doi:10.6041/j.issn.1000-1298.2024.03.023

基于改进 YOLO v7 轻量化模型的自然果园环境下苹果识别方法

张　震　周　俊　江自真　韩宏琪
（南京农业大学工学院，南京 210031）

摘要：针对自然果园环境下苹果果实识别中，传统的目标检测算法往往很难在检测模型的检测精度、速度和轻量化方面实现平衡，提出了一种基于改进 YOLO v7 的轻量化苹果检测模型。首先，引入部分卷积（Partial convolution，PConv）替换多分支堆叠模块中的部分常规卷积进行轻量化改进，以降低模型的参数量和计算量；其次，添加轻量化的高效通道注意力（Efficient channel attention，ECA）模块以提高网络的特征提取能力，改善复杂环境下遮挡目标的错检漏检问题；在模型训练过程中采用基于麻雀搜索算法（Sparrow search algorithm，SSA）的学习率优化策略来进一步提高模型的检测精度。试验结果显示：相比于 YOLO v7 原始模型，改进后模型的精确率、召回率和平均精度分别提高 4.15、0.38、1.39 个百分点，其参数量和计算量分别降低 22.93% 和 27.41%，在 GPU 和 CPU 上检测单幅图像的平均用时分别减少 0.003 s 和 0.014 s。结果表明，改进后的模型可以实时准确地识别复杂果园环境中的苹果，模型参数量和计算量较小，适合部署于苹果采摘机器人的嵌入式设备上，为实现苹果的无人化智能采摘奠定了基础。

关键词：苹果识别；自然果园环境；YOLO v7；PConv；高效通道注意力机制；麻雀搜索算法
中图分类号：TP391.4　　**文献标识码**：A　　**文章编号**：1000-1298(2024)03-0231-12　　**OSID**：

Lightweight Apple Recognition Method in Natural Orchard Environment Based on Improved YOLO v7 Model

ZHANG Zhen　ZHOU Jun　JIANG Zizhen　HAN Hongqi
(*College of Engineering*, *Nanjing Agricultural University*, *Nanjing* 210014, *China*)

Abstract: In the task of apple recognition in natural orchard environments, it is difficult for traditional object detection algorithms to achieve a balance between the accuracy, speed, and lightweight of the detection model. Therefore, a lightweight apple detection model based on improved YOLO v7 model was proposed. Firstly, partial convolution (PConv) was introduced in the multi branch stacking module to replace regular convolution, in order to reduce the parameter and computation of the model. Then a lightweight efficient channel attention (ECA) module was introduced to enhance the feature extraction ability and improve the problem of false and missed detection of occluded targets in complex environments. Finally, a learning rate optimization strategy based on sparrow search algorithm (SSA) was adopted in model training to further increase the detection accuracy of the model. The experimental results showed that compared with the original YOLO v7 model, the precision, recall, and average accuracy of the improved model was raised by 4.15 percentage points, 0.38 percentage points and 1.39 percentage points respectively; the number of parameters and computations were decreased by 22.93% and 27.41%, respectively; and the average time to detect each image under GPU and CPU was decreased by 0.003 s and 0.014 s, respectively. The results indicated that the improved model can quickly and accurately detect apple fruits in natural orchard environments, and the number of parameters and computations were less, which was suitable to be deployed on the embedded devices of apple harvesting robots, and laying the foundation for unmanned intelligent apple picking.

Key words: apple recognition; natural orchard environments; YOLO v7; PConv; efficient channel attention module; sparrow search algorithm

收稿日期：2023-08-06　修回日期：2023-10-21
基金项目：江苏省重点研发计划项目（BE2017370）和国家自然科学基金项目（31471419）
作者简介：张震（1992—），男，博士生，主要从事农业采摘机器人研究，E-mail：673308156@qq.com
通信作者：周俊（1974—），男，教授，博士生导师，主要从事农业装备智能化技术和农业机器人研究，E-mail：zhoujun@njau.edu.cn

0 引言

苹果具有极高的经济和食用价值,在全世界被广泛种植[1]。然而,苹果采摘作业依然主要依靠手工,耗时耗力,且需要高昂的劳动成本[2]。因此,大力研发和推广苹果自动化采摘装备,可以节约劳动成本、提高生产效率并促进苹果产业的发展[3-4]。尽管苹果自动采摘装备不断更新迭代,但苹果果实的识别检测依然是一个薄弱环节,成为制约采摘机器人在实际果园采摘中推广应用的主要因素[5]。果园自然环境的复杂性和水果的非结构化特征是苹果收获机器人目标检测面临的主要问题。

果实检测速度及准确率是影响机器人高效采摘的关键因素。传统目标检测算法以手工特征提取与分类器相结合为基础,严重依赖于水果形态、颜色和纹理等特征[6-7],如 LIU 等[8]使用基于颜色和形状特征的 SVM 分类器实时检测苹果,准确率为 90.6%。虽然这类方法可以实现在特定场景下对特定水果进行识别,但是其识别精度普遍受限于果实特征信息的提取质量,不适用于实际果园中的水果实时识别检测,而且手工获取特征的步骤较为复杂,鲁棒性较差。

随着计算机系统性能和运算能力的不断提升,以卷积神经网络为代表的深度学习算法被广泛应用于农作物和水果果实的识别检测[9-10]。WAN 等[11]对现有 Faster R-CNN 模型的池化层和卷积层结构进行改进,设计了一种多类别水果检测方法,平均检测精度为 91%。FU 等[12]针对多类别遮挡下的苹果识别任务,设计了一种基于 Faster R-CNN 的苹果识别方法,并通过使用深度特征过滤背景物体来提高苹果检测准确性。测试结果显示,在无遮挡、树枝、树叶和果实重叠遮挡下,模型平均精度分别为 90.9%、85.8%、89.9% 和 84.8%,处理单幅图像平均需要 0.241 s。虽然以 Faster R-CNN 为代表的二阶段目标识别算法在果园环境下的果实识别任务中获得了可以接受的结果,但是其网络结构复杂,消耗大量的计算资源,可能导致在实时检测任务中处理高分辨率图像时出现检测失败的问题,达不到实时性要求。

YOLO 和 SSD 系列作为一种先进的一阶段目标检测算法,已被广泛应用于自然环境中的果实识别检测[13-15]。赵德安等[16]基于 YOLO v3 设计了一种用于自然果园环境下的苹果识别方法,并压缩模型的网络深度来加快检测速度,该模型可以很好地检测存在遮挡和套袋的苹果,召回率为 90%,准确率达 97%,在 GPU 下检测单幅图像的平均时间为 16.69 ms。闫彬等[17-18]将果园中的苹果划分为无法采摘、可迂回采摘和可直接采摘 3 种类别,为了准确识别并区分不同枝干遮挡情况下的苹果,提出了一种基于改进 YOLO v5m 的检测模型,识别准确率为 81.0%,召回率为 85.9%,平均精度为 80.7%,检测单幅图像平均需要 25 ms。基于 YOLO 系列的目标检测算法能够很好地识别复杂果园环境中的水果,且检测速度较快。此外,一些学者在检测模型的卷积层添加 CA[19]、SE[20]、CBAM[21]等注意力机制,以进一步提高模型在复杂果园环境中的鲁棒性和泛化性。尤燕等[22]在 YOLO v7 的小目标检测层中引入多头自注意力机制,设计了一种疏果期的苹果目标检测模型,提高了模型检测性能和鲁棒性。LU 等[23]通过添加 CBAM 注意力机制模块,提出了一种改进的 YOLO v4 模型用于识别检测自然果园环境下不同成熟度的苹果果实,对于早期、中期和收获期的水果,模型检测精度分别为 86.2%、87.5% 和 92.6%。注意力机制的引入,提高了模型检测精度,但同时也增加了模型参数量和计算复杂度,增加了模型在采摘机器人嵌入式设备上的部署成本和难度。

因此,研究人员致力于开发适用于嵌入式设备的轻量化目标检测模型,并取得了一定的进展。WANG 等[24]采用 MobileNetV3 结构替换 YOLO v5 模型的骨干网络,并增加小目标检测层,设计了一种轻量化番茄检测模型,降低了模型参数量和计算量,提高了模型对小目标果实的检测精度。HU 等[25]设计了一种基于 Ghost Bottleneck 的轻量级 YOLO v5 改进模型用于梨果实目标检测,并将 SE 模块嵌入到设计的网络中,在减少网络参数量的同时提高了特征提取能力。模型轻量化设计,降低了模型体积,提高了模型运行性能,但精度方面仍有一定程度的损失,并不能达到轻量化与精度相平衡。

上述研究推动了水果识别检测技术的发展,在检测精度、检测速度和轻量级方面展现了各自的见解。然而,苹果检测模型需要部署在性能较低的嵌入式设备上供采摘机器人使用,在复杂果园环境中准确、快速地识别出苹果果实,需要全面解决这 3 个问题。同时,在自然环境中,由于苹果自然生长的非结构化特征,可能会出现枝叶和果实重叠遮挡的问题,很难准确地识别检测到目标苹果。

为了解决上述问题,本文设计一种基于改进 YOLO v7 的轻量化苹果检测方法,在保证识别精度和速度的同时使模型尽可能轻量化,以降低计算成本并满足实时检测需求。

1　自然果园环境下苹果图像数据集构建

1.1　苹果图像采集

为了获取能够较好地反映自然果园环境中苹果状态的图像数据集，图像采集地点选在江苏省徐州市丰县华山镇杏花村乐丰苹果园。该果园栽培模式为主干型宽行密植栽培，栽培模式比较标准，规划合理并且便于管理；苹果树横向枝都不长，因此大部分果实裸露在外，果树底部和侧向空间较大。这种果树种植结构使得机器人的相机能够从果树侧面或底部向上采集苹果图像进行识别，并可为采摘机器人提供足够大的作业空间。

图像采集设备选用 Intel RealSense D435 型相机，其分辨率为 1 280 像素 ×720 像素，苹果图像采集过程如图 1 所示。为保证所采集图像数据的多样性，分别在不同天气（晴天和阴天）情况下对摘袋后着色 5 ~ 15 d 的苹果进行图像采集，拍摄时相机至果树的距离在 0.3 ~ 1 m 内并选择多角度进行采集，采集时间为 2021 年 11 月 8—11 日每天的 08:00—18:00。图 2 为在自然果园环境下采集的部分苹果图像，包含顺光、侧光和逆光 3 种不同光照条件，未成熟和成熟 2 种不同苹果颜色，以及树叶、树枝和果实重叠遮挡 3 种不同遮挡类别。共采集并保存 JPG 格式的苹果图像 4 000 幅。

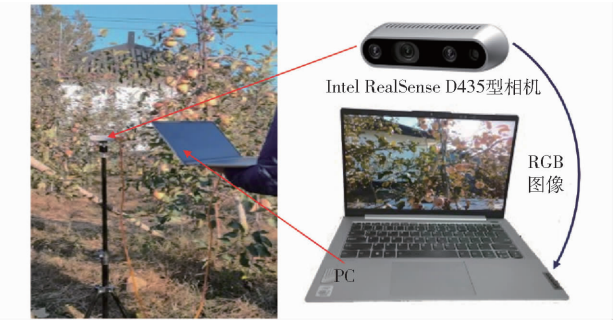

图 1　实际果园场景中的苹果图像采集

Fig. 1　Apple image acquisition in real orchard scenes

1.2　苹果图像划分与处理

首先，将 4 000 幅苹果图像按比例 8:1:1分为训练集、验证集和测试集，以保证三者之间无重复图像，防止模型过拟合。测试集的 400 幅图像包含不同复杂场景下的苹果图像（图 2）各 50 幅，用于测试模型检测性能。

在进行苹果果实标注时，综合考虑苹果的各种外观、形态及采摘机器人末端执行器的容错能力，对遮挡面积超过 50% 和图像边缘处裸露面积小于 50% 的果实不予标注。使用软件 LabelImg 在图像中绘制苹果果实的最小外接矩形进行标注，以确保矩形中包含的背景区域尽可能少。设置矩形框的类别属性为“Apple”，保存注释后生成 xml 格式的标签文件，包含矩形框的高度、宽度及类别信息。



图 2　不同复杂场景下的苹果图像

Fig. 2　Apple images in different complex scenes

1.3　苹果图像数据增强

在检测模型的训练任务中，如果用于训练的图像数据不足，会产生过拟合的情况；此外，训练集中的 3 200 幅原始苹果图像难以涵盖自然环境下所有的光照强度、天气、噪声、清晰度等因素，因此需要对苹果图像数据进行扩增以获得充足的样本数据，提高目标检测模型的检测性能和泛化能力。通过 Python 对采集的原始苹果图像添加随机对比度、随机亮度、随机翻转、随机高斯噪声、随机饱和度，以尽可能多地涵盖自然果园环境下各种复杂背景的苹果图像，部分示例如图 3 所示。在图像中加入高斯噪声，可以模仿在实际应用中由于设备或树枝的晃动导致所捕捉到的图像模糊这一情况。扩增后苹果图像数量比原始图像多 4 倍，最终增强训练集由 16 000 幅图像组成，包括原始图像 3 200 幅和增强图像 12 800 幅。

2　基于改进 YOLO v7 的检测模型

复杂果园环境下的苹果果实实时检测，需要选择性能优越、识别速度快、识别精度高的检测模型。作为单阶段目标检测算法之一，YOLO v7 具有精度

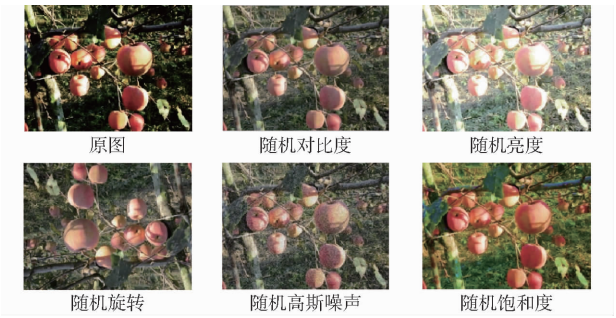

图 3 图像增强的示例

Fig. 3 Examples of image augmentation

高、易于训练和部署、检测速度快等特点[26]。虽然该网络表现出较好的检测性能,但在处理复杂果园环境中存在遮挡情况的苹果检测任务中,依然存在错检漏检的问题,仍有提升和改进的空间。

2.1 YOLO v7 网络结构

YOLO v7 模型的结构主要包括:主干网络(Backbone)、颈部网络(Neck)和检测头(Head)。Backbone 首先对输入尺寸为 640 像素×640 像素的苹果图像进行特征提取;再由 Neck 进行特征融合以结合不同尺度的特征信息,并进一步提取特征,获得 3 个加强过的有效特征层;最后,Head 对每个特征层的特征点进行判断,得出目标物体的位置、置信度和类别信息。

主干特征提取网络主要利用多分支堆叠模块 Multi_Concat_Block 和下采样模块 Transition_Block 对输入的图像不断地进行长度和宽度的压缩以及通道的扩张,以实现特征提取。Multi_Concat_Block 模块的结构如图 4 所示,包含 4 个分支,每个分支对输入的特征层进行不同个数的标准化卷积操作,最后,将 4 个分支的结果堆叠后再进行一次标准化卷积操作,得到 1 个固定尺寸的输出特征层。该模块以增加一定网络深度的方式来提高精度,同时,多分支堆叠及其跳跃连接结构,有效避免了增加深度所导致的梯度消失问题。Transition_Block 模块结构如图 5 所示,主要对输入的特征进行下采样操作。该模块通过 2 个分支对输入特征进行并行提取和压缩,左分支分别进行 1 个步长为 2×2 的池化操作和 1 个 1×1 的卷积运算;右分支进行 1 个 1×1 的卷积运算后再进行 1 个步长为 2×2 的卷积运算,最后将 2 个分支的结果堆叠在一起。

Neck 网络中,YOLO v7 构建了与 YOLO v5 相同的特征金字塔结构对主干网络的 3 个有效特征层进行加强特征提取。该部分同样也使用 Multi_Concat_Block 和 Transition_Block 模块进行特征提取,并结合上采样和下采样操作将不同尺度的特征层进行传递和融合,以便提取出更好的特征层用于目标检测。

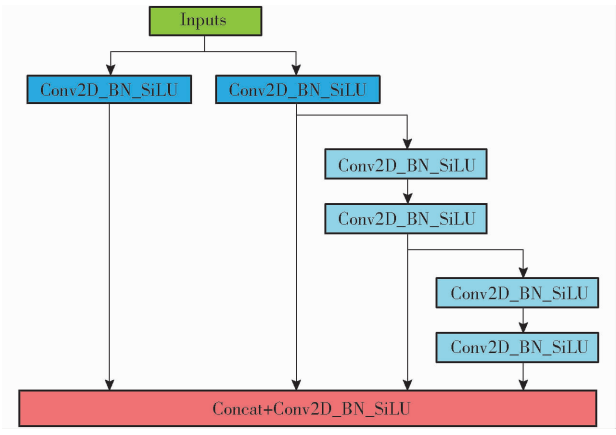

图 4 多分支堆叠模块结构

Fig. 4 Structure of multi branch stacking module

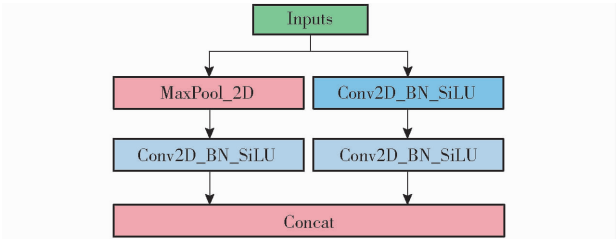

图 5 下采样模块结构

Fig. 5 Structure of downsampling module

在此之前,YOLO v7 利用特殊的 SPPCSPC 模块对尺寸为(20,20,1024)的特征层进一步提取特征信息,以提高网络层的感受野。SPPCSPC 模块是 CSP 和 SPP 模块的结合,首先通过 CSP 模块将特征层分为 2 部分,一部分进行标准化卷积操作后直接传递到模块结构的末端;另一部分通过 SPP 模块进行 4 种不同大小卷积核的池化处理,将任意尺寸的特征层堆叠为固定尺寸的输出特征层;最后将 2 部分堆叠并进行 1 次标准化卷积运算。这样不仅可以提高网络层的感受野和精度,还可以有效避免特征信息的重复提取,减少计算量,使得速度更快。

Head 部分是 YOLO v7 的分类器与回归器,选用表示大、中、小 3 种目标尺寸的检测头,将精炼的特征信息进行解耦,通过 RepConv 结构对不同尺寸的增强特征层进行通道数调整,结合 1×1 卷积进行类别的预测并得出目标的位置和置信度。

2.2 PConv

部分卷积(Partial convolution, PConv)通过减少冗余计算和内存访问的数量,可以更有效地提取空间特征[27]。其工作原理如图 6c 所示,不同于常规卷积和深度/群组卷积,PConv 只对少部分输入通道使用过滤器进行空间特征提取,其余通道保持不变。因此,PConv 相比于常规卷积和深度/群组卷积具有更低的浮点运算次数和更高的浮点计算速度。

为了设计快速轻量化的检测模型,许多改进

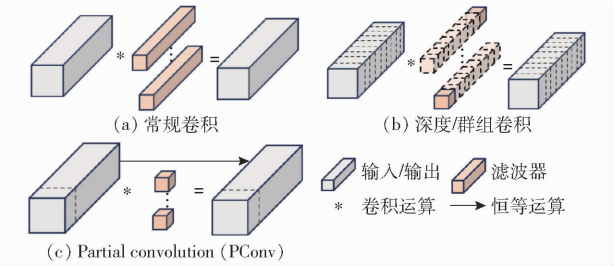

图 6　PConv 的工作原理图

Fig. 6　Working principle of PConv

都集中于减少浮点运算次数上，但是浮点运算次数的减小不一定能带来相应的运算速度提升，因为普通卷积操作对内存的频繁访问使得浮点运算的效率较低。由于检测目标只有苹果这一个类别，并不需要深度卷积操作提取特征，因此，将 PConv 引入模型的多分支堆叠模块中，以实现更加轻量化的网络模型，得到的 P－Multi_Concat_Block 结构如图 7 所示。

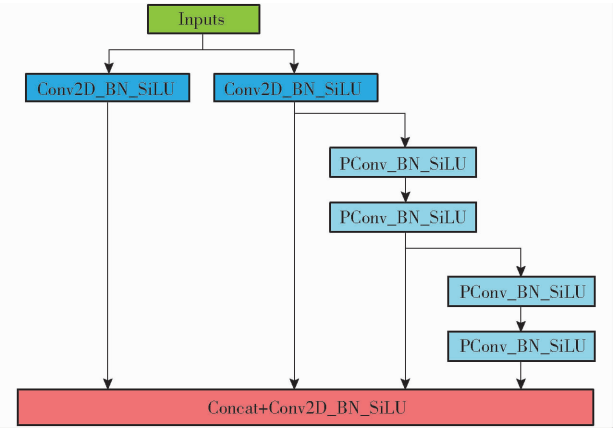

图 7　改进的 P－Multi_Concat_Block 结构

Fig. 7　Structure of improved P－Multi_Concat_Block

2.3　ECA 模块

ECA 模块[28]在 SE 模块的基础上进行改进，去除了全连接层，使用 1×1 的卷积来实现局部跨通道交互，避免了降维对数据的影响，在提升模型性能的同时，可减少模型的计算量和参数量。该模块从通道间的特征信息中自适应学习，对特征进行重构和分配，以提高小目标特征图的比例和权重，缓解卷积模块在通道范围内的不足，增强网络的特征提取能力，使模型利用全局特征来区分图像信息层次。因此，该模块在检测过程中能够抑制背景信息干扰，更好地关注复杂背景下的苹果目标，实现特征图的通道自适应校准。

如图 8 所示，该模块通过使用不同大小的卷积核将输入特征分割成多尺度特征；然后，在每个通道进行全局平均池化后，采用大小为 k 的快速 1 维卷积获取局部跨通道交互信息；最后，将整合后的通道信息权值与原始输入特征相乘，得到具有通道关系

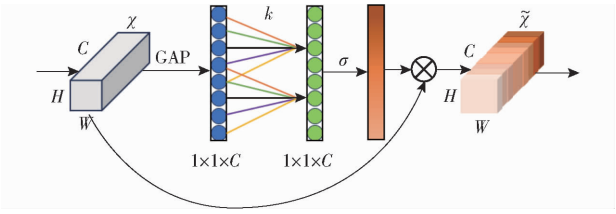

图 8　ECA 模块结构

Fig. 8　Structure of ECA module

的特征图。其中，k 为局部跨通道交互的覆盖范围；H、W、C 分别为特征图高度、宽度和通道数。该模块结构简单，是一种轻量化的通道注意力机制，在提升性能的同时可有效降低模型复杂程度。

2.4　基于麻雀搜索算法的学习率优化

学习率是影响模型识别精度的一个重要参数，合适的学习率可以提高模型的精度并加快模型的收敛。在 YOLO v7 模型的训练过程中，学习率通常根据经验来设定，当设定较小的学习率时，会导致模型在训练过程中收敛速度过慢；当设定的学习率较大时，会造成模型在训练过程中的优化幅度跳过最优点。因此，提出一种基于麻雀搜索算法（Sparrow search algorithm，SSA）的学习率优化方法来获得最优平均精度所对应的学习率，以提高模型的检测精度。

SSA 是一种模拟麻雀觅食和躲避捕食者行为的群智能优化算法[29]。每只麻雀在觅食过程中存在 3 种可能的行为：①作为发现者，率领麻雀群搜索食物。②作为加入者，跟从发现者觅食并相互竞争以提高捕食率。③作为警觉者，察觉危险就放弃觅食。且每只麻雀只有一个属性，即找到食物的位置。其算法流程如图 9 所示。

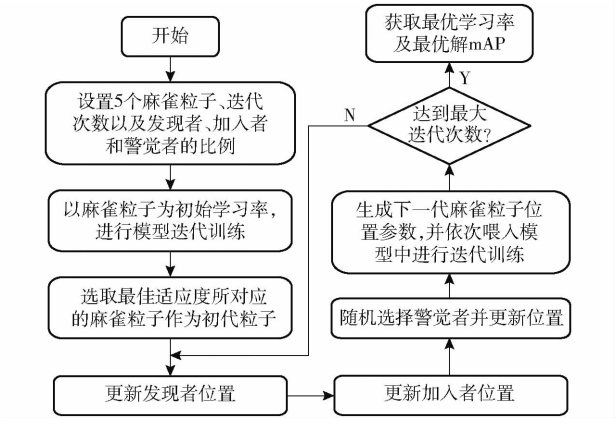

图 9　基于 SSA 的学习率优化流程图

Fig. 9　Process of learning rate optimization based on SSA

主要步骤如下：

（1）设置麻雀粒子、迭代次数以及加入者、发现者和警觉者的比例。

假设在 d 维搜索空间中有 N 个麻雀种群，则初

始种群为 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{iD})$；其中，$i=1,2,\cdots,N$；$d=1,2,\cdots,D$；$x_{iD}$表示第 i 只麻雀在 D 维搜索空间中的位置；适应度为 $f_i=f(x_1,x_2,\cdots,x_D)$。

(2)以麻雀粒子为初始学习率，验证集的平均精度均值(mAP)作为适应度，进行模型迭代训练，得到具有最优适应度的麻雀位置。

(3)更新发现者位置。发现者位置更新计算公式为

$$\boldsymbol{x}_{id}^{t+1}=\begin{cases}\boldsymbol{x}_{id}^{t}\exp\left(\dfrac{-i}{\alpha t_{\max}}\right) & (R_2<S_T)\\ \boldsymbol{x}_{id}^{t}+P\boldsymbol{L} & (R_2\geqslant S_T)\end{cases}$$

$$(\alpha,P\in(0,1];R_2\in[0,1];S_T\in[0.5,1]) \quad (1)$$

式中 $\boldsymbol{x}_{id}^{t}$——第 i 只麻雀在第 t 次迭代中的位置

α、P——均匀随机数

$t_{\max}$——最大迭代次数

R_2——预警值

S_T——安全值

$\boldsymbol{L}$——元素为 1 的矩阵

(4)更新加入者位置。加入者位置更新计算公式为

$$\boldsymbol{x}_{id}^{t+1}=\begin{cases}P\exp\left(\dfrac{\boldsymbol{x}_{wd}^{t}-\boldsymbol{x}_{id}^{t}}{i^2}\right) & \left(i>\dfrac{N}{2}\right)\\ \boldsymbol{x}_{bd}^{t+1}+|\boldsymbol{x}_{id}^{t}-\boldsymbol{x}_{bd}^{t+1}|\boldsymbol{A}^{+}\boldsymbol{L} & \left(i\leqslant\dfrac{N}{2}\right)\end{cases} \quad (2)$$

其中 $$\boldsymbol{A}^{+}=\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}$$

式中 $\boldsymbol{x}_{wd}^{t}$——麻雀在第 t 次迭代中的最差位置

$\boldsymbol{x}_{bd}^{t+1}$——麻雀在第 $t+1$ 次迭代中的最佳位置

$\boldsymbol{A}$——元素被任意分配为 1 或 -1 的 $1\times d$ 矩阵

(5)随机选择警觉者并更新位置。设定警觉者的数量占麻雀种群的 20%。其位置更新计算公式为

$$\boldsymbol{x}_{id}^{t+1}=\begin{cases}\boldsymbol{x}_{bd}^{t}+\beta(\boldsymbol{x}_{id}^{t}-\boldsymbol{x}_{bd}^{t}) & (f_i\neq f_g)\\ \boldsymbol{x}_{id}^{t}+K\dfrac{\boldsymbol{x}_{id}^{t}-\boldsymbol{x}_{wd}^{t}}{|f_i-f_w|+\tau} & (f_i=f_g)\end{cases}$$

$$(K\in[-1,1]) \quad (3)$$

式中 $\boldsymbol{x}_{bd}^{t}$——麻雀在第 t 次迭代中的最佳位置

β——步长参数

f_i——第 i 只麻雀的适应度

f_g——当前全局最优适应度

f_w——当前全局最差适应度

K——麻雀运动方向的随机数

τ——无穷小常数

(6)生成下一代麻雀粒子位置参数，并依次喂入模型中进行迭代训练。

(7)判断是否达到最大迭代次数。若达到，则程序结束，输出结果；否则，重复步骤(3)~(6)。

2.5 改进 YOLO v7 苹果检测模型框架设计

自然果园环境中，由于光照条件和遮挡等复杂场景，导致图像中干扰信息与苹果特征信息混淆，影响识别的准确性，出现错检漏检的情况，故为进一步解决检测模型在复杂果园环境下的检测性能不佳、硬件部署条件苛刻等问题，本研究提出一种改进 YOLO v7 的苹果检测模型，总体框架如图 10 所示。主要改进方法如下：

(1)引入 PConv 替换多分支堆叠模块 Multi_Concat_Block 中的部分常规卷积进行轻量化改进，以设计快速轻量化的网络模型。

(2)在 YOLO v7 的 Neck 网络中引入轻量化的 ECA 模块，提高模型的特征提取能力。YOLO v7 的 Neck 部分存在 3 个增强特征层，尺寸分别为(80、80、512)、(40、40、1024)和(20、20、512)。由于 ECA 模块的作用与其在卷积神经网络中的位置有关，当将 ECA 模块融合在网络的早期卷积层时，新网络更加关注图像的局部特征，只负责区分图像的主要和次要信息；当融合在网络的深度卷积层时，图像的全局特征会得到更好地表征，有助于增强特征提取。因此，分别在维度为 1 024 和 512 的特征层后端融合 3 个 ECA 模型，以获得最佳精度模型。

(3)基于 SSA 算法对学习率进行优化改进。由于 YOLO v7 原始模型在训练过程的学习率通常根据经验人为设定，很难找到最佳的学习率。因此采用麻雀搜索算法对模型训练过程中的学习率进行优化改进，得到最优平均检测精度所对应的学习率，从而进一步提高模型的检测精度。算法流程如图 9 所示。

2.6 模型评价指标

为了准确客观地评估网络模型检测性能，选取评价指标为：精确率(P)、召回率(R)、F1 值、平均精度均值(Mean average precision, mAP)和每秒传输帧数(Frames per second, FPS)。

3 试验结果与分析

3.1 环境配置与模型训练

模型训练在系统为 ubuntu 18.04 的云服务器上进行。主要的硬件配置为：12 核 Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz，43 GB 运行内存，Nvidia GeForce RTX 3090 显卡，24 GB 显存，并配置了与显卡驱动相适配的 CUDA 11.1 和 cudnn 8.0.4，调用 GPU 以加速模型训练；选用 Python 版本为 3.8.5，框架为 PyTorch 1.8.1 版本。

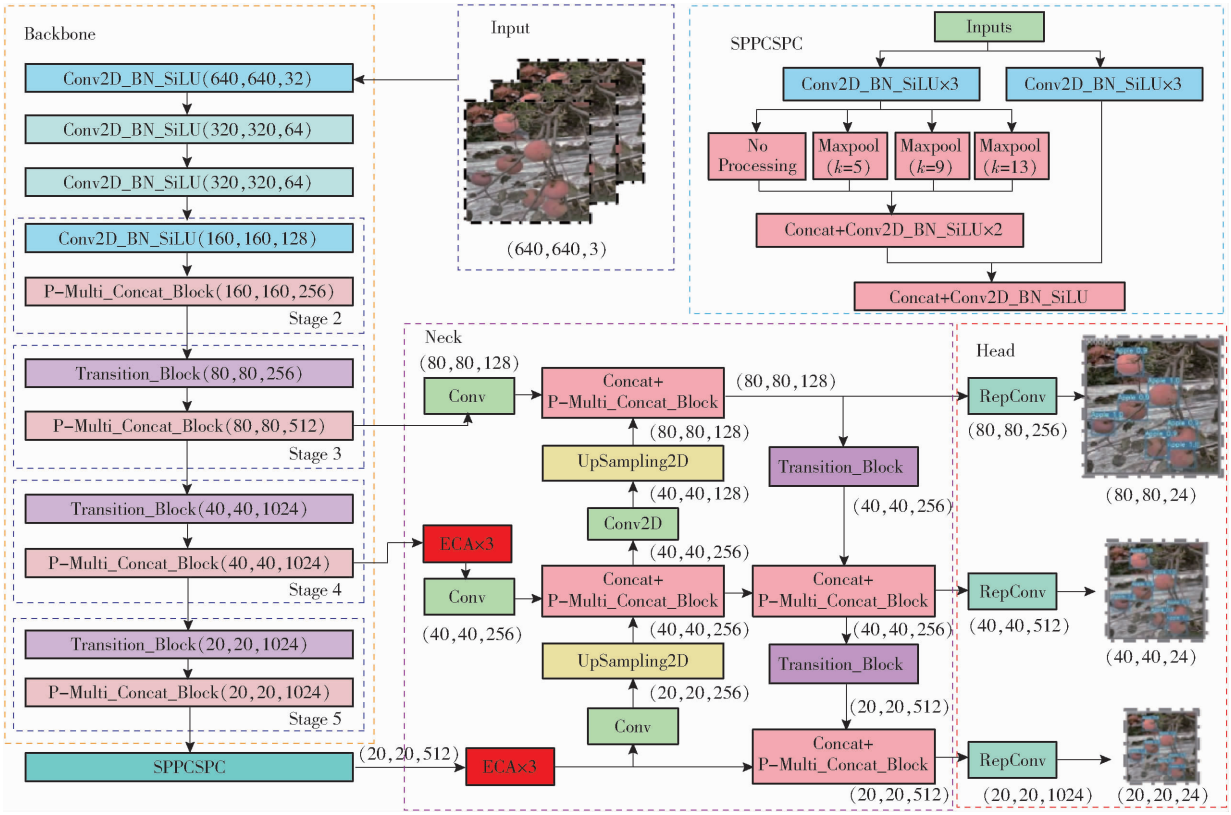

图 10　改进 YOLO v7 苹果检测模型的总体框架

Fig. 10　Overall framework of improved YOLO v7 apple detection mode

设置模型训练时迭代数(Epoch)为 100，批次大小为 32。经过 100 轮迭代训练后，模型的损失值收敛趋势如图 11 所示。结果显示，模型在前 20 轮训练中快速收敛于 0.07 后趋于稳定，此时验证损失的数值低于训练损失，这是由于模型可能陷入局部收敛状态；接着，模型在第 65 轮附近又二次收敛于 0.04 并趋于稳定，此时模型跳出局部收敛，进入全局收敛，验证损失与训练损失基本相等，表明训练得到的模型稳定有效。本研究提出的改进模型精确率为 90.96%，召回率为 91%，平均精度均值 mAP@0.5 为 97%。

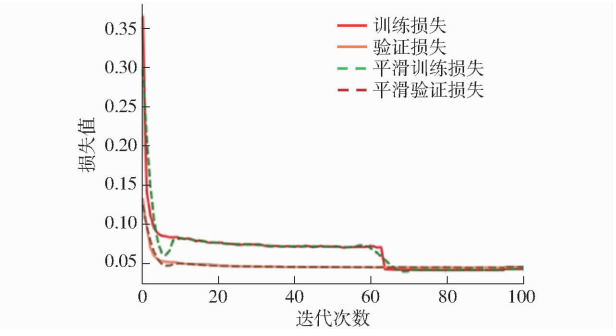

图 11　改进模型的损失变化曲线

Fig. 11　Loss variation curves of improved model

3.2　不同注意力机制性能对比试验

为了验证 ECA 注意力模块对于 YOLO v7 检测模型在苹果数据集上的提升效果，分别采用常用的 CBAM 和 SE 注意力模块对 YOLO v7 模型进行改进，并在相同配置环境下对同一苹果数据集进行检测性能的对比试验。试验结果如表 1 所示。

表 1　引入不同注意力模块的模型检测结果

Tab. 1　Detection results of models with different attention modules

模型	P/%	R/%	mAP/%	计算量	FPS/(f·s^{-1})
YOLO v7 + CBAM	87.90	91.40	94.20	$1.154\,00\times10^{11}$	68.6
YOLO v7 + SE	88.80	90.80	94.10	$1.133\,00\times10^{11}$	69.7
YOLO v7 + ECA	89.06	91.82	96.69	$1.064\,85\times10^{11}$	69.8

分析对比检测结果可以得出，YOLO v7 + ECA 模型的精确率、召回率和平均精度均值比 YOLO v7 + CBAM 模型分别高出 1.16、0.42、2.49 个百分点；比 YOLO v7 + SE 模型分别高出 0.26、1.02、2.59 个百分点。同时，相比于其他 2 种模型，YOLO v7 + ECA 模型的计算量较小，检测速度较快，表明 ECA 注意力模块对 YOLO v7 模型的性能提升效果较为明显。

3.3　消融试验和对比试验

3.3.1　消融试验

为了验证所提出的 3 种改进方法的有效性，在相同的配置环境下对同一苹果数据集展开消融试验，进行检测性能的对比分析。在 YOLO v7 模型的基础上分别依次引入 3 种改进方法，试验结果如

表 2 所示,其中,A 为引入 PConv 对 Multi_Concat_Block 模块中的部分常规卷积进行替换,B 为添加轻量化的 ECA 模块,C 为进行基于 SSA 的学习率优化改进。

表 2 不同改进方法的消融试验

Tab. 2 Ablation experiments of different improvement methods

模型	P/%	R/%	mAP@0.5/%	参数量	计算量	单幅图像平均用时/s	
						GPU	CPU
YOLO v7	86.81	90.62	95.61	$3.762\,0\times10^{7}$	$1.064\,7\times10^{11}$	0.014 2	0.320
YOLO v7 + A	86.81	90.62	95.61	$2.910\,5\times10^{7}$	$7.764\,3\times10^{10}$	0.013 6	0.294
YOLO v7 + B	89.48	90.93	96.64	$3.762\,0\times10^{7}$	$1.064\,8\times10^{11}$	0.014 5	0.386
YOLO v7 + C	88.69	90.02	96.21	$3.762\,0\times10^{7}$	$1.064\,7\times10^{11}$	0.014 3	0.334
YOLO v7 + A + B	89.48	90.93	96.64	$2.910\,5\times10^{7}$	$7.766\,1\times10^{10}$	0.014 1	0.303
YOLO v7 + A + C	88.69	90.02	96.21	$2.910\,5\times10^{7}$	$7.764\,3\times10^{10}$	0.013 8	0.295
YOLO v7 + B + C	90.96	91.00	97.00	$3.719\,5\times10^{7}$	$1.051\,3\times10^{11}$	0.015 1	0.389
YOLO v7 + A + B + C	90.96	91.00	97.00	$2.899\,3\times10^{7}$	$7.728\,6\times10^{10}$	0.013 9	0.306

结果显示,相比于 YOLO v7 初始模型,YOLO v7 + A 模型的检测精度保持不变,其参数量和计算量分别降低 22.63% 和 27.08%,在 GPU 和 CPU 上检测每幅图像平均用时分别减少 0.006 s 和 0.026 s,证明了轻量化改进的有效性。相比于 YOLO v7 初始模型,YOLO v7 + B 模型的精确率、召回率和 mAP@0.5 分别提高 2.67、0.31、1.03 个百分点,其参数量保持不变,计算量略有提高,在 GPU 和 CPU 上检测每幅图像的平均用时分别增加 0.003 s 和 0.066 s。相比于 YOLO v7 初始模型,YOLO v7 + C 模型的参数量和计算量保持不变,其召回率降低 0.6 个百分点,精确率和 mAP@0.5 分别提高 1.81、0.6 个百分点,在 GPU 和 CPU 上检测每幅图像平均用时分别增加 0.001 s 和 0.014 s。可以发现,相比于单独引入 ECA 模块,单独引入 PConv 后的模型检测性能提升表现较强。这是由于引入 ECA 后,增强了网络模型的特征提取能力,可以关注到待检测目标更多容易被淹没的特征信息,而基于 SSA 的学习率优化方法在模型训练过程中迭代搜寻出最优检测精度所对应的学习率,因此提升幅度较小。

相比于 YOLO v7 + B 模型,YOLO v7 + A + B 模型的参数量和计算量分别降低 22.63% 和 27.06%,其精确率、召回率和 mAP@0.5 没有发生改变,在 GPU 和 CPU 上检测每幅图像平均用时分别减少 0.004 s 和 0.083 s;相比于 YOLO v7 原始模型,YOLO v7 + A + B 模型在 GPU 和 CPU 上检测每幅图像平均用时分别减少 0.001 s 和 0.017 s。相比于 YOLO v7 + C 模型,YOLO v7 + A + C 模型精确率、召回率和 mAP@0.5 保持不变,其参数量和计算量分别降低 22.63% 和 27.08%,在 GPU 和 CPU 上检测每幅图像平均用时分别减少 0.005 s 和 0.039 s。相比于 YOLO v7 原始模型,在 GPU 和 CPU 上检测每幅图像平均用时分别减少 0.004 s 和 0.025 s。

相比于 YOLO v7 初始模型,YOLO v7 + B + C 模型精确率、召回率和 mAP@0.5 分别提高 4.15、0.38、1.39 个百分点,其参数量和计算量分别降低 1.13% 和 1.26%,在 GPU 和 CPU 上检测单幅图像平均用时分别增加 0.009 s 和 0.069 s。对比发现,YOLO v7 + B + C 模型检测性能高于 YOLO v7 + B 和 YOLO v7 + C 模型,表明单独引入一种改进方法对模型的检测性能提升有限。相比于 YOLO v7 + B + C 模型,YOLO v7 + A + B + C 模型精确率、召回率和 mAP@0.5 保持不变,其参数量和计算量分别降低 22.05% 和 26.48%,在 GPU 和 CPU 上检测单幅图像平均用时分别降低 0.012 s 和 0.083 s。相比于 YOLO v7 原始模型,其参数量和计算量分别降低 22.93% 和 27.41%,在 GPU 和 CPU 上检测单幅图像平均用时分别降低 0.003 s 和 0.014 s。

3.3.2 不同检测模型性能对比试验

为了进一步验证改进后模型检测性能,选用了常用的 Faster R - CNN、SSD、YOLO v5、YOLO v7、YOLO v8 模型以及轻量化模型 YOLO v5s、YOLO v7 - tiny、YOLO v8s 与本研究所提出的改进模型在相同的配置环境下对同一苹果数据集进行检测性能的对比试验。各模型检测试验结果如表 3 所示。

由表 3 可知,与 Faster R - CNN 模型相比,本研究所提出模型参数量和计算量分别降低 30.56% 和 7.77%,平均精度均值提高 15.3 个百分点,在 GPU 上的检测速度提升 51.55 f/s。相比于 SSD 模型,虽改进后模型参数量增加 18.89%,但其计算量减少 11.71%,平均精度均值高 7.4 个百分点,在 GPU 上的检测速度提升 19.95 f/s。相比于 YOLO v5 模型,改进后模型参数量和计算量分别减少 37.12% 和 28.17%,平均精度均值高出3.2个百分点,在GPU

表 3　不同检测模型检测结果

Tab. 3　Detection results of different detection models

模型	mAP@0.5/%	参数量	计算量	FPS/(f·s^{-1})
Faster R－CNN	81.70	4.175 3×10^7	8.380 1×10^{10}	24.30
SSD	89.60	2.438 6×10^7	8.754 1×10^{10}	55.90
YOLO v5	93.80	4.610 8×10^7	1.076 0×10^{11}	65.40
YOLO v5s	91.50	7.350 0×10^6	1.650 0×10^{10}	78.10
YOLO v7	95.61	3.762 0×10^7	1.064 7×10^{11}	70.03
YOLO v7－tiny	93.30	7.130 0×10^6	1.331 0×10^{10}	80.07
YOLO v8	95.10	4.369 1×10^7	1.657 0×10^{11}	77.14
YOLO v8s	92.70	1.321 0×10^7	2.971 0×10^{10}	82.14
本文模型	97.00	2.899 3×10^7	7.728 6×10^{10}	75.85

上的检测速度提升 10.45 f/s。相比于 YOLO v7－tiny 模型，虽改进后的模型在 GPU 上的检测速度降低 4.22 f/s，参数量和计算量分别降低 33.64% 和 53.35%，但是其平均精度均值提高 2.7 个百分点。相比于 YOLO v8 模型，虽然改进后的模型在 GPU 上的检测速度降低 1.29 f/s，但是其参数量和计算量分别降低 33.64% 和 53.35%，平均精度均值提高 2.9 个百分点。

与 YOLO v5s、YOLO v7－tiny、YOLO v8s 轻量化模型相比，虽然本研究所提出模型参数量和计算量较高，在 GPU 上的检测速度分别降低 2.25、4.22、6.29 f/s，但是平均精度均值分别高出 5.5、3.7、4.3 个百分点。表明本研究所提出的改进模型相比于这 3 种轻量化模型仅损失少量的检测速度，就可具有较高的检测精度。

结果表明本研究提出的改进模型相比于其他模型有较为明显的优势，在保持一定的检测精度和速率的同时，减少了一定的计算量和参数量，消耗的内存资源较少，适合部署在采摘机器人的嵌入式设备上。

3.4　不同场景下检测效果对比试验

为了验证改进后的检测模型对实际果园环境中苹果果实的检测效果及泛化性能，随机挑选成熟和未成熟 2 种不同成熟度，顺光、侧光和逆光 3 种不同光照条件，以及树叶、树枝和果实重叠遮挡 3 种不同遮挡类别的苹果图像各 50 幅，采用 Faster R－CNN、YOLO v5、YOLO v8 和 YOLO v7 原始模型与本研究提出的改进模型进行检测性能的对比试验。统计各个模型的检测结果，并在检测效果图中分别使用黄色圆圈和蓝色方框标注出错检和漏检的苹果目标。

3.4.1　不同成熟度下检测效果对比试验

以成熟和未成熟为控制变量，测试模型在不同成熟度下的检测性能，结果如表 4 所示，检测效果如图 12 所示。

表 4　不同成熟度下检测结果

Tab. 4　Test results under different maturity levels

%

成熟度	模型	*P*	*R*	mAP@0.5
成熟	Faster R－CNN	78.60	73.20	82.70
	YOLO v5	90.20	87.10	94.20
	YOLO v7	91.51	88.60	96.48
	YOLO v8	92.14	88.71	96.51
	本文模型	96.69	86.45	97.81
未成熟	Faster R－CNN	77.50	71.20	80.60
	YOLO v5	89.70	86.80	93.30
	YOLO v7	90.50	88.56	96.23
	YOLO v8	89.50	86.40	95.70
	本文模型	94.07	86.09	96.71

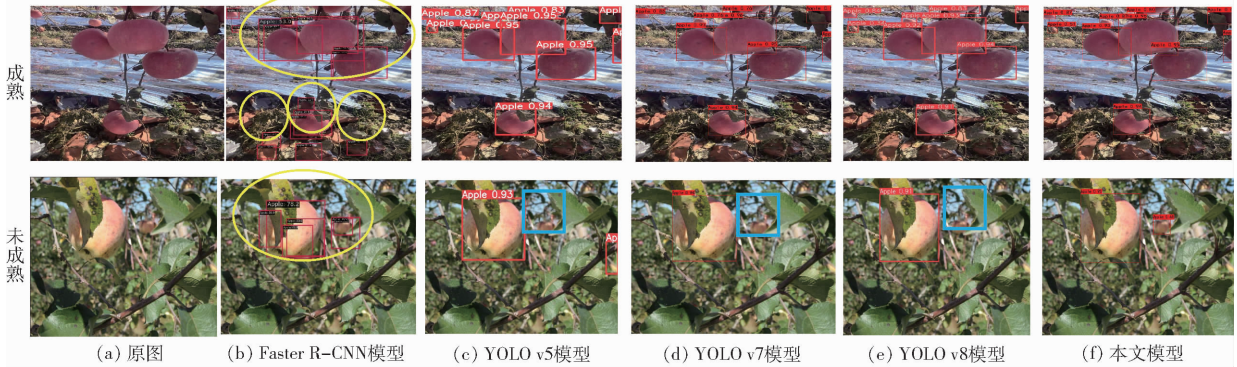

图 12　不同成熟度下检测效果

Fig. 12　Detection effects under different maturity levels

对比表 4 可知,在成熟条件下,本研究提出的改进模型的平均精度均值比 Faster R - CNN、YOLO v5、YOLO v7 和 YOLO v8 模型分别提高 15.11、3.61、1.33、1.3 个百分点;精确率比 Faster R - CNN、YOLO v5、YOLO v7 和 YOLO v8 模型分别提高 18.09、6.49、5.18、4.55 个百分点;召回率比 Faster R - CNN 提高 13.25 个百分点,比 YOLO v5、YOLO v7 和 YOLO v8 模型分别降低 0.65、2.15、2.26 个百分点。在未成熟条件下,本研究提出的改进模型平均精度均值比 Faster R - CNN、YOLO v5、YOLO v7 和 YOLO v8 模型分别提高 16.11、3.41、0.48、1.01 个百分点;精确率比 Faster R - CNN、YOLO v5、YOLO v7 和 YOLO v8 模型分别提高 16.57、4.47、3.57、4.57 个百分点;召回率比 Faster R - CNN 提高 15.89 个百分点,比 YOLO v5、YOLO v7 和 YOLO v8 模型分别降低 0.71、0.47、0.31 个百分点。

从图 12 可以看出,Faster R - CNN 模型对于不同成熟度的单个苹果目标均容易识别出重复检测框,且容易将枯黄色的树叶误识别为苹果。YOLO 系列的检测模型可以很好地检测出成熟苹果目标,这是由于成熟苹果具有鲜明的纹理和颜色特征,与背景颜色区分度较为明显,检测难度较小。而 YOLO v5、YOLO v7 和 YOLO v8 模型在检测未成熟的苹果时容易出现漏检的情况,这是由于其颜色呈青色或黄绿色,与背景颜色相近,两者之间的分界线不明显,检测难度较大。

3.4.2 不同光照条件下检测效果对比试验

为了验证改进后的模型对于果园环境中不同光照情况的实用性和鲁棒性,分别以顺光、侧光和逆光为控制变量,测试 5 种模型在不同光照条件下的检测性能。统计各个模型的测试结果如表 5 所示,检测效果如图 13 所示。

表 5 不同光照条件下检测结果

Tab. 5 Test results under different lighting conditions

%

光照条件	模型	P	R	mAP@0.5
顺光	Faster R - CNN	78.80	72.10	82.10
	YOLO v5	90.10	87.20	93.80
	YOLO v7	90.93	88.51	96.24
	YOLO v8	91.25	88.52	96.35
	本文模型	97.31	83.22	97.18
侧光	Faster R - CNN	77.90	71.60	81.80
	YOLO v5	90.10	87.30	93.60
	YOLO v7	90.50	88.56	96.23
	YOLO v8	90.43	88.35	96.21
	本文模型	94.17	85.71	96.48
逆光	Faster R - CNN	77.30	71.40	80.30
	YOLO v5	89.80	87.40	93.30
	YOLO v7	90.93	88.51	96.24
	YOLO v8	90.43	88.35	96.21
	本文模型	94.11	85.32	96.36

分析表 5 可知,本研究提出的改进模型在 3 种光照条件下的综合检测性能均优于 Faster R - CNN、YOLO v5、YOLO v7 和 YOLO v8 模型。结果表明提出的改进模型可以适用于实际果园环境中大部分的光照情况,对于不同光照条件下的苹果检测具有一定的实用性与鲁棒性。

从图 13 可以看出,在顺光条件下,Faster R - CNN 模型容易出现错检的情况,而 YOLO 系列检测模型的检测效果较好,这是由于苹果表面的光照度均匀,图像中的苹果与背景颜色区分度较为明显。在侧光条件下,由于苹果表面的亮度增加且产生光

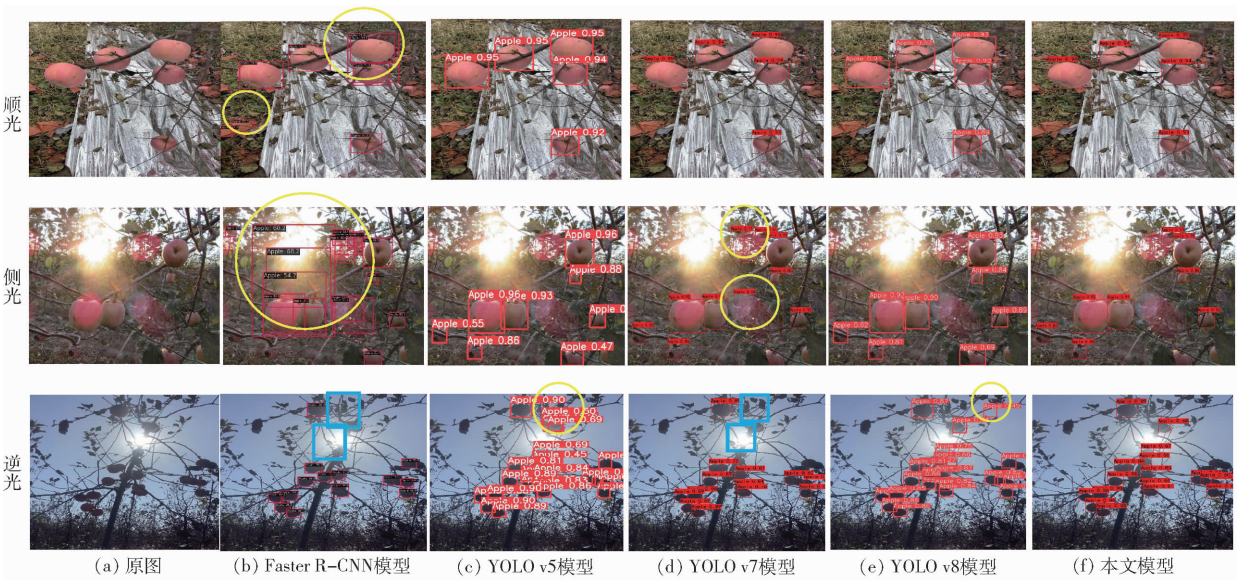

图 13 不同光照条件下检测效果

Fig. 13 Detection effects under different lighting conditions

晕和亮斑，Faster R－CNN 模型对于单个苹果目标出现重复检测框，YOLO v7 模型容易将图像中的亮斑和光晕误识别为苹果，而 YOLO v5、YOLO v8 和本研究提出的改进模型的检测效果较好。在逆光条件下，Faster R－CNN 和 YOLO v7 容易产生漏检的情况，YOLO v5 和 YOLO v8 容易产生错检的情况，这是由于苹果图像亮度较低，与背景颜色区分度不大，检测难度较大。

3.4.3　不同遮挡类别下检测效果对比试验

果园中的苹果遮挡类别主要有树叶、树枝和果实重叠遮挡 3 种，分别测试各个模型在 3 种不同遮挡类别下的检测性能。测试结果如表 6 所示，检测效果如图 14 所示。

由图 14 可以看出，Faster R－CNN、YOLO v5、YOLO v7 和 YOLO v8 模型对于存在树叶遮挡、树枝遮挡和果实重叠遮挡的苹果检测均存在不同程度的错检漏检现象。对于果实重叠遮挡的情况，由于重叠苹果之间的分界线不明显，往往会把相互粘连的多个苹果识别为 1 个苹果目标，或仅把未遮挡的部位误识别为整个被遮挡的苹果目标，造成识别和定位的不准确。

表 6　不同遮挡类别下的检测结果

Tab. 6　Test results under different occlusion categories　%

遮挡类别	模型	P	R	mAP@0.5
树叶遮挡	Faster R－CNN	76.70	70.40	78.90
	YOLO v5	89.60	87.30	92.90
	YOLO v7	91.13	88.67	96.34
	YOLO v8	90.71	88.35	96.27
	本文模型	94.27	86.19	96.63
树枝遮挡	Faster R－CNN	77.10	71.20	79.60
	YOLO v5	90.10	87.80	93.30
	YOLO v7	91.32	88.62	96.42
	YOLO v8	91.35	88.60	96.45
	本文模型	96.75	87.08	97.93
果实重叠遮挡	Faster R－CNN	76.30	70.10	76.80
	YOLO v5	89.10	86.70	92.60
	YOLO v7	90.86	88.65	96.38
	YOLO v8	90.75	88.51	96.31
	本文模型	94.19	86.20	96.72

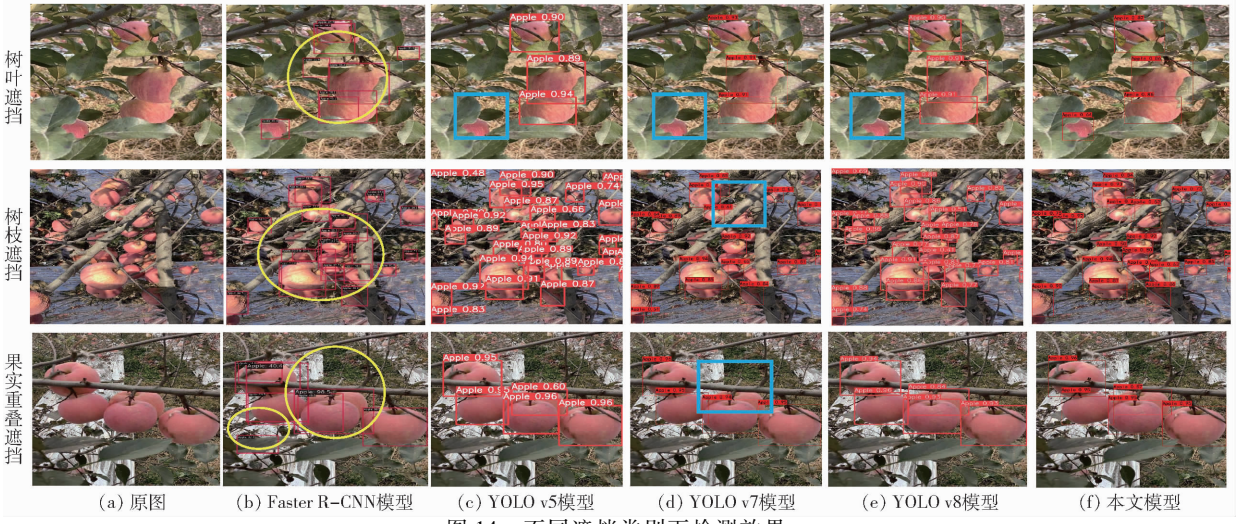

图 14　不同遮挡类别下检测效果

Fig. 14　Detection effects under different occlusion categories

对比分析表 6 可知，由于 Faster R－CNN 模型在检测过程中存在严重的错检漏检现象，其精确率、召回率和平均精度均值均普遍较低。相比于 YOLO v5、YOLO v7 和 YOLO v8 模型，在不同遮挡类别下，虽然改进模型的召回率较低，但其精确率和平均精度均值均有较大的优势。因此，本研究提出的改进模型在 3 种不同遮挡类别下的综合检测性能均优于 Faster R－CNN、YOLO v5、YOLO v7 和 YOLO v8 模型。

4　结论

（1）为了实现实际果园环境中苹果目标的快速准确识别，本研究提出了一种基于改进 YOLO v7 的苹果识别方法，通过引入 PConv 替换多分支堆叠模块中的部分常规卷积进行轻量化改进，然后引入轻量化的 ECA 模块来提高模型的特征提取能力，改善复杂环境下遮挡目标的错检漏检问题；并采用麻雀搜索算法对模型训练过程中的学习率进行优化，得到最优权重文件，以进一步提高模型的检测精度。通过拍摄与数据增强的方式建立自然果园场景下的苹果图像数据集，用于模型的训练和测试。将改进后的模型与原始模型以及几种常用网络模型进行检测性能的对比试验。

（2）消融试验检测结果表明，本研究提出的改进模型具有最佳的性能提升优势，其精确率、召回率、平均精度均值分别为 90.96%、91.00% 和

97.00%;比 YOLO v7 原始模型分别提高 4.15、0.38、1.39 个百分点。表明本研究提出的改进模型具有更好的检测性能。

(3)不同检测模型的检测结果表明,本研究提出的改进模型的平均精度均值比 Faster R-CNN、SSD、YOLO v5 和 YOLO v8 模型分别高 15.3、7.4、3.2、1.9 个百分点。与 YOLO v5s、YOLO v7-tiny 和 YOLO v8s 等轻量化模型相比,仅损失了少量的检测速度,就表现出较明显的检测精度优势。同时,在不同光照条件、不同成熟度和不同遮挡类别等复杂场景的实例检测对比试验中,改进后的模型均具有一定的优势和较高的鲁棒性。

(4)改进后的模型检测速度较快,在 GPU 下的检测速度为 75.85 f/s,比 YOLO v7 原始模型提高 5.82 f/s,在 CPU 和 CPU 下检测单幅图像平均用时比 YOLO v7 原始模型分别减少 0.003 s 和 0.014 s。相比于 Faster R-CNN、SSD、YOLO v5 模型,在 GPU 上的检测速度分别提高 51.55、19.95、10.45 f/s。同时,改进后的模型参数量和计算量较小,分别为 $2.899\,3\times10^{7}$ 和 $7.728\,6\times10^{10}$,比 YOLO v7 原始模型分别降低 22.93% 和 27.41%。模型占用的内存资源较少,适合部署于苹果采摘机器人的嵌入式设备上,为实现苹果的无人化智能采摘奠定了基础。

参 考 文 献

[1] 王丹丹,何东健.基于 R-FCN 深度卷积神经网络的机器人疏果前苹果目标的识别[J].农业工程学报,2019,35(3):156-163.
WANG Dandan, HE Dongjian. Recognition of apple targets before fruits thinning by robot based on R-FCN deep convolution neural network[J]. Transactions of the CSAE, 2019, 35(3): 156-163. (in Chinese)

[2] CHU P, LI Z, LAMMERS K, et al. Deep learning-based apple detection using a suppression Mask R-CNN[J]. Pattern Recognition Letters, 2021, 147: 206-211.

[3] 苑进.选择性收获机器人技术研究进展与分析[J].农业机械学报,2020,51(9):1-17.
YUAN Jin. Research progress analysis of robotics selective harvesting technologies[J]. Transactions of the Chinese Society for Agricultural Machinery, 2020, 51(9): 1-17. (in Chinese)

[4] ZHANG Z, IGATHINATHANE C, LI J, et al. Technology progress in mechanical harvest of fresh market apples[J]. Computers and Electronics in Agriculture, 2020, 175: 105606.

[5] TANG Y, CHEN M, WANG C, et al. Recognition and localization methods for vision-based fruit picking robots: a review[J]. Frontiers in Plant Science, 2020, 11: 510.

[6] LIN G, TANG Y, ZOU X, et al. In-field citrus detection and localisation based on RGB-D image analysis[J]. Biosystems Engineering, 2019, 186: 34-44.

[7] LIN G, TANG Y, ZOU X, et al. Color-, depth-, and shape-based 3D fruit detection[J]. Precision Agriculture, 2020, 21: 1-17.

[8] LIU X, ZHAO D, JI W, et al. A detection method for apple fruits based on color and shape features[J]. IEEE Access, 2019, 7: 67923-67933.

[9] ALTAHERI H, ALSULAIMAN M, MUHAMMAD G. Date fruit classification for robotic harvesting in a natural environment using deep learning[J]. IEEE Access, 2019, 7(1): 117115-117133.

[10] 宋怀波,尚钰莹,何东健.果实目标深度学习识别技术研究进展[J].农业机械学报,2023,54(1):1-19.
SONG Huaibo, SHANG Yuying, HE Dongjian. Review on deep learning technology for fruit target recognition[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(1):1-19. (in Chinese)

[11] WAN Shaohua, GOUDOS S. Faster R-CNN for multi-class fruit detection using a robotic vision system[J]. Computer Networks, 2019, 168: 107036.

[12] FU L, MAJEED Y, ZHANG X, et al. Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting[J]. Biosystems Engineering, 2020, 197: 245-256.

[13] TANG Y, QIU J, ZHANG Y, et al. Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment:a review[J]. Precision Agriculture, 2023, 24: 1183-1219.

[14] YAO J, QI J, ZHANG J, et al. A real-time detection algorithm for kiwifruit defects based on YOLO v5[J]. Electronics, 2021, 10(14): 1711.

[15] WU D, LV S, JIANG M, et al. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments[J]. Computers and Electronics in Agriculture, 2020, 178: 105742.

[16] 赵德安,吴任迪,刘晓洋,等.基于 YOLO 深度卷积神经网络的复杂背景下机器人采摘苹果定位[J].农业工程学报,2019,35(3):164-173.
ZHAO Dean, WU Rendi, LIU Xiaoyang, et al. Apple positioning based on YOLO deep convolutional neural network for picking robot in complex background[J]. Transactions of the CSAE, 2019, 35(3): 164-173. (in Chinese)

[17] 闫彬,樊攀,王美茸,等.基于改进 YOLO v5m 的采摘机器人苹果采摘方式实时识别[J].农业机械学报,2022,53(9):28-38,59.
YAN Bin, FAN Pan, WANG Meirong, et al. Real-time apple picking pattern recognition for picking robot based on improved YOLO v5m[J]. Transactions of the Chinese Society for Agricultural Machinery, 2022,53(9): 28-38, 59. (in Chinese)

(下转第 262 页)