软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(6):2906−2921 [doi: 10.13328/j.cnki.jos.006493]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于网格切分的单阶段实例分割方法*

王文海, 李志琦, 路 通

(计算机软件新技术国家重点实验室 (南京大学), 江苏 南京 210023)

通信作者: 路通, E-mail: lutong@nju.edu.cn

摘 要: 近年来, 与二阶段实例分割方法相比, 单阶段实例分割方法由于实时性强, 已在实际应用中取得了初步进展, 但目前仍然存在以下两个主要缺点. (1) 精度较低: 单阶段方法缺少多轮优化环节, 因此其精度离实际应用仍存在差距; (2) 不够灵活: 目前大多数单阶段方法是独立设计的, 难以兼容实际应用中不同类型的物体检测框架, 因此适用范围相对有限. 提出了一种精确且灵活的单阶段实例分割框架——网格实例分割方法 (GridMask), 其中两个关键步骤如下: (1) 为了提高实例分割精度, 提出了一种网格切分二值化算法, 将物体边界框内的区域划分为多个独立的网格, 然后在每个网格上进行实例分割. 该步骤将物体分割任务简化成了多个网格切片的分割, 有效降低了特征表示的复杂程度, 进而提高了实例分割的精度; (2) 为了兼容不同的物体检测方法, 设计了一个可以即插即用的子网络模块. 该模块可以无缝地接入到目前大多数主流物体检测框架中, 以增强这些方法的分割性能. 所提方法在公共数据集 MS COCO 上取得了出色的性能, 优于现有的大部分单阶段方法, 甚至一些二阶段方法.

关键词: 实例分割; 物体检测; 卷积神经网络; 网格切分; 计算机视觉

中图法分类号: TP391

中文引用格式: 王文海, 李志琦, 路通. 基于网格切分的单阶段实例分割方法. 软件学报, 2023, 34(6): 2906–2921. http://www.jos.org.cn/1000-9825/6493.htm

英文引用格式: Wang WH, Li ZQ, Lu T. Grid Dividing for Single-stage Instance Segmentation. Ruan Jian Xue Bao/Journal of Software, 2023, 34(6): 2906–2921 (in Chinese). http://www.jos.org.cn/1000-9825/6493.htm

Grid Dividing for Single-stage Instance Segmentation

WANG Wen-Hai, LI Zhi-Qi, LU Tong

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Abstract: In recent years, single-stage instance segmentation methods have made preliminary progress in real-world applications due to their high efficiency, but there are still two drawbacks compared to two-stage counterparts. (1) Low accuracy: the single-stage method does not have multiple rounds of refinement, so its accuracy is some distance away from real-world applications; (2) Low flexibility: most existing single-stage methods are specifically designed models, which are not compatible with object detectors. This study presents an accurate and flexible framework for single-stage instance segmentation, which contains the following two key designs. (1) To improve the accuracy of instance segmentation, a grid dividing binarization algorithm is proposed, where the bounding box region is firstly divided into several grid cells and then instance segmentation is performed on each grid cell. In this way, the original full-object segmentation task is simplified into the sub-tasks of grid cells, which significantly reduces the complexity of feature representation and further improves the instance segmentation accuracy; (2) To be compatible with object detectors, a plug-and-play module is designed, which can be seamlessly plugged into most existing object detection methods, thus enabling them to perform instance segmentation. The proposed method achieves excellent performance on the public dataset, such as MS COCO. It outperforms most existing single-stage methods and even some two-stage methods.

Key words: instance segmentation; object detection; convolutional neural network (CNN); grid dividing; computer vision

---

* 基金项目: 国家自然科学基金 (61672273, 61832008)
收稿时间: 2021-06-04; 修改时间: 2021-07-23, 2021-09-09; 采用时间: 2021-09-22; jos 在线出版时间: 2022-10-14
CNKI 网络首发时间: 2022-11-15

实例分割 (instance segmentation) 是照片编辑、自动驾驶、航空图像处理等诸多计算机视觉应用[1,2]的一项基本任务. 在深度学习时代[3−6], 研究人员见证了实例分割领域的快速发展. 现有实例分割方法大致可以分为两类: 二阶段方法和单阶段方法. 其中, 二阶段方法[7−9]基本上是由物体检测方法[10]演进而来的, 这些方法通常使用特征提取器 (如: RoI Pooling[11]和 RoI Align[7]) 来提取物体边界框 (bounding box) 内的特征, 然后利用这些特征来分割出前景. 由于两个阶段相互分离, 因此这类方法实时性较差, 难以满足实际应用的需求. 和二阶段方法不同, 单阶段方法通常基于像素相似度学习 (pixel affinity learning)[12]或者新的物体掩模 (mask) 表示方法[13−16]. 这些方法没有特征提取步骤, 推理速度可以很快, 但其实例分割精度通常较低, 而且大多是为特定目的而设计的模型, 难以灵活地迁移到现实中不同的物体检测方法中 (如: 场景文字分割领域中的 PAN[17]和 PAN++[18]). 针对上述问题, 在前期研究[17,18]基础上, 本文拟进一步探索可达到二阶段方法精度, 同时兼具较高实时性、且能与大多数现有物体检测方法灵活衔接的新的单阶段实例分割算法框架.

不妨从最简单的二值图像分割开始思考. 如图 1(a) 为二值图像可以直接通过对比每个像素与前景/背景之间的距离来进行分割. 如果图中前景和背景的像素值已经确定, 只需要通过一个简单的对比函数 $R(\cdot)$ 就可以对其进行分割. 这个对比函数可以表示为以下公式:

$$R(x,f,b)=[D(x,f)<D(x,b)] \tag{1}$$

其中, x, f 和 b 分别表示像素值, 前景值和背景值. $D(\cdot)$ 表示广义上的距离函数, 可以为欧氏距离、余弦距离等. 这个规则相对简单, 如果当前像素值 x 与前景值 f 的距离比背景值 b 近, 则 $R(x,f,b)$ 为真, 表示当前像素为前景像素, 否则为背景像素. 但是实际场景中的图像 (如: RGB 图像) 往往比二值图像复杂得多, 公式 (1) 显然不能处理这些图像的实例分割问题. 最近, Bolya 等人[13]提出了 YOLACT 算法, 将图像中的每个像素值映射成高维的嵌入向量 (embedding vector), 然后根据每个像素与物体嵌入向量的距离来判断它是属于前景还是背景 (参考图 1(b) 为 YOLACT[13]通过将像素映射成高维嵌入向量来进行实际场景图像的实例分割), 从而达到实例分割的目的. 但是在存在同类物体遮挡的复杂场景中, 由于 YOLACT 算法中每个物体实例仅由一个总体的嵌入向量来表示, 其表示能力非常有限, 因此分割结果质量不高. 由图 1(c) 可见, 本文的方法通过网格切分二值化算法来简化实际场景图像的实例分割.

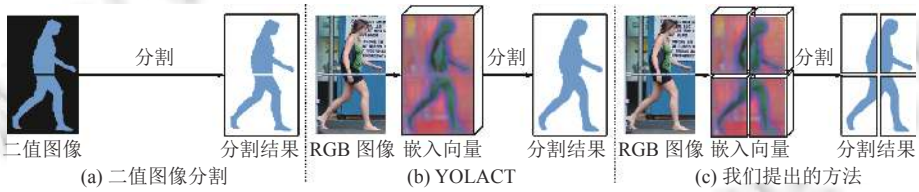


图 1 不同的实例分割方法之间的对比

为了解决这个问题, 本文从网格切分二值化这一新思路展开了研究, 将复杂的待分割区域均匀切分成 $k\times k$ 个小块, 然后在每个小块中利用嵌入向量距离来进行分割. 从图 2 的例子可见, 在没有进行切块前, 待分割区域有 6 种不同的语义类别 (如: 人、建筑、金属、路面和塑料), 但是, 如果只看其中的一个小块, 则语义类别会大大减少, 甚至直接减少成 2 类 (如: 人和建筑). 进行切分之后, 每个小块中的语义类别从 6 类显著减少到 2 类, 被简化了许多. 所以如果只在小块中做分割, 即使嵌入向量的表达能力不强, 也能取得很好的分割效果, 从而有效提升复杂场景下的实例分割精度.

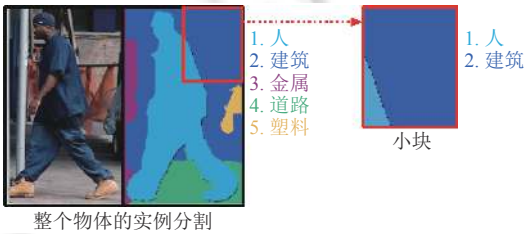


图 2 网格切分的动机

基于上述思路 (参考图 1 和图 2), 本文提出了一种新的单阶段实例分割方法——网格实例分割 (GridMask). 本文方法独立于检测分支模块, 其中包含了一个简单的全卷积网络模块和一个后处理模块 (即网格切分二值化算法), 通过将需要预测的实例掩模进行切分来单独预测物体的每个小块, 从而有效简化了问题的难度, 提高了模型精度. 与现有实例分割方法[7,13]相比, 本文提出的方法有以下 3 个优点: (1) 与二阶段方法相比, 本文的方法继承了单阶段方法推理速度快的优点, 对实际应用更加友好; (2) 与大多数单阶段方法相比, 本文方法的实例分割精度更高, 甚至可以和二阶段方法的精度相媲美; (3) 本文方法可看作一个即插即用的模块, 能非常方便地应用在目前绝大部分主流的物体检测方法上, 从而有效提高现有方法的实例分割性能.

为了验证本文方法的有效性, 本文在目前最常用的公共数据集 MS COCO [19]上进行了充分的实验, 并与现有代表性实例分割方法进行了详细比较, 实验结果表明, 在同样的硬件/软件环境下, 本文的单阶段方法可以取得与二阶段方法媲美甚至更好的实例分割精度. 例如, 以 ResNeXt-101[20]为骨干网络 (backbone network), 本文的方法在 MS COCO test-dev 数据集上的平均精度 (average precision, AP) 为 37.3, 比二阶段方法 Mask R-CNN[7]高, 并且推理速度是单阶段方法 TensorMask[14]的 5 倍.

总的来说, 本文主要贡献如下.

(1) 提出了一种精确且灵活的单阶段实例分割方法——网格实例分割 (GridMask), 该方法能与目前主流的物体检测方法兼容.

(2) 提出了一种新的后处理方法, 即网格切分二值化算法, 将物体的分割任务简化成了多个网格块的分割任务, 从而提高了实例分割的精度.

(3) 提出的单阶段方法在保证较快的推理速度的情况下, 取得了与二阶段方法媲美甚至更好的实例分割精度.

1 研究背景

实例分割是计算机视觉中的一项基本任务, 实例分割需要定位出不同的实例, 相比于物体检测, 实例分割需要预测物体的掩模 (Mask), 相比于语义分割, 实例分割需要区分同一类别下的不同实例, 因此实例分割任务一直以来都是计算机视觉领域中极具挑战性的任务. 由于实例分割能够预测出物体的掩模并且能够区分不同的实例, 所以实例分割结果通常比物体检测结果包含更丰富的信息, 并可应用于照片编辑、自动驾驶、航空图像处理等. 随着深度学习的发展[1,3−6,21], 实例分割领域也取得了长足的进步. 现有的实例分割方法大致分为两类: 二阶段方法和单阶段方法. 二阶段方法和单阶段方法的区分依据主要为整体处理流程中是否使用“先检测后分割”的范式.

1.1 二阶段实例分割方法

大多数二阶段实例分割方法[7−9,21,22]按照“先检测后分割”的范式进行实例分割. 具体来说, 这些方法首先检测物体的边界框, 然后在每个边界框内进行前景区域分割. 最基础的二阶段方法是 Mask R-CNN[7], 它的核心思想是在 Faster R-CNN[10]的物体检测分支的基础上添加一个额外的分割分支, 用于在物体的边界框内进行实例分割. PANet[8]和 MS R-CNN[9]是基于 Mask R-CNN 的拓展工作. 其中, PANet 通过一个自下而上的路径增强了 FPN[23], 并缩短了语义信息路径. MS R-CNN 根据实例分割结果的质量对其重新打分, 提高了实例分割的精度. 得益于“先检测后分割”的范式, 这些二阶段的方法通常具有较高的精度和灵活性. 但是, 上述的二阶段方法都采用了特征提取器 (如: RoI Pooling[11]和 RoI Align[7]) 来提取物体边界框内的特征, 然后将其用于后续的实例分割中, 这类特征提取器在实际应用中可能会成为速度优化的瓶颈.

1.2 单阶段实例分割方法

与两阶段方法不同, 单阶段方法主要基于像素相似度学习[12]或者新的物体掩模表示方法[13−16], 且不依赖于特征提取器. Deep watershed transform[12]借助全卷积神经网络预测整个图像的能量分布图 (energy map), 然后通过分水岭算法 (watershed transform) 进行实例分割. InstanceFCN[15]则先预测一组对实例敏感的得分图, 然后对滑动窗口内的分割结果进行组合, 得到实例分割结果. YOLACT[13]则是为实时的实例分割而设计的, 它通过每个像素的嵌入向量与物体实例的嵌入向量的内积来进行实例分割. ESE-Seg[16]也是一种实时的实例分割方法, 它利用一个中

心点和一组从中心点出发的射线来表示物体掩模, 减少了预测结果的计算量. 与 ESE-Seg 类似, PolarMask[24,25] 提出了一种基于极坐标的物体掩模表示方法, 将实例分割问题拆分为物体中心点分类和边缘极坐标回归两个子问题. PolarMask 将实例分割视为物体检测的扩展, 相比物体检测, 只增加较低的计算量就能够实现实例分割, 但是受限于模型的建模方式, PolarMask 分割结果的质量并不高. TensorMask[14]是一种基于滑动窗口的实例分割方法, 它通过 4 维张量来表示物体掩模, 并采用 Align2nat 操作和张量双金字塔 (tensor bipyramid) 来提高实例分割的精度. 然而, 上面提到的大多数单阶段方法在精度上都比两阶段方法低得多, 而且与主流的物体检测方法不能很好地兼容, 难以实现即插即用. 本文目标正是设计一个精确且灵活的单阶段实例分割方法.

2 方法介绍

在本节中, 本文将详细介绍所提出的网格实例分割方法 (GridMask). 本文首先对该方法的框架作总体的描述, 然后具体介绍其中的物体检测模块、嵌入向量生成模块、网格切分二值化算法和损失函数等细节.

2.1 总体框架

从图 3 中可以看到, 本文的网格实例分割方法包含 3 个关键模块: 检测模块、嵌入向量生成模块和后处理模块 (即网格切分二值化算法). 其中: (1) 物体检测模块可以由大部分主流的物体检测器 (如: RetinaNet[26]和 FCOS[27]) 实现, 用于预测物体的边界框. (2) 嵌入向量生成模块用于为图像中的每个像素和物体实例 (边界框) 生成高维的嵌入向量. (3) 后处理模块则由网格切分二值化算法实现. 对于每个物体实例, 网格切分二值化算法会根据其边界框和嵌入矢量生成分割结果.

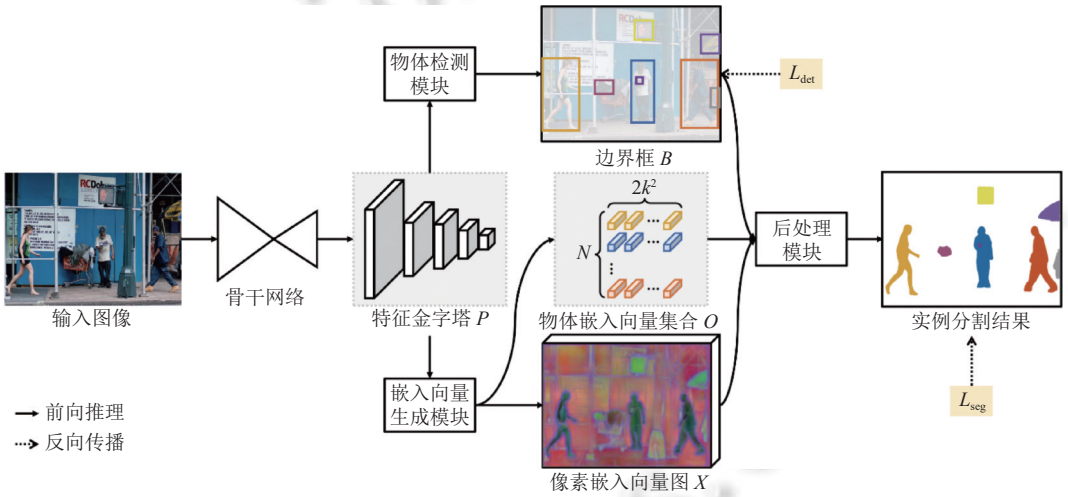


图 3 网格实例分割方法的总体结构

本文方法的前向推理的过程如下.

(1) 给定一张输入图像 $I \in \mathbb{R}^{H\times W\times 3}$ (H 为图高, W 为图宽), 本文方法首先将其输入到骨干网络 (如: 带 FPN[23]的 ResNet-50[4]) 中, 得到特征金字塔 $P=\{P_3,P_4,\ldots,P_7\}$, 这些特征图的宽高分别为原图的 1/8、1/16 和 1/128, 通道 (channel) 数均为 256.

(2) 然后, 在特征金字塔 P 的基础上, 物体检测模块预测物体的边界框 $B \in \mathbb{R}^{N\times 4}$, 这里 N 表示图像中物体的个数, 4 表示一个四元组描述边界框的位置.

(3) 与此同时, 嵌入向量生成模块也基于特征金字塔 P 生成: 像素嵌入向量图 $X \in \mathbb{R}^{H/8\times W/8\times C}$, 其中每个位置表示对应像素的 C 维嵌入向量; 物体嵌入向量集合 $O \in \mathbb{R}^{N\times 2k^2\times C}$, 集合中有 N 个物体, 每个物体均由 $2k^2$ 个 C 维嵌入向量表示. 这里之所以用 $2k^2$ 个嵌入向量表示每个物体, 是因为在本文的方法中每个物体均被切分成了 k^2 网格单

元, 每个网格单元的前景和背景均由一个嵌入向量表示.

(4) 最后, 基于物体边界框 B、像素嵌入向量图 X 和物体嵌入向量集合 O, 后处理模块利用网格切分二值化算法生成实例分割结果 $M \in \mathbb{B}^{N\times H\times W}$. 这里 $\mathbb{B}$ 为布尔量, 即 0 或 1.

关于过程中的物体检测模块、嵌入向量生成模块和后处理模块的具体细节, 本文会在后续的第 2.2、2.3 和 2.4 节中展开描述.

在训练过程中, 本文模型通过端到端训练获得, 其中物体检测模块由损失函数 L^{det} 进行优化, 嵌入向量生成模块和后处理模块由损失函数 L^{seg} 优化, 骨干网络由两个损失函数 L^{det} 和 L^{seg} 联合优化. 这些损失函数的具体细节将在后续的第 2.5 节具体介绍.

2.2 物体检测模块

物体检测模块用于预测物体的边界框, 它可以由现有的大多数物体检测器[10,26–30]实现. 本文选择一种常用的物体检测方法 FCOS[27]来实现物体检测模块. 按照 FCOS 的设置, 本文方法在特征金字塔 $\{P_3,P_4,\ldots,P_7\}$ 上进行像素级别的物体分类、中心度 (centerness) 预测和边界框回归, 然后通过非极大值抑制算法 (non-maximum suppression)[11]去掉重复的边界框, 保留 N 个边界框作为检测结果 $B \in \mathbb{R}^{N\times 4}$.

2.3 嵌入向量生成模块

嵌入向量生成模块用于为输入图像中的每个物体和像素生成嵌入向量. 因为骨干网络和物体检测模块已经占用了很多计算资源, 所以在设计嵌入式矢量生成模块时, 本文尽量在显存和计算方面使该模块保持轻量化. 如图 4 所示, 嵌入向量生成模块有两个分支, 其中一个分支用于生成物体嵌入向量集合 $O \in \mathbb{R}^{N\times 2k^2\times C}$, 另外一个分支用于生成像素嵌入向量图 $X \in \mathbb{R}^{H/8\times W/8\times C}$.

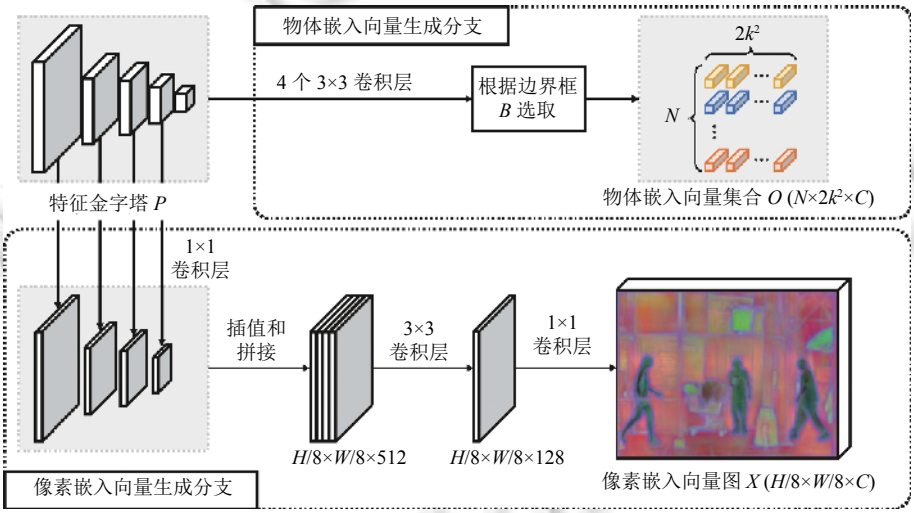


图 4 嵌入向量生成模块的细节图, 其中包含两个分支: 物体嵌入向量生成分支和像素嵌入向量生成分支

物体嵌入向量生成分支由 4 个 3×3 的卷积层构成. 和物体检测模块一样, 这个分支也直接作用在特征金字塔的 5 个特征图 (即, $\{P_3,P_4,\ldots,P_7\}$) 上, 得到 5 个通道数为 $2k^2C$ 的嵌入向量特征图 $\{Q_3,Q_4,\ldots,Q_7\}$. 这些嵌入向量特征图 Q 的分辨率与原特征图 P 的分辨率相同. 因此对于每个预测的边界框 $b \in B$, 假设该边界框是从特征图 P_i 的 (x,y) 位置处预测得到的, 均能从嵌入向量特征图 Q_i 中同样的位置处找到一组 $2k^2C$ 维的嵌入向量与其对应. 这里的 $2k^2C$ 维可以理解为 $2k^2$ 个 C 维的嵌入向量. 因为在本文方法中, 每个物体均被切分成了 k^2 网格单元, 每个网格单元的前景和背景各有一个嵌入向量, 所以每个物体都需要 $2k^2$ 个 C 维的嵌入向量来表示.

像素嵌入向量生成分支的输入为特征金字塔的前 4 个特征图 (即, $\{P_3,P_4,\ldots,P_6\}$). 首先, 这个分支通过 1×1 的卷积层将输入的特征图的通道数减少到 128. 然后, 通过双线性插值 (bilinear interpolation) 将这些特征图的大小

都插值到宽高为输入图像的 1/8, 紧接着在通道 (channel) 维度上将插值后的特征图拼接起来, 得到一个通道数为 512 的特征图. 最后, 通过一个 3×3 和 1×1 的卷积层生成像素嵌入向量图 $X \in \mathbb{R}^{H/8\times W/8\times C}$, 其中每个位置表示对应像素的 C 维嵌入向量.

2.4 网格切分二值化算法

网格切分二值化算法用于将物体检测模块和嵌入向量生成模块生成的物体边界框 B、像素嵌入向量图 X 和物体嵌入向量集合 O 拼装成最终的实例分割结果. 图 5 描述了网格切分二值化的详细过程, 具体分为以下 3 个步骤.

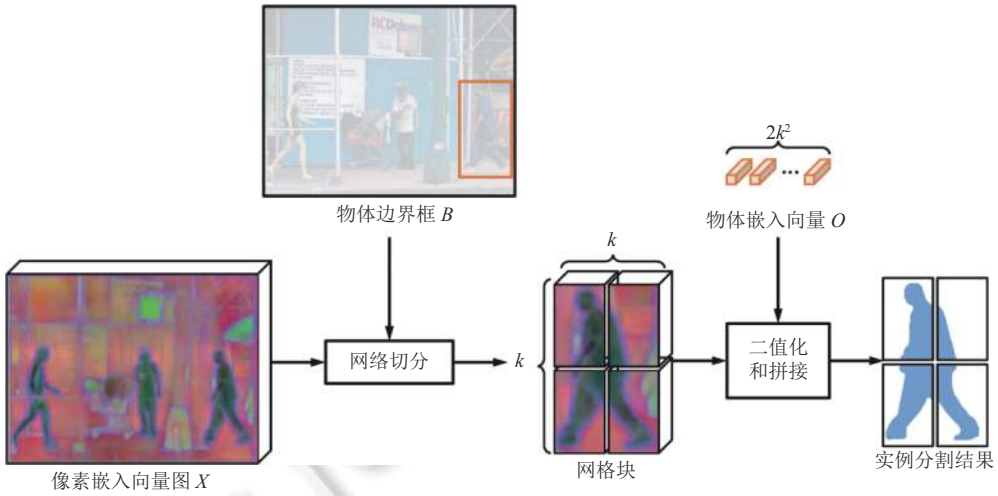


图 5 网格切分二值化算法的细节图

(1) 首先, 对于任意一个物体边界框 $b \in B$, 截取像素嵌入向量图 X 中物体边界框 b 内的区域, 并将该区域均匀切分为 k^2 个网格块. 这里, 不妨令网格块为 $G = \{g_i | i = 1,2,\ldots,k^2\}$.

(2) 然后, 对于每个网格块 g_i, 根据下述的公式 (2) 对其进行前景背景分割.

$$R\left(X_p, O_b^{2i}, O_b^{2i+1}\right) = \left[D\left(X_p, O_b^{2i}\right) < D\left(X_p, O_b^{2i+1}\right)\right] \tag{2}$$

其中, $D(\cdot)$ 表示广义的距离函数. X_p 表示网格块 g_i 中的某个像素 p 对应的嵌入向量. O_b 表示物体边界框 b 对应的一组物体嵌入向量, O_b^{2i} 表示物体边界框 b 的第 i 个网格的前景嵌入向量, 相对地 O_b^{2i+1} 表示物体边界框 b 的第 i 个网格的背景嵌入向量. 不难发现, 公式 (2) 其实是公式 (1) 的推广, 如果像素嵌入向量 X_p 与前景嵌入向量 O_b^{2i} 的距离比背景嵌入向量 O_b^{2i+1} 近, 则 $R\left(X_p, O_b^{2i}, O_b^{2i+1}\right)$ 为真, 表示当前像素为前景像素.

(3) 最后, 分割好的 k^2 个网格块拼接起来, 得到一个完整的实例分割结果 M_b.

另外, 在模型训练过程中, 可以直接使用目标函数直接优化网格分割二值化算法生成的结果. 本文的解决方案就是基于公式 (2) 设计一个对应的目标函数.

在本文中, 不妨采用余弦相似度 $COS(\cdot)$ 作为公式 (2) 中的距离函数 $D(\cdot)$, 因为两个向量越相似, 余弦相似度越大, 所以公式 (2) 需要改写成:

$$R\left(X_p, O_b^{2i}, O_b^{2i+1}\right) = \left[COS\left(X_p, O_b^{2i}\right) > COS\left(X_p, O_b^{2i+1}\right)\right] \tag{3}$$

为了让公式 (3) 成立, 对于前景像素, 本文希望 $COS\left(X_p, O_b^{2i}\right)$ 尽量趋近于 1, $COS\left(X_p, O_b^{2i+1}\right)$ 尽量趋近于−1. 相反地, 对于背景像素, 本文则希望 $COS\left(X_p, O_b^{2i}\right)$ 尽量趋近于−1, $COS\left(X_p, O_b^{2i+1}\right)$ 尽量趋近于 1.

因此, 可以把公式 (3) 拆成两项, 即前景项 $COS\left(X_p, O_b^{2i}\right)$ 和背景项 $COS\left(X_p, O_b^{2i+1}\right)$, 这两项可以通过不同的目标函数分别进行优化. 因为余弦相似度 $COS(\cdot)$ 的值域为 [−1, 1], 不适合用二值交叉熵 (binary cross entropy, BCE) 优化, 所以本文定义了一个映射函数 $Proj(\cdot)$, 把前景项和背景项的值域映射到 [0, 1], 其具体公式如下:

$$Proj(x)=\sigma(\alpha x) \tag{4}$$

其中, $\sigma(\cdot)$为 Sigmoid 函数, α 为缩放因子. 在本文的实验中缩放因子α默认设置为 5. 有了映射函数$Proj(\cdot)$, 前景项和背景项的目标函数可以写成:

$$L_f\left(X_p,O_b^{2i},Y_p\right)=BCE\left(Proj\left(COS\left(X_p,O_b^{2i}\right)\right),Y_p\right) \tag{5}$$

$$L_b\left(X_p,O_b^{2i+1},Y_p\right)=BCE\left(Proj\left(COS\left(X_p,O_b^{2i+1}\right)\right),1-Y_p\right) \tag{6}$$

其中, $L_f(\cdot)$和$L_b(\cdot)$为前景项和背景项的目标函数. Y_p 为像素 p 的标注信息, 如果像素 p 是前景像素, 这标准信息为 1 否则为 0.

有了公式 (5) 和公式 (6) 所示的目标函数, 就可以端到端优化网格切分二值化算法的分割结果, 从而使得模型训练更加容易, 使得分割结果更加精确. 与基于度量学习的分割目标函数[30]相比, 本文方法, (1) 可以同时对前景和背景嵌入向量同时进行优化, 有利于精细的分割; (2) 不需要额外的阈值对分割结果进行二值化.

2.5 损失函数设计

本文方法的总体损失函数可以写成如下公式:

$$L=L^{\text{det}}+L^{\text{seg}} \tag{7}$$

其中, L^{det} 是检测损失函数, 用于优化物体检测模块. 在文本中, 本文采用与物体检测器 FCOS[27]相同的损失函数. L^{seg} 是分割损失函数, 用于优化嵌入向量生成模块, 其具体公式如下:

$$L^{\text{seg}}=\frac{1}{N}\sum_{j=1}^{N}L_j^{\text{seg}} \tag{8}$$

$$L_j^{\text{seg}}=\frac{1}{k^2}\sum\nolimits_{i=0}^{k^2}\sum\nolimits_{X_p\in g_{i,j}}\frac{L_f\left(X_p,O_j^{2i},Y_p\right)+L_b\left(X_p,O_j^{2i+1},Y_p\right)}{\left|g_{i.j}\right|} \tag{9}$$

其中, N 是预测的物体边界框的数目. L_j^{seg}第 j 个边界框的分割损失函数. G_j 是第 j 个物体边界框网格块集合, 其数量为 k^2. $g_{i,j}$ 表示G_j中的第 i 个网格块, $\left|g_{i.j}\right|$ 是其中像素点的数量. X_p 是网格块 $g_{i,j}$ 中某个像素 p 对应的嵌入向量, 损失函数 L_f 和 L_b 分别是公式 (5) 和公式 (6) 所定义的损失函数.

3 实 验

本文在目前最常用的公共数据集 MS COCO[19]上验证本文方法的有效性. 按照领域内常用做法[7,14], 本文在 8 万张训练集图片和 3.5 万张验证集图片 (即, trainval35k) 上训练模型, 然后在 5 千张验证集图片 (即 minival) 上验证结果. 另外, 本文方法的结果还提交至测试集 (即 test-dev) 上与其他主流方法的结果进行了对比.

3.1 实现细节

本文的方法采用带有 FPN[23]的 ResNet-50[4]作为骨干网络. 在训练的时候遵循常用训练策略[7,31], 本文使用随机梯度下降 (SGD) 作为优化器 (optimizer), 并以批大小 (batch size) 为 16 训练 9 万次迭代 (即, 1 训练策略). 初始学习率 (learning rate) 为 0.01, 并在第 6 万次和第 8 万次迭代减小为之前学习率的 1/10. 权重衰减 (weight decay) 和动量 (momentum) 分别设置为 0.000 1 和 0.9. 本文使用在 ImageNet[32]上预训练的权重初始化骨干网络. 在训练和测试阶段, 输入图像的尺寸通过放缩保证短边为 800 像素且长边不超过 1 333 像素.

3.2 消融实验

3.2.1 为什么网格切分策略是有效的?

本文计算了当物体边界框切分为不同数量的网格时, 每个网格中包含的平均实例数量 $\overline{N}$, 因为 $\overline{N}$ 能够粗略反映出网格中图像内容的复杂度. $\overline{N}$ 越大, 图像内容越复杂, 对其进行分割难度也随之增大. 如图 6 所示, 本文让网格数量的平方根 k 从 1 增加到 10, 然后在 MS COCO trainval35k 和 minival[19,33]上统计平均实例数量 $\overline{N}$. 注意, 当 $k=1$时, 物体边界框被切分成 1×1 的网格, 所以这个网格就等价于整个物体边界框. 从图 6 中可以看出, 在 $k>1$ 的

情况下, 网格中的平均实例数量比整个物体边界框 (即, $k=1$) 中的平均实例数量少很多, 这说明了相比在整个物体边界框中进行分割, 切分成多个网格之后的分割难度被大大降低. 与此同时, 本文发现 $\overline{N}$ 随 k 值的增大而不断减小, 并且当 $k>3$ 时, $\overline{N}<2$, 这意味着每个网格内的图像内容已经变得非常简单, 简单到只需要进行二分类分割即可, 没有必要再取更大的 k.

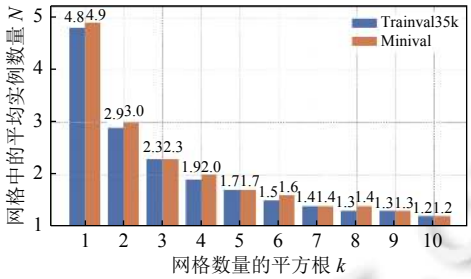
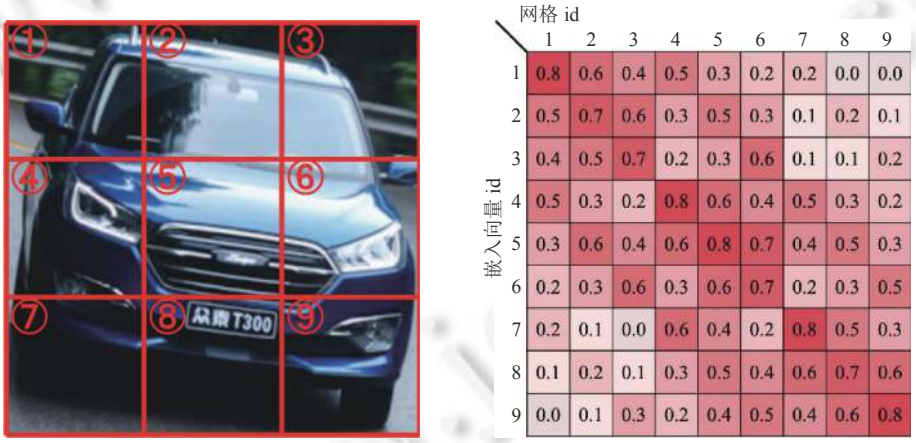


图 6 在不同网格数量的平方根 k 下每个格子包含的平均实例数量 $\overline{N}$

3.2.2 网格嵌入向量之间的关系

为了研究网格嵌入向量之间的关系, 本文对 3×3 网格的嵌入向量进行交叉测试. 如图 7(a) 所示, 交叉测试包含 9 轮, 在第 i 轮的测试中, 本文通过网格 i 的嵌入向量对所有 9 个网格进行分割, 并计算每个网格的分割结果与真实标注的 mIoU. 图 7(b) 展示了 MS COCO minival 中所有物体每个网格嵌入向量交叉测试的 mIoU 的平均值. 从图中可以看到, 每个网格的嵌入向量在自己负责的网格上的分割结果质量都很好, 但是在其他网格上的分割结果质量较差. 这个结果是符合预期的, 因为不同位置网格的嵌入向量负责的语义信息不同. 如图 7(a) 所示, 对于一辆轿车来说, 1、2、3 号网格通常负责车窗信息; 4、5、6 号网格通常负责车身信息; 7、8、9 号网格通常负责车轮信息. 这个结果说明了网格嵌入向量是位置敏感的. 一些前人工作 [22,34] 也验证了这一点. 结合第 3.2.1 节的结论, 网格切分策略有以下两个优点: (1) 可以降低待分割内容的复杂度, 使得实例分割结果更准确; (2) 可以学习到位置敏感语义表示, 有利于提高实例分割结果的精细程度.



(a) 网格编号 (b) 网格嵌入向量交叉测试的 mIoU

图 7 网格编号和网格嵌入向量交叉测试的 mIoU

3.2.3 网格数量 k^2 对实例分割精度的影响

为了研究网格数量 k^2 对实例分割精度的影响, 本文让 k 从 1 逐步增大到 5, 然后比较不同 k 值下的模型在 MS COCO minival 上的平均精度 (AP). 从表 1 中可以看到, 采用网格切分策略 (即, $k>1$) 的模型的平均精度比不采用网格切分策略 (即, $k=1$) 的模型高 1.1 个百分点, 这验证了网格切分策略的有效性. 另外可以注意到, 在 $k\leqslant3$ 时,

平均精度不断增长. 当$k>3$时, 平均精度开始震荡. 这个现象和本文在第 3.2.1 节的结论相对应. 因此, 在后续的实验中, 本文将k的默认值设置为 3. 在这个设置下, 模型的平均精度为 32.7, 比不采用网格切分策略的模型高 1.6 个百分点.

表 1 网格数量k^2对实例分割精度的影响 (%)

k	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1	31.1	51.7	32.4	13.0	34.3	46.7
2	32.2	52.6	33.7	13.2	35.3	48.7
3	**32.7**	**53.0**	**34.5**	13.9	**35.9**	48.4
4	32.4	52.4	34.2	13.9	35.4	49.0
5	32.6	52.7	34.2	**14.0**	35.4	**49.8**

另外, 为了直观地展示网格切分策略的作用, 本文比较了采用和不采用网格切分策略的模型的可视化结果, 如图 8 为采用 ($k=3$) 和不采用 ($k=1$) 网格切分策略的模型 (以 ResNet50-FPN 为骨干网络) 的可视化结果. 第 1 列是没有重叠实例的样例, 第 2 列和第 3 列展示了带有重叠实例的样例. 从图 8 第 1 列中可以看出, 当实例没有出现重叠时, 两者的结果是相似的. 然而, 当出现重叠情况时 (见图 8 第 2 列和第 3 列), 不采用网格切分策略的模型的结果有很多错误, 而采用网格切分策略的模型可以准确地进行实例分割, 这说明了网格切分策略能够处理带有重叠实例的复杂场景.

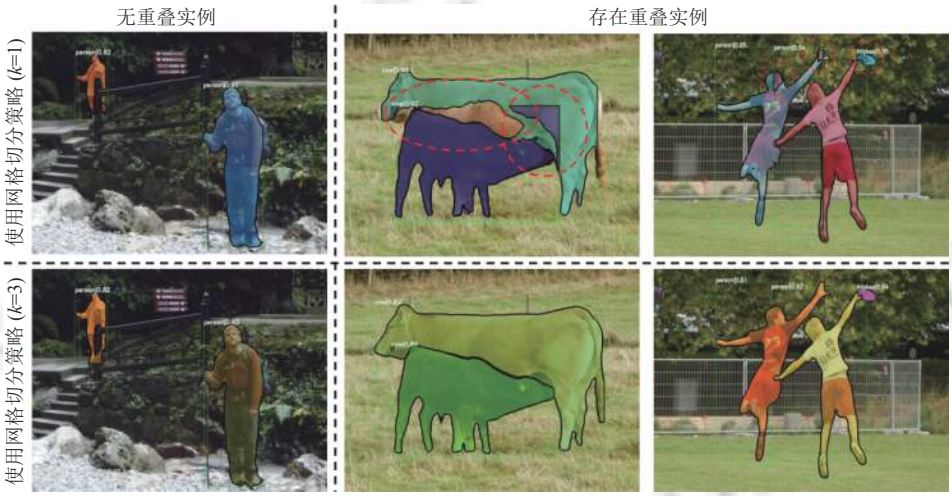


图 8 采用和不采用网格切分策略的模型的可视化结果比较

3.2.4 网格实例分割方法的灵活性

为了展示本文方法的灵活性, 本文将 GridMask 迁移到其他主流的算法上, 如: Faster R-CNN[10](带 FPN[23]和 RoIAlign[10])、CondInst[35]、和 RDSNet[30]. 其中, Faster R-CNN 是一个基于锚框 (anchor-based) 的物体检测方法, CondInst 是一个基于动态卷积的算法, RDSNet 采用了物体检测和实例分割相互辅助的框架. 不同方法的设置如下.

(1) 在 Faster R-CNN 中, 本文首先将物体嵌入向量生成分支接入到 RoIAlign 后与 Faster RCNN 的检测分支并行, 生成与检测结果一一对应的物体嵌入向量. 另外, 本文还将像素嵌入向量生成分支接入到 FPN 后, 提取像素嵌入向量图 X. 最后再辅以网格切分二值化算法进行实例分割.

(2) 在 CondInst 中, 本文将网格化的思想拓展到动态卷积上. 因为嵌入向量本质上可以看作是一层 1×1 的动态卷积, 所以为了适配 CondInst, 本文首先通过物体嵌入向量生成分支为每个物体的每个网格生成一组动态卷积权重w_i^j, 这里i和j表示第i个物体的第j个网格. 然后, 将像素嵌入向量生成分支接入到 FPN 后, 提取像素嵌入向量图 X. 接着, 通过每个网格的动态卷积权重w_i^j在像素嵌入向量图 X 上分割每个网格的前景内容. 最后将网格的

内容拼接成完整的实例分割结果.

(3) 在 RDSNet 中, 本文将 RDSNet 的 Object Stream 的 Representation 分支替换为本文的物体嵌入向量生成分支, 将 RDSNet 的 Pixel Stream 的 FCN 网络替换为本文的像素嵌入向量生成分支, 将 RDSNet 生成掩膜的后处理替换为本文的网格切分二值化算法. 除此之外, RDSNet 的其它设置 (包括物体检测-实例分割交互的过程) 保持不变.

如后文表 2 所示, GridMask 在 4 种截然不同的检测框架 (即 FCOS、Faster R-CNN、CondInst 和 RDSNet) 中都能取得良好的精度. 值得一提的是, 本文的 GridMask 与 CondInst 的动态卷积机制和 RDSNet 的物体检测-实例分割的交互机制是兼容的, 能在 CondInst 和 RDSNet 原来有精度的基础上进一步提高实例分割精度 (大于 1AP). 这个实验结果也表明了本文的方法可以非常灵活地接入到不同的物体检测和实例分割方法中, 从而提高这些方法实例分割的性能.

3.2.5 特征融合方法对像素嵌入向量生成分支的影响

为了研究特征融合方法对像素嵌入向量生成分支的影响, 本文通过两种不同的方法分别对输入像素嵌入向量生成分支的特征进行融合, 并对比最终的实例分割精度. 具体来说, 本文对比的两种特征融合方法分别为: (1) 拼接金字塔特征 $\{P_3, P_4, P_5, P_6\}$; (2) 与 PolarMask++[25]一样直接采用 P_3 作为融合特征. 如表 3 所示, 第 1 种特征融合方法在 AP 上比第 2 种方法稍高 (32.7 vs. 32.2), 但是在模型的整体速度上却几乎相同 (12.1 vs. 12.4). 因此, 第 1 种方法 (即, 拼接金字塔特征 $\{P_3, P_4, P_5, P_6\}$) 的性价比比第 2 种方法 (即, 直接采用 P_3) 要稍好一些, 但对最终结果的影响不大.

表 2 网格实例分割方法的灵活性 (%)

方法	AP	AP_{50}	AP_{75}
GridMask+FCOS[27]	32.7	53.0	34.5
GridMask+Faster R-CNN[10]	33.0	54.5	34.6
CondInst[35]	34.9	55.4	36.8
GridMask+CondInst[35]	36.0	56.5	38.0
RDSNet[30]	32.1	52.4	33.7
GridMask+RDSNet[30]	33.3	54.7	34.9

表 3 特征融合方法对像素嵌入向量生成分支的影响

方法	AP (%)	AP_{50} (%)	AP_{75} (%)	FPS
拼接金字塔特 $\{P_3, P_4, P_5, P_6\}$	**32.7**	**53.0**	**34.5**	12.1
直接采用 P_3	32.2	52.0	34.4	**12.4**

3.2.6 骨干网络的对实例分割精度的影响

为了更好地分析本文方法的有效性, 本文使用更重的骨干网络 (如: ResNet-101-FPN, ResNeXt-101-FPN) 替换默认的骨干网络 (ResNet-50-FPN). 如后文表 4 所示, 在相同的设置下, ResNet-101-FPN 带来了 1.8 个百分点的提升, ResNeXt-101-FPN 能够带来 4.4 个百分点的提升.

3.2.7 嵌入向量维度 d 对实例分割精度的影响

为了研究嵌入向量维度 d 对实例分割精度的影响, 本文比较了在不同的嵌入向量维度 d 下, 本文方法的平均精度. 本文让嵌入向量维度 d 从 8 增加到 24, 然后对比不同嵌入向量维度下的模型在 COCO minival 数据集上的平均精度. 从表 5 中可以看到, 本文方法的平均精度随着维度 d 的增加而不断提升. 当 $d > 16$ 时, 结果平均精度开始饱和. 又因为嵌入向量维度的增大会带来额外的计算和显存的消耗, 为了权衡平均精度和计算量, 本文将 d 的默认值设置为 16.

表 4 骨干网络的影响 (%)

骨干网络	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50[4]	32.7	53.0	34.5	13.9	35.9	48.4
ResNet-101[4]	34.5	55.4	36.4	14.6	37.9	52.2
ResNeXt-101[20]	**37.1**	**59.1**	**39.3**	**15.6**	**41.1**	**55.8**

表 5 嵌入向量维度 d 对结果的影响 (%)

d	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
8	32.3	52.8	33.8	13.6	34.9	49.0
12	32.5	52.8	34.2	13.3	35.5	**49.1**
16	**32.7**	53.0	**34.5**	**13.9**	35.9	48.4
20	32.6	52.9	34.2	13.8	35.9	48.6
24	**32.7**	**53.2**	34.2	13.8	**36.0**	48.9

3.2.8 公式 (4) 中缩放因子α对实例分割精度的影响

本文还研究了映射函数$Proj(\cdot)$中缩放因子α在不同取值下实例分割精度的影响. 本文让缩放因子α从 1 增大到 9, 然后对比不同α值下的模型在 COCO minival 数据集上的平均精度, 结果如表 6 所示. 当$\alpha=1$时, 本文方法的平均精度掉到 31.0. 当$\alpha=5$时, 模型取得最好平均精度 32.7.

为了解释这一现象, 本文在图 9 中绘制了映射函数$Proj(x)$在不同缩放因子α下的函数曲线. 从图中可以发现, 随着缩放因子α的增大, 函数曲线变得越发陡峭, 映射函数$Proj(\cdot)$的值域也越来越接近 (0, 1). 当$\alpha=1$时, $Proj(\cdot)$的值域大致为 [0.27, 0.73], 与监督信号的区间 [0, 1] 有较大的区别, 导致二值交叉熵损失函数没有充分收敛. 当$\alpha>5$时, 映射函数$Proj(\cdot)$的值域基本等于 [0, 1], 这与二值交叉熵的要求相符合.

表 6 缩放因子α的影响 (%)

α	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
1	31.0	52.0	32.0	13.0	34.3	46.2
3	32.3	52.6	33.8	13.7	35.1	48.4
5	**32.7**	**53.0**	**34.5**	**13.9**	**35.9**	48.4
7	32.6	52.6	34.2	**13.9**	35.7	49.3
9	32.4	52.6	34.2	13.8	35.5	**49.8**

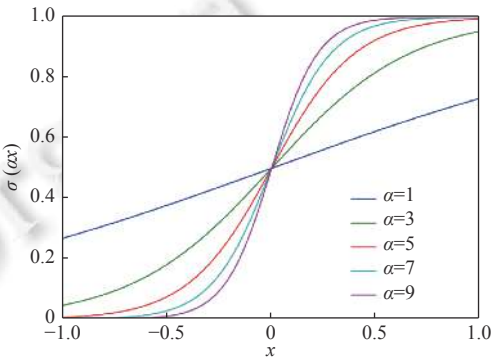


图 9 不同α取值下映射函数$Proj(x)$的曲线

3.2.9 嵌入向量生成模块的有效性

为了验证本文的嵌入向量生成模块的有效性, 本文将其与 Panoptic FPN[36]进行对比. Panoptic FPN 是一个轻量的像素级预测的生成模块. 正如在第 2.3 节中所讨论的, 嵌入向量生成模块应该在存储和计算上非常高效, 因此本文并没有使用重量级的分割方法 (如: PSPNet[37]) . 为了公平比较, 在同样的测试条件下, 本文在预测像素的嵌入向量时, 将本文的嵌入向量生成模块替换为 Panoptic FPN. 如表 7 所示, 本文的嵌入向量生成模块在使用更少的存储和计算量的情况下取得了和 Panoptic FPN 近似的表现 (32.7 对比 32.8), 其中浮点运算次数 (floating point operations, FLOPs) 是在输入图片尺寸为 800×800×3 的情况下计算得到.

表 7 嵌入向量生成模块的有效性

嵌入向量生成模块	参数量 (M)	FLOPs (G)	时间 (ms)	AP (%)
Ours	**0.7**	**6.3**	**2.3**	32.7
Panoptic FPN[36]	1.8	25.2	9.4	**32.8**

3.3 与主流实例分割方法的对比

本文展示了提出的网格实例分割方法 (GridMask) 在 MS COCO test-dev 数据集上的实验结果, 并且将其与现有主流的单阶段和二阶段方法进行了比较.

为了与二阶段方法进行公平的比较, 本文在没有使用数据增广的条件下, 使用 1×训练策略 (详见第 3.1 节) 来训练本文的模型. 如表 8 所示, 本文的方法取得了与二阶段方法相媲美的平均精度 (AP), 表 8 中, “Aug”表示数据增广, “√”表示在训练过程中使用了数据增广, “—”表示在训练过程中没有使用数据增广. 当骨干网络是 ResNet-101-FPN 时, 本文方法的平均精度为 34.8, 仅比 Mask R-CNN 低不到 1 个百分点. 当骨干网络是 ResNeXt-101-FPN 时, 本文方法的平均精度提高到了 37.3, 比 Mask R-CNN 高.

当与单阶段方法比较时, 因为现有的单阶段方法[13,14]都采用了数据增广和更多的迭代次数, 本文采用 2×训练策略 (18 万次迭代) 来训练本文的模型, 并且对图像的短边进行抖动, 让其在 [640, 800] 区间内随机取值. 如表 5 所示, 当骨干网络是 ResNet-101-FPN 时, 本文方法取得了 36.8 的平均精度, 这个结果比主流方法 YOLACT 高 5.6 个

百分点. 与 TensorMask[14]相比, 本文的方法和 TensorMask 的平均精度相差不大 (36.8 vs. 37.1), 但是在相同的硬件/软件条件下, 本文的“GridMask+ResNet-101-FPN”在使用英伟达 V100 显卡上能够达到 11.6 f/s, 比 TensorMask 快 4 倍. 如果本文使用 4 线程优化网络切分二值化, “GridMask+ResNet-101-FPN”的速度可以提高到 13.5 f/s, 比 TensorMask 快 5 倍. 值得一提的是, GridMask 也适用于一些最新的方法 (如: RDSNet[30]和 CondInst[35]), 可以进一步提高这些算法的精度. 从表 8 中可以看到, “GridMask+RDSNet”在 MS COCO test-dev 上的 AP 为 37.5, 比原 RDSNet 高 1.1 个百分点, 尤其在大物体的实例分割结果 (AP_L) 上, GridMask 可以带来约 2 个百分点的提升. “GridMask+CondInst”在 MS COCO test-dev 取得 40.1 的 AP, 比原 CondInst 高 1 个百分点, 比 SOLO[31]高 2.3 个百分点. 这些结果验证了本文提出的 GridMask 的有效性.

表 8 在 COCO test-dev 数据集上的实例分割结果 (%)

方法		骨干网络	训练轮数	Aug	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
二阶段方法	MNC[38]	ResNet-101-C4	12	—	24.6	44.3	24.8	4.7	25.9	43.6
	FCIS[22]	ResNet-101-C5-dilated	12	—	29.2	49.5	—	7.1	31.3	50.0
	Mask R-CNN[7]	ResNet-101-FPN	12	—	35.7	58.0	37.8	15.5	38.1	52.4
	Mask R-CNN[7]	ResNeXt-101-FPN	12	—	37.1	60.0	39.4	15.9	39.9	53.5
单阶段方法	TensorMask[14]	ResNet-101-FPN	72	√	37.1	59.3	39.4	17.1	39.1	51.6
	YOLACT[13]	ResNet-101-FPN	48	√	31.2	50.6	32.8	12.1	33.3	47.1
	PolarMask[24]	ResNet-101-FPN	24	√	32.1	53.7	33.1	14.7	33.8	45.3
	PolarMask++[25]	ResNet-101-FPN	24	√	33.8	57.5	34.6	16.6	35.8	46.2
	RDSNet[30]	ResNet-101-FPN	24	√	36.4	57.9	39.0	16.4	39.5	51.5
	SipMask[34]	ResNet-101-FPN	72	—	38.1	60.2	40.8	17.8	40.8	54.3
	SOLO[31]	ResNet-101-FPN	36	√	37.8	59.5	40.4	16.4	40.6	54.2
	CondInst[35]	ResNet-101-FPN	36	√	39.1	60.9	42.0	21.5	41.7	50.9
本文的方法	GridMask	ResNet-101-FPN	12	—	34.8	55.9	36.8	14.5	37.3	50.6
	GridMask	ResNeXt-101-FPN	12	—	37.3	59.5	39.7	16.7	40.4	54.0
	GridMask	ResNet-101-FPN	24	√	36.8	58.7	39.1	16.0	39.7	53.0
	GridMask	ResNeXt-101-FPN	24	√	38.6	61.3	40.9	18.0	41.6	**55.1**
	GridMask+RDSNet[30]	ResNet-101-FPN	24	√	37.5	59.8	40.0	17.1	40.8	53.7
	GridMask+CondInst[35]	ResNet-101-FPN	36	√	**40.1**	**62.3**	**43.0**	**21.9**	**42.7**	52.7

另外, 本文在图 10 和图 11 中还展示了本文方法的一些可视化结果. 可以看到, 本文方法即使在一些复杂场景下也能取得质量较高的实例分割结果. 本文的网格切分策略能够通过将实例切块来简化分割的难度, 对于复杂场景 (如, 实例之间出现遮挡) 也具有很好的处理能力.

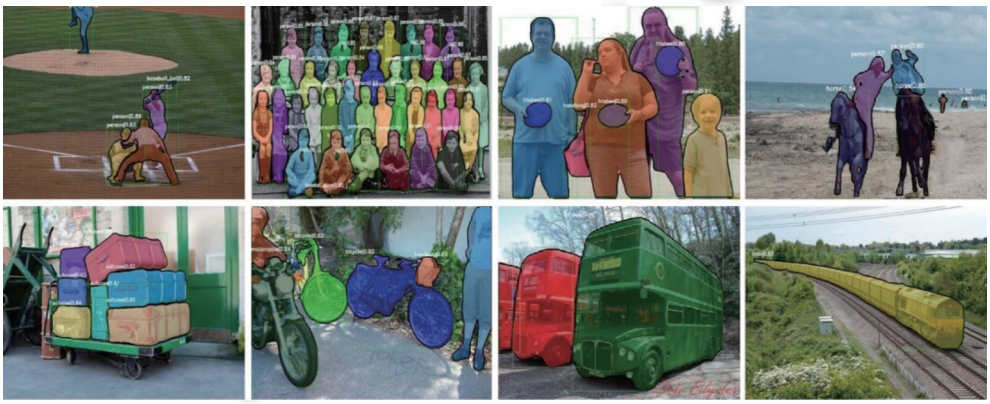


图 10 ResNet-50-FPN+GridMask 在 MS COCO minival 数据集上的可视化结果

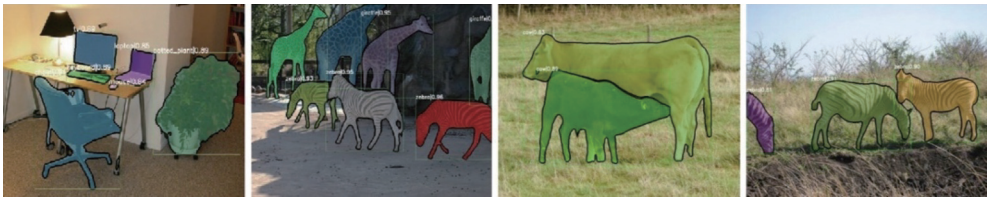


图 10 ResNet-50-FPN+GridMask 在 MS COCO minival 数据集上的可视化结果 (续)

图 11 MS COCO minival 数据集上使用“YOLACT+ResNet-101-FPN”“Mask R-CNN+ResNet-101-FPN” 和“GridMask+ResNet-101-FPN”得到的可视化结果

3.4 速度分析

本文分析了提出的网格实例分割方法中不同模块的前向推理所需的时间. 如后文表 9 所示, 本文测试了在图像短边为{800, 600, 400}情况下, 本文方法在 MS COCO minival 数据集中所有图像的平均速度. 从表中可以看到, 与骨干网络和检测模块消耗的时间相比, 嵌入向量生成模块的耗时非常低 (1.3–2.3 ms), 这说明本文方法所需要的资源仅稍微高于它的物体检测模块 (如: FCOS). 另外, 网格切分二值化算法是可以并行的, 使用多线程能够其进行加速. 例如, 本文可以使用 4 个线程将网络切分二值化算法的运行时间降低为原来的 1/4. 这些实验结果均是在使用批处理大小为 1, 英伟达 V100 显卡和主频为 2.20 GHz 的 CPU 的情况下得到.

4 总结和展望

本文提出了一个精确且灵活的单阶段实例分割方法——网格实例分割 (GridMask), 该方法能够达到与二阶段方法媲美的实例分割精度, 而且能够非常方便地移植到不同的物体检测方法中. 在 MS COCO 数据集上的实验充

分证明了本文方法的优越性. 该单阶段实例分割相比两阶段实例分割而言, 模型更加轻量, 处理流程更加简洁, 在分割速度上更具优势, 并且在实际部署应用上更加友好. 诸多实际应用 (如自动驾驶) 需要实时处理输入数据, 在这些领域中, 本文所提出的单阶段实例分割方法具有较强的应用价值. 本文认为未来单阶段实例分割的研究方向应该为在保证具有能够进行实时分割的能力下, 不断提高模型的分割能力, 使单阶段实例分割方法能够同时在速度和精度上超过两阶段的实例分割方法. 本文希望该工作能抛砖引玉, 对后续的单阶段实例分割研究有所启发.

表 9 “GridMask+ResNet-50-FPN”在 MS COCO minival 数据集上的时间消耗

图像尺寸	网络模块 (ms)			后处理 (ms)		AP (%)	速度 (f/s)
	Back	Det	Dec	Det	Dec		
800	18.7	22.4	2.3	13.8	25.2/6.8	**32.7**	12.1/15.6
600	10.9	13.6	1.7	12.4	21.9/6.3	30.8	16.5/22.3
400	**7.6**	**10.4**	**1.3**	**11.4**	**19.5/5.8**	25.6	**19.9/27.4**

注: “Back”“Det”和“Dec”分别表示骨干网络, 检测模块和嵌入向量生成模块. 带下划线的结果是用4个线程优化的网络切分二值化算法得到的

References:

[1] Jiang F, Gu Q, Hao HZ, Li N, Guo YW, Chen DX. Survey on content-based image segmentation methods. Ruan Jian Xue Bao/Journal of Software, 2017, 28(1): 160–183 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5136.htm [doi: 10.13328/j.cnki.jos.005136]

[2] Yao RQ, Tang JF, Yu JQ, Wang ZK. Player detection and segmentation in sports video. Ruan Jian Xue Bao/Journal of Software, 2015, 26: 155–164 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15026.htm

[3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego: ICLR, 2015.

[4] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

[5] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 2261–2269. [doi: 10.1109/CVPR.2017.243]

[6] Zhang S, Gong YH, Wang JJ. The development of deep convolution neural network and its applications on computer vision. Chinese Journal of Computers, 2019, 42(3): 453–482 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2019.00453]

[7] He KM, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Vision. Venice: IEEE, 2017. 2980–2988. [doi: 10.1109/ICCV.2017.322]

[8] Liu S, Qi L, Qin HF, Shi JP, Jia JY. Path aggregation network for instance segmentation. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8759–8768. [doi: 10.1109/CVPR.2018.00913]

[9] Huang ZJ, Huang LC, Gong YC, Huang C, Wang XG. Mask scoring R-CNN. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 6402–6411. [doi: 10.1109/CVPR.2019.00657]

[10] Ren SQ, He KM, Girshick RB, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Proc. of the 2015 Advances in Neural Information Processing Systems. Montreal: NIPS, 2015. 91–99.

[11] Girshick R. Fast R-CNN. In: Proc. of the 2015 IEEE Int'l Conf. on Computer Vision. Santiago: IEEE, 2015. 1440–1448. [doi: 10.1109/ICCV.2015.169]

[12] Bai M, Urtasun R. Deep watershed transform for instance segmentation. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 2858–2866. [doi: 10.1109/CVPR.2017.305]

[13] Bolya D, Zhou C, Xiao FY, Lee YJ. YOLACT: Real-time instance segmentation. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 9156–9165. [doi: 10.1109/ICCV.2019.00925]

[14] Chen XL, Girshick R, He KM, Dollar P. TensorMask: A foundation for dense object segmentation. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 2061–2069. [doi: 10.1109/ICCV.2019.00215]

[15] Dai JF, He KM, Li Y, Ren SQ, Sun J. Instance-sensitive fully convolutional networks. In: Proc. of the 14th European Conf. on Computer Vision. Amsterdam: Springer, 2016. 534–549. [doi: 10.1007/978-3-319-46466-4_32]

[16] Xu WQ, Wang HY, Qi FB, Lu CW. Explicit shape encoding for real-time instance segmentation. In: Proc. of the 2019 IEEE/CVF Int'l

Conf. on Computer Vision. Seoul: IEEE, 2019. 5167–5176. [doi: 10.1109/ICCV.2019.00527]

[17] Wang WH, Xie EZ, Song XG, Zang YH, Wang WJ, Lu T, Yu G, Shen CH. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 8439–8448. [doi: 10.1109/ICCV.2019.00853]

[18] Wang WH, Xie EZ, Li X, Liu XB, Liang D, Yang ZB, Lu T, Shen CH. PAN++: Towards efficient and accurate end-to-end spotting of arbitrarily-shaped text. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2021.

[19] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft COCO: Common objects in context. In: Proc. of the 13th European Conf. on Computer Vision. Zurich: Springer, 2014. 740–755. [doi: 10.1007/978-3-319-10602-1_48]

[20] Xie SN, Girshick R, Dollár P, Tu ZW, He KM. Aggregated residual transformations for deep neural networks. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 5987–5995. [doi: 10.1109/CVPR.2017.634]

[21] Wang ZY, Yuan C, Li JC. Instance segmentation with separable convolutions and multi-level features. Ruan Jian Xue Bao/Journal of Software, 2019, 30(4): 954–961 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5667.htm [doi: 10.13328/j.cnki.jos.005667]

[22] Li Y, Qi HZ, Dai JF, Ji XY, Wei YC. Fully convolutional instance-aware semantic segmentation. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 4438–4446. [doi: 10.1109/CVPR.2017.472]

[23] Lin TY, Dollár P, Girshick R, He KM, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 936–944. [doi: 10.1109/CVPR.2017.106]

[24] Xie EZ, Sun PZ, Song XG, Wang WH, Liu XB, Liang D, Shen CH, Luo P. PolarMask: Single shot instance segmentation with polar representation. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 12190–12199. [doi: 10.1109/CVPR42600.2020.01221]

[25] Xie EZ, Wang WH, Ding MY, Zhang RM, Luo P. PolarMask++: Enhanced polar representation for single-shot instance segmentation and beyond. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2021.

[26] Lin TY, Goyal P, Girshick R, He KM, Dollár P. Focal loss for dense object detection. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Vision. Venice: IEEE, 2017. 2999–3007. [doi: 10.1109/ICCV.2017.324]

[27] Tian Z, Shen CH, Chen H, He T. FCOS: Fully convolutional one-stage object detection. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 9626–9635. [doi: 10.1109/ICCV.2019.00972]

[28] LeCun Y, Haffner P, Bottou L, Bengio Y. Object recognition with gradient-based learning. In: Forsyth DA, Mundy JL, Di Gesú V, Cipolla R, eds. Shape, Contour and Grouping in Computer Vision. Berlin: Springer, 1999. 319–345. [doi: 10.1007/3-540-46805-6_19]

[29] Wu YX, He KM. Group normalization. In: Proc. of the 15th European Conf. on Computer Vision. Munich: Springer, 2018. 3–19. [doi: 10.1007/978-3-030-01261-8_1]

[30] Wang SR, Gong YC, Xing JL, Huang LC, Huang C, Hu WM. RDSNet: A new deep architecture forreciprocal object detection and instance segmentation. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. New York: AAAI, 2020. 12208–12215. [doi: 10.1609/aaai.v34i07.6902]

[31] Wang XL, Kong T, Shen CH, Jiang YN, Li L. SOLO: Segmenting objects by locations. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 649–665. [doi: 10.1007/978-3-030-58523-5_38]

[32] Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. ImageNet: A large-scale hierarchical image database. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255. [doi: 10.1109/CVPR.2009.5206848]

[33] Caesar H, Uijlings J, Ferrari V. COCO-stuff: Thing and stuff classes in context. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1209–1218. [doi: 10.1109/CVPR.2018.00132]

[34] Cao JL, Anwer RM, Cholakkal H, Khan FS, Pang YW. SipMask: Spatial information preservation for fast image and video instance segmentation. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 1–18. [doi: 10.1007/978-3-030-58568-6_1]

[35] Tian Z, Shen CH, Chen H. Conditional convolutions for instance segmentation. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 282–298. [doi: 10.1007/978-3-030-58452-8_17]

[36] Kirillov A, Girshick R, He KM, Dollár P. Panoptic feature pyramid networks. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 6392–6401. [doi: 10.1109/CVPR.2019.00656]

[37] Zhao HS, Shi JP, Qi XJ, Wang XG, Jia JY. Pyramid scene parsing network. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 6230–6239. [doi: 10.1109/CVPR.2017.660]

[38] Dai JF, He KM, Sun J. Instance-aware semantic segmentation via multi-task network cascades. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 3150–3158. [doi: 10.1109/CVPR.2016.343]

附中文参考文献:

[1] 姜枫, 顾庆, 郝慧珍, 李娜, 郭延文, 陈道蓄. 基于内容的图像分割方法综述. 软件学报, 2017, 28(1): 160–183. http://www.jos.org.cn/1000-9825/5136.htm [doi: 10.13328/j.cnki.jos.005136]

[2] 姚沁汝, 唐九飞, 于俊清, 王赠凯. 体育视频中的运动员检测与分割. 软件学报, 2015, 26: 155–164. http://www.jos.org.cn/1000-9825/15026.htm

[6] 张顺, 龚怡宏, 王进军. 深度卷积神经网络的发展及其在计算机视觉领域的应用. 计算机学报, 2019, 42(3): 453–482. [doi: 10.11897/SP.J.1016.2019.00453]

[21] 王子愉, 袁春, 黎健成. 利用可分离卷积和多级特征的实例分割. 软件学报, 2019, 30(4): 954–961. http://www.jos.org.cn/1000-9825/5667.htm [doi: 10.13328/j.cnki.jos.005667]

王文海(1994－), 男, 博士, CCF 学生会员, 主要研究领域为场景文字检测, 实例分割, 神经网络.

路通(1976－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为计算机视觉, 场景文本检测.

李志琦(1998－), 男, 博士生, 主要研究领域为实例分割, 全景分割.