江苏农业学报(*Jiangsu J.of Agr.Sci.*),2024,40(3):478-489
http://jsnyxb.jaas.ac.cn

任守纲,朱勇杰,顾兴健,等. 基于稀疏实例与位置感知卷积的植物叶片实时分割方法[J].江苏农业学报,2024,40(3):478-489.
doi:10.3969/j.issn.1000-4440.2024.03.010

基于稀疏实例与位置感知卷积的植物叶片实时分割方法

任守纲[1,2], 朱勇杰[1], 顾兴健[1], 武鹏飞[3], 徐焕良[1]
(1.南京农业大学人工智能学院,江苏 南京 210095; 2.国家信息农业工程技术中心,江苏 南京 210095; 3.新疆兴农网信息中心/新疆维吾尔自治区农业气象台,新疆 乌鲁木齐 830002)

摘要: 植物叶片分割在高通量植物表型数据获取任务中起着关键作用。目前,多数植物叶片分割方法专注于提高模型分割精度,却忽视模型复杂度和推理速度。针对该问题,本研究提出一种基于稀疏实例激活与有效位置感知卷积的实例分割模型(ePaCC-SparseInst),实现植物叶片实时、精确分割。在 ePaCC-SparseInst 中引入 1 组稀疏实例激活图作为叶片对象表示方式,并使用二部图匹配算法实现预测对象与实例激活图的一一映射,从而避免了繁琐的非极大值抑制(Non-maximum suppression,NMS)运算,提高了模型的推理速度。此外,在实例分支中引入有效位置感知卷积(ePaCC)模块,在增大模型全局感受野的同时提高了模型的推理速度。在 Komatsuna 数据集上,ePaCC-SparseInst 平均分割精度(*AP*)达到 85.33%,每秒传输帧数达到 43.52。在相同训练条件下,ePaCC-SparseInst 的性能优于 SparseInst、Mask R-CNN、CondInst 等实例分割算法。此外在 CVPPP A5 数据集上,ePaCC-SparseInst 较上述算法同样取得了更好的分割精度和推理速度。本研究提出的方法采用纯卷积的架构实现了叶片的实时分割,可以为在移动端或边缘设备上获取植物表型数据提供技术支持。

关键词: 实例分割; 计算机视觉; 植物表型; 叶片分割
中图分类号: TP391.41 **文献标识码:** A **文章编号:** 1000-4440(2024)03-0478-12

Real-time segmentation of plant leaves based on sparse instances and position aware convolution

REN Shou-gang[1,2], ZHU Yong-jie[1], GU Xing-jian[1], WU Peng-fei[3], XU Huan-liang[1]
(1.*College of Artificial Intelligence, Nanjing Agricultural University, Nanjing* 210095, *China*; 2.*National Engineering and Technology Center for Information Agriculture, Nanjing* 210095, *China*; 3.*Xinjiang Xingnong Network Information Center/Xinjiang Uygur Autonomous Region Agricultural Meteorological Observatory, Urumqi* 830002, *China*)

Abstract: The segmentation of plant leaves plays a crucial role in high-throughput plant phenotyping data acquisition tasks. Currently, most methods for plant leaf segmentation focus on improving the accuracy of the segmentation model but overlook the model's complexity and inference speed. In response to this issue, this study proposed an instance segmentation model (ePaCC-SparseInst) based on sparse instance activation and efficient position-aware convolution to achieve real-time and accurate segmentation of plant leaves. In ePaCC-SparseInst, a set of sparse instance activation maps was introduced as the representation of leaf objects. A bipartite graph matching algorithm was employed to establish a one-to-one mapping between predicted objects and instance activation maps, thereby avoiding the cumbersome non-maximum suppression (NMS) operation and improving the model's inference speed. Additionally, an effective position-aware circulate convolution (ePaCC) module was introduced into

收稿日期:2023-01-13
基金项目:国家自然科学基金项目(61806097)
作者简介:任守纲(1977-),男,山东日照人,博士,副教授,主要从事软件工程、人工智能研究。(E-mail)rensg@njau.edu.cn
通讯作者:武鹏飞,(E-mail)445305370@qq.com

the instance branch, which increased the model's global receptive field and enhanced its inference speed. On the Komatsuna dataset, ePaCC-SparseInst achieved an average segmentation precision (*AP*) of 85.33% and an inference speed of 43.52 frames per second (*FPS*). Under the same training conditions, its performance surpassed instance segmentation algorithms such as SparseInst, Mask R-CNN, and CondInst. Furthermore, on the CVPPP A5 dataset, ePaCC-SparseInst achieved better segmentation accuracy and inference speed than the aforementioned algorithms. The proposed method used a pure convolutional architecture to achieve real-time leaf segmentation, which could provide technical support for obtaining plant phenotypic data on mobile or edge devices.

Key words: instance segmentation; computer vision; plant phenotypes; leaf segmentation

植物叶片在揭示植物生长发育状况的过程中起到关键作用。例如叶面积和叶片的形状(叶长、叶宽、叶倾角等)与植物的光合作用、呼吸作用、蒸腾作用以及碳和养分同化等多种生理活动密切相关[1]。因此借助非破坏性方法获取叶片特征数据,与植物基因数据相结合,能为改善植株的遗传性状提供参考,从而提高作物产量[2]。在高通量植物表型任务中,分割出植物叶片是获得生物量特征的重要前提之一[3]。因此叶片分割是植物表型任务中的关键步骤,精准且快速地从复杂的背景中分割出叶片区域,有助于植物叶片表型参数的快速提取,同时也对植物生长状况的同步观察以及病虫害的实时监督等方面有着重要的意义[4]。叶片分割任务主要分为单目标叶片分割和多目标叶片分割2类[5-7]。

在单目标叶片分割方面,通常优先采用传统的图像分割方法或者基于深度学习的语义分割方法。蒋焕煜等[8]将基于形态学的分水岭算法用于叶片边缘分割,借此提取穴孔中的番茄幼苗的叶片面积和叶片周长。孙国祥等[9]提出了基于边缘链码信息的番茄苗重叠叶面分割算法,对72孔和128孔的分割准确度达到了100%和96%,分割单张图片的平均耗时分别达到0.835 s和0.990 s。王纪章等[10]借助Kinect相机对彩色叶片图像进行预处理、阈值分割、形态学运算以及连通分量统计等操作,实现了幼苗的无损检测,该算法估算出的叶面积与实际叶面积的平均误差仅为2.15%。针对绿色叶片图像分割困难的问题,伍艳莲等[11]提出一种改进的均值漂移算法,实现了对绿色作物的精确分割,性能上优于常规的均值漂移算法且图像的错分率小于6.5%。面对不同的背景、阴影等成像环境,单一模型往往难以很好地解决植物叶片分割问题,胡静等[12]提出一种基于鲁棒随机游走的交互式植物叶片分割方法,通过人机交互获取前景的像素信息,获得了光滑、鲁棒的植物叶片分割结果图。Kan等[13]提出的基于U-net的叶片分割模型,采用了ResNet-50作为主干网络用于编码图片,并在解码器中引入特征融合模块,更好地融合了来自底层的位置信息以及来自高层的语义信息。从上述的方法可以看出,传统的图像分割算法和基于深度学习的语义分割算法,虽然大多有着不错的分割效果,但是无法分割出植物中的每一张叶片。而实例分割算法则能将图片中的每一个实例对象都分割出来,即使这些实例对象属于同一个类别。

在多目标叶片分割方面,目前已有较多基于实例分割的研究。Yin等[14]提出一种基于倒角匹配的多叶片对齐算法,该算法能找到输入图像中多个叶片的最佳对齐方式,且能应用于叶片的实例分割当中。Ren等[15]率先将循环神经网络(Recurrent neural network, RNN)应用在叶片实例分割任务中,该算法能一次一个地查询叶片实例并对其进行实例分割,即使在遮挡场景下性能也不会出现衰退。随着He等[16]提出Mask R-CNN,基于卷积神经网络(Convolutional neural network, CNN)的实例分割网络占据了绝大部分的实例分割任务,与上述2种算法相比,以Mask R-CNN为代表的实例分割算法有着更高的分割精度。乔虹等[17]采用Mask R-CNN对大田环境下的葡萄叶片进行了检测与分割,结果显示不同的葡萄叶片分割精度都大于等于0.88。该葡萄叶片检测与分割算法展示了Mask R-CNN对不同天气状况以及各种复杂背景下的叶片分割任务都有着较好的鲁棒性和较高的分割精度。但是该算法在叶片边缘的分割精度依旧不够理想,这是由Mask R-CNN经过感兴趣区域(ROI)裁剪导致特征图分辨率降低等原因导致的。针对以上问题,袁山等[18]采用结合了注意力机制的Cascade mask R-CNN,解决叶片遮挡以及叶片边缘特征不明显等问题。以Mask R-CNN

为代表的两阶段实例分割算法,虽然在精度上有着很大的优势,但是其在推理速度方面仍存在着较大的缺陷。例如在袁山等的工作中可以看到,Mask R-CNN 推理 1 张图片需要耗时 0.383 s,Cascade mask R-CNN 需要耗时 0.387 s,以及其改进模型需要耗时 0.562 s。而 Guo 等[4]提出了一种端到端无锚的单阶段实例分割算法,以提高叶片的分割精度。该算法省去了大多数实例分割算法所依赖的锚框,节约了大量实例筛选的时间,但其在解码器中加入的多尺度注意力模块以及掩码细化模块又降低了推理的速度。由上述方法可以看出,模型分割精度与推理速度之间存在一定的对立性和矛盾性。若将上述模型应用在移动端或边缘设备上时,会对模型的部署和使用造成较大的困难[19]。SparseInst 模型[20]作为现有实时实例分割的基准模型,有着较快的推理速度和较高的分割精度,同时很好地控制模型的大小,但在实例识别时,由于其感受野的局限性,会导致面积较小的叶片分割错误。

Hu 等[21]在 Faster R-CNN[22]目标检测算法的检测分支中引入一个带有自注意力机制的对象关系模块,该模块通过目标对象的外观特征和形状之间的潜在联系同时处理这一组对象,对这一组对象之间的关系进行建模,提高了算法的检测精度。在实例分割的研究中,Guo 等[23]在 SOLOv2[24]的核分支(Kernel branch)中引入了 Vision Transformer(ViT),试图捕获长距离的上下文依赖,获得全局特征信息,以更好地区分具有相同语义类别的重叠实例。受这 2 个工作的启发,本研究拟在 SparseInst 实例分支中引入一个带有全局感受野的有效位置感知卷积(ePaCC)模块替换原有的普通卷积层,通过该模块获取长距离的上下文语义信息,输出全局性、语义信息性更强的实例激活图,在保持精度不降低的情况下,提高叶片实例分割的速度。此外,在 ePaCC 模块中引入 eSE 注意力模块,缓解模型通道信息丢失问题的同时进一步提升模型的推理速度。最后使用基于匈牙利算法的二部图匹配算法实现预测对象与真实标签(GT)的一一映射,避免繁琐的参数设计和后处理操作,以期实现植物叶片实时实例分割。

1 材料与方法

1.1 试验数据集

Komatsuna 植物叶片数据集[7]总共有 900 张小松菜各个生长阶段的图片。本研究随机选取 720 张图片作为训练集,剩余的 180 张图片作为测试集。其中每一张图片的分辨率为480×480。图 1 展示了部分 Komatsuna 数据集的原始图像以及对应的标注信息。

CVPPP[6]中的 A5 数据集由不同尺寸的烟草花和拟南芥图像组成,共计 810 张图像。其中,拟南芥原始图像分辨率为:500×530、530×565 和441×441,烟草花原始图像的分辨率为:2 448×2 048。本研究将 810 张图像划分为训练集(648 张)和测试集(162 张)。

1.2 数据增强

数据增强是解决深度学习样本较少问题的常用方法之一[25]。本研究使用 Detectron2 自带的随机缩放图像的最小边长进行数据增强。输入图像的最小边长会随机调整为 352、384、416、448、480,使得输入模型训练的叶片图像为 720 张,而实际训练的图片数量为3 600张,这基本满足了实例分割训练模型对图片数量的要求。

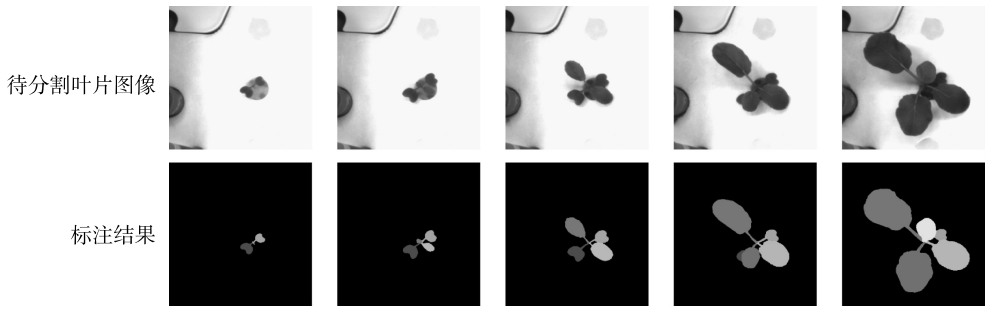
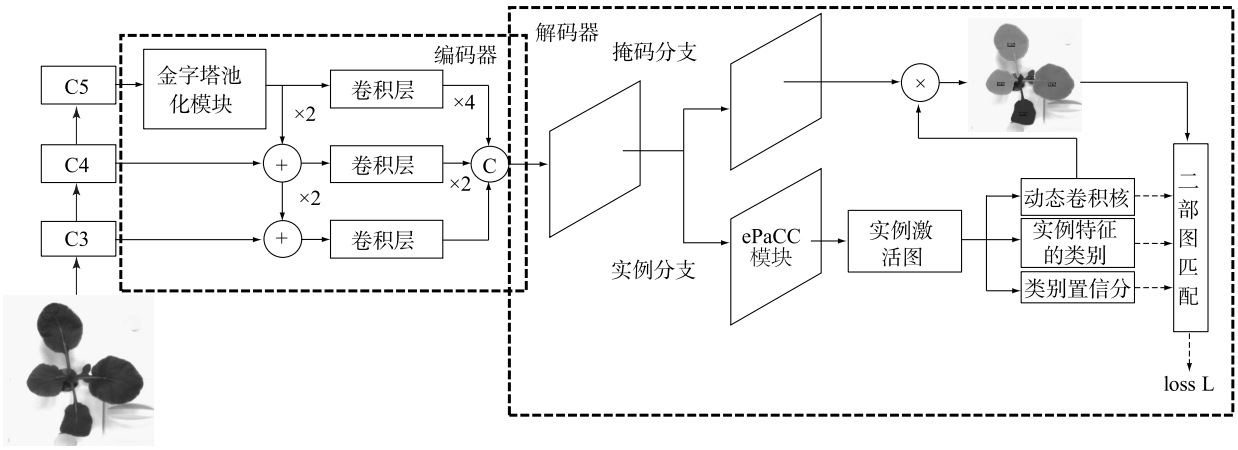


图 1 Komatsuna 数据集部分待分割叶片图像及其标注

Fig.1 Part of leaf images to be segmented and their annotations in Komatsuna dataset

1.3 ePaCC-SparseInst 模型概述

ePaCC-SparseInst 模型主要由 3 个部分组成(图 2),分别为用于特征提取的主干网络、实例上下文编码器以及基于实例激活图的分割解码器。主干网络用于从输入图像中提取多尺度特征。实例上下文编码器将主干网络中生成的多尺度特征融合,以生成上下文信息更加丰富的单尺度特征图。最后将生成的特征图送入基于实例激活图的分割解码器中,生成叶片实例的激活图,以突出叶片实例,用于后续网络的识别和分割。



C3、C4 和 C5 分别表示 ResNet 特征提取网络中第三层特征层、第四层特征层以及第五层特征层;loss L:损失函数;C:特征拼接。

图 2 ePaCC-SparseInst 实例分割网络

Fig.2 ePaCC-SparseInst instance segmentation network

1.4 实例上下文编码器和基于实例激活图的分割解码器

ePaCC-SparseInst 的实例上下文编码器不同于以往的实例分割算法所使用的特征金字塔网络(Feature pyramid network,FPN)[26],而是对其进行重构。实例上下文编码器在主干网络输出的特征层 C5 之后采用金字塔池化模块(Pyramid pooling module,PPM)[27]放大模型的感受野,旨在输出语义信息更为丰富的特征图。最后输出的单级特征图融合了 P3、P4 和 P5 3 层特征图,其分辨率为原始图像的八分之一。

基于实例激活图的分割解码器由实例分支和掩码分支构成。如图 2 所示,实例分支生成实例激活图和 3 个$N\times D$(N:实例激活图的个数;D:向量的长度)的向量用于后续实例识别和分割。实例激活图位于实例分支中,旨在全局范围内突出每个实例对象的信息区域。突出区域的实例特征语义信息丰富,在实例识别中,有很强的实例感知能力。而掩码分支则用于解码特征图并生成叶片对象的分割结果。

1.5 实例分支

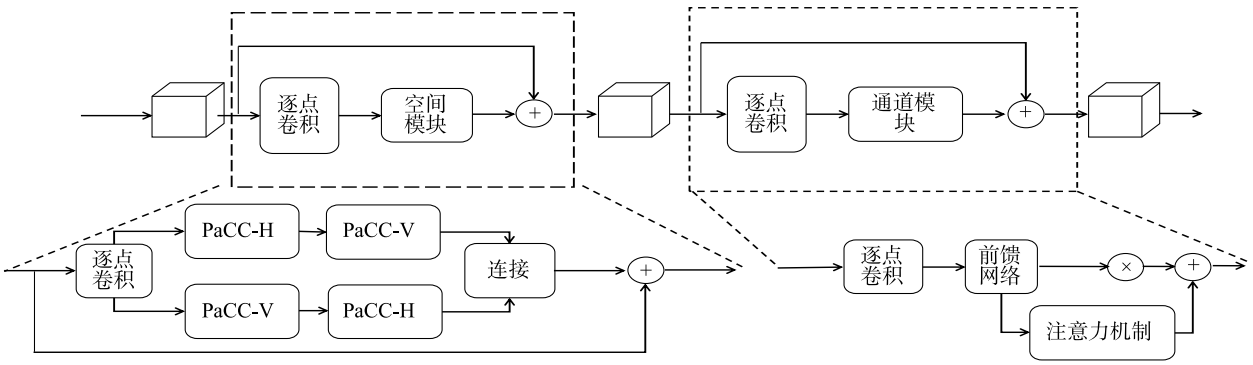
与现有的实例分割模型不同,ePaCC-SparseInst 既不依赖边界框[22]也不依赖稠密的锚框[28]定位实例对象,而是通过一组稀疏的实例激活图(Instance activate maps,IAM)来定位潜在的实例对象。实例激活图作为一种新的实例对象表示方式,可以动态地显示每个实例对象的区域信息。实例激活图是由 ePaCC 模块和 Sigmoid 激活函数在实例分支中激活后生成。给定 ePaCC 生成的特征图 $X\in R^{D\times(H\times W)}$($D$:向量的长度;$H$:向量的高度;$W$:向量的宽度),经过实例激活函数后输出的实例激活图为 $A=f_{\text{iam}}(X)\in R^{N\times(H\times W)}$。$A$ 为 N 个实例激活图的稀疏集,$f_{\text{iam}}(\cdot)$ 为非线性的 Sigmoid 激活函数。实例激活图根据区域位置信息聚合特征,以获得每个实例的语义信息,用于识别和分割实例对象。然后通过 1 个全连接层生成 3 个$N\times D$ 的向量,这 3 个向量分别是:掩码分支生成实例掩码的动态卷积核、实例特征的类别及其类别置信分。

1.6 ePaCC 模块

本研究在实例分支中引入一个带有全局感受野的 ePaCC 模块替代普通卷积层,试图捕获长距离的上下文依赖,获得全局特征信息以更好地区分具有相同语义类别的重叠叶片。ePaCC 改进自一种全新

的轻量级卷积模块位置感知卷积(PaCC)[29],其凭借全卷积的结构完美融合了 Vision Transformer 的优点。该模块具有全局感受野的同时,也具备普通卷积的位置敏感性。ePaCC 与普通的轻量级卷积神经网络和 ViT 相比,拥有更好的性能、更少的参数和更快的推理速度。其结构如图 3 所示。

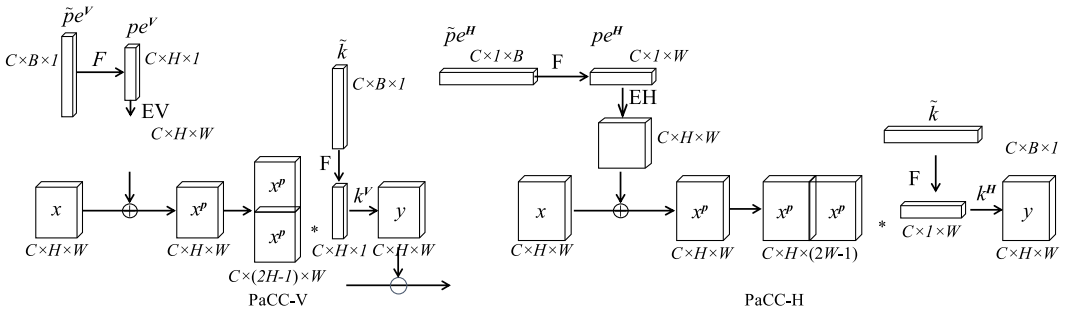
ePaCC 模块提取特征的主要结构为 PaCC 算子,该算子主要有 2 种类型,分别是垂直方向的 PaCC 算子(PaCC-V)以及水平方向的 PaCC 算子(PaCC-H)。如图 4 所示,PaCC-H 的感受野能覆盖一整行,而 PaCC-V 的感受野则能覆盖一整列。结合 PaCC-H 和 PaCC-V 则能实现提取全局特征的功能。以 PaCC-V 为例,PaCC 算子的实现见公式(1)~公式(5):



PaCC-H、PaCC-V 分别表示水平方向的位置感知卷积算子和垂直方向的位置感知卷积算子。

图 3 ePaCC 模块结构

Fig.3 Structure diagram of ePaCC module



B:实例位置编码的高度或者宽度;C:实例位置编码双线性插值后张量的通道数;H:实例位置编码双线性插值后张量的高度;W:实例位置编码双线性插值后张量的宽度;x:输入特征图;y:输出特征图;$\tilde{pe}^V$:垂直方向基本位置编码;$\tilde{pe}^H$:水平方向基本位置编码;F:双线性插值函数;pe^V:垂直方向实例位置编码;pe^H:水平方向实例位置编码;EV:垂直方向延展;EH:水平方向延展;x^p:包含位置信息的特征图;$\tilde{k}$:基本实例卷积核;k^V:垂直方向的实例卷积核;k^H:水平方向的实例卷积核。

图 4 垂直方向的位置感知卷积算子(PaCC-V)和水平方向的位置感知卷积算子(PaCC-H)结构

Fig.4 Structure of vertical position-aware circulate convolution (PaCC-V) and horizontal position-aware circulate convolution (PaCC-H)

$$pe^V = F(\tilde{pe}^V) = [pe_0^V, pe_1^V, \cdots, pe_{h-1}^V]^{\mathrm{T}} \quad (1)$$

$$pe_{\mathrm{e}}^V = EV(pe^V, w) \quad (2)$$

$$k^V = F(\tilde{k}) = [k_0^V, k_1^V, \cdots, k_{h-1}^V] \quad (3)$$

$$x^p = x + pe_{\mathrm{e}}^V \quad (4)$$

$$y_{i,j} = \sum_{t \in (0,h-1)} k_t^V x^p[(i+t) \bmod h, j] \quad (5)$$

其中 pe^V 是指垂直方向实例位置编码,通过 F(双线性插值函数)由一个垂直方向基本位置编码 $\tilde{pe}^V$ 生成。$\tilde{pe}^V$ 由垂直方向的实例位置编码延展得到;pe_{e}^V 是垂直方向位置编码的展开形式;k^V 是每一个 PaCC 的实例卷积核,由基本的实例卷积核 $\tilde{k}$ 通过双线性插值函数得到。x 是输入特征图;x^p 是包

含了位置信息的输入特征图;$y_{i,j}$表示 PaCC 模块在位置(i,j)处输出的特征。EV:垂直方向延展;w:实例位置编码双线性插值后张量的宽度;mod:模运算;h:高度。

Metaformer 结构是 Vision Transformer 取得成功的关键之一[30],它通常由 2 个重要的组件构成:令牌混合器和通道混合器。前者用于在不同空间位置的令牌之间交换信息,后者用于在不同通道之间交换信息。ePaCC 模块则使用一组串并联的 PaCC 算子替换 Metaformer 中的自注意力模块,构成一个新的空间模块。为了使纯卷积架构的 PaCC 模块也具有数据驱动性,引入 eSE 注意力模块作为 ePaCC 模块的通道混合器,使网络模型可以关注输入特征图中更加重要的特征信息,同时抑制不必要的干扰信息,从而输出语义信息更丰富的特征图。

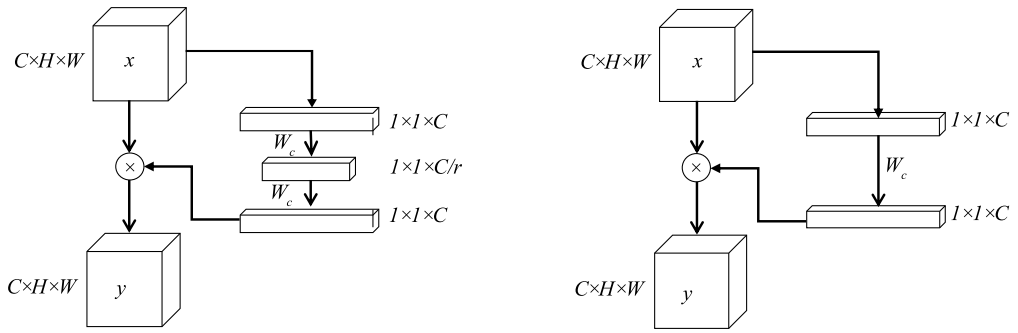
1.7 eSE 注意力模块

SE 注意力模块是卷积神经网络中最常用的注意力模块[31]。如图 5 所示,SE 注意力模块中 2 个压缩通道维数的全连接层会导致通道信息丢失的问题。因此本研究采用 eSE 通道注意力模块[32]替代 PaCC 模块中的 SE 通道注意力模块。该模块只使用一个通道数为 C 的全连接层,保证了通道信息不会因为通道压缩而丢失,进一步增强了实例激活图的质量,并提升了模型的推理速度。假设给定特征图 $X_i \in R^{C\times H\times W}$,经过 eSE 注意力模块后输出的特征图可以用公式表示为:

$$A_{\mathrm{eSE}}(X_i)=\sigma\{W_C[F_{\mathrm{gap}}(X_i)]\} \tag{6}$$

$$X_0=X_i \otimes A_{\mathrm{eSE}}(X_i) \tag{7}$$

其中 $F_{\mathrm{gap}}(\cdot)$为全局平均池化;W_C 是全连接层;σ 是一个 Sigmoid 激活函数。



C:张量的通道数;H:张量的高度;W:张量的宽度;r:压缩常量;x:输入特征图;y:输出特征图;W_c:全连接层。

图 5 SE 注意力模块和 eSE 注意力模块结构

Fig.5 Structural diagram of SE attention module and eSE attention module

1.8 掩码分支

本研究的掩码分支沿用了 SparseInst 的掩码分支,首先通过卷积层生成掩码特征图 $M \in R^{D\times H\times W}$,结合来自实例分支的动态卷积核$\{w_i\}^N$生成每一个实例对象的分割掩码 $m_i=w_i\times M$。最后通过双线性插值恢复到原始比例的特征图。

实例激活图会随输入特征的改变而改变,因此很难手工制定规则来分配标签进行模型训练。针对该问题,本研究将标签分配问题转化为二部图匹配问题,并使用匈牙利算法[33]实现预测对象与实例激活图的一一映射。二部图匹配有助于实例激活图突出显示单个实例对象并抑制冗余预测,因此避免了计算繁琐的非极大值抑制(NMS),节约了大量模型推理时间。

2 结果与分析

2.1 模型训练

试验环境为 Nvidia GTX 1080Ti GPU、显存 11 G 和 Linux Ubuntu 18.04 操作系统。本研究中的所有网络模型都基于开源框架 detectron2 和 Pytorch 进行部署,同时配备了 Python3.7、Cuda10.2 并行计算架构以及 Cudnn7.6 GPU 加速库。

参数设置:每次迭代训练的图片为 4 张,ePaCC-SparseInst、SparseInst、SOLOV2、CondInst 均迭代 25 000次,而 Mask R-CNN 仅迭代10 000次,这是因为 ePaCC-SparseInst 的二部图匹配算法需要更多训练轮次来收敛网络模型,而 Mask R-CNN 则会因为迭代轮次过多而产生过拟合的现象,导致分割精度

剧烈降低。ePaCC-SparseInst 的初始学习率为0.000 050 0,因为二部图匹配算法对学习率十分敏感,初始学习率太大会导致模型在训练开始时就梯度爆炸。当网络迭代到16 000次和20 000次时,学习率会分别衰减为0.000 005 0和0.000 000 5;SOLOv2 和 CondInst 的初始学习率为0.000 200 0,学习率衰减规则同 ePaCC-SparseInst 一致;Mask R-CNN 的初始学习率为0.005 000 0,当网络迭代到8 000次和9 000 次时,学习率会衰减为 0.000 500 0 和 0.000 050 0。此外 SparseInst 和 ePaCC-SparseInst 中的超参数 N 设为 100,超参数 D 设为 256。为了保证试验结果的可信性,本研究中所有展示的评价指标都是 3 次训练结果的平均值。

2.2 评价指标

本研究使用实例分割算法中最常用的评价指标平均分割精度(AP)来衡量网络模型的分割精度,包括 AP_{50}、AP_{75}、AP_s、AP_m以及 AP_l。其中 AP_{50}和 AP_{75}表示真实标签和预测结果的交并比为 50%以及 75%时的平均分割精度;AP_s表示像素小于 32^2实例对象的平均分割精度,在本研究中用来衡量网络对小叶片的检测效果;AP_l表示像素大于 96^2实例对象的平均分割精度,在本研究中用来评估网络对大叶片的检测效果;而 AP_m表示像素则介于 32^2和 96^2之间实例对象的平均分割精度。AP 的计算公式如下所示:

$$P=\frac{TP}{TP+FP}\times 100\% \tag{8}$$

$$R=\frac{FP}{TP+FN}\times 100\% \tag{9}$$

$$AP=\int_0^1 P(R) \tag{10}$$

式中,TP 表示正样本被预测正确的像素总和;FP 表示正样本被预测错误的像素总和;FN 表示负样本被预测错误的像素总和;P 表示准确率;R 表示召回率;AP 表示平均分割精度。

为了衡量模型的推理速度,本研究使用模型在1080Ti 服务器上每秒传输帧数(Frames per second,FPS)作为模型推理速度的评价指标。

2.3 模型主干网络的选择与分析

主干网络(Backbone)是模型提取特征的关键部件之一,能提取输入图片的信息,生成不同尺度的特征图,供后续网络的检测与分割。选择正确的主干网络以及合适的网络深度可以让模型的性能得到进一步提升。本研究选用 ResNet 系列中 4 种不同深度的模型作为 ePaCC-SparseInst 的主干网络并进行了一系列的试验分析,试验结果如表 1 所示。结果表明,ResNet-50 作为 ePaCC-SparseInst 的主干网络时,分割效果明显优于其他几种模型,同时模型 FPS 达到 43.52,满足实时性的需求。ResNet-18 和 ResNet-34 虽有较快的推理速度,但是分割精度都低于 ResNet-50。而 ResNet-101 则因为网络模型的进一步加深,出现了一定程度的性能衰退。因此,综合考虑模型的精度和推理速度,本研究选用 ResNet-50 作为 ePaCC-SparseInst 的主干网络。

表 1 不同主干网络对 ePaCC-SparseInst 的性能影响

Table 1 Effects of different backbone networks on the performance of ePaCC-SparseInst

主干网络	AP(%)	AP_{50}(%)	AP_{75}(%)	AP_s(%)	AP_m(%)	AP_l(%)	FPS
ResNet-18	79.70	96.72	89.28	51.21	84.57	98.59	83.57
ResNet-34	82.57	97.91	89.71	54.87	87.82	99.44	49.33
ResNet-50	85.33	98.91	92.88	61.54	90.49	99.74	43.52
ResNet-101	81.05	97.39	89.96	55.10	86.77	99.58	27.40

AP:平均分割精度;AP_{50}:真实标签与预测结果交并比为 50%时的平均分割精度;AP_{75}:真实标签与预测结果交并比为 75%时的平均分割精度;AP_s:像素小于 32^2实例对象的平均分割精度;AP_l:像素大于 96^2实例对象的平均分割精度;AP_m:像素则介于 32^2和 96^2之间实例对象的平均分割精度;FPS:每秒传输帧数。

2.4 模型消融试验

消融试验可以探究网络中的各个子结构对模型性能产生的影响。表 2 展示了实例上下文编码器中各个组件对网络模型性能的影响。为了增大感受野和获取更多的实例上下文信息,增加金字塔池化模块使模型的 AP 较基准模型 SparseInst 提高了 0.75 个百分点。融合特征金字塔 P3 到 P5 的多尺度特征,增强了输出单尺度特征的表达能力并将 ePaCC-SparseInst 的平均分割精度进一步提高至 85.33%。此外为了验证 ePaCC-SparseInst 中新增模块的效果,本研究设置了消融试验,将 ePaCC-SparseInst 中改进或者添加的模块逐个删除后进行训练。试验结果如表 3 所示。

表 2 编码器的消融试验结果

Table 2 Ablation experiment results of encoder

PPM	特征融合	AP(%)	AP_{50}(%)	AP_{75}(%)	AP_s(%)	AP_m(%)	AP_l(%)	FPS
×	×	84.06	97.93	91.71	59.56	89.07	99.74	45.23
×	√	84.74	97.89	92.19	58.80	90.04	99.63	44.82
√	×	84.81	97.97	92.71	60.26	89.82	99.78	43.64
√	√	85.33	98.91	92.88	61.54	90.49	99.74	43.52

AP、AP_{50}、AP_{75}、AP_s、AP_l、AP_m、FPS 见表 1 注。PPM:金字塔池化模块。

从表 3 可以看出,使用 PaCC 模块替换普通卷积层解码特征之后,模型的 AP 提高了 0.29 个百分点,FPS 提高了 1.19,AP_s 从 60.33% 提升至 61.63%,这是因为引入的 PaCC 模块有着普通卷积层所不具备的全局感受野,能在全局的范围内,找到普通卷积网络不能感知的小叶片。将 PaCC 中的 SE 注意力模块替换成 eSE 注意力模块之后,模型的性能又获得了进一步的提升,虽然平均分割精度仅提高了 0.17 个百分点,但模型的推理速度提升了 1.33。这是因为 eSE 注意力模块中的单层全连接层缓解了 PaCC 模块通道丢失问题,同时进一步提高了模型的推理速度。通过上述消融试验可以发现,本研究对 SparseInst 的改进都取得了积极的作用,特别是引入了 eSE 注意力模块的 ePaCC 模块对模型的精度和速度都有较大提升。

表 3 解码器消融试验结果

Table 3 Ablation experiment results of decoder

方法	AP(%)	AP_{50}(%)	AP_{75}(%)	AP_s(%)	AP_m(%)	AP_l(%)	FPS
SparseInst	84.87	98.44	91.78	60.33	90.08	99.73	41.00
SpaseInst+PaCC	85.16	98.86	92.78	61.63	89.91	99.88	42.19
SpaseInst+ePaCC	85.33	98.91	92.88	61.54	90.49	99.74	43.52

SparseInst:SparseInst 模型;SpaseInst+PaCC:使用 PaCC 模块替换普通卷积层后的模型;SpaseInst+ePaCC:使用 ePaCC 模块替换普通卷积层后的模型。AP、AP_{50}、AP_{75}、AP_s、AP_l、AP_m、FPS 见表 1 注。

2.5 ePaCC 的层数对模型性能的影响及分析

为了设计出更好的实例分支解码器,本研究进行了如表 4 所示的对比试验。该试验评估了不同层数的 ePaCC 模块对模型性能的影响。

由表 4 可以看出,当 ePaCC 的层数从 1 层增加到 2 层时,模型的分割精度得到略微的提升,但是模型的 FPS 降低了 2.97 。而 ePaCC 的层数增加到 3 层时,模型的分割精度反而出现一定程度的降低,其中最为明显的是 AP_s,较 1 层 ePaCC 降低 0.82 个百分点,这表明模型对小叶片的分割精度出现了大幅降低。因此为了平衡分割精度和模型的推理速度,本研究选用层数为 1 的 ePaCC 作为实例分支中的解码器。同时也证明了 ePaCC 有着较好的拟合能力,仅需 1 层就能很好地拟合数据。

表 4 ePaCC 的层数对模型性能的影响

Table 4 Effects of the number of layers of ePaCC on model performance

层数	AP(%)	AP_{50}(%)	AP_{75}(%)	AP_s(%)	AP_m(%)	AP_l(%)	FPS
1	85.33	98.91	92.88	61.54	90.49	99.74	43.52
2	85.43	98.94	92.89	61.57	90.52	99.72	40.55
3	85.00	98.95	92.74	60.72	90.00	99.73	37.39

AP、AP_{50}、AP_{75}、AP_s、AP_l、AP_m、FPS 见表 1 注。

2.6 ePaCC-SparseInst 与不同模型的对比试验

使用 Komatsuna 数据集对比 Mask R-CNN 以及目前较为先进的实时实例分割方法。最后在相同的测试集下进行推理测试,最终的试验结果如表 5 所示。

由表 5 可知,ePaCC-SparseInst 实例分割算法在

Komatsuna 数据集上的 *FPS* 达到 43.52,也就是单帧图片的分割时间仅需 22.97 ms,同时平均分割精度达到了 85.33%。而经典的实例分割算法 Mask R-CNN 单帧图片的分割时间需要 40.93 ms,平均分割精度仅为 75.15%。如果直接将 Mask R-CNN 应用于农业场景中,将无法满足实时分割植物叶片的需求。对比当前较为流行的实时实例分割算法 CondInst 和 SOLOv2,ePaCC-SparseInst 在推理模型推理速度以及叶片分割精度上都有一定程度的优势。因此可以得出结论,ePaCC-SparseInst 在推理速度和分割精度上都有着不错的效果,为模型在移动端或算力较低的边缘设备上部署提供了一种可行的方案。

表 5 5 种实例分割算法在 Komatsuna 测试集和 CVPPP A5 上的测试结果

Table 5 Test results of five instance segmentation algorithms on Komatsuna test dataset and CVPPP A5 dataset

方法	数据集	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_s (%)	AP_m (%)	AP_l (%)	*FPS*
Mask R-CNN	Komatsuna	75.15	95.24	86.79	47.13	81.76	94.09	24.43
CondInst		80.12	97.92	90.98	58.23	87.70	99.53	29.59
SOLOv2		82.64	97.87	91.04	56.33	88.65	99.54	30.52
SparseInst		84.87	98.44	91.78	60.33	90.08	99.73	41.00
ePaCC-SparseInst		85.33	98.91	92.88	61.54	90.49	99.74	43.52
Mask R-CNN	CVPPP A5	48.00	84.68	49.86	40.05	56.71	28.28	22.22
CondInst		59.48	90.42	65.63	46.04	78.91	64.00	22.95
SOLOv2		52.20	72.65	60.50	32.79	71.61	51.75	26.68
SparseInst		60.72	88.72	65.99	40.68	75.82	77.90	37.13
ePaCC-SparseInst		63.12	88.44	68.22	42.86	78.32	80.64	41.63

AP、AP_{50}、AP_{75}、AP_s、AP_l、AP_m、*FPS* 见表 1 注。

此外,本研究还对上述几种实例分割方法进行了可视化分析,如图 6 所示。其中 Mask R-CNN 的分割结果是最差的,这主要由 Mask R-CNN 的 2 个缺陷导致的:首先,Mask R-CNN 是两阶段的实例分割算法,这导致掩码的分割结果极度依赖检测结果的准确性。而检测框出现偏差时,就会对分割结果产生影响。从图 6 也可以看出这一现象,检测框的不准确导致部分叶片的叶尖部分没有被分割出来。为了提高检测的精度,有些方法在 Faster R-CNN 的基础上进行了一定的改进,如 Cascade R-CNN[34] 等。但这些方法大多会增加额外的开销,导致模型的复杂度以及推理时间大大增加。其次,Mask R-CNN 在 ROI Align 之后,特征图的分辨率仅为 28×28,导致部分语义信息丢失。而以 CondInst 为代表的实例分割算法都抛弃了 ROI Align,在原始特征层上动态地生成分割掩码,尽可能地保留更多的语义信息。但 CondInst 实例分割算法从本质上来说还是两阶段实例分割算法,只是将检测算法从 Faster R-CNN 替换成了检测精度更高的 FCOS[28],因此也会因为检测结果不佳影响最终掩码的分割精度。SOLOv2 将分割问题很好地转化为位置分类问题,做到了不需要目标检测来引导实例分割。但是其在小面积叶片的分割与检测上,很容易出现漏检的情况。由表 5 中 SOLOv2 AP_s 指标以及图 6 也可以看出这个缺陷。而 ePaCC-SparseInst 在模型分割精度上较 SparseInst 有着略微的提高。从 SparseInst 的第二张可视化图中可以看出,叶子细小的叶柄没有被完整的分割出来,而改进的 SparseInst 则将细小的叶柄分割出来了。同时也可以看出 ePaCC-SparseInst 分割出的叶片较 SparseInst 一般有着更高的置信分。这说明 ePaCC 模块使 ePaCC-SparseInst 的实例分支输出的实例激活图更具全局性,使得模型的分割精度得到了进一步的提升。

另外,在 CVPPP A5 数据集上,ePaCC-SparseInst 的 *FPS* 也达到了 41.63 ,比 SparseInst 快 4.5。引入带有全局感受野的 ePaCC 模块后,全局感受野对叶面积较大的叶片感知能力得到了提升,ePaCC-SparseInst 的 AP_l 比 SparseInst 平均分割精度高 2.74 个百分点。补充的 CVPPP A5 数据集也说明了本研究提出的 ePaCC-SparseInst 在实时分割叶片上效果较其他算法有较大的优势。

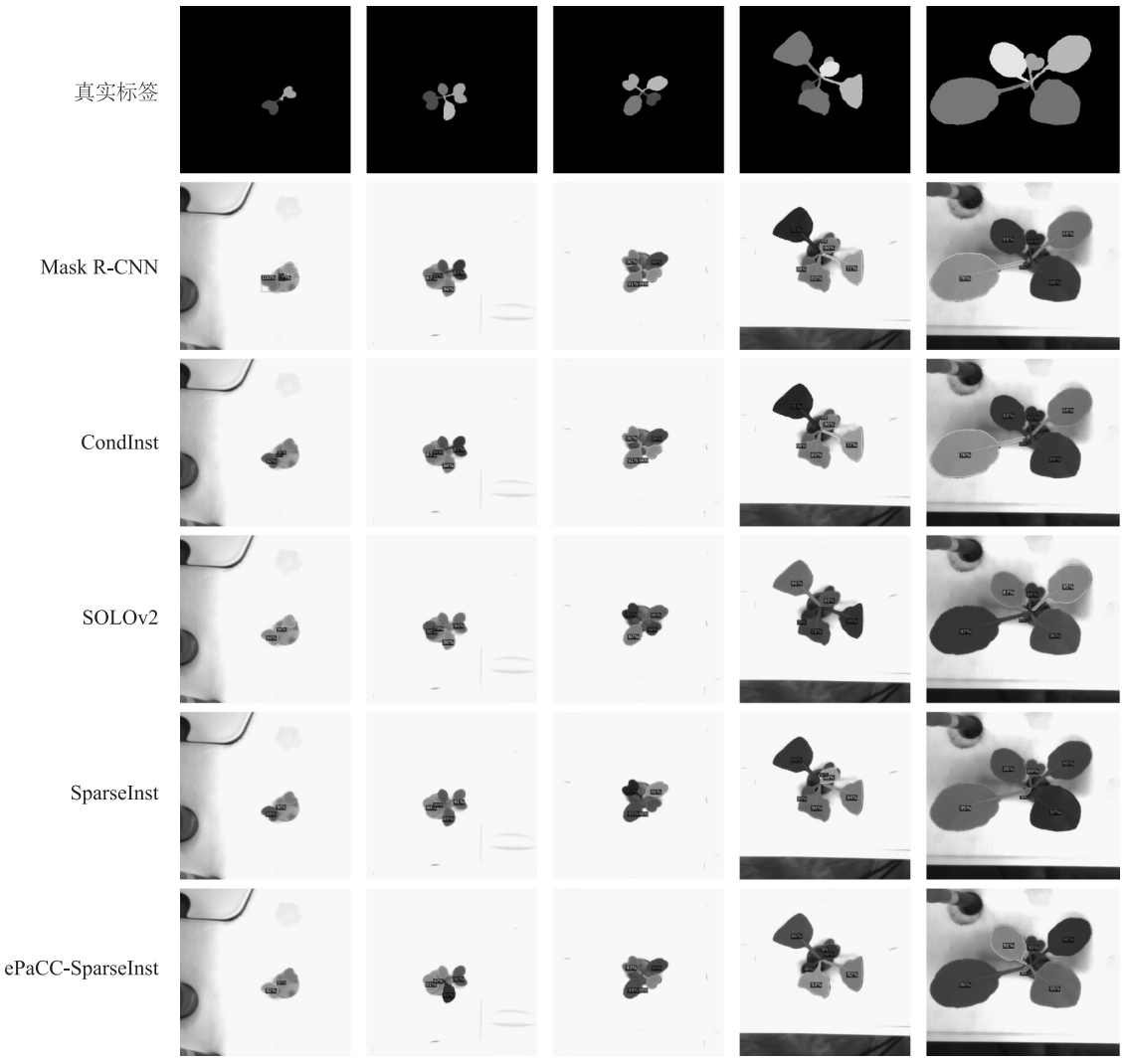


图 6　5 种实例分割模型的分割结果与真实标签的对比

Fig.6　Comparison between segmentation results of five instance segmentation algorithms and ground truth

2.7　模型复杂度分析

为了对比本研究的模型与现有的实例分割模型的复杂度以及模型参数量的大小,选用 ResNet-50 为主干网络,输入模型的图像尺寸为 480×480 像素的小松菜植物图像,对比各个模型的每秒浮点计算数(Floating point operations per second,*FLOPs*)和模型参数量。

由表 6 可知,与 Mask R-CNN、SOLOv2 等实例分割算法相比,ePaCC-SparseInst 的模型每秒浮点计算数以及参数量都得到了大幅的降低;同时, ePaCC-SparseInst 较 SparseInst 增加的模型参数量可以忽略不计,但模型每秒浮点计算数减少了 16.06 G。因此,全卷积结构的 ePaCC-SparseInst 大幅削减了模型每秒浮点计算数和参数量,同时提高了模型的分割精度,为在算力较低的设备上实现实时实例分割提供了一种可行的参考方案。

表 6　5 种不同模型的计算复杂度和参数量对比

Table 6　Comparison of computational complexity and the number of parameters of five different models

方法	每秒浮点计算数(G)	模型参数量(M)	平均分割精度(%)
Mask R-CNN	101.89	44.3	75.15
CondInst	93.11	34.1	80.12
SOLOv2	102.85	36.7	82.64
SparseInst	73.64	31.6	84.87
ePaCC-SparseInst	57.58	31.7	85.33

3 讨 论

本研究以 Komatsuna 数据集中的小松菜各个生长阶段的图片为主要研究对象,以 SparseInst 实例分割网络为基础,兼顾模型精度和推理速度,对 SparseInst 实例分割网络进行了改进,构建了 ePaCC-SparseInst 实时实例分割网络,并进行了一定的试验分析和评估。最终得到了如下结论:

1) 引入了稀疏的实例激活图作为对象的表示方式,并使用匈牙利算法实现标签和预测结果一一映射,大大地提高了模型的推理速度。同时在解码器中使用带有全局感受野的 ePaCC 模块。该模块把 PaCC 模块中的 SE 注意力模块替换成能更好得保留通道信息的 eSE 注意力模块,进一步降低了模型在单张图片上的推理时间。

2) 基于 Komatsuna 数据集中原先和数据增强后的共3 600张图片,对 ePaCC-SparseInst 实例分割网络进行试验。结果表明:ePaCC-SparseInst 实例分割网络的平均精度为 85.33%,推理速度每秒传输帧数为 43.52。

3)通过消融试验证明了 ePaCC-SparseInst 各个改进模块的有效性。并将其与 Mask R-CNN、CondInst 和 SOLOv2 等实例分割算法进行对比试验,结果表明改进后的 SparseInst 实例分割算法无论在分割精度上还是模型推理速度上,都优于其他实例分割算法。

本研究为基于图像实例分割的植物表型研究提供了一种兼顾精度和速度的改进思路,对植物叶片的表型研究和实际应用领域都具有重要的应用价值和前景。

参考文献:

[1] ZHOU Y, SRINIVASAN S, MIRNEZAMI S V, et al. Semiautomated feature extraction from RGB images for sorghum panicle architecture GWAS[J]. Plant Physiology,2019,179(1):24-37.

[2] PIERUSCHKA R, SCHURR U. Plant phenotyping:past,present, and future[J]. Plant Phenomics,2019,2019(3):1-6.

[3] SCHARR H, MINERVINI M, FRENCH A P, et al. Leaf segmentation in plant phenotyping:a collation study[J]. Machine Vision and Applications,2016,27(4):585-606.

[4] GUO R, QU L, NIU D, et al. LeafMask:towards greater accuracy on leaf segmentation[C]. Montreal,BC,Canada:IEEE,2021.

[5] GRAND-BROCHIER M, VACAVANT A, CERUTTI G, et al. Tree leaves extraction in natural images:comparative study of preprocessing tools and segmentation methods[J]. IEEE Transactions on Image Processing,2015,24(5):1549-1560.

[6] SCHARR H, PRIDMORE T, TSAFTARIS S A. Computer vision problems in plant phenotyping[C]. Venice,Italy:IEEE,2017.

[7] UCHIYAMA H, SAKURAI S, MISHIMA M, et al. An easy-to-setup 3D phenotyping platform for KOMATSUNA dataset[C]. Venice,Italy:IEEE,2017.

[8] 蒋焕煜,施经挥,任 烨,等. 机器视觉在幼苗自动移钵作业中的应用[J]. 农业工程学报,2009,25(5):127-131.

[9] 孙国祥,汪小旦,何国敏. 基于边缘链码信息的番茄苗重叠叶面分割算法[J]. 农业工程学报,2010,26(12):206-211.

[10] 王纪章,顾容榕,孙 力,等. 基于 Kinect 相机的穴盘苗生长过程无损监测方法[J]. 农业工程学报,2021,52(2):227-235.

[11] 伍艳莲,赵 力,姜海燕. 基于改进均值漂移算法的绿色作物图像分割方法[J]. 农业工程学报,2014,30(24):161-167.

[12] 胡 静,陈志泊,张荣国, 等. 基于鲁棒随机游走的交互式植物叶片分割[J]. 模式识别与人工智能,2018,31(10):933-940.

[13] KAN J, GU Z, MA C, et al. Leaf segmentation algorithm based on improved U-shaped network under complex background[C]. Chongqing,China:IEEE,2021.

[14] YIN X, LIU X, CHEN J, et al. Multi-leaf alignment from fluorescence plant images[C]. Steamboat Springs, CO, USA: IEEE, 2014.

[15] REN M, ZEMEL R S. End-to-end instance segmentation with recurrent attention[C]. Honolulu,HI,USA:IEEE,2017.

[16] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]. Venice,Italy:IEEE,2017.

[17] 乔 虹,冯 全,赵 兵,等. 基于 Mask R-CNN 的葡萄叶片实例分割[J]. 林业机械与木工设备,2019,47(10):15-22.

[18] 袁 山,汤 浩,郭 亚. 基于改进 Mask R-CNN 模型的植物叶片分割方法[J]. 农业工程学报,2022,38(1):212-220.

[19] 邢洁洁,谢定进,杨然兵,等.基于 YOLOv5s 的农田垃圾轻量化检测方法[J]. 农业工程学报,2022,38(19):153-161.

[20] CHENG T, WANG X, CHEN S, et al. Sparse instance activation for real-time instance segmentation[C]. New Orleans, LA, USA: IEEE,2022.

[21] HU H, GU J, ZHANG Z, et al. Relation networks for object detection[C]. Salt Lake City,UT,USA:IEEE,2018.

[22] REN S, HE K, GIRSHICK R, et al. Faster R-CNN:towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2017,39(6):1137-1149.

[23] GUO R, NIU D, QU L, et al. SOTR:segmenting objects with transformers[C]. Montreal,QC,Canada:IEEE,2021.

[24] WANG X, ZHANG R, KONG T, et al. SOLOv2: dynamic and fast instance segmentation[C]. Red Hook,NY,USA:Curran Associates Inc.,2020.

[25] BUSLAEV A, IGLOVIKOV V I, KHVEDCHENYA E, et al. Albumentations:fast and flexible image augmentations:2[J]. Infor-

mation,2020,11(2):125.

[26] LIN T-Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]. Honolulu,HI,USA:IEEE,2017.

[27] ZHAO H, SHI J, QI X, et al. Pyramid scene parsing network [C]. Honolulu,HI,USA:IEEE,2017.

[28] TIAN Z, SHEN C, CHEN H, et al. FCOS: fully convolutional one-stage object detection[C]. Seoul,South Korea:IEEE,2019.

[29] ZHANG H, HU W, WANG X. ParC-Net: position aware circular convolution with merits from ConvNets and transformer[C]. Tel Aviv,Israel:Springer Nature Switzerland,2022.

[30] YU W, LUO M, ZHOU P, et al. Metaformer is actually what you need for vision[C]. New Orleans,LA,USA:IEEE,2022.

[31] HU J, SHEN L, ALBANIE S, et al. Squeeze-and-excitation networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2020,42(8):2011-2023.

[32] LEE Y, PARK J. CenterMask:real-time anchor-free instance segmentation[C]. Seattle,WA,USA:IEEE,2020.

[33] STEWART R, ANDRILUKA M, NG A Y. End-to-end people detection in crowded scenes[C]. Las Vegas,NV,USA:IEEE,2016.

[34] CAI Z, VASCONCELOS N. Cascade R-CNN: delving into high quality object detection[C]. Salt Lake City, UT, USA: IEEE, 2018.

(责任编辑:陈海霞)