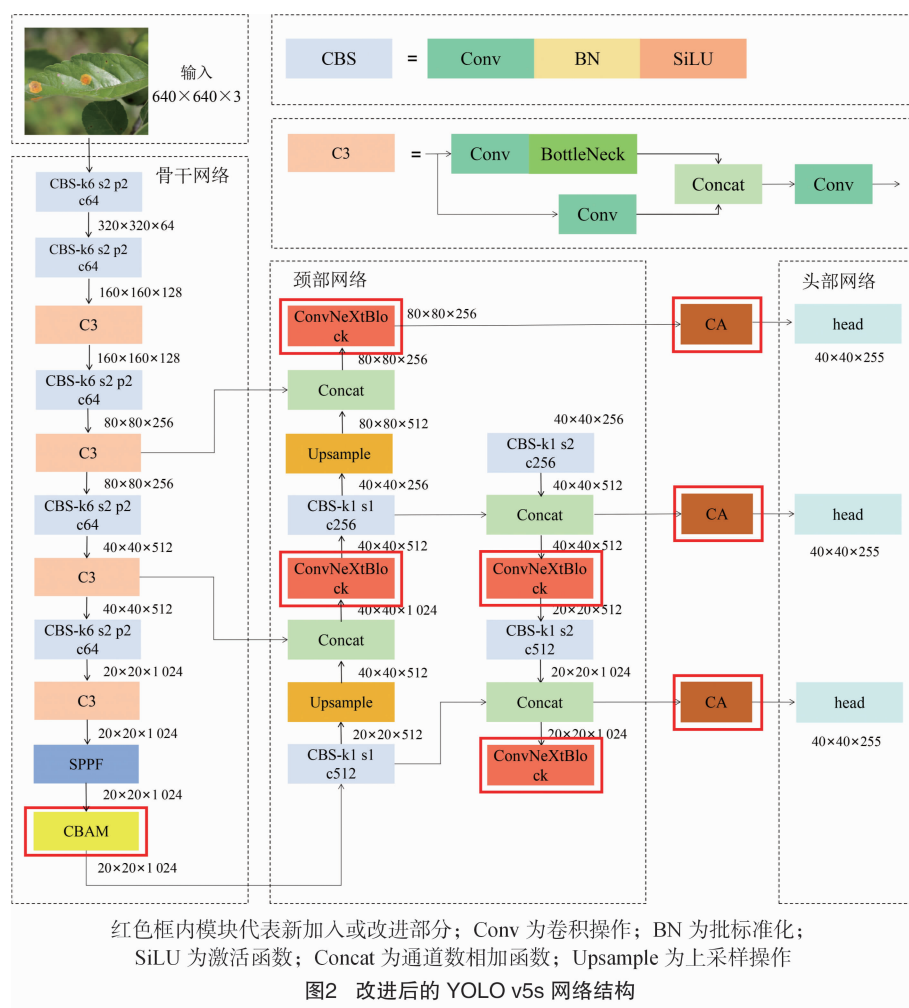
表 1 4 种苹果叶片病害图片数据集划分

病害种类	标注名称	图片数量(张)		
		训练集	验证集	总数
链格孢叶斑	A	315	75	390
蛙眼叶斑病	F	280	69	349
锈病	R	271	68	339
灰斑病	G	349	87	436

框的尺寸能够更好地代表数据集中目标的实际尺寸分布,从而在训练过程中使模型更容易学习到目标的特征提高检测精度。其次,在骨干网络中引入卷积块注意力模块(convolutional block attention module, CBAM),CBAM 通过卷积操作在特征图上计算注意力权重,这些权重可以分为通道注意力和空间注意力。重要特征是指那些对目标识别至关重要的特征,如病害斑点的纹理、形状、颜色等。在通道注意力机制下,CBAM 根据每个通道的重要性分配权重,强化对目标识别有帮助的通道特征;在空间注意力机制下,CBAM 根据每个空间位置的重要性分配权重,强化对目标识别有帮助的空间特征,抑制背景噪声和无关区域的特征。这种处理可以提高模型对重要特征的关注度,从而增强判别能力。在颈部网络使用 ConvNeXtBlock 模块替换原有的 C3 模块,ConvNeXtBlock 是一个改进的卷积模块,它通过更深的卷积操作和更广的感受野来捕捉多尺度特征。ConvNeXtBlock 采用一种新颖的网络架构设计,包括更深层次的卷积、跳跃连接以及归一化层,这些设计能够更好地捕捉和融合不同尺度

的特征信息。通过这种改进,模型在不同尺度下的特征表达能力得到了显著提升,使其能够更准确地识别大小不同的目标,从而提高目标检测的准确性和鲁棒性。模型还在颈部网络中引入了 CA 模块,该模块是一种基于坐标的注意力机制,它通过将注意力权重与坐标信息结合,能够更精准地定位目标区域并强化目标特征。CA 模块的核心思想是利用

坐标信息引导注意力权重的计算,从而更好地捕捉目标的空间布局和上下文关系。这样模型不仅能够关注到目标本身,还能考虑目标周围的环境特征,提高目标检测的鲁棒性。最后,为使模型收敛速度更快,使用 ECIoU 损失函数替换原始的 CIoU 损失函数。本研究改进的 YOLO v5s 模型网络结构如图 2 所示。



**1.2.1 KMeans++ 聚类算法生成锚框** 在 YOLO v5 中,锚框是一种预设的矩形框,用在特征图上预测目标的位置和大小。本方法采用 KMeans++ 聚类算法生成锚框,其目的是使生成的锚框更接近数据集中目标的真实大小。YOLO v5s 默认使用 3 个不同的锚框,锚框的大小为  $[10, 13, 16, 30, 33, 23]$ 、 $[30, 61, 62, 45, 59, 119]$ 、 $[116, 90, 156, 198, 373, 326]$ 。其中  $[10, 13, 16, 30, 33, 23]$  等这些数字表示锚框的宽度和高度。具体来说,  $[10, 13]$  中的 10 表示锚框的宽度,13 表示锚框的高度。其他锚框的数字组合也是如此,分别表示不同锚框的宽度和高

度。这 3 个不同大小的锚框对应  $80 \times 80$ 、 $40 \times 40$ 、 $20 \times 20$  等 3 个不同尺度的特征图。但给定的锚框是使用 MSCOCO 数据集聚类得出的结果<sup>[27]</sup>, MSCOCO 数据集有 80 多个类别,试验中病害类别并不包含在内。虽然 YOLO v5s 引入了自适应锚框算法,但算法依旧存在局限性。自适应锚框算法在模型训练前需遍历整个数据集的标注信息,根据这些信息计算标注框的尺寸分布。若计算得到的最优召回率大于或等于 0.98,则不进行进一步的更新;反之,若最优召回率低于 0.98,则触发锚框尺寸的重新计算和更新流程。



为使在训练之前得到更符合真实标注框的锚框,本研究选择使用 KMeans++ 聚类算法生成新的锚框,从而优化模型训练过程。KMeans++ 聚类生成锚框如算法 1 所示。

算法 1 KMeans++ 生成锚框:

输入:数据集  $D$ , 聚类数量  $k$ ;

输出:锚框的中心点集合  $C$ 。

(1)初始化中心点集合  $C$  为空集;

(2)从数据集  $D$  中随机选择 1 个样本作为初始聚类中心,并加入到  $C$  中;

(3)while  $|C| < k$  do;

(4)对  $D$  中每个样本  $x$  计算最小距离平方  $d(x)^2$ ;

(5)以  $d(x)^2$  成正比概率选择新聚类中心,并加入  $C$ ;

(6)根据  $C$  中  $k$  个中心点确定锚框宽高;

(7)返回锚框中心点集合  $C$ 。

通过 KMeans++ 聚类算法生成锚框的具体大小如表 2。

表 2 KMeans++ 聚类算生成锚框尺寸

尺寸	锚框大小
小	$15 \times 22, 19 \times 32, 24 \times 40$
中	$32 \times 34, 34 \times 46, 41 \times 59$
大	$66 \times 68, 59 \times 96, 105 \times 121$

1.2.2 引入注意力机制 注意力机制融入到目标识别领域<sup>[28]</sup>,可以帮助模型更加专注于图像中的关键部分,从而提高目标检测的准确性。目前已有多种注意力机制如 EMA (exponential moving average)<sup>[29]</sup>、SA (spatial attention)<sup>[30]</sup>、CA (coordinate attention)<sup>[31]</sup>、CBAM 等被广泛应用于目标检测中<sup>[32]</sup>。具体到本研究,为使模型在特征提取阶段增强对缺乏纹理颜色的小目标病害的提取能力,使得不同尺度的病害特征都能被有效地利用。选择在骨干网络中加入 CBAM 模块,在颈部网络加入 CA 模块,CBAM 模块由 Woo 等提出,结构如图 3 所示。

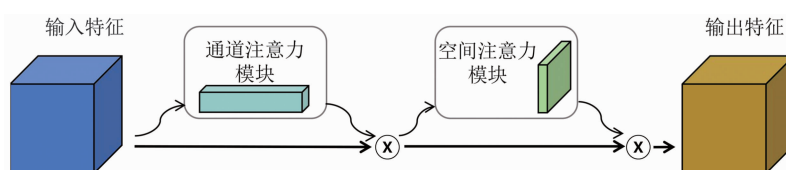


图3 CBAM 结构

CBAM 模块主要包括 CAM 与 SAM 两个部分,模块首先通过通道注意力机制 (channel attention module, CAM) 增强网络对不同通道特征的表达能

力,随后通过空间注意力机制 (spatial attention module, SAM) 进一步细化每个位置的特征响应。其中通道注意力模块 (CAM) 结构如图 4 所示。

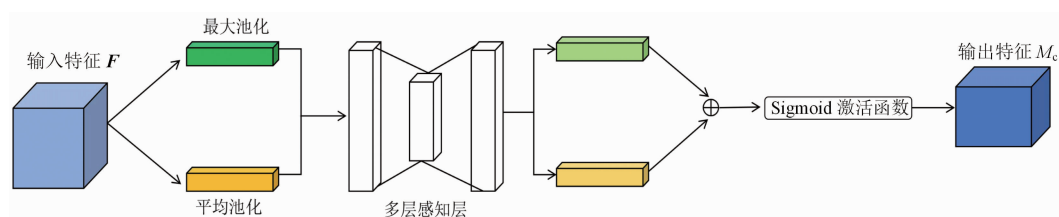


图4 CAM 结构

输入特征图  $F$  是前置卷积层的输出,包含多个通道,每个通道编码了不同的特征信息。CAM 首先对特征图  $F$  进行全局最大池化和全局平均池化操作。全局最大池化操作捕获特征图中最为显著的特征,而全局平均池化则提取特征图中的平均信息。池化后的结果被送入共享的多层感知器 (multilayer perceptron, MLP), MLP 具有 2 层全连接层,并在这 2 层之间使用 ReLU 激活函数。共享的 MLP 对 2 种池化信息进行进一步的抽象和重构,生

成具有描述性的特征通道权重。经过 MLP 处理后的 2 组特征图进行逐元素相加,通过 Sigmoid 激活函数处理,生成通道注意力图  $M_c$ 。得到的通道注意力图  $M_c$  为每个通道指定 1 个权重,通过这个权重对输入特征图  $F$  中每个通道的响应进行放大或缩减。通道注意力计算公式为式(1):

$$M_c(F) = \sigma \{ MLP[ \text{Avg}(F) ] + MLP[ \text{MaxPool}(F) ] \} = \sigma [ W_1 (W_0(F_{\text{avg}})) + W_1 [ W_0(F_{\text{max}}) ] ] \quad (1)$$

通道注意力这样的机制使得网络能够强化对

重要特征通道的关注,而抑制不重要的通道,进而更有效地利用特征信息。SAM 则关注于在空间维

度上的重要区域,以强化局部的特征表达。SAM 的结构如图 5 所示。

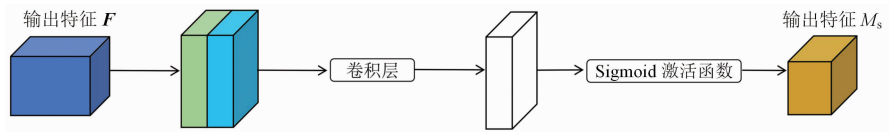


图5 SAM 结构

在 CAM 模块之后, SAM 模块首先对细化后的特征图  $F'$  执行最大池化和平均池化操作。池化操作后得到的 2 个特征图在通道维度上进行堆叠, 形成 1 个两通道的特征图, 堆叠后的特征图通过 1 个  $7 \times 7$  的卷积层进行处理。通过卷积层后, 使用 Sigmoid 函数激活得到空间注意力图  $M_s$ 。最后, 空间注意力图  $M_s$  被用来对输入的通道细化特征图  $F'$  进行加权。空间注意力权重矩阵  $M_s$  计算公式为式 (2)。

$$M_s(F) = \sigma(f^{7 \times 7} \{ [\text{AvgPool}(F); \text{MaxPool}(F)] \}) = \sigma[f^{7 \times 7} ([F_{\text{avg}}^s; F_{\text{max}}^s])]. \quad (2)$$

除了在骨干网络加入 CBAM 模块之外, 为进一步强化模型对于关键空间位置的响应, 使得不同尺度的特征都能被更有效地利用。本研究选择在颈部网络融入 CA 模块, 其网络结构如图 6 所示。

首先输入的是具有 1 个  $C \times H \times W$  维度的特征图, 包含  $C$  个通道的特征图分别在  $x$  轴和  $y$  轴上进行平均池化操作。经过池化操作的特征图被拼接起来, 并通过 1 个二维卷积 (Conv2d) 操作, 拼接完的特征图经过批量归一化和非线性函数处理。之后被拆分成两部分, 并分别通过另外的卷积层进行处理, Sigmoid 函数将卷积的结果映射到  $(0, 1)$  区间, 生成权重。再次使用 Sigmoid 函数对原始输入特征图进行加权, 突出重要的通道特征。

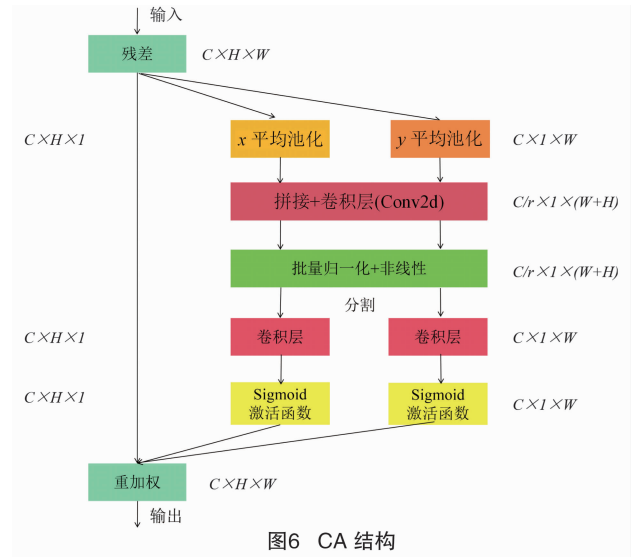


图6 CA 结构

**1.2.3 ConvNeXtBlock 结构** 当病害特征被骨干网络提取后, 在颈部网络进行特征融合, 但在深层网络中随着卷积的增加, 特征图的分辨率逐步降低, 这使小目标信息丢失, 尤其是在处理苹果叶片病害识别这种细粒度的任务中, 而且因不同阶段不同大小的病害可能需要多尺度的特征来进行有效识别, 因此本研究选择使用 ConvNeXtBlock 模块替换颈部网络中的 C3 模块, ConvNeXt 结构由 Liu 等提出<sup>[33]</sup>, 它是基于 Transformer 架构上的卷积神经网络模块, ConvNeXt 网络结构如图 7 所示。



图7 ConvNeXt 网络结构

模块首先通过 1 个  $7 \times 7$  的深度可分离卷积处理输入特征图, 紧接着通过层标准化进行归一化操作, 这一步有助于加速整个训练过程并加快模型的收敛速度。之后使用  $1 \times 1$  卷积来聚合深度可分离卷积后的特征, 并进行通道上的信息融合。这一步使通道数量变为原来的 4 倍, 增强了网络对于局部特征的捕捉能力, ReLU 激活函数的加入给予了模

型的非线性。接着再次使用  $1 \times 1$  卷积来进一步调整特征。最后输入的特征图与最后一次卷积的输出进行残差连接输出结果。

**1.2.4 损失函数改进** 损失函数的改进可以进一步提升模型的收敛速度和精度, YOLO v5s 默认将 CIoU 作为损失函数。CIoU 的设计主要考虑的关键因素包括边框的重叠面积、中心点距离以及长宽

比。CIoU 损失通过增加中心点距离,提供了当 IoU 相同时更有区分度的梯度。CIoU 的计算公式为:

$$\text{CIoU} = \text{IoU} - \frac{p^2(b, b^{\text{gt}})}{c^2} \alpha \nu. \quad (3)$$

式中:  $\alpha, \nu$  分别作为权重函数和宽高比的度量值,其中  $\alpha$  的计算公式为式(4),  $\nu$  的计算公式为式(5)。

$$\alpha = \frac{\nu}{(1 - \text{IoU}) + \nu}; \quad (4)$$

$$\nu = \frac{4}{\pi} \left( \arctan \frac{\omega^{\text{gt}}}{h^{\text{gt}}} - \arctan \frac{\omega}{h} \right)^2. \quad (5)$$

式中:  $\omega, h$  是真实框的宽和高;  $\omega^{\text{gt}}, h^{\text{gt}}$  是预测框的宽和高。但  $\alpha, \nu$  反映的只是边界框长、宽的差异,并没有反映出宽、高对于置信度的差异,且当  $\omega = k\omega^{\text{gt}}, h = kh^{\text{gt}}$  惩罚项将失去效果。EIoU 在此基础上将纵横比的影响因子拆分,分别计算真实框和预测框的长、宽, EIoU 损失函数主要包括重叠损失、中心距离损失、宽高损失 3 个部分<sup>[34]</sup>。EIoU\_Loss 的计算公式为:

$$\text{Loss}_{\text{EIoU}} = L_{\text{IoU}} + L_{\text{dis}} + l_{\text{asp}} = 1 - \text{IoU} + \frac{p^2(b^{\text{gt}}, b)}{c^2} + \frac{p^2(h^{\text{gt}}, h)}{c_h^2} + \frac{p^2(\omega^{\text{gt}}, \omega)}{c_\omega^2}. \quad (6)$$

本研究结合了 2 种损失函数的优点选择使用 ECIoU 作为损失函数,在此前的研究中,已有研究者将此损失函数应用到目标识别领域,如曹超等在 YOLO v8 中使用 ECIoU 损失函数来加速模型收敛<sup>[35]</sup>, ECIoU 损失函数计算公式为:

$$\text{Loss}_{\text{ECIoU}} = 1 - \text{IoU} + \alpha \nu + \frac{p^2(b^{\text{gt}}, b)}{c^2} + \frac{p^2(h^{\text{gt}}, h)}{c_h^2} + \frac{p^2(\omega^{\text{gt}}, \omega)}{c_\omega^2}. \quad (7)$$

### 1.3 算法流程和分析

改进后的算法具体包括:在训练前使用 KMeans++ 聚类算法生成新锚框、在骨干网络中加入 CBAM 模块、在颈部网络使用 ConvNeXtBlock 模块替换 C3 模块并引入 CA 模块、使用 ECIoU 损失函数替换 CIoU 损失函数。改进后算法的具体训练流程为算法 2 所示。

算法 2 改进后的算法流程:

输入:训练数据集、模型参数;

输出:目标检测结果。

(1)初始化改进后的 YOLO v5s 模型;

(2)New anchorKMeans++ /\* 使用 KMeans++ 算法生成新锚框 \*/;

(3)for  $i = 0$  to  $N - 1$ ;

(4)trainNew anchor & ECIoU; /\* 使用改进后的锚框和损失函数进行训练 \*/;

(5)通过正向传播生成预测框,然后根据预测和真实标签计算损失,损失计算公式为式(7);

(6)通过式(7)计算结果进行反向传播,优化模型权重;

(7)for end。

首先初始化改进后的模型实例,并加载模型参数,包括加载预训练的权重、设置学习率、迭代次数以及批处理大小。随后模型使用优化后的锚框和改进后的损失函数,在数据集上进行多次迭代训练。在训练中模型接收输入图像并执行正向传播来生成预测框,然后根据标注框和预测结果之间的差异,计算损失函数。计算得出损失之后,使用反向传播算法来计算梯度,最后使用这些梯度来更新模型权重。

### 1.4 试验环境

为了验证本算法的有效性,于 2024 年 2—4 月在铜仁市大数据智能计算及应用重点实验室铜仁学院人工智能服务器上进行试验。试验系统为 Ubuntu,试验硬件为浪潮高性能人工智能平台,配置为: Intel® Xeon® Platinum 8358 PCPU@2.60 GHz, A40GPU × 2, 512 GB 内存,单张加速卡显存为 45 GB 共 90 GB。代码编译环境为 Python 3.8.10,加速卡 CUDA 版本为 11.2,深度学习框架选择 Pytorch 2.2.1。

### 1.5 模型评价指标

在苹果叶片病害的识别任务中,选择使用的评价指标有精准率(precision,  $P$ )、召回率(recall,  $R$ )以及平均精准率  $mAP$  值,精准率计算公式为式(8),召回率计算公式为式(9)。平均精准率  $mAP$  值是基于精准率和召回率得出的,可以更全面地反映模型的性能。为进一步测试模型全面的精度性能指标,在本试验中纳入了  $mAP_{0.5:0.9}$  的计算,  $mAP_{0.5:0.9}$  不仅考虑到了模型在容易的情况(如交并比  $\text{IoU} = 0.5$ )下的性能,还评估了在更困难的情况(如  $\text{IoU} = 0.9$ )下的表现。

$$P = \frac{TP}{TP + FP}; \quad (8)$$

$$R = \frac{TP}{TP + FN}. \quad (9)$$

其中:  $TP$  代表为真正例,指模型正确地将正类别的

样本识别为正类别; $FP$  为假正例,是指模型错误地将负类别的样本识别为正类别; $FN$  为假负例,是指模型错误地将正类别的样本识别为负类别。

## 2 结果与分析

### 2.1 损失函数对比

通过对损失函数的改进,使用  $ECIoU$  损失函数替换原始的  $CIoU$  损失函数, $ECIoU$  损失函数结合了  $EIoU$  和  $CIoU$  的优点,既保证了惩罚项的有效性,又加快了模型的收敛速度,训练结果如图 8 所示, $ECIoU$  损失函数在 Bounding box 的回归速度上要优于  $CIoU$  损失函数。

### 2.2 消融试验结果

为保证研究算法的 4 点改进是有效的,在其他

参数不变的条件下进行了 4 种改进方式的消融试验,结果表明每一种改进对模型的精度都有所提升,详细的试验结果如表 3 所示。

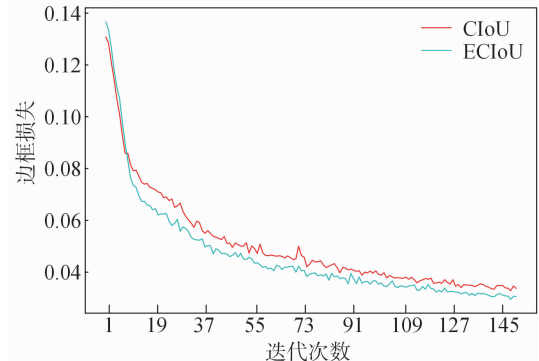


图8  $ECIoU$  与  $CIoU$  回归对比

表 3 改进方法的消融实验结果

改进方法	KMeans ++	ECIOU	CBAM + CA	ConvaNeXt Block	准确率 (%)	召回率 (%)	$mAP_{0.5}$ (%)	$mAP_{0.5:0.9}$ (%)
YOLO v5s	—	—	—	—	83.6	84.2	85.4	46.1
改进 1	✓	—	—	—	83.4	84.2	85.8	46.6
改进 2	—	✓	—	—	83.7	84.1	86.1	47.5
改进 3	—	—	✓	—	83.7	85.9	86.2	47.7
改进 4	—	—	—	✓	86.7	84.9	87.1	48.2
本研究算法	✓	✓	✓	✓	85.9	85.5	91.7	51.3

注:“✓”代表在 YOLO v5s 中加入了此模块;“—”代表在 YOLO v5s 中未加入此模块。

“改进 1”是在原始的 YOLO v5s 模型基础上使用 KMeans ++ 聚类算法生成预测锚框,改进 1 使得模型的  $mAP_{0.5}$ 、 $mAP_{0.5:0.9}$  分别提升了 0.4、0.5 百分点。“改进 2”是在在原始的 YOLO v5s 模型的基础上使用  $ECIoU$  损失函数替换原始的  $CIoU$  损失函数,改进 2 使得模型的  $mAP_{0.5}$ 、 $mAP_{0.5:0.9}$  分别提升了 0.7、1.4 百分点。“改进 3”在 YOLO v5s 的骨干网络 SPPF 层后加入 CBAM 模块,在颈部网络加入 CA 模块。对比原始 YOLO v5s 网络,其准确率、召回率分别提升 0.1、1.7 百分点, $mAP_{0.5}$ 、 $mAP_{0.5:0.9}$  分别提升 0.8、1.6 百分点。“改进 4”使用 ConvNeXt 模块替换颈部网络中的 C3 模块,相比较于原始 YOLO v5s 模型, $mAP_{0.5}$ 、 $mAP_{0.5:0.9}$  分别提升 1.7、2.1 百分点。最终改进模型也就是本研究算法,试验结果对比原始 YOLO v5s 模型准确率、召回率分别提升 2.3、1.3 百分点, $mAP_{0.5}$ 、 $mAP_{0.5:0.9}$  分别提升 6.3、5.2 百分点,其各类指标也优于其他 3 种改进方式。

### 2.3 对比试验

为了更进一步证明本算法对其他模型的有效

性,试验将本算法与一阶段目标检测算法 SDD、YOLO v7、YOLO v8、两阶段目标检测算法 Faster R-CNN 进行对比,结果如表 4 所示。本研究算法  $mAP_{0.5}$  值要高于其他一阶段目标识别算法和两阶段算法。链格孢叶斑病、蛙眼叶斑病、锈病的识别准确率要优于其他模型;但在灰斑病的识别中两阶段目标检测算法 Faster R-CNN 尽管拥有微弱的优势,但其庞大的参数量和计算代,价依旧没有办法使其与本研究算法相媲美。

### 2.4 对比分析

为更加直接观察模型在复杂环境下的识别效果,本试验还通过对比部分病害图片的检测结果来评估模型的有效性。在结果对比试验中,将本模型与原始的 YOLO v5s 模型进行对比测试,结果如图 9 所示。图 9-a 为原始的 YOLO v5s 检测结果,可以明显看出原始 YOLO v5s 模型对于 A 种病斑的识别置信度较低,对于 R 种小尺寸病斑识别效果较差,存在漏检的情况。而在图 9-b 中可以看到,改进后的 YOLO v5s 模型对于以上病斑识别的置信度远高



表 4 不同检测模型对比

模型	$mAP_{0.5}$ (%)	参数量 (MB)	准确率(%)			
			链格孢叶斑病	蛙眼叶斑病	灰斑病	锈病
Faster R - CNN	91.1	137.2	86.6	82.8	88.4	88.6
SSD	87.1	24.2	81.3	80.5	86.2	86.7
YOLO v7	90.0	6.0	84.3	84.2	88.1	87.3
YOLO v8	90.4	11.2	85.1	83.4	87.5	87.4
YOLO v5s	85.4	7.0	84.1	81.5	83.2	85.2
本研究算法	91.7	33.0	85.3	84.8	88.2	89.3

于 YOLO v5s 模型,在小目标病害识别的场景中没有出现漏检的情况。为验证本算法对复杂场景下的鲁棒性,图 10 展示了病害在强光照、模糊、暗光 3 种非正常环境下的检测结果。结果表明,原始的 YOLO v5s 模型在强光照、模糊的场景下出现了多处漏检情况,而改进后的模型未出现漏检,且检测置信度要远高于 YOLO v5s 模型。图 9 和图 10 的 2 组试验结果证明了改进后的算法对复杂场景和小目标病斑识别的有效性。



图 9 改进前后检测结果对比

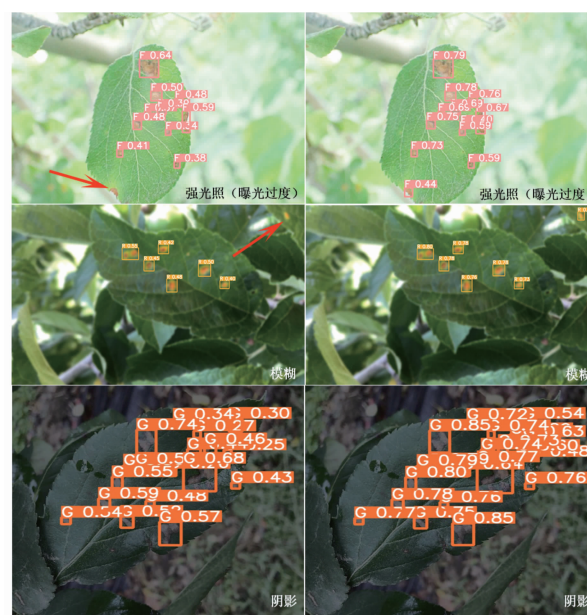


图 10 不同环境下检测效果对比

行识别,通过对原始的 YOLO v5s 模型的多个网络部分进行改进,改进后的模型与对比 Faster R - CNN、SSD、YOLO v7、YOLO v8 和 YOLO v5s 目标检测模型, $mAP_{0.5}$ 值分别提升 0.6、4.6、1.7、1.3、6.3 百分点。对于小尺寸病害锈病的检测准确率提升 0.7、2.6、2.1、9、4.1 百分点,具有较高的识别精度。并通过在强光照、模糊、暗光 3 种非正常环境下与 YOLO v5s 模型进行对比测试,改进后的模型表现出显著优势,鲁棒性较强。可以应用于复杂场景和小目标病斑的识别场景。

但检测模型要达到实际的应用水平,只针对这 4 种病害还远远不够,对于精确识别生长在不同时间阶段的苹果叶片病害也是亟需解决的问题。所以,后续的试验将加入不同阶段的多种病害和更复杂场景下的病害图片进行研究,继续提高模型的识别精度。

### 3 结论与讨论

本研究针对复杂环境下小尺度苹果叶片病害进



## 参考文献:

- [1] 王 壹. 中国苹果产量稳居世界第一[N]. 农民日报, 2023-11-18(7).
- [2] Zhang Y K, Huang S J, Zhou G X, et al. Identification of tomato leaf diseases based on multi-channel automatic orientation recurrent attention network[J]. Computers and Electronics in Agriculture, 2023, 205: 107605.
- [3] Gadekallu T R, Rajput D S, Reddy M P K, et al. A novel PCA-whale optimization-based deep neural network model for classification of tomato plant diseases using GPU[J]. Journal of Real-Time Image Processing, 2021, 18(4): 1383-1396.
- [4] Zhang D Y, Huang Y, Wu C, et al. Detecting tomato disease types and degrees using multi-branch and destruction learning[J]. Computers and Electronics in Agriculture, 2023, 213: 108244.
- [5] 鲍文霞, 黄雪峰, 胡根生, 等. 基于改进卷积神经网络模型的玉米叶部病害识别[J]. 农业工程学报, 2021, 37(6): 160-167.
- [6] Zeng W H, Li H D, Hu G S, et al. Lightweight dense-scale network (LDSNet) for corn leaf disease identification[J]. Computers and Electronics in Agriculture, 2022, 197: 106943.
- [7] Qian X F, Zhang C Q, Chen L, et al. Deep learning-based identification of maize leaf diseases is improved by an attention mechanism: self-attention[J]. Frontiers in Plant Science, 2022, 13: 864486.
- [8] 毛 锐, 张宇晨, 王泽玺, 等. 利用改进 Faster-RCNN 识别小麦条锈病和黄矮病[J]. 农业工程学报, 2022, 38(17): 176-185.
- [9] 马超伟, 张 浩, 马新明, 等. 基于改进 YOLO v8 的轻量化小麦病害检测方法[J]. 农业工程学报, 2024, 40(5): 187-195.
- [10] 李伟豪, 詹 炜, 周 婉, 等. 轻量型 YOLOv7-TSA 网络在茶叶病害检测识别中的研究与应用[J]. 河南农业科学, 2023, 52(5): 162-169.
- [11] 曾晏林, 贺壹婷, 蔺 瑶, 等. 基于 BCE-YOLO v5 的苹果叶部病害检测方法[J]. 江苏农业科学, 2023, 51(15): 155-163.
- [12] Mathew M P, Yamuna Mahesh T. Determining the region of apple leaf affected by disease using YOLO v3[C]//2021 International Conference on Communication, Control and Information Sciences. Idukki: IEEE, 2021: 1-4.
- [13] Liu B, Zhang Y, He D J, et al. Identification of apple leaf diseases based on deep convolutional neural networks[J]. Symmetry, 2017, 10(1): 11.
- [14] Wang Y W, Wang Y J, Zhao J B. MGA-YOLO: a lightweight one-stage network for apple leaf disease detection[J]. Frontiers in Plant Science, 2022, 13: 927424.
- [15] 赵嘉威, 田光兆, 邱 畅, 等. 基于改进 YOLO v4 算法的苹果叶片病害检测方法[J]. 江苏农业科学, 2023, 51(9): 193-199.
- [16] 黄 炜, 王娟娟, 殷学丽. 基于特征分离的小样本苹果病害叶片检测[J]. 江苏农业科学, 2023, 51(23): 195-202.
- [17] 李鑫然, 李书琴, 刘 斌. 基于改进 Faster R-CNN 的苹果叶片病害检测模型[J]. 计算机工程, 2021, 47(11): 298-304.
- [18] 公徐路, 张淑娟. 基于改进 YOLO v5s 的苹果叶片小目标病害轻量化检测方法[J]. 农业工程学报, 2023, 39(19): 175-184.
- [19] 孙长兰, 林海峰. 一种基于集成学习的苹果叶片病害检测方法[J]. 江苏农业科学, 2022, 50(20): 41-47.
- [20] 王瑞鹏, 陈锋军, 朱学岩, 等. 采用改进的 EfficientNet 识别苹果叶片病害[J]. 农业工程学报, 2023, 39(18): 201-210.
- [21] 李红光, 于若男, 丁文锐. 基于深度学习的小目标检测研究进展[J]. 航空学报, 2021, 42(7): 100-118.
- [22] 龙 阳, 肖小玲. 基于多注意力机制的苹果叶部病害检测方法[J]. 江苏农业科学, 2023, 51(23): 178-186.
- [23] Yang Q, Duan S K, Wang L D. Efficient identification of apple leaf diseases in the wild using convolutional neural networks[J]. Agronomy, 2022, 12(11): 2784.
- [24] Hughes D P, Salathe M. An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing[EB/OL]. (2016-04-12)[2023-06-10]. <https://arxiv.org/abs/1511.08060>.
- [25] Feng J Z, Chao X F. Apple tree leaf disease segmentation dataset[DS/OL]. (2022-03-31)[2023-06-10]. <https://www.scidb.cn/en/detail?dataSetId=0e1f57004db842f99668d82183afd578>.
- [26] Thapa R, Zhang K, Snaveley N, et al. The Plant Pathology Challenge 2020 data set to classify foliar disease of apples[J]. Applications in Plant Sciences, 2020, 8(9): e11390.
- [27] Chen X L, Fang H, Lin T Y, et al. Microsoft coco captions: data collection and evaluation server[J/OL]. [2024-03-15]. <https://arxiv.org/abs/1504.00325>.
- [28] Loureiro C, Filipe V, Gonçalves L. Attention mechanism for classification of melanomas[C]//International Conference on Optimization, Learning Algorithms and Applications, 2022: 65-77.
- [29] Li X, Zhong Z S, Wu J L, et al. Expectation-Maximization Attention Networks for Semantic Segmentation[C]//International Conference on Computer Vision, 2019.
- [30] Guo M H, Liu Z N, Mu T J, et al. Beyond self-attention: external attention using two linear layers for visual tasks[C]//IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021: 5436-5447.
- [31] Hou Q, Zhou D, Feng J. Coordinate Attention for Efficient Mobile Network Design[C]//2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 13708-13717.
- [32] Woo S, Park J, Lee J Y, et al. CBAM: convolutional block attention module[C]//European Conference on Computer Vision. Cham: Springer, 2018: 3-19.
- [33] Liu Z, Mao H Z, Wu C Y, et al. A ConvNet for the 2020s[C]//2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. New Orleans, 2022: 11966-11976.
- [34] Zhang Y F, Zhang Z, Jia Z, et al. Focal and efficient IoU loss for accurate bounding box regression[J]. Neurocomputing, 2022, 506(28): 146-157.
- [35] 曹 超, 顾幸生. 基于改进 YOLO v8 的行李追踪技术[J]. 计算机工程与应用, 2024, 60(9): 151-158.