

DOI: 10.16861/j.cnki.zggc.202423.0595

基于改进 YOLOv5 的番茄果实识别估产方法

杨 健, 杨啸治, 熊 串, 刘 力

(成都理工大学机电工程学院 成都 610059)

摘要: 为了实现大棚环境中番茄的智能在线产量估算, 提出了一种基于改进的 YOLOv5 (You Only Look Once v5) 番茄识别算法, 对自然生长状态下的番茄果实产量进行统计和估算。首先, 使用可分离视觉转换器 (Separable Vision Transformer, SepViT) 替换骨干网络的最后一层, 以增强骨干网络与全局信息之间的联系并提取番茄特征; 其次, 引入 WIOU (Wise Intersection over Union) 损失函数和 Mish 激活函数, 以提高收敛速度和精度。试验结果表明, 改进后的检测模型在平均精度 (mean Average Precision, mAP) 方面达到了 99.5%, 相较传统的 YOLOv5 模型提高了 1.1 个百分点, 每张图像的处理时间为 15 ms。此外, 改进后的 YOLOv5 算法对密集和遮挡情况下的番茄果实识别效果更好。

关键词: 大棚番茄; YOLOv5; 注意力机制; 损失函数

中图分类号: S641.2

文献标志码: A

文章编号: 1673-2871(2024)06-061-08

An improved YOLOv5-based method for tomato fruit identification and yield estimation

YANG Jian, YANG Xiaozhi, XIONG Chuan, LIU Li

(School of Mechanical and Electrical Engineering, Chengdu University of Technology, Chengdu 610059, Sichuan, China)

Abstract: In pursuit of intelligent real-time yield estimation for tomatoes in greenhouse environments, we introduce an enhanced YOLOv5 tomato recognition algorithm aimed at the statistical assessment and estimation of tomato fruit yield in their natural growth conditions. Our approach involved two key enhancements: firstly, we substituted the final layer of the backbone network with a Separable Vision Transformer to augment the connectivity between the backbone network and global context, thereby facilitating tomato feature extraction; secondly, we incorporated the WIOU loss function and employ the Mish activation function to enhance convergence speed and accuracy. Experimental findings demonstrate that the improved detection model achieves a mAP score of 99.5%, reflecting 1.1 percentage points enhancement compared to the conventional YOLOv5 model, and the processing time for every image is 15ms. Furthermore, the improved YOLOv5 algorithm exhibits superior recognition performance for densely populated and occluded tomato fruits.

Key words: Greenhouse tomato; YOLOv5; Attention mechanism; Loss function

随着我国经济的发展, 农业正在向现代化转型, 温室大棚已成为现代农业的重要支撑, 而大棚番茄种植也在全国范围内快速发展。准确估算大棚番茄的产量是种植规划和营销策略的重要组成部分, 实时统计和预测番茄果实产量信息, 并加以相应的生产控制, 可以解决番茄产能波动和生产过程中的不连续性问题。然而, 大棚内的番茄茎叶和果实密集且相互重叠^[1], 限制了果实图像特征的识别, 进而影响了番茄产量的准确估算。因此, 实现准确获取番茄果实视觉信息是支持智能估产的重要前提。

传统的番茄识别方法通常是手动设计特征提取器来提取目标特征, 如梯度直方图 (Histogram of Oriented Gradient, HOG)、尺度不变特征变换 (Scale-invariant feature transform, SIFT) 和 Haar-like 特征等, 这些特征再输入支持向量机 (Support Vector Machine, SVM)、AdaBoost、随机森林 (Random Forest, RF) 等分类器进行分类和识别^[2]。Liu 等^[3]应用伪色去除技术和 SVM 达到了 94.41% 的识别率, 但在番茄遮挡和重叠等复杂情况下, 效果不佳。李寒等^[4]采用 RGB-D 相机拍摄图像, 通过预处理得到水果轮廓, 采用 K 均值聚类与自组织映射算法进行

收稿日期: 2023-09-09; 修回日期: 2023-10-30

基金项目: 四川省科技创新苗子工程 (2022040)

作者简介: 杨 健, 女, 副教授, 研究方向为智能机器人。E-mail: 1459555755@qq.com

共同第一作者: 杨啸治, 男, 在读硕士研究生, 研究方向为目标检测与深度学习。E-mail: 2021051189@stu.cdut.edu.cn

番茄识别,识别率达到了 87.2%,但轮廓提取易受光照影响。Malik 等^[5]采用改进的 HSV 颜色空间算法和分水岭分割算法分离红色番茄,但识别率只有 81.6%。而马翠花等^[6]采用基于密度的局部自适应阈值分割算法提高番茄检测精度,对未成熟番茄的识别率只达到 77.6%。这些传统方法泛化性差、计算复杂度高,难以提取出合理的特征。此外,它们往往没有考虑到大棚环境中的复杂影响因素,对各种特征的变化鲁棒性不足,难以满足实际需求。而近年来兴起的深度卷积神经网络为获取番茄果实视觉信息提供了更好的方法。

随着深度学习的不断发展,目前的目标检测算法可以分为两类。第一类是基于候选区域的 two stage 算法,例如 R-CNN^[7]、Fast R-CNN^[8]和 Faster R-CNN^[9]等。这类算法通过两个步骤来进行计算:首先选择候选框,然后对候选框进行分类或回归。这种方法具有高鲁棒性和低识别错误率,但运算时间长,占用磁盘空间大,并且对图像信息进行了重复计算,不适合实时检测应用。第二类是基于网络模型的 one stage 算法,例如 YOLO^[10-11]和 SSD^[12]等。这类算法通过在图像上以不同尺度进行遍历抽样,并利用卷积神经网络提取特征后直接进行回归。这种方法的网络模型较为简单,识别速度快,特别适合实时检测应用^[13]。其中,YOLO 模型是 Redmon 等^[10]于 2017 年提出的对象检测模型,与 two stage 算法相比,YOLO 模型使用单个前馈网络直接预测边界框及其对应的类,作为一种基于回归的检测器,不仅支持快速实时识别,其精度也能够满足一般应用要求。

2018 年,周云成等^[14]提出了一种基于 Fast R-CNN 的双卷积链算法,对番茄果实识别的平均精度为 63.99%。2021 年,张文静等^[15]将 AlexNet 作为 Faster R-CNN 的骨干网络,在 50%的重叠度下,番茄检测的平均精度为 83.9%,每幅图像的处理时间为 245 ms。2020 年,刘芳等^[16]通过改进 YOLO 网络模型,可在复杂环境中有效地检测番茄果实,识别准确率达到 97.13%。同年,Liu 等^[17]提出一种用于检测番茄果实的改进 YOLOv3 模型,在轻度遮挡条件下,番茄果实的识别准确率达到 94.58%,但在遮挡严重时表现较差。2021 年,成伟等^[18]使用广义交并比 (Generalized Intersection over Union^[19], GIOU),作为改进 YOLOv3 算法的位置损失函数,在番茄果实检测方面的平均精度达到 95.7%,每幅图像的平均处理时间为 15 ms^[19]。

综上所述,笔者针对大棚环境下番茄果实重叠、枝叶遮挡、光照不均等复杂情况,提出一种改进的 YOLOv5 目标检测算法。首先,使用可分离视觉转换器 (SepViT) 替换骨干网络的最后一层,以增强骨干网络与全局信息的联系和提取番茄的特征;其次,利用 WIOU 损失函数和 Mish 激活函数来同时提高收敛速度和精度,使得 YOLOv5 能够在大棚环境下获得对番茄果实较高的检测精度。

1 番茄图像数据集制作

在笔者研究中使用的番茄数据集部分是在四川省成都市拍摄的,部分是从参考文献中获得的。这些图像有效地反映了番茄生长的复杂性,如大小和光线的差异、番茄之间的遮挡以及枝叶的遮挡。在各种条件下,包括侧光、背光、阴影等,随机拍摄了 1000 余张番茄图像,图 1 所示为已建立的番茄数据集,拍摄时间为 2022 年 7 月,该番茄识别数据集已经在 Zenodo 上公开发布,可以在 <https://zenodo.org/record/8429227> 上获取。

为了增强原始图像数据,提高数据集的普适性和鲁棒性,笔者采用了空间级和像素级数据增强方法,且同时使用了这两类增强方法,如图 2 所示。空间级变换技术修改边界框和图像,而像素级变换技术修改图像,同时保持边界框不变。

2 基于改进 YOLOv5 的番茄识别模型

2.1 YOLOv5 目标检测算法

YOLOv5 是一种基于卷积神经网络 (Convolutional Neural Networks, CNN) 的无锚点对象检测算法,相比其他对象检测算法具有更高的效率和准确性,其原理如图 3 所示。

YOLOv5 算法的基本原理是将输入的图片分成 $S \times S$ 个栅格化单元格。如果检测到的目标中心落入特定的单元格中,则该单元格负责检测目标,即自身有目标的概率为 $P_{obj} = 1$ 。然后,每个单元格预设生成 B 个先验边界框,每个边界框与真实值边界框的交并比为 IOU。因此,图像中的目标位置和类别预测可以表示为 $S \times S \times B \times (4 + 1 + C)$ 的张量,其中 4 表示先验边界框的坐标 (x, y) 和宽高 (w, h) , 1 表示置信度得分,总共有 5 个特征参数, C 表示使用的数据集中目标的类别数。通过连续回归训练真实边界框,可以得到最终预测的目标位置、置信度和类别信息。最后,保留置信度最高的



图 1 番茄数据集

Fig. 1 Tomato dataset



图 2 数据增强

Fig. 2 Data enhancement

边界框,以筛选出最佳的识别结果^[20]。

2.2 深度可分离视觉注意力机制

最近几年,视觉转换器(Vision Transformer,

ViT)在各种计算机视觉任务中取得了巨大的成功,其性能在关键领域甚至超过了传统的卷积神经网络^[21-22]。然而,这些性能提升通常是以增加计算复

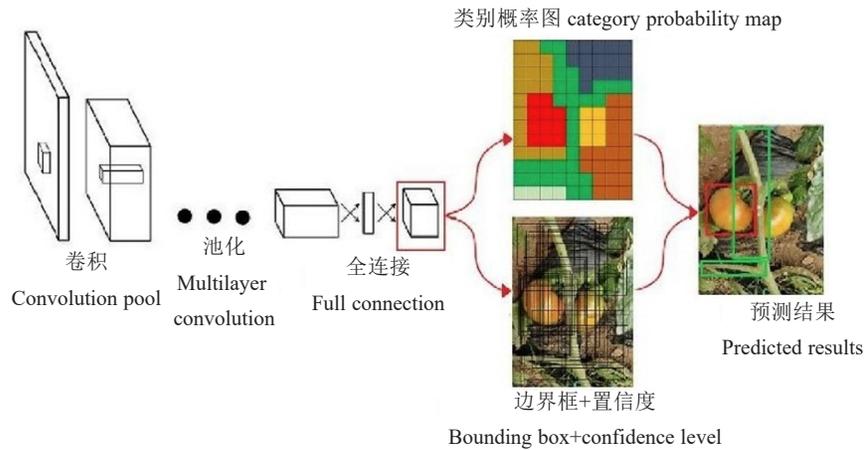


图3 YOLO 算法原理

Fig. 3 Principle of YOLO algorithm

杂度和参数数量为代价的,而深度可分离视觉注意力机制可以通过平衡计算成本和准确性来提高模型的性能^[23]。

笔者将骨干网络的最后一层替换为 SepViT Block,以增强模型的特征提取能力,并优化与全局信息的联系。在 SepViT Block 中,Depthwise selfattention (DWA)和 Pointwise selfattention(PWA)可以减少计算量,并实现窗口内的局部信息通信和全局信息交互。首先,输入的特征图划分为多个窗口,每个窗口被认为是特征图的一个输入通道,不同类型的信息被包含在每个窗口中;然后,对每个窗口及其像素标记执行 DWA,这有助于融合通道的空间信息,类似于 MobileNet^[24-26]中的深度卷积。DWA 的操作如下:

$$DWA(f) = Attention(f \cdot W_Q, f \cdot W_K, f \cdot W_V)$$

其中 f 是特征标记,由窗口标记和像素标记组成, W_Q 、 W_K 和 W_V 代表常规自注意力任务中的 query、key、value 三个线性层,PWA 类似于 MobileNet 中的 pointwise convolution 操作,不同之处在于 point-

wise convolution 用于融合不同通道的信息,而 PWA 则是在窗口之间建立连接。完成 DWA 操作后,PWA 建立窗口之间的关系,并通过 Layer Normalization (LN)和 Gelu 激活函数生成注意力图。PWA 的运行过程如下:

$$PWA(f, wt) = Attention\{Gelu[LN(wt)] \cdot W_Q, Gelu[LN(wt)] \cdot W_K, f\}$$

其中 wt 代表窗口标记。

SepViT Block 可以表示为:

$$\tilde{f}^n = Concat(f^{n-1}, wt);$$

$$\dot{j}^n = DWA[LN(\tilde{f}^n)];$$

$$\dot{f}^n, \dot{wt} = Slice(\dot{j}^n);$$

$$\hat{f}^n = PWA(\dot{f}^n, \dot{wt}) + f^{n-1};$$

$$f^n = MLP[LN(\hat{f}^n)] + \hat{f}^n$$

其中 f^n 表示 SepViT 块, \dot{j}^n 和 \dot{wt} 是特征映射和学习的窗口标记,Concat 表示串联操作,Slice 表示切片操作,图 4 为 SepViT 块的结构。

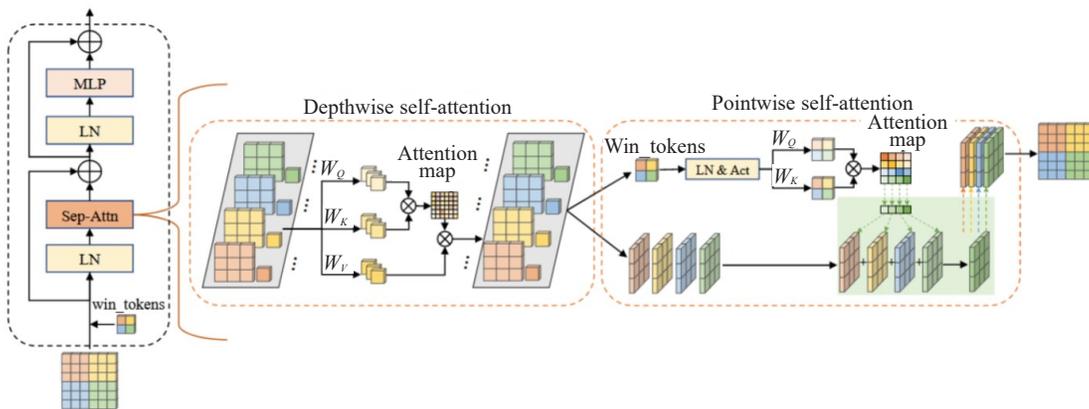


图4 SepVit 块结构

Fig. 4 SepVit block structure

2.3 损失函数与激活函数

YOLOv5 的损失函数由两部分组成:分类损失和边界框回归损失。边界框回归损失最经典的形式是 IOU(Intersection over Union)^[27]损失, YOLO 系列中最常用的版本是 CIOU(Complete Intersection over Union)^[28]。CIOU 通过预测框中心点欧氏距离和重叠率参数作为预测框偏差的偏差指标。

$$\begin{cases} CIOU_{ij} = \frac{J}{U} - \frac{\rho^2(b, b^{gt})}{c^2} - \alpha v; \\ \alpha = \frac{v}{(1-IoU) + v}; \\ v = \frac{4}{\pi^2} \left(\arctan \frac{\omega^{gt}}{h^{gt}} - \arctan \frac{\omega}{h} \right)^2. \end{cases}$$

其中 (ω^{gt}, h^{gt}) 、 (ω, h) 分别表示预测框和真实框的高与宽, b, b^{gt} 分别表示预测框和真实框的中心点, ρ 表示两个中心点间的欧氏距离, c 表示最小外接矩形框对角线距离, α 表示权重函数, v 表示真实框与预测框矩形对角线倾斜角的方差, 如图 5 所示。

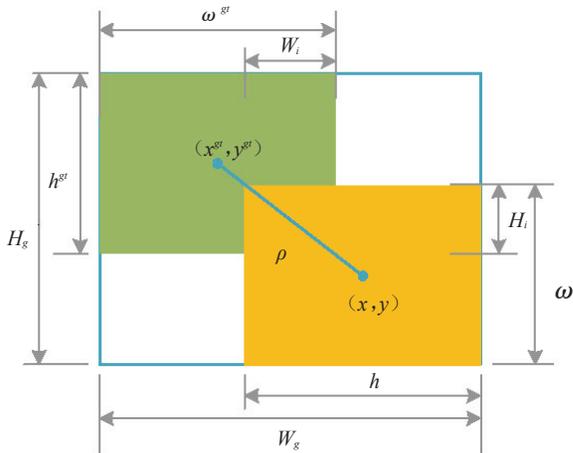


图 5 损失函数边框图

Fig. 5 Loss function border diagram

由于大棚内的复杂环境,训练数据不可避免地包含了低质量示例,从而降低模型的泛化性能。一个好的损失函数应该在 anchor box 和 target box 很好重合的情况下减弱几何因素的惩罚。因此,笔者选择了 WIOU^[29]替换 CIOU,来提高模型的整体性能。

$$L_{WIOU} = R_{WIOU} L_{IOU};$$

$$R_{WIOU} = \exp \left[\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*} \right].$$

式中 W_g, H_g 是最小封闭框的尺寸。为了防止 R_{WIOU} 产生阻碍收敛的梯度,将 W_g, H_g 从计算图中

分离出来(上标*表示该操作)。因为它有效地消除了阻碍收敛的因素,所以不需要引入纵横比等指标。

在 YOLOv5 中,原先使用 ReLU 作为激活函数,但 ReLU 在负值时会直接截断,导致梯度的损失。因此,笔者使用 Mish^[30]激活函数代替 ReLU 激活函数,并将其应用于训练和推理过程中。Mish 是一种光滑且非单调的激活函数,其定义为:

$$f(x) = x \tanh[\text{softplus}(x)] = x \tanh[\ln(1 + e^x)].$$

Mish 激活函数的优势在于它没有上界,可以避免梯度饱和的问题。从图 6 中可以看出,当输入值为负时, Mish 允许较小的负梯度流入,从而保证信息的流动性,同时稳定了网络梯度的流动,起到了强正则化的作用。此外, Mish 函数是一个光滑函数,允许更多的信息深入神经网络中,因此,使用 Mish 函数可以提高网络的精确度和泛化能力。

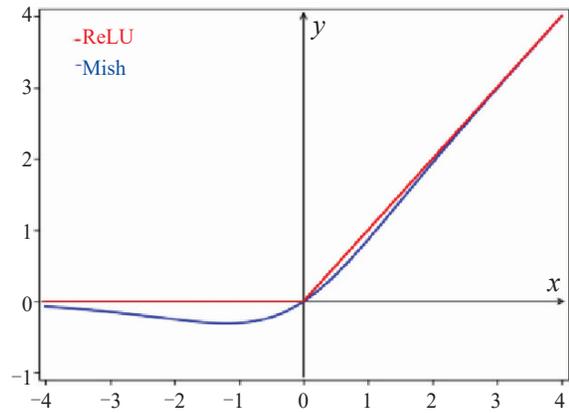


图 6 激活函数图

Fig. 6 Activation function graph

3 结果与分析

3.1 番茄检测算法试验

试验的主要硬件平台为 Dell Precision 3660 Tower 工作站,搭载 CPU 为 Intel 12500×3 GHz,显卡为 NVIDIA RTX A4000,软件平台包括 CUDA11.1、Python3.8.8 和 PyTorch1.8.0。在试验中,将数据集按照 8:1:1 的比例随机分为训练集、验证集和测试集,并使用 SGD 优化器进行模型训练。设置批处理大小为 64,初始学习率为 0.01,输入图像的大小为 640×640,训练周期为 900。

使用可视化工具 TensorBoard 记录训练过程中损失函数值和精度值的变化,图 7 和图 8 展示了损失函数和 mAP 的变化趋势。

在前 200 个迭代循环中,损失函数的值呈明显

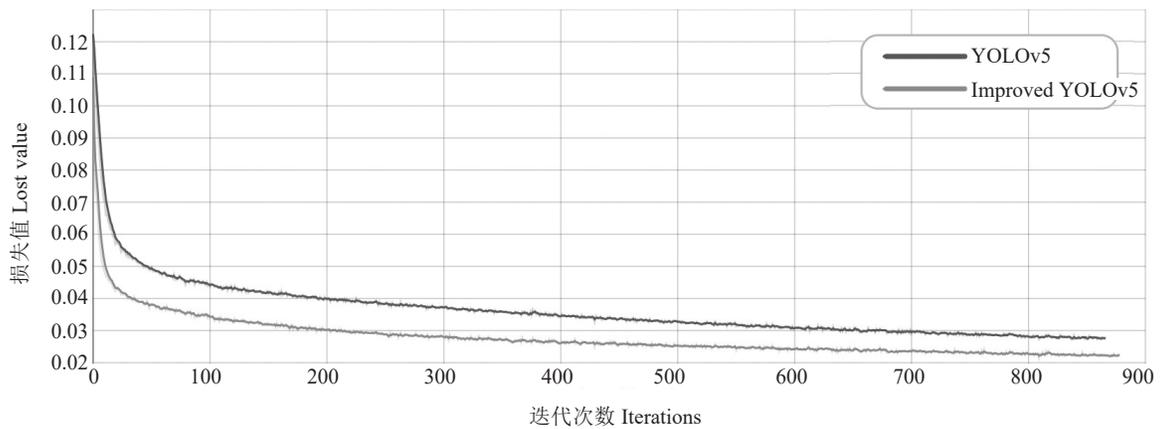


图7 损失函数值变化趋势

Fig. 7 Trend of loss function values

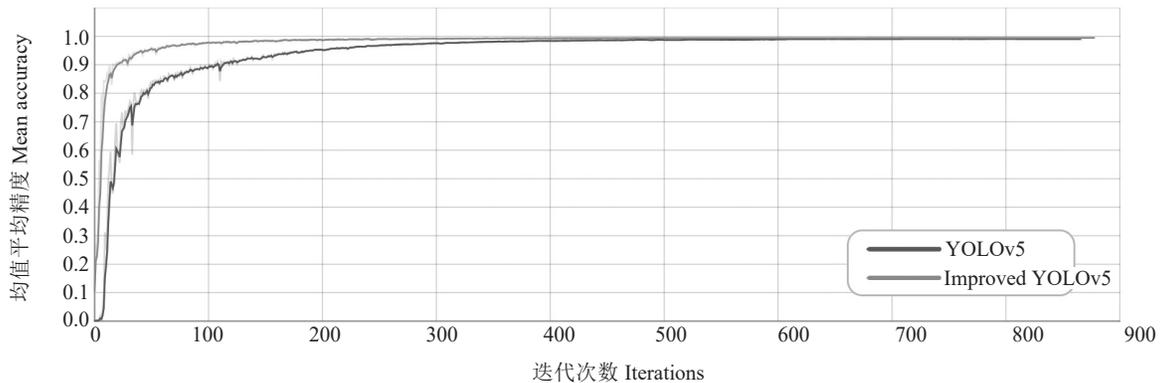


图8 均值平均精度变化趋势

Fig. 8 Trend in mean average precision

下降的趋势,随后下降速度减缓。经过 600 个迭代循环后,损失函数的值降至 3 附近,表明模型已经达到稳定收敛状态。在训练过程中,每隔 10 个迭代周期输出 1 次模型,并对测试集图像进行识别和处理。使用平均精度(mAP)作为评价指标,选择具有最高精度的模型作为最优模型。表 1 是与改进模型相比,使用相同训练过程的 YOLOv5 和改进模型的对比结果。

表 1 算法对比

Table 1 Comparison of algorithms

算法 Algorithm	平均精度值 Mean average precision/ %	检测速度 Detection speed/ms
YOLOv5	98.4	10
Improved YOLOv5	99.5	15

传统 YOLOv5 的 mAP 值为 98.4%,改进后的 YOLOv5 为 99.5%,提高了 1.1 个百分点,检测改进算法的效率与传统算法基本相同。可以看出,改进后的模型具有更高的准确度,能够更好地识别和定位番茄果实。

3.2 番茄产量估计试验

试验方法如下:(1)选择不同光线和尺度的视野,随机采集番茄植株图像,保证数据的多样性;(2)特意选择一些稀疏、密集和模糊果实的特殊图像样本,人工统计试验图像中红色和绿色果实的数量作为对照。(3)利用 YOLOv5 模型及改进后的 YOLOv5 模型对采集图像进行模式识别,对视野内的水果进行计数,并与人工计数结果进行对比,评价产量估算模型的准确性。

通过参考红色和绿色番茄的人工计数结果,笔者评估了 YOLOv5 和改进的 YOLOv5 对番茄的识别计数结果。对红番茄、绿番茄和总番茄的识别计数结果与人工计数进行比较,并以计算的比值作为红番茄、绿番茄和总番茄估产准确率的衡量指标。识别结果如图 9、10 所示,统计结果如表 2 所示。

可以看出,改进后的 YOLOv5 模型对密集红色番茄和绿色番茄的产量估算精度都有明显提高:密集红色番茄的估算精度提高了 1.5 个百分点,密集绿色番茄的估算精度提高了 2.7 个百分点。这表明改进后的 YOLOv5 模型在识别和估算密集的番茄



图 9 YOLOv5 算法
Fig. 9 YOLOv5 algorithm

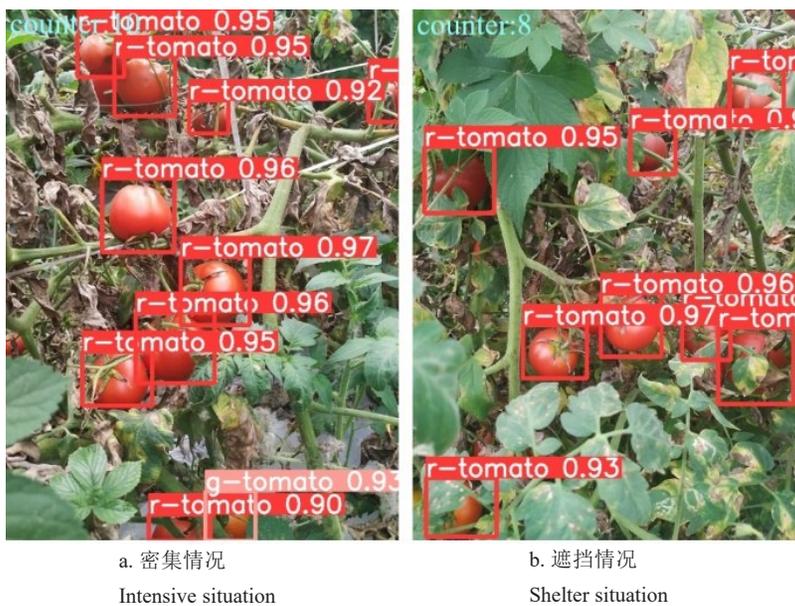


图 10 改进的 YOLOv5 算法
Fig. 10 Improved YOLOv5 algorithm

表 2 番茄果实产量估算准确率统计
Table 2 Statistics of accuracy of tomato fruit yield estimation

算法 Algorithm	红色番茄 Red tomato/%		绿色番茄 Green tomato/%		总产量 Total yield/%
	密集 Dense	遮挡 Shelter	密集 Dense	遮挡 Shelter	
	YOLOv5	98.2	97.4	96.4	
Improved YOLOv5	99.7	99.2	99.1	98.9	99.3

群体时更为准确。且改进后的 YOLOv5 模型对遮挡红色番茄和绿色番茄的产量估算精度也有明显提高:遮挡红色番茄的估算精度提高了 1.8 个百分

点,遮挡绿色番茄的估算精度提高了 3.2 个百分点。这意味着改进后的 YOLOv5 模型在处理遮挡情况下的番茄估算更为可靠。

4 讨论与结论

笔者提出了一种用于复杂环境下的改进 YOLOv5 算法,对自然生长条件下的番茄果实产量进行统计和估算。通过改进传统的 YOLOv5 模型的骨干网络和损失函数,构建了一种用于自然生长下番茄果实识别的模型。改进后的 YOLOv5 模型识别准确率得到提升,最终模型的 mAP 值为 99.5%,

比未改进的 YOLOv5 模型提高了 1.1 个百分点。改进后的 YOLOv5 模型对处于密集和遮挡条件下的番茄果实有更明显的识别效果和鲁棒性。密集条件下的红色番茄和绿色番茄的估算准确率分别为 99.7% 和 99.1%，遮挡条件下的红色番茄和绿色番茄的估算准确率分别为 99.2% 和 98.9%。通过应用笔者提出的番茄识别网络模型，能够有效解决传统图像检测方法在处理番茄果实重叠、枝叶遮挡和光照不均等复杂情况下定位检测精度低的问题，为后续研究提供了更有力的技术支持。

随着技术不断进步，未来的研究方向包括利用迁移学习和自监督学习来增强模型的泛化性能，实施目标跟踪以追踪番茄状态，进一步优化精细的番茄定位和计数，以及将番茄识别与农业决策支持系统紧密融合，以提供实时的农田管理建议。

参考文献

- [1] 张俊宁, 毕泽洋, 闫英, 等. 基于注意力机制与改进 YOLO 的温室番茄快速识别[J]. 农业机械学报, 2023, 54(5): 236-243.
- [2] 孙皓泽, 常天庆, 王全东, 等. 一种基于分层多尺度卷积特征提取的坦克装甲目标图像检测方法[J]. 兵工学报, 2017, 38(9): 1681-1691.
- [3] LIU G X, MAO S Y, KIM J H. A mature-tomato detection algorithm using machine learning and color analysis[J]. Sensors, 2019, 19(9): 2023.
- [4] 李寒, 陶涵斌, 崔立昊, 等. 基于 SOM-K-means 算法的番茄果实识别与定位方法[J]. 农业机械学报, 2021, 52(1): 23-29.
- [5] MALIK M H, ZHANG T, LI H, et al. Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm[J]. IFAC-Papers On Line, 2018, 51(17): 431-436.
- [6] 马翠花, 张学平, 李育涛, 等. 基于显著性检测与改进 Hough 变换方法识别未成熟番茄[J]. 农业工程学报, 2016, 32(14): 219-226.
- [7] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2014.
- [8] GIRSHICK R. Fast R-CNN[C]. IEEE International Conference on Computer Vision, 2015.
- [9] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. Advances in Neural Information Processing Systems, 2015.
- [10] REDMON J, FARHADI A. YOLO9000: Better, faster, stronger[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017.
- [11] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [12] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: Single shot multibox detector[C]. Computer Vision - ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11- 14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016.
- [13] 何斌, 张亦博, 龚健林, 等. 基于改进 YOLO v5 的夜间温室番茄果实快速识别[J]. 农业机械学报, 2022, 53(5): 201-208.
- [14] 周云成, 许童羽, 邓寒冰, 等. 基于双卷积链 Fast R-CNN 的番茄关键器官识别方法[J]. 沈阳农业大学学报, 2018, 49(1): 65-74.
- [15] 张文静, 赵性祥, 丁睿柔, 等. 基于 Faster R-CNN 算法的番茄识别检测方法[J]. 山东农业大学学报(自然科学版), 2021, 52(4): 624-630.
- [16] 刘芳, 刘玉坤, 林森, 等. 基于改进型 YOLO 的复杂环境下番茄果实快速识别方法[J]. 农业机械学报, 2020, 51(6): 229-237.
- [17] LIU G X, NOUAZE J C, TOUKO P L, et al. YOLO-Tomato: A robust algorithm for tomato detection based on YOLOv3[J]. Sensors-Basel, 2020, 20(7): 2145.
- [18] 成伟, 张文爱, 冯青春, 等. 基于改进 YOLOv3 的温室番茄果实识别估产方法[J]. 中国农机化学报, 2021, 42(4): 176-182.
- [19] REZATOFIHI H, TSOI N, GWAK J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019: 658-666.
- [20] 成伟. 基于深度学习的番茄植株多目标识别方法研究[D]. 太原: 太原理工大学, 2021.
- [21] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]. Proceedings of the 31st Information Processing Systems, 2017: 6000-6010.
- [22] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16×16 words: Transformers for image recognition at scale[J]. arXiv: 2010.11929, 2020.
- [23] LI W, WANG X, XIA X, et al. Sepvit: Separable vision transformer[J]. arXiv: 2203.15380, 2022.
- [24] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv: 1704.04861, 2017.
- [25] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: Inverted residuals and linear bottlenecks[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018: 4510-4520.
- [26] ANDREW H, MARK S, GRACE C, et al. Searching for mobile-netv3[C]. Proceedings of the IEEE/CVF international conference on computer vision, 2019: 1314-1324.
- [27] YU J H, JIANG Y N, WANG Z Y, et al. Unitbox: An advanced object detection network[C]. Proceedings of the 24th ACM international conference on Multimedia, 2016: 516-520.
- [28] ZHENG Z H, WANG P, REN D W, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation[J]. IEEE Transactions on Cybernetics, 2021, 52(8): 8574-8586.
- [29] TONG Z J, CHEN Y H, XU Z W, et al. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism[J]. arXiv: 2301.10051, 2023.
- [30] MISRA D. Mish: A self regularized non-monotonic activation function[J]. arXiv: 1908.08681, 2019.