

Pointly-Supervised Instance Segmentation

Bowen Cheng

摘要

We propose an embarrassingly simple point annotation scheme to collect weak supervision for instance segmentation. In addition to bounding boxes, we collect binary labels for a set of points uniformly sampled inside each bounding box. We show that the existing instance segmentation models developed for full mask supervision can be seamlessly trained with point-based supervision collected via our scheme. Remarkably, Mask R-CNN trained on COCO, PASCAL VOC, Cityscapes, and LVIS with only 10 annotated random points per object achieves 94% - 98% of its fully-supervised performance, setting a strong baseline for weakly-supervised instance segmentation. The new point annotation scheme is approximately 5 times faster than annotating full object masks, making high-quality instance segmentation more accessible in practice.

Inspired by the point-based annotation form, we propose a modification to PointRend instance segmentation module. For each object, the new architecture, called Implicit PointRend, generates parameters for a function that makes the final point-level mask prediction. Implicit PointRend is more straightforward and uses a single point-level mask loss. Our experiments show that the new module is more suitable for the point-based supervision.

关键词：Instance Segmentation; Mask R-CNN; CoCo; PointRend

1 引言

在计算机视觉领域，实例分割是一个很重要的研究主题，在地理信息系统、医学影像、自动驾驶、机器人等领域有着很重要的应用技术支持作用，具有十分重要的研究意义。

2 相关工作

2.1 Region CNN 算法

RCNN (Regions with CNN features) 是在 2014 年提出的一种目标检测算法，RCNN 是将 CNN 方法应用到目标检测问题上的一个里程碑，借助 CNN 良好的特征提取和分类的性能，通过 REgionPropo 方法实现目标检测^[1]。

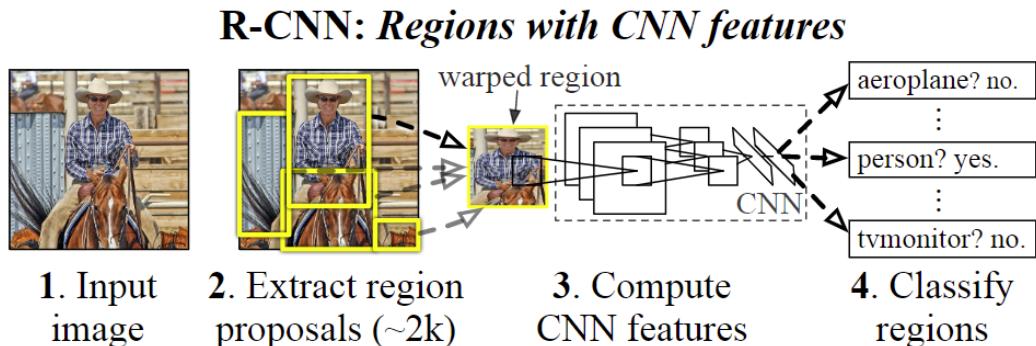


图 1: RCNN 算法流程

RCNN 算法分为 4 个步骤：

- 1、一张图像生成 1K~2K 个候选区域
- 2、对每个候选区域，使用深度网络提取特征
- 3、特征送入每一类的 SVM 分类器，判别是否属于该类
- 4、使用回归器精细修正候选框位置

虽然在当时 RCNN 算法在目标检测领域获得了很好的检测效果，但是还是存在诸多问题，如候选框选择算法耗时严重、重叠区域特征重复计算、分步骤进行，过程繁琐。

2.2 Fast-RCNN 算法

在了解 Fast-RCNN 之前先了解 SPP (Spatial Pyramid Pooling, 空间金字塔池化)

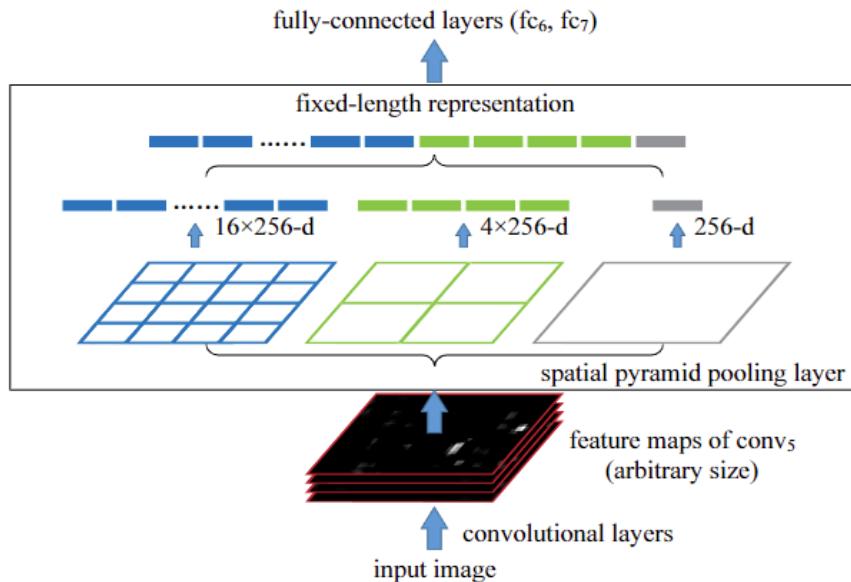


图 2: SPP 算法流程

SPP 就是将一个输入特征图得到多个大小不一样的输出特征图，然后将这些特征图进行 flatten (铺平，展成一维数组)，然后这些一维数组再进行拼接 (concatenate) 最后就可以将拼接得到的特征图送到 FC (全连接层) 去进行分类^[2]。

上图里面就是一个输入特征图分别得到一个 4*4 的输出特征图，一个 2*2 特征图以及一个 1*1 特征图（通道数都为 256），展平，拼接后得到 256 (16+4+1) *1 大小的特征图。不管你输入特征图 size 是多少，都会得到一个 4*4 的输出特征图，一个 2*2 特征图以及一个 1*1 特征图。

SPP 显著特点：1、不管输入尺寸是怎样，SPP 可以产生固定大小的输出 2、使用多个窗口 (pooling window)3、SPP 可以使用同一图像不同尺寸 (scale) 作为输入，得到同样长度的池化特征。

在 RCNN 之后，SPPNet 解决了重复卷积计算和固定输出尺寸两个问题，SPPNet 的主要贡献是在整张图像上计算全局特征图，然后对于特定的建议候选框，只需要在全局特征图上取出对应坐标的特征图就可以了。但 SPPNet 仍然存在一些弊端，如仍然需要将特征保存在磁盘中，速度还是很慢。

Fast RCNN 算法在 RCNN 和 SPPNet 的基础上进行了改进。其训练步骤实现了端到端，基于 VGG16 网络，其训练速度比 RCNN 快了 9 倍，测试速度快了 213 倍，在 PASCAL VOC2012 数据集达到了 68.4% 的准确率。相比 R-CNN，主要两处不同：(1) 最后一层卷积层后加了一个 ROI pooling layer；(2) 损失函数使用了多任务损失函数 (multi-task loss)，将边框回归直接加入到 CNN 网络中训练^[3]。

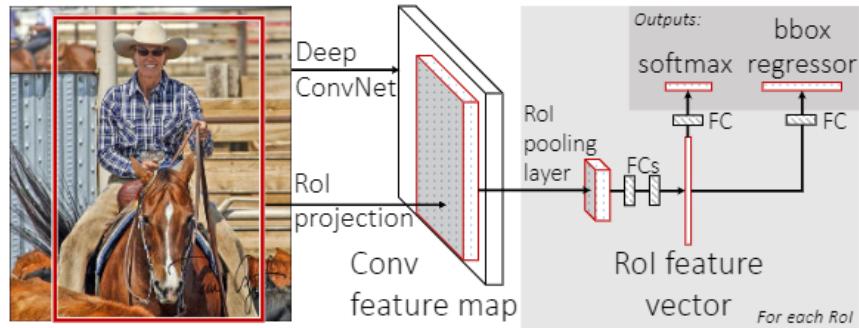


图 3: Fast-RCNN 算法流程

2.3 Faster-RCNN 算法

Faster RCNN 是第一个端到端和第一个近实时深度学习探测器。Faster-RCNN 的主要贡献是引入了 Region Proposal Network (RPN)，该网络使几乎无代价的 region proposal 成为可能。从 R-CNN 到 Faster RCNN，对象检测系统的大多数独立模块，例如提议检测，特征提取，边界框回归等，已逐步集成到统一的端到端学习框架中^[4]。

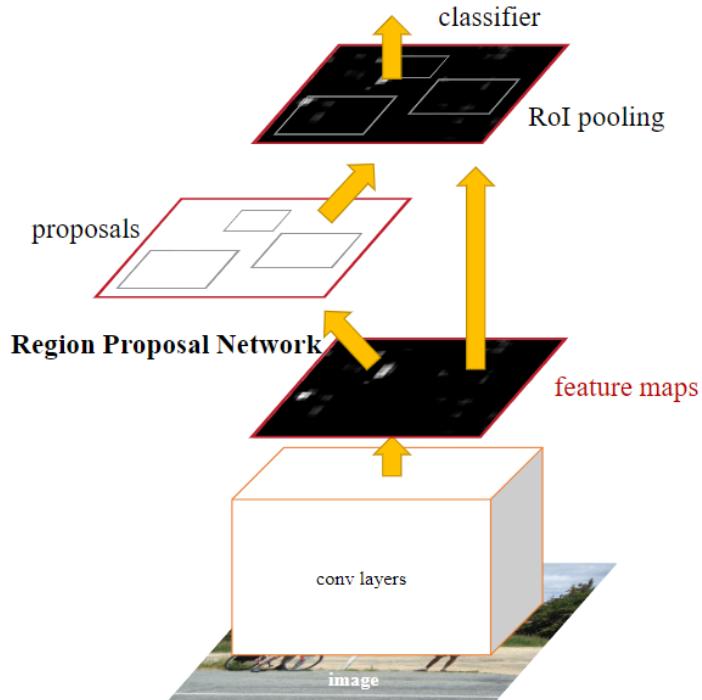


图 4: Faster-RCNN 算法流程

Fast-RCNN 存在测试时速度慢、训练时速度慢、训练所需空间大的问题。Faster RCNN 相比 FAST-RCNN，主要两处不同：

- (1) 使用 RPN(Region Proposal Network) 代替。原来的 Selective Search 方法产生建议窗口；
- (2) 产生建议窗口的 CNN 和目标检测的 CNN 共享。

复现工作中我也根据网上开源的代码进行了 Faster-RCNN 的复现，实验结果如下。

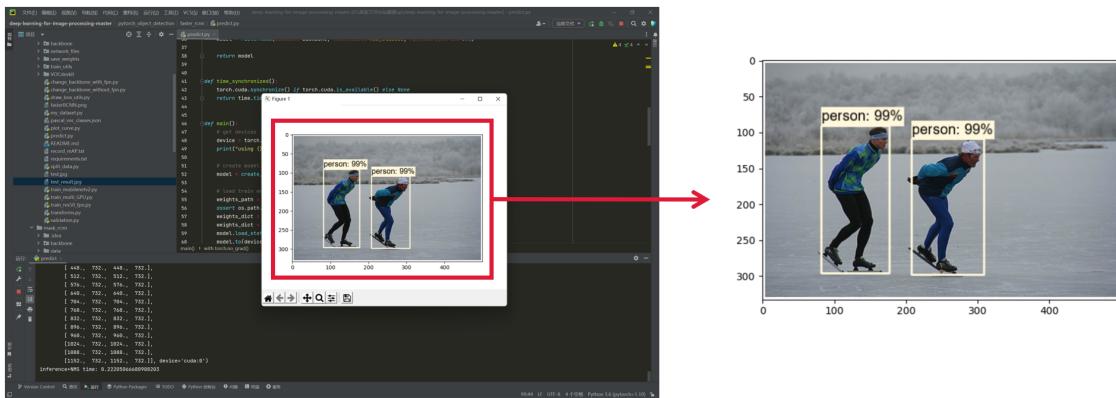


图 5: Faster-RCNN 算法流程

2.4 Mask-RCNN 算法

Mask R-CNN 是在 Faster R-CNN 的基础上加了一个用于预测目标分割 Mask 的分支（即可预测目标的 Bounding Boxes 信息、类别信息以及分割 Mask 信息）。

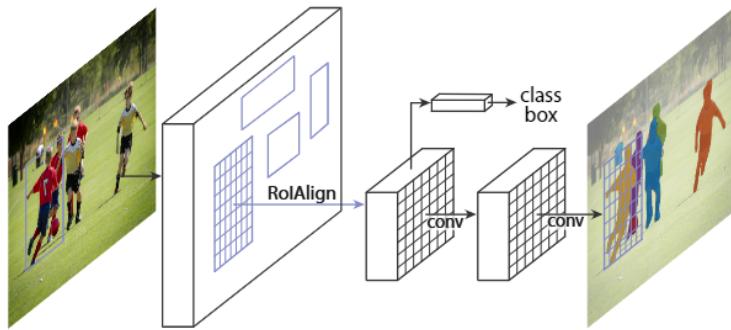


图 6: Mask-RCNN 算法流程

Mask R-CNN 的结构也很简单，就是在通过 RoIAlign（在原 Faster R-CNN 中是 RoIPool）得到的 RoI 基础上并行添加一个 Mask 分支（小型的 FCN）。见下图，之前 Faster R-CNN 是在 RoI 基础上接上一个 Fast R-CNN 检测头，即图中 class, box 分支，现在又并行了一个 Mask 分支^[5]。

复现工作中我也根据网上开源的代码进行了 Mask-RCNN 的复现，实验结果如下。

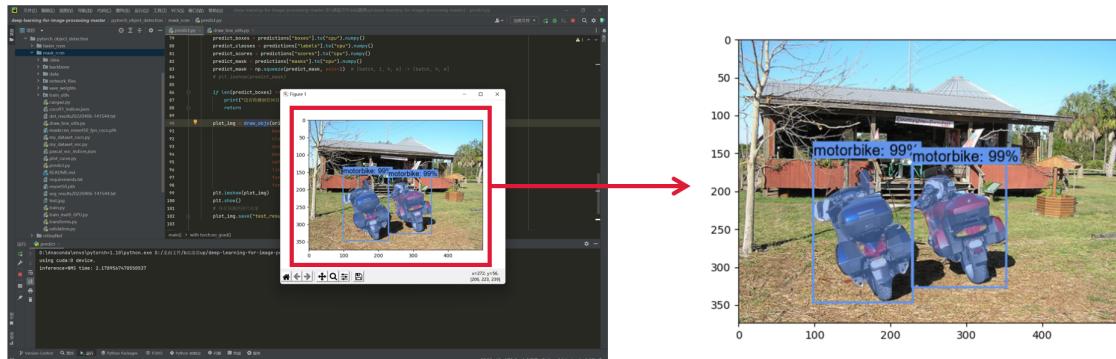


图 7: Mask-RCNN 算法流程

3 本文方法

3.1 基本方法概述

PointRend (Pointbased Rendering, 基于点的渲染)，用点的特征表示来解决图像分割问题^[6]。

图像分割步骤：

1、一个标准的分割网络（实心红色箭头）输入一张图像，使用轻量级的分割头，对每个检测到的对象（红色框）进行粗略的 mask 预测（例如 7×7 ）。

2、PointRend 选择一组点（红色点），并用一个小规模的多层次感知器（MLP）为每个点进行独立预测。MLP 使用在这些点处计算的插值特征（红色虚线箭头）进行预测。该特征包含细粒度特征和粗略预测特征。

3、迭代执行细分 mask 渲染算法，来细化预测 mask 的不确定区域。

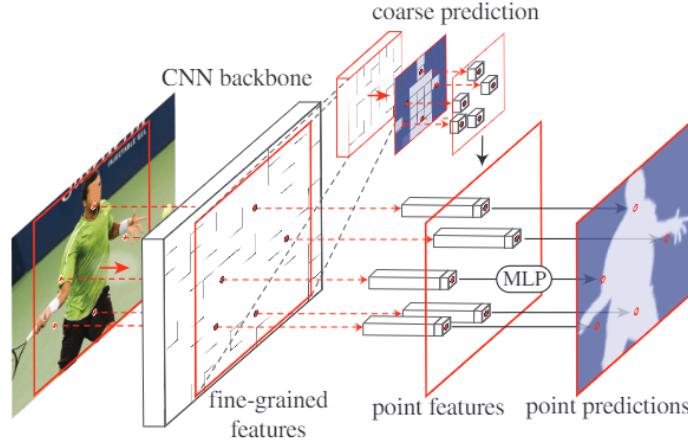


图 8: PointRend 算法流程

3.2 本文方法概述

Implicit PointRend 相比 PointRend 主要变化在于，前者先在图像上选好点，然后在选好的点上进行训练，后者则是在 feature map 上随机取点后再进行训练^[7]。

Implicit PointRend 比 PointRend 更简单：

- (1) 它在训练过程中不需要重要点采样
- (2) 它使用单点级掩码损失而不是两个掩码损失。Implicit PointRend 可以直接使用选好的点进行训练，而无需任何中间预测插值步骤。实验表明，新模块在点监督方面优于 PointRend。

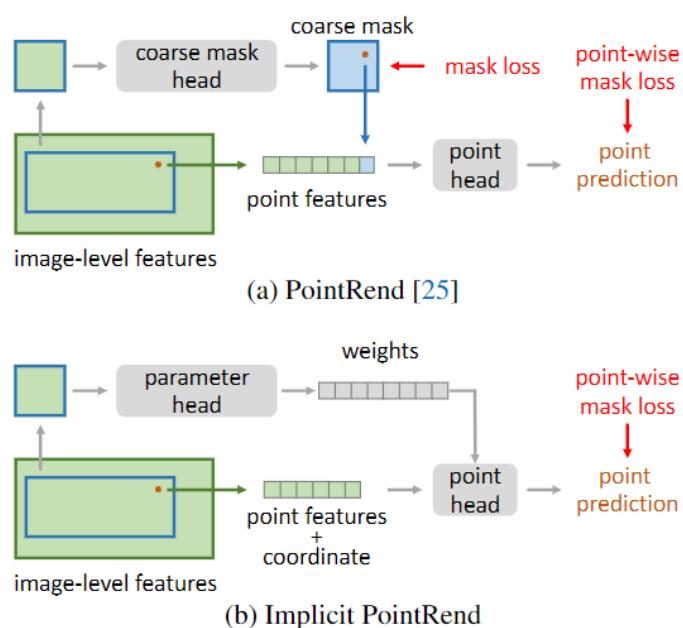


图 9: Implicit PointRend 和 PointRend 模型对比

3.3 损失函数定义

点头输出部分采用二元交叉熵 (binary cross entropy)，由于 i 互不干扰，该损失函数一般用于多分类问题，输出部分采用该函数可以更好的针对类别对预测参数进行修正。

$$\text{Loss} = -\frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log (1 - \hat{y}_i)$$

点头的预测参数上使用 L2loss，以避免预测参数变得无界。这种损失起到了权重衰减的作用，否则动态参数就不存在权重衰减。

$$loss(x, y) = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

4 复现细节

4.1 复现工作细节

由于之前对该方向没有了解，研究方向与该方向不同，复现该篇论文的过程中学习了 RCNN、Fast-RCNN、Faster-RCNN 和 Mask-RCNN，在此过程中通过网上的开源代码复现了 Faster-RCNN 和 Mask-RCNN，并得出实验结果，并用 ide 进行单步跟踪细致的学习了 Faster-RCNN 模型。

完成上述工作后我根据 Facebook 目标检测大模型 detectron2 源代码库再次复现了 Faster-RCNN 和 Mask-RCNN 模型，在此基础上对本文的方法进行复现工作，并将之前的数据集由 PASCAL VOC 换成了 COCO。

由于本文的方法基于 PointRend，在复现本文之前我又了解了 PointRend 并查看了该篇论文，随后在 detectron2 上复现了本文方法。本文的方法在标注和模型上都对 PointRend 进行了更新升级，所以我在新模型上进行了两个实验，第一个是在旧标注上的实验，第二个是在新标注上的实验。

4.2 实验环境搭建

在 github 上下载 Facebook 目标检测大模型 detectron2 源代码库 detectron2，根据源代码库搭建本文目标检测模型。

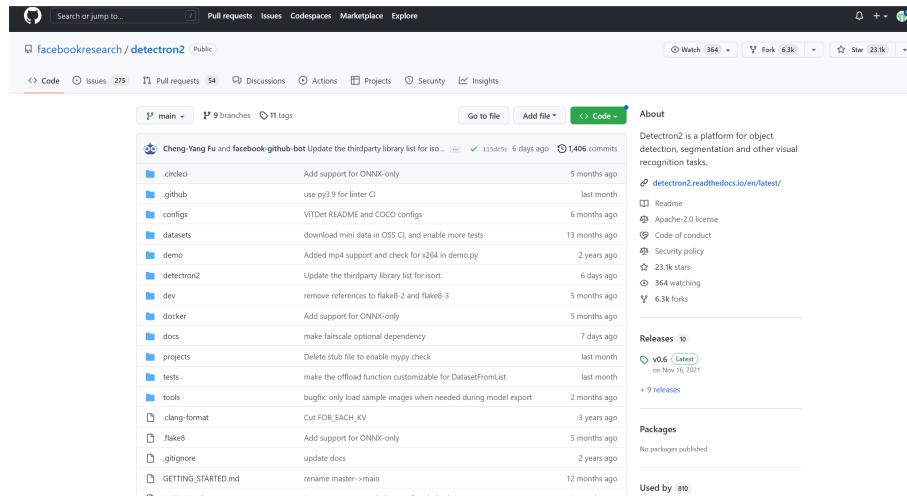


图 10: detectron2 源代码库

4.3 创新点

由于复现本文需要了解该方向的模型变迁和原理，此过程消耗大量时间，最后来不及创新并修改论文。在复现本文的过程中发现了可创新点，因为在论文模型中点头的预测参数上使用 L2loss，虽然相比 smooth L1loss 有更快的运算时间，但是该损失不是最优办法。在修改该损失的过程中发现 detectron2 大模型对该部分封装程度高，使得原有 L2loss 中的 y 和 x 无法拆分，同时由于时间原因导致修改失败。

5 实验结果分析

本文模型在旧标注方案上的复现结果如下图。

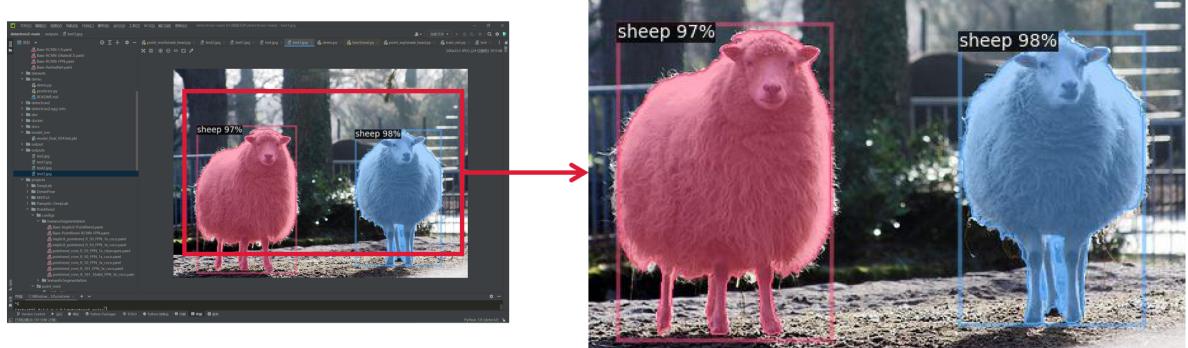


图 11: Implicit PointRend 在旧标注上的实验结果

本文模型在新标注方案上的复现结果如下图。

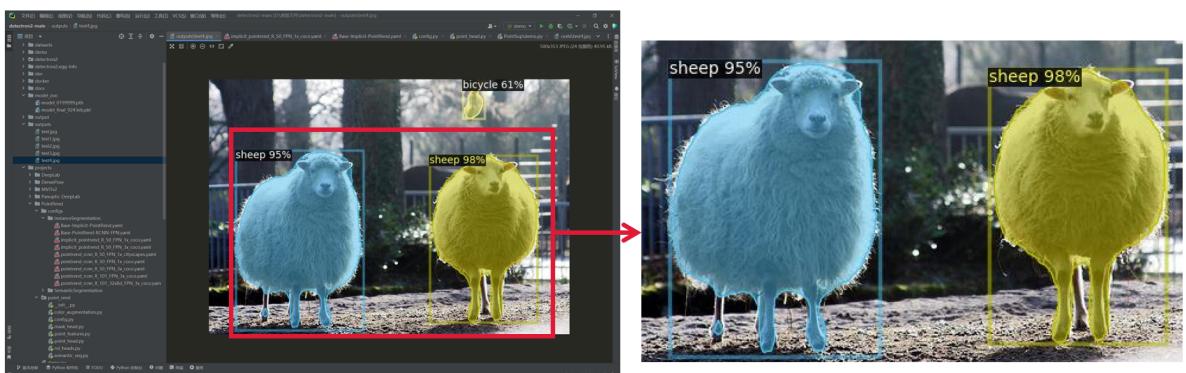


图 12: Implicit PointRend 在新标注上的实验结果

6 总结与展望

本文模型对 PointRend 模型进行了改进并引入了新的重点标注方案，虽然相比原有随机取点的 PointRend 方案在训练速度有了很大的提升，但是边缘的精度却有所下降。而新模型在原有的旧标注方案边缘精度上表现更好，但是训练速度上缺 da 大打折扣。结合这两点我们可以在新模型上采用更进一步的标注方法，增加新模型在物体边缘的目标检测精度，寻找取点的数量和训练时间之间的平衡点。

参考文献

- [1] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

- [2] HE K, ZHANG X, REN S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904-1916.
- [3] GIRSHICK R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [4] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
- [5] HE K, GKIOXARI G, DOLLÁR P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.
- [6] KIRILLOV A, WU Y, HE K, et al. Pointrend: Image segmentation as rendering[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9799-9808.
- [7] CHENG B, PARKHI O, KIRILLOV A. Pointly-supervised instance segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 2617-2626.