倪敏腾, 黄影平. 基于蛇模型的弱监督实例分割网络[J]. 智能计算机与应用,2026,16(8):82-90. DOI:10.20169/j.issn.2095-2163.24121601

基于蛇模型的弱监督实例分割网络

倪敏腾, 黄影平
(上海理工大学 光电信息与计算机工程学院, 上海 200093)

摘 要: 实例分割旨在从像素级层面分割图像中不同类别的单个对象。目前大多数实例分割网络都是全监督的,使用像素级标注样本进行有监督的模型训练,但逐像素地标注数据费时费力。现有的基于蛇模型的实例分割网络通过轮廓变形网络迭代变形初始轮廓来完成实例分割。本文在蛇模型的基础上提出了一种弱监督实例分割网络,仅使用边界框标注的数据集就可以完成训练,而无需对数据集进行耗时的逐像素标注。本文设计了两个损失函数,包括投影损失项和颜色成对损失项,以弱监督的方式进行模型训练。在初始轮廓的选择上,通过对 4 种初始轮廓变形方案进行实验比较,选出椭圆轮廓作为蛇模型的初始轮廓。与其他实例分割网络的对比实验表明,本文方法在精度和速度方面具有综合优势。

关键词: 实例分割; 蛇模型; 弱监督
中图分类号: TP391.4 **文献标志码**: A **文章编号**: 2095-2163(2026)08-0082-09

Weakly supervised instance segmentation network based on the snake model

NI Minteng, HUANG Yingping
(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Instance segmentation aims to segment individual objects of different categories in an image at the pixel level. Currently, most instance segmentation networks are fully supervised, which adopt pixel-level annotations for supervised model training, but it is time-consuming and laborious to label data pixel by pixel. Existing snake model-based instance segmentation networks complete instance segmentation by iteratively deforming an initial contour via a contour deformation network. Based on the snake model, this paper proposes a weakly supervised instance segmentation network, which can be trained with only bounding-box annotations without the need for time-consuming pixel-by-pixel annotation of the dataset. This paper designs two loss functions, including a projection loss term and color pairwise loss term, to train the model in a weakly supervised manner. In the selection of the initial contour, the elliptical contour is selected as the initial contour of the snake model through experimental comparison of four initial contour deformation schemes. The comparative experiments with other instance segmentation networks show that the proposed method has comprehensive advantages in terms of accuracy and speed.

Key words: instance segmentation; snake model; weak supervision

0 引 言

实例分割是计算机视觉的重要研究内容,旨在像素级地分割出图像中不同类别的单个对象,被广泛应用于自动驾驶、医疗、工业控制等领域。目前大多数实例分割方法都是全监督的,使用像素级的标注样本进行有监督的模型训练。这些方法虽然取得了显著的效果,但逐像素地标注数据费时费力。因此,一些研究者提出了弱监督的实例分割方法,使用仅有边界框标注的数据集对实例分割模型进行训练,如 Khoreva 等[1]提出 SDI 模型,但这些弱监督实例分割模型大多难以达到全监督实例分割模型的性能。

受蛇模型或活动轮廓方法的影响,最近的一些深度学习实例分割方法使用活动轮廓代替逐像素的掩膜表示实例分割的结果,如 Curve-GCN[2] 和 Deep Snake[3]。Deep Snake 根据检测器的结果框生成八边形的初始轮廓,对轮廓上的输入特征进行圆周卷积运算,得到轮廓的输出特征并预测轮廓点与目标点之间的偏移,通过不断迭代得到最终的实例分割

作者简介: 倪敏腾(1997—),男,硕士研究生。主要研究方向:弱监督实例分割。
通信作者: 黄影平(1966—),男,博士,教授。主要研究方向: 智能汽车障碍物探测辨识,汽车电子系统仿真测试,视觉检测技术。Email: huangyingping@usst.edu.cn。

收稿日期: 2024-12-16

轮廓。基于蛇模型的这一类方法，有更快的运行速度，并且取得了和传统实例分割方法相似的性能。

针对上述问题，本文提出了一种基于蛇模型的弱监督实例分割网络（Weakly Supervised Snake Model，WSS），其核心思想是使用两个损失项替代原有的全监督平滑损失函数以实现弱监督的模型训练。第一项基于如下假设：物体的边界框与其轮廓在垂直、水平两个方向的投影应当一致，最小化轮廓和边界框在垂直和水平两个方向上的差距。第二项利用了邻近像素点之间的颜色关系，即目标体内或背景中邻近的两个点的颜色应该相似，最小化目标体内的颜色差异。

利用上述的两个损失函数，仅使用边界框标注的数据集作为监督，得到了与全监督实例分割方法相似的分割性能。本文的 WSS 在 Cityscapes[4] 和 SBD[5] 数据集上表现出较好的实例分割性能，同时可以满足实时实例分割的要求，在 RTX 3080ti GPU 上的运行速度为 34.6 fps。

总结来说，本文有以下贡献：

1）重新设计了蛇模型的损失函数，实现了以弱监督的方式训练蛇模型完成实例分割任务，即在只有边界框标注的数据集中完成实例分割的训练。与其他实例分割网络的对比实验表明，本文方法在精度和速度方面具有综合优势。

2）提出了笛卡尔坐标系和伪极坐标系下的 4 种初始轮廓变形方案，并使用 SBD 数据集对 4 种方案进行对比实验，实验结果表明，使用了椭圆形初始轮廓的方案 D 实例分割效果最佳。

1　相关工作

1.1　实例分割

目前的实例分割网络可以分为两阶段和单阶段两类方法：

自上而下的两阶段实例分割，如 He 等提出的 Mask R-CNN[6] 为代表，该方法通过在已有目标检测网络 Faster R-CNN[7] 基础上新增一个并行分支来预测目标掩膜，最终实现了 5 fps 的推理速度。但此类方法严重依赖候选区域生成，整体性能存在明显瓶颈。

单阶段的实例分割不需要生成候选区域，可以直接对输入图像进行分割，如 Tian 等提出的全卷积单阶段框架 FCOS[8] 为代表，该方法以逐像素全卷积的方式完成实例分割任务，网络结构相较两阶段方法更简洁，训练和运行速度也更快。但上述早期提出的方法均属于全监督实例分割模型，训练过程需要耗费大量成本标注像素掩膜。

1.2　轮廓分割

1988 年 Kass 等[9] 提出活动轮廓的蛇模型，利用能量最小化模型使活动轮廓贴近物体的轮廓。2020 年 Peng 等[3] 提出 Deep Snake 模型利用神经网络迭代变形初始轮廓，使其匹配目标对象轮廓，在 Cityscapes 和 SBD 数据集上实现了实时实例分割，相较于传统逐像素的实例分割模型，实例分割精度有所提高，处理速度也更快。但 Deep Snake 也是一种全监督实例分割模型，训练 Deep Snake 依然需要逐像素掩膜标注的数据集。

1.3　弱监督实例分割

通过卷积网络训练实例分割模型可以获得相较于传统方法更好的效果，但这类方法依赖大量的训练样本或预训练模型。Tian 等[10] 提出 BoxInst 模型，在完全监督网络模型中重新设计了损失函数，通过投影损失项和颜色成对损失项改进了弱监督网络的分割结果，实现了掩膜级别的实例分割。

但 BoxInst 使用掩膜表示实例分割的结果，相较于基于活动轮廓（蛇模型）的实例分割方法，网络更为复杂，计算参数更多，运行速度较慢。本文提出的基于蛇模型的弱监督实例分割网络 WSS，可以兼具蛇模型与弱监督的优势。

2　方法

本文的 WSS 模型由检测器、初始轮廓生成模块、轮廓形变模块三部分组成，方法整体框图如图 1 所示。首先，将预处理后的原始图像输入检测器，得到目标对象的边界框与图像特征，随后基于边界框生成由 64 个采样点组成的初始轮廓，初始轮廓再经过轮廓变形网络的三次迭代，最终输出目标对象的预测轮廓。整个模型基于蛇模型框架完成图像的实例分割，沿用文献[3]中的圆周卷积算法处理对象的轮廓特征，训练时采用重新设计的投影损失项与颜色成对损失项，实现 WSS 模型仅依靠边界框标注数据集即可完成弱监督训练。

2.1　检测器

检测器使用了 CenterNet 网络[11]，网络结构如图 2 所示。CenterNet 是一种单阶段的检测模型，由主干部分和预测头部分组成。CenterNet 主干部分使用改进后的 DLA-34 网络[12] 提取特征，并在底层使用了可变形卷积[13]；预测部分的 3 个通道分别预测对象中心点的分布、3D 位置和对象边界框的长宽。

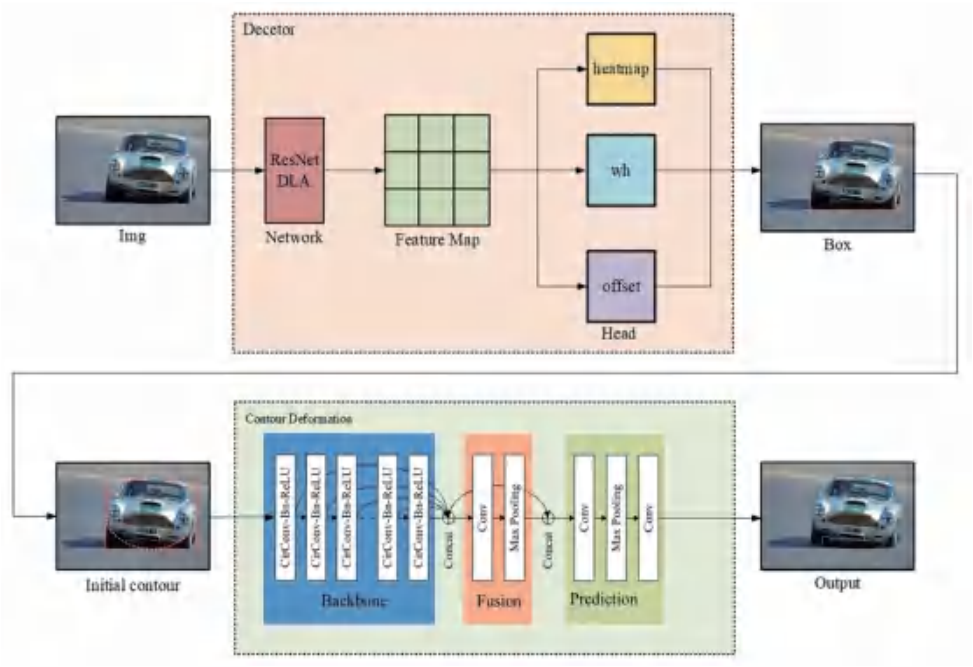


图 1 网络结构图

Fig. 1 Network structure diagram

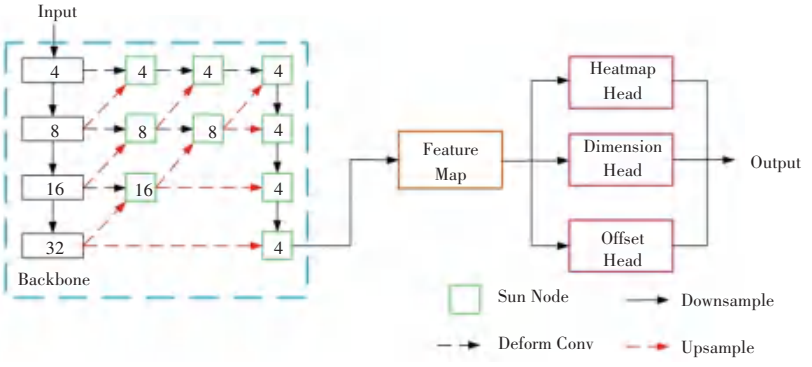


图 2 检测网络

Fig. 2 Detection network

检测器的损失函数由中心点损失 L_k、对象大小损失 L_{size} 与偏置损失 L_{off} 三部分组成:

$$L_{\text{det}} = L_k + \lambda_{\text{size}} L_{\text{size}} + \lambda_{\text{off}} L_{\text{off}} \quad (1)$$

其中, $\lambda_{\text{size}} = 0.1$, $\lambda_{\text{off}} = 1$。

2.2 初始轮廓

在蛇模型网络中,初始轮廓的设计是影响网络分割效果的重要因素之一。本文设计了四边形、八边形和椭圆形 3 种初始轮廓,采样点数量 N 设置为 64。

四边形轮廓由目标物体的 4 个极值点相连构成,采样点均匀分布在四边形轮廓上,如图 3(a)所示。八边形轮廓在 4 个极值点的基础上,分别在轮廓框上延伸出轮廓框长度的 1/4 形成 4 条线段,连结 4 条线段的端点形成八边形轮廓,采样点均匀分布在八边形轮廓上,如图 3(b)所示。椭圆形轮廓由内切轮廓框的椭圆形成,采样点均匀分布在椭圆形轮廓上,如图 3(c)所示。



(a) 四边形初始轮廓 (b) 八边形初始轮廓 (c) 椭圆初始轮廓

图 3 初始轮廓

Fig. 3 Initial outline

笛卡尔坐标系中采样点 P_i 的坐标表示为：

$$P_i(x_i,y_i) \tag{2}$$

伪极坐标系中采样点 P_i 的坐标表示为：

$$P_i=\left(x_0+x\cdot\cos\frac{\theta\pi}{32},y_0+y\cdot\sin\frac{\theta\pi}{32}\right)=(x_0,y_0,\rho_i,\theta_i) \tag{3}$$

其中，$x_0=\frac{x_{\min}+x_{\max}}{2}$，$y_0=\frac{y_{\min}+y_{\max}}{2}$，$x=\frac{\text{Width}}{2}$，$y=\frac{\text{Length}}{2}$，$\theta_i=\frac{\theta\pi}{32}$，$\rho_i=\sqrt{(x\cdot\cos\theta_i)^2+(y\cdot\sin\theta_i)^2}$。

2.3　轮廓变形

2.3.1　基于轮廓的图像特征学习

传统蛇模型方法将采样点作为变量，通过手工设置的能量函数得到目标的边界，但由于能量函数是非凸的且能量函数都是基于低级别的图像特征，只能得到局部最优解。

本文的轮廓变形网络采用端到端的方式进行轮廓变形。每个采样点 P_i 的输入特征 f_i 由图像特征与采样点相对坐标级联而成，形式为：$[F(P_i);p_i]$，F 表示输入图像在 CenterNet 检测网络最终得到的特征图，p_i 表示采样点的相对坐标。

与传统二维图像特征不同，为充分利用轮廓首尾相连接的循环特征，使用循环卷积学习轮廓的循环特征。循环卷积的计算如图 4 所示，蓝色节点是输入特征 f_i，黄色节点是卷积核，绿色节点是输出特征，输出特征的长度与输入特征的长度相同。与标准卷积相同，输出特征是输入特征与卷积核的乘积。

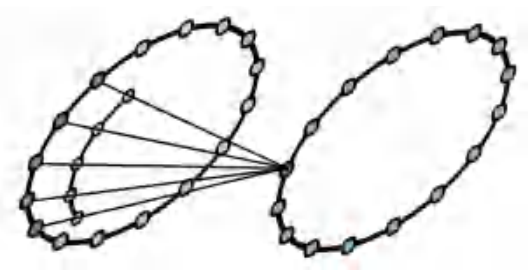


图 4　循环卷积

Fig. 4　Circular convolution

网络基于邻近采样点的图像特征生成每一个采样点的预测偏移量。计算圆周卷积时，将输入特征 f 重定义为周期信号，圆周卷积的运算公式为：

$$(f\cdot k)_i=\sum_{j=-r}^{r}f_{i-j}k_j \tag{4}$$

其中 $k:[-r,r]$ 是可学习的卷积核，设置卷积长度为 9，即 $r=4$。

2.3.2　轮廓变形网络

轮廓变形网络由主干部分、融合部分和预测部分组成，如图 5 所示。主干部分由 5 组圆周卷积层、批量规范化层(BN)和 ReLU 激活层组成，融合部分由一个卷积层和一个最大池化层基本块组成，预测部分由一个卷积层、一个最大池化层和一个卷积层组成。各层之间通过跳跃连接，再将不同尺度的特征级联作为后一层的输入，最后输出的结果是采样点的偏移量。

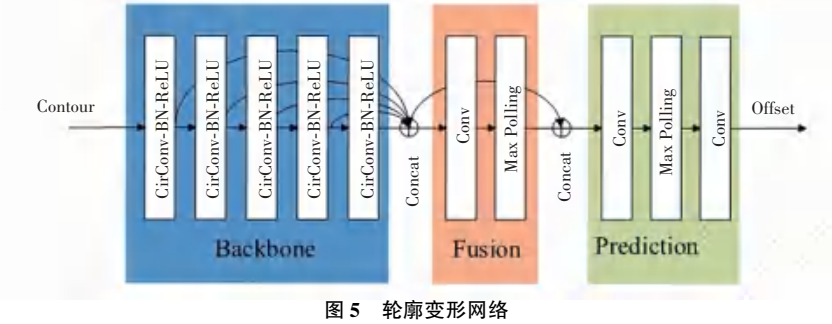


图 5　轮廓变形网络

Fig. 5　Contour deformation network

2.3.3　轮廓变形

WSS 基于笛卡尔坐标系和伪极坐标系设计了 4 种轮廓变形方案：

方案 A 和方案 B 分别使用笛卡尔坐标系的四边形初始轮廓和八边形初始轮廓，轮廓变形网络输出采样点在 x 轴和 y 轴的偏移量 Δx_i 和 Δy_i，变形后的采样点的坐标为

$$P_i^{'}=(x_i+\Delta x_i,y_i+\Delta y_i) \tag{5}$$

方案 C 使用伪极坐标系，利用了椭圆形的初始轮廓结构，保持采样点变形前后的极角 θ_i 不变，轮廓变形网络输出采样点极径的偏移量 $\Delta\rho_i$，每个采样点的坐标为：

$$P_i^{'}=(x_0,y_0,\rho_i+\Delta\rho_i,\theta_i) \tag{6}$$

其中，方案 A、方案 B 和方案 C 均用作方案 D 的对比实验。

方案 D 使用伪极坐标系，轮廓变形网络输出采样点极角和极径的偏移量 $\Delta\theta_i$ 和 $\Delta\rho_i$，每个采样点的坐标为：

$$P_i' = (x_0, y_0, \rho_i + \Delta\rho_i, \theta_i + \Delta\theta_i) \quad (7)$$

一次轮廓变形往往不能得到很好的实例分割结果,将前一次的轮廓变形结果作为初始轮廓再次输入轮廓变形网络,重复迭代多次可以优化模型,迭代次数 n 设置为 3。

2.4 投影损失与成对颜色损失

WSS 沿用了原有的蛇模型实例分割网络,仅通过重新设计损失函数实现了弱监督训练,新设计的损失函数包括投影损失项与成对颜色损失项。其中,投影损失项用于最小化对象的真实边界框和预测轮廓投影之间的差异,成对颜色损失项则利用成对像素的颜色特征对网络的实例分割结果进行监督。

2.4.1 投影损失项

投影损失项使用边界框的真值标注来监督实例分割网络生成的轮廓,预测轮廓应与边界框具有相同的水平和垂直投影。记 $b \in \{0,1\}^{H\times W}$ 为标注的掩膜,掩膜在 x 轴和 y 轴方向上的投影分别为:$\mathrm{Proj}_x(b) = l_x$, $\mathrm{Proj}_y(b) = l_y$,其中 $l_x \in \{0,1\}^W$, $l_y \in \{0,1\}^H$。记 $\tilde{m} \in \{0,1\}^{H\times W}$ 为实例分割网络生成的预测掩膜,其投影为 $\tilde{l}_x$ 和 $\tilde{l}_y$,如图 6 所示。则投影损失项的计算公式为:

$$L_{\mathrm{Proj}} = L(\mathrm{Proj}_x(\tilde{m}), \mathrm{Proj}_x(b)) + L(\mathrm{Proj}_y(\tilde{m}), \mathrm{Proj}_y(b)) = L(\tilde{l}_x, l_x) + L(\tilde{l}_y, l_y) \quad (8)$$

其中,L 为 Dice Loss,用于衡量投影向量的相似度。

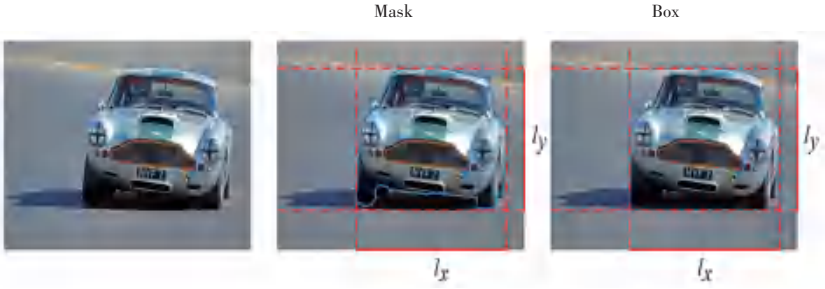


图 6 投影损失项

Fig. 6 Projection loss term

2.4.2 成对颜色损失项

由于没有使用像素级掩膜标注的数据集训练模型,本网络模型以成对的方式监督生成的掩膜。依据经验,相似颜色的邻近点很可能具有相同的标签,即在一定颜色阈值内邻近的两个点很可能都属于前景或背景。本文采用文献[11]中的成对损失函数公式如下:

$$L_{\mathrm{pairwise}} = -\frac{1}{N}\sum_{e\in E_{\mathrm{in}}} \mathbb{I}_{\{S_e>\tau\}} \log P(y_e = 1) \quad (9)$$

其中,$y_e \in \{0,1\}$ 表示一对像素点 e 的标签,$y_e = 1$ 表示两个像素点具有相同的标签,反之 $y_e = 0$;E_{in} 是 e 的集合,N 是 E_{in} 中的成对像素点 e 数量;S_e 是 e 的颜色相似度;$\mathbb{I}_{\{S_e>\tau\}}$ 是指示函数,如 $S_e > \tau$,则$\mathbb{I}$为 1,否则,$\mathbb{I}$为 0。

由于 WSS 采用轮廓变形的方式实现实例分割,因此本文取轮廓上点 P_i 与其径向邻近点,在 LAB 色彩空间中计算成对损失:针对采用笛卡尔坐标系的四边形初始轮廓与八边形初始轮廓,将邻近点定义为采样点周围、间隔不同扩张率的 8 个点;针对采用伪极坐标系的椭圆形初始轮廓,将邻近点定义为与采样点在同一条极径上一定领域范围内的点。如,$P_i(x_0, y_0, \rho_i, \theta_i)$ 的 4 个径向邻近点 Q_{i1}, Q_{i2}, Q_{i3} 与 Q_{i4} 分别为 $(x_0, y_0, \rho_i - 2\Delta\rho, \theta_i)$, $(x_0, y_0, \rho_i - \Delta\rho, \theta_i)$, $(x_0, \Delta\rho, \theta_i)$, $(x_0, y_0, \rho_i + 2\Delta\rho, \theta_i)$,如图 7 所示。



图 7 径向邻近点

Fig. 7 Radially adjacent points

活动轮廓的总损失 L 为:

$$L = L_{\mathrm{Proj}} + L_{\mathrm{pairwise}} \quad (10)$$

其中,L_{Proj} 为投影损失,L_{pairwise} 为成对颜色损失。

3 实验及结果分析

实验基于 Ubuntu 20.04 操作系统开展,硬件环境为 Intel(R) Core(TM) i7-7700K CPU @ 4.20 GHz 处理器,32 GB 内存,20 GB NVIDIA GeForce

RTX 3080Ti GPU，网络框架采用 Pytorch 搭建。

实验数据来自 SBD[5] 和 Cityscapes[4] 数据集。SBD 数据集有 11 355 张来自 PASCAL VOC 2011 的图像，包括交通工具、动物等 20 个类别。数据集对这些图像进行了像素级的实例标注。实验中使用 AP_{vol}、AP_{50}、AP_{70} 进行评估，AP_{vol} 是 AP 的平均值。

Cityscapes 数据集是由在 50 个城市晴朗天气下拍摄的 25 000 张街道实景图像组成，包含车辆、行人、背景等 8 个语义类。针对 Cityscapes 数据，本文以 8 个语义类的平均精度 AP 与模型运行时间综合评价分割网络性能。

实验主要分为 3 个部分：第一，对比弱监督蛇模型网络 WSS 和全监督蛇模型网络 Deep Snake[3] 在 SBD 数据集的分割结果，验证弱监督网络 WSS 与相同结构的全监督网络的分割性能相近。第二，对 4 种初始轮廓变形方案和不同的像素邻域分别进行实验，确定最佳的初始轮廓和像素邻域。第三，比较 WSS 与其他实例分割网络在 SBD 数据集和 Cityscapes 数据集的分割结果，验证弱监督网络 WSS 与全监督网络在复杂数据集的分割性能相近。

3.1　消融实验

在 SBD 数据集中对 WSS 进行消融实验，将 WSS 与具有相同网络结构的全监督网络 Deep Snake[3] 进行对比实验。两个网络采用同样的图像增强算法训练，经过 160 个周期的训练，初始学习率设置为 $1e^{-4}$，分别在第 80 和第 120 周期对学习率进行衰减（衰减系数为 0.5）。消融实验的结果见表 1 所示，从表 1 中可以看出，仅使用边界框标注的数据完成弱监督训练的 WSS，与全监督训练的 Deep Snake[3] 相比，AP_{vol} 仅相差 0.3，同时 WSS 运行速度略快于 Deep Snake。消融实验结果示例如图 8 所示，在相同的网络结构下，本文采用的弱监督实例分割训练能够达到与全监督训练相近的实例分割性能。

表 1　消融实验

Table 1　Ablation experiment

模型	AP_{vol}	AP_{50}	AP_{70}
Deep Snake	54.4	62.1	48.3
WSS	54.1	61.7	47.8



图 8　消融实验结果示例

Fig. 8　Example of ablation experiment results

3.2　初始轮廓与像素临域

在 SBD 数据集中对 2.3.3 中方案 A-D 4 种轮廓变形策略进行对比实验，方案 A 和方案 B 分别使用笛卡尔坐标系的四边形初始轮廓和八边形初始轮廓，方案 C 和方案 D 均使用伪极坐标系的椭圆形初始轮廓，但是变形策略不同。

方案 A-D 使用相同的弱监督训练网络，实验结果见表 2，对比实验结果示例如图 9 所示。与方案 C 相比，同样具有弱监督网络使用径向邻近点监督颜色相似性损失的优势，由于方案 D 在极角方向有偏移量回归，弥补了初始轮廓的采样点均是均匀分布的局限性，因此实例分割效果更好。

表 2　不同轮廓变形策略的性能比较

Table 2　Performance comparison of different contour deformation strategies

方案	AP_{vol}	AP_{50}	AP_{70}
方案 A	52.0	59.3	46.0
方案 B	53.5	60.8	47.1
方案 C	52.4	59.8	46.4
方案 D	**54.1**	**61.7**	**47.8**



图 9 不同轮廓变形策略的对比实验结果示例

Fig. 9 Comparison results of different contour deformation strategies

在 SBD 数据集中对 2.4 节中椭圆形初始轮廓中成对颜色损失项的不同像素邻域进行实验。增大像素邻域可以扩大成对颜色损失项的计算范围,本实验中选取了 $(-\Delta\rho,\Delta\rho)$、$(-2\Delta\rho,2\Delta\rho)$ 和 $(-4\Delta\rho,4\Delta\rho)$ 三个像素邻域,实验结果见表 3。结果表明,$(-2\Delta\rho,2\Delta\rho)$、$(-4\Delta\rho,4\Delta\rho)$ 像素邻域的实验结果相近,且均优于 $(-\Delta\rho,\Delta\rho)$ 像素邻域,因此最终选用了 $(-2\Delta\rho,2\Delta\rho)$ 像素邻域。

表 3 不同像素邻域的性能比较

Table 3 Performance comparison of different pixel neighborhoods

像素邻域	AP_{vol}	AP_{50}	AP_{70}
$(-\Delta\rho,\Delta\rho)$	53.0	61.4	47.3
$(-2\Delta\rho,2\Delta\rho)$	**54.1**	**61.7**	**47.8**
$(-4\Delta\rho,4\Delta\rho)$	53.3	61.5	46.3

3.3 SBD 数据集和 Cityscapes 数据集的实验结果

在 SBD 数据集和 Cityscapes 数据集的实验中使用了与 Deep Snake[3] 相同的训练配置,先对检测器单独训练 140 个周期,初始学习率设置为 $1e^{-4}$,并在第 80 和第 120 周期各下降一半;然后对轮廓变形网络进行端到端训练 200 个周期,初始学习率设置为 $1e^{-4}$,在第 80、第 120 和第 150 周期各下降一半。

表 4 在 SBD 数据集中对 WSS 和其他基于轮廓的实例分割网络进行了比较,STS[14] 使用径向矢量表示目标的轮廓,ESE[15] 使用显式形状编码近似目标的轮廓。在使用相同的主干网络和训练条件下,仅使用弱监督训练的 WSS 实验结果优于其他类型的完全监督的实例分割网络。WSS 在验证集上的性能稍逊于 Deep Snake,仅相差 0.6 AP。WSS 在 SBD 数据集中的测试结果如图 10 所示。可以看出 WSS 对于各个尺度的车辆和小尺度的行人这类轮廓较为简单的目标处理得很好,但是由于初始轮廓和轮廓变形的流程设计对于大尺度的行人目标,其较大内凹的部分存在一些误检。由此可见,弱监督训练网络 WSS 在一般场景中依然具有优势,处理速度达 34.6 fps,可以满足实时实例分割的要求。

表 4 SBD 测试结果

Table 4 SBD test results

Method	AP_{vol}	AP_{50}	AP_{70}	Time/ms	fps
STS[14]	29.0	30.0	6.5	28.0	35.7
ESE-20[15]	35.3	40.7	12.1	**24.0**	**41.7**
Deep Snake	54.4	62.1	48.3	31.0	32.3
WSS	53.8	60.8	45.3	28.9	34.6



图 10 BD 测试结果示例

Fig. 10 Example of SBD test results

表 5 在 Cityscapes 数据集中对 WSS 和其他的实例分割网络进行了比较，在使用相同的主干网络和训练条件下，仅使用弱监督训练的 WSS 实验结果甚至优于一些完全监督的实例分割网络。WSS 在验证集上的性能稍逊于 Deep Snake ，仅落后 0.2 AP，但在车辆类别上，WSS 反而比 Deep Snake 高出 0.2 AP。WSS 在 Cityscapes 数据集中的测试结果如图 11 所示。在随机选取的 6 张测试图片中对车辆目标和行人目标进行检测，本文的方法没有发生误检。可以看出 6 张测试图片中的目标轮廓大多是简单非凹的，WSS 对于这类大尺度的车辆目标、小尺度的车辆目标以及中小尺度的行人目标处理得都很好。由此可见，弱监督训练网络 WSS 在复杂场景中依然具有优势，处理速度达 4.9 fps，可满足实时实例分割的要求。

表 5　Cityscapes 测试结果

Table 5　Cityscapes test results

Method	AP_{val}	AP	AP_{50}	person	car	bicycle	fps
SGN	29.2	25.0	44.9	21.8	39.4	12.4	0.6
Polygon RNN+		25.5	45.5	29.4	48.3	13.6	
Mask R-CNN	31.5	26.2	49.9	30.5	46.9	16.0	2.2
PA Net[16]	36.5	**31.8**	57.1	36.8	54.8	20.8	
Deep Snake	**37.4**	31.7	**58.4**	**37.2**	56.0	**16.4**	4.6
WSS	37.2	31.6	58.0	37.0	**56.2**	15.9	**4.9**



图 11　Cityscapes 测试结果示例

Fig. 11　Example of Cityscapes test results

4　结束语

本文提出了一种基于蛇模型的弱监督实例分割方法，用投影损失项和颜色成对损失项代替蛇模型原有的损失函数，实现了只使用边界框标注的数据集完成对实例分割网络的训练。初始轮廓是影响蛇模型效果的一个重要因素，经过对比实验，4 种初始轮廓变形方案中使用椭圆形初始轮廓的方案 D 结果最好。在 SBD 数据集和 Cityscapes 数据集中进行实验验证，证明了本文的方法适用于多种实例分割场景，相较于其他实例分割方法在精度和运行速度方面具有综合优势，且可以满足实时实例分割的要求。

参考文献

[1] Khoreva A, Benenson R, Hosang J, et al. Simple does it: Weakly supervised instance and semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 876-885.

[2] Ling H, Gao J, Kar A, et al. Fast interactive object annotation with curve-gcn[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2019: 5257-5266.

[3] Peng S, Jiang W, Pi H, et al. Deep snake for real-time instance segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 8533-8542.

[4] Cordts M, Omran M, Ramos S, et al. The cityscapes dataset for

semantic urban scene understanding[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 3213-3223.

[5] Hariharan B, Arbeláez P, Bourdev L, et al. Semantic contours from inverse detectors [C]//Proceedings of 2011 International Conference on Computer Vision. Piscataway, NJ: IEEE, 2011: 991-998.

[6] He K, Gkioxari G, Dollár P, et al. Mask R-CNN [C]// Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE, 2017: 2961-2969.

[7] 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, 2016, 39(6): 1137-1149.

[8] Tian Z, Shen C, Chen H, et al. FCOS: A simple and strong anchor-free object detector [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(4): 1922-1933.

[9] Kass M, Witkin A, Terzopoulos D. Snakes:Active contour models [J]. International Journal of Computer Vision, 1988, 1(4): 321-331.

[10] Tian Z, Shen C, Wang X, et al. Boxinst: High-performance instance segmentation with box annotations [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2021: 5443-5452.

[11] Duan K, Bai S, Xie L, et al. Centernet: Keypoint triplets for object detection[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ: IEEE, 2019: 6569-6578.

[12] Yu F, Wang D, Shelhamer E, et al. Deep layer aggregation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 2403-2412.

[13] Dai J, Qi H, Xiong Y, et al. Deformable convolutional networks [C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway, NJ: IEEE, 2017: 764-773.

[14] Jetley S, Sapienza M, Golodetz S, et al. Straight to shapes: Real-time detection of encoded shapes [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2017: 6550-6559.

[15] Xu W, Wang H, Qi F, et al. Explicit shape encoding for real-time instance segmentation [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway, NJ: IEEE, 2019: 5168-5177.

[16] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 8759-8768.