



在VOC数据集上训练并测试模型 Mask R-CNN 和 Sparse R-CNN

陈杨 大数据学院 22307140022

数据预处理(格式转换)

将VOC数据集转换为coco格式进行训练，使用框架中自带的将voc数据格式转化为coco格式的函数，存在问题，直接将bbox作为segmentation，无法处理实例分割所需要的segmentation标注，而只能用于目标检测，在voc数据集中segmentation object用png图片表示，而在cooc数据集中由**polygen** (**iscrowd**=0, 实例为单一对象) 和**RLE** (**iscrowd**=1, 同一类别多个实例重叠，对于二进制掩膜0-1矩阵进行编码压缩) 格式的segmentation标注来表示，也即无法获取mask-rcnn模型训练需要的mask，因此需要自己在原框架的基础上实现数据格式转换。另外，并不是每一张图片都有对应png segmentation object，因此在实例分割任务中只选择了有segmentation object的图片进行训练，而在目标检测任务中选择了更多图片进行训练，两个任务对应的json文件分别位于**object detection**和**segmentation**文件夹下。

参考原先的数据格式转化脚本，自主实现了 `pascal_voc_seg.py`，位于 `tools/dataset_converters` 目录下，确保能将png mask转化为需要的标注形式。此外，将数据集按照官方文档的要求进行摆放，这里的annotation有两种类型，一种是目标检测对应的，一种是实例分割对应的（两个任务使用的数据集不完全相同）。

```
mmdetection/
├── mmdet/
├── tools/
├── configs/
└── data/
    ├── coco/
    │   ├── annotations
    │   ├── train2017
    │   ├── val2017
    └── VOCdevkit/
        ├── VOC2007
        └── VOC2012
```

训练设置

详细配置文件位于work_dirs目录下

一、基础训练配置对比

参数项	Mask R-CNN	Sparse R-CNN
Backbone	ResNet-50 (冻结stage1)	ResNet-50 (冻结stage1)
预训练源	torchvision官方权重	torchvision官方权重
优化器	SGD (momentum=0.9)	AdamW (带梯度裁剪, max_norm=1)
初始学习率	0.02	2.5e-5
权重衰减	0.0001	0.0001
训练周期	12 epochs	12 epochs
学习率调度策略	线性预热500iter + MultiStepLR (8/11 epoch时下降0.1x)	同左
批量大小	训练: 2/GPU, 测试: 1/GPU	训练: 2/GPU, 测试: 1/GPU

二、核心结构对比

1. neck结构

```

# Mask R-CNN的FPN配置
neck=dict(
    in_channels=[256,512,1024,2048],      # ResNet各stage输出通道
    num_outs=5,                          # 生成5层特征金字塔
    out_channels=256                      # 统一输出通道数
)
# Sparse R-CNN的改进FPN
neck=dict(
    add_extra_convs='on_input',           # 在输入特征上添加额外卷积
    num_outs=4,                          # 生成4层特征图
    start_level=0                        # 从stage0开始输出
)

```

2. 检测头

组件	Mask R-CNN	Sparse R-CNN
RPN	传统锚框生成器	动态生成100个提案
RoI提取	RoIAlign (7x7)	RoIAlign (7x7)+动态卷积
分类头	2个全连接层	6个级联动态头
回归头	4个坐标回归器	GIoU损失+坐标精修
后处理	需要NMS	无需NMS

三、训练策略

1. 数据增强

```

# 共同使用的增强策略
train_pipeline=[

    dict(type='LoadImageFromFile'),      # 图像加载
    dict(type='LoadAnnotations'),        # 标注加载
    dict(type='Resize',                # 尺寸调整
        scale=(1333,800), keep_ratio=True),
    dict(type='RandomFlip', prob=0.5),   # 随机翻转
    dict(type='PackDetInputs')          # 数据打包
]

```

2. 损失函数

损失类型	Mask R-CNN	Sparse R-CNN
分类损失	CrossEntropyLoss (权重1.0)	FocalLoss ($\alpha=0.25, \gamma=2.0$)
回归损失	L1 Loss (权重1.0)	L1 Loss + GIoU Loss (总权重7.0)
分割损失	Mask CrossEntropy (权重1.0)	无

task1 目标检测

数据集划分

由于显存的限制，选择了VOC07和12的训练集作为总的训练集（共8218张图片），VOC07的验证集作为总的验证集（共2510张图片），VOC07的测试集作为总的测试集（共4952张图片）

task2 实例分割

数据集划分

选择选择了VOC07和12的训练集作为总的训练集（共209+1464张图片），VOC07和12的验证集作为总的验证集（共213+1449张图片），VOC07的测试集作为总的测试集（共210张图片）

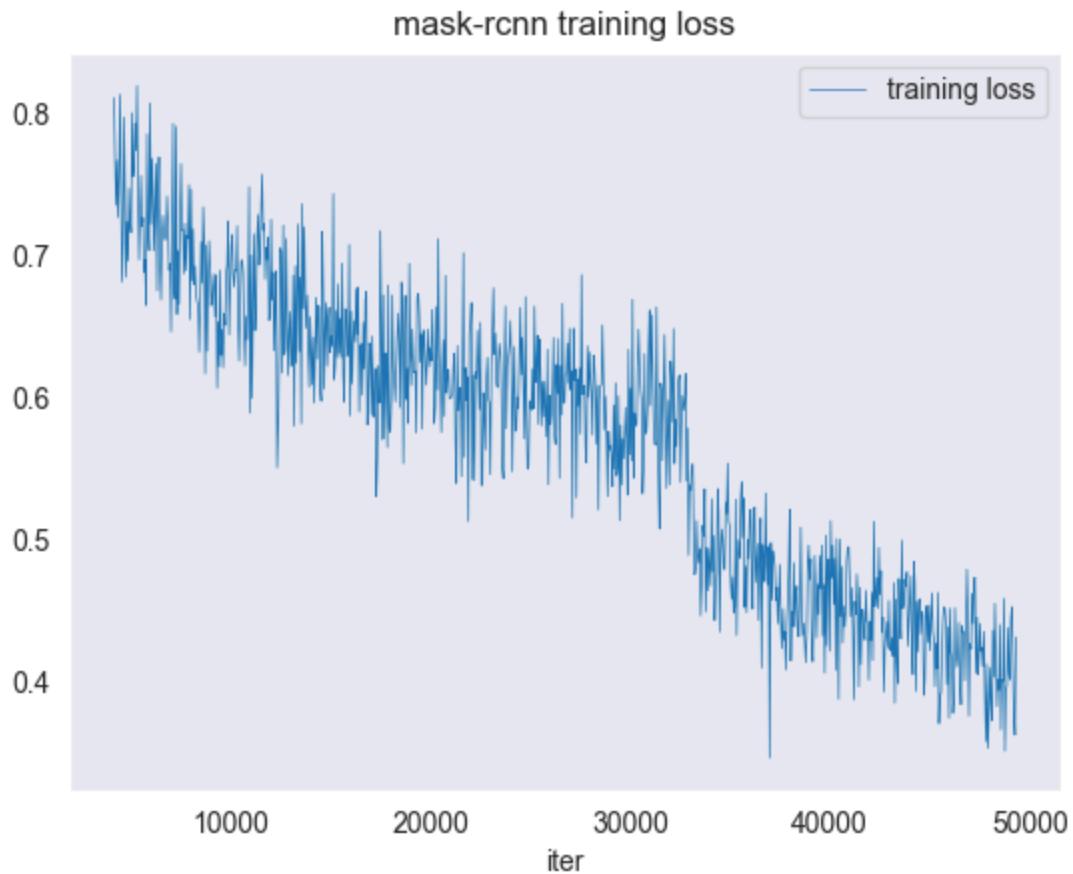
实验结果

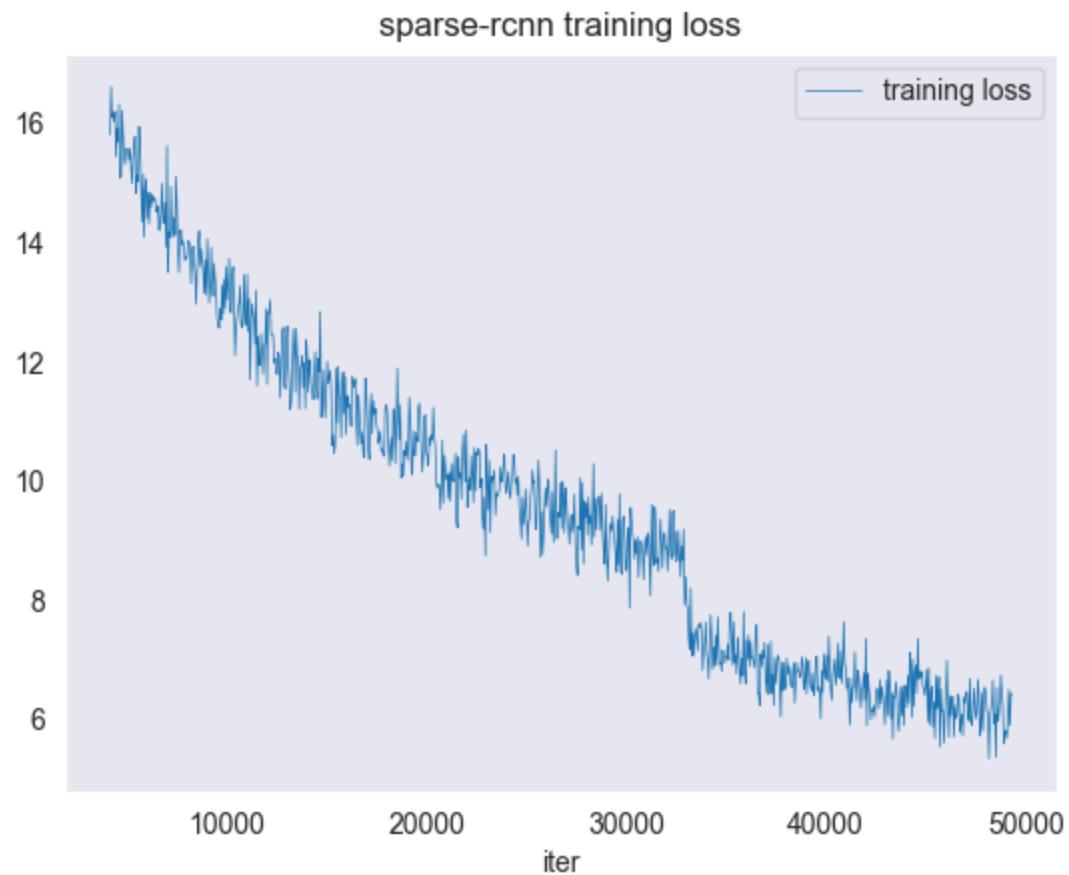
目标检测

这里loss以及mAP曲线的绘制是分析log得到的，由于训练时忘记启用Tensorboard，而且训练时间过长，因此这里是分析log得到的曲线

超参数设置

训练集loss曲线



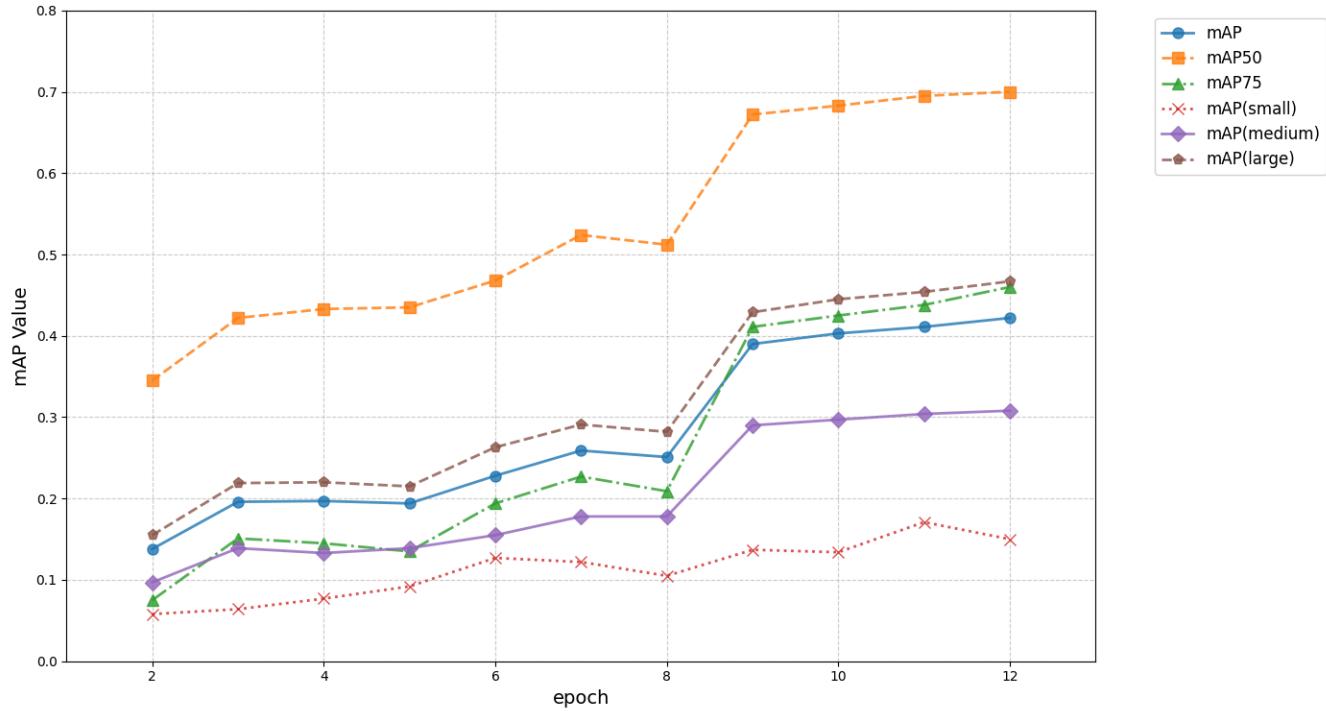


中间loss骤降对应的是学习率减少

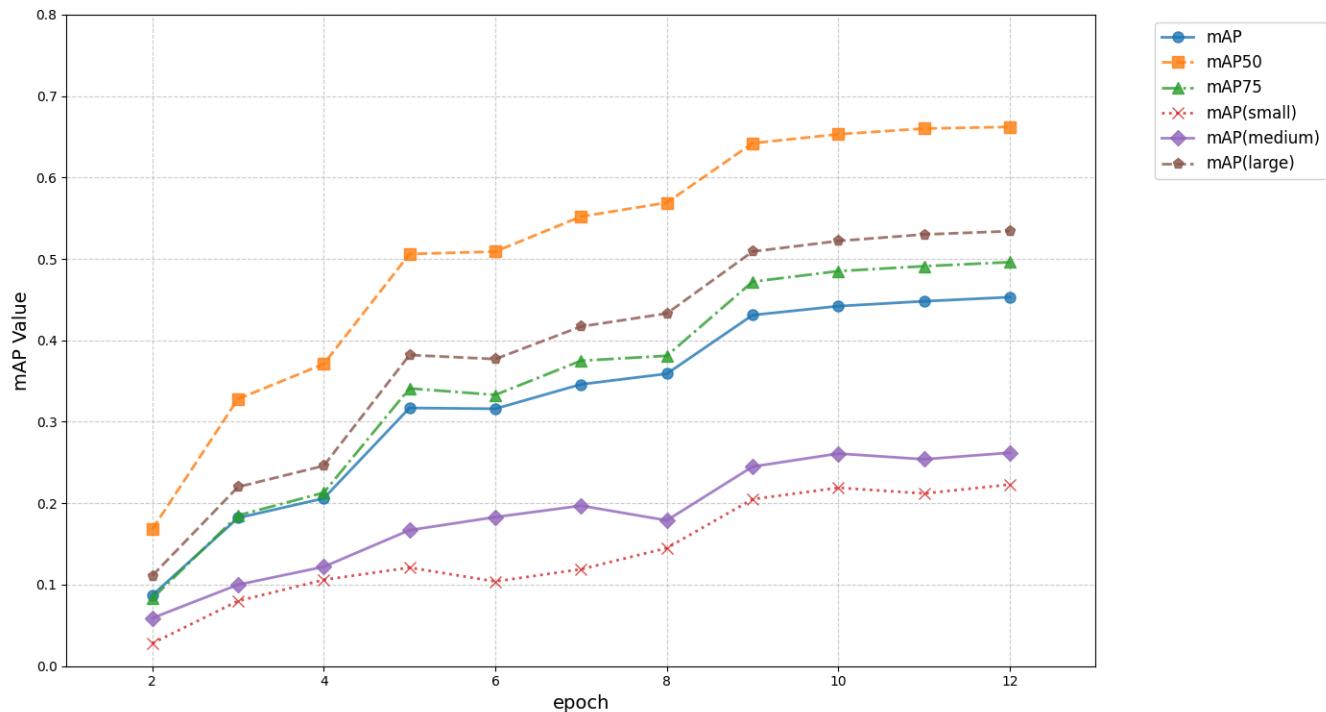
验证集mAP曲线

横轴为epoch数，纵轴为mAP值，可以看到基本都在处于上升阶段，未出现过拟合的情况，可以继续增加epoch数，使得模型达到更好的效果。

Mask-rcnn Evaluation Metrics Progression



Sparse-rcnn Evaluation Metrics Progression



测试集各项指标

模型	mAP	mAP@50	mAP@75	mAP_s	mAP_m	mAP_l	data_time	time
Mask-RCNN	0.424	0.713	0.448	0.225	0.302	0.461	0.1756	0.862
Sparse-RCNN	0.458	0.683	0.498	0.145	0.262	0.532	0.0746	0.190

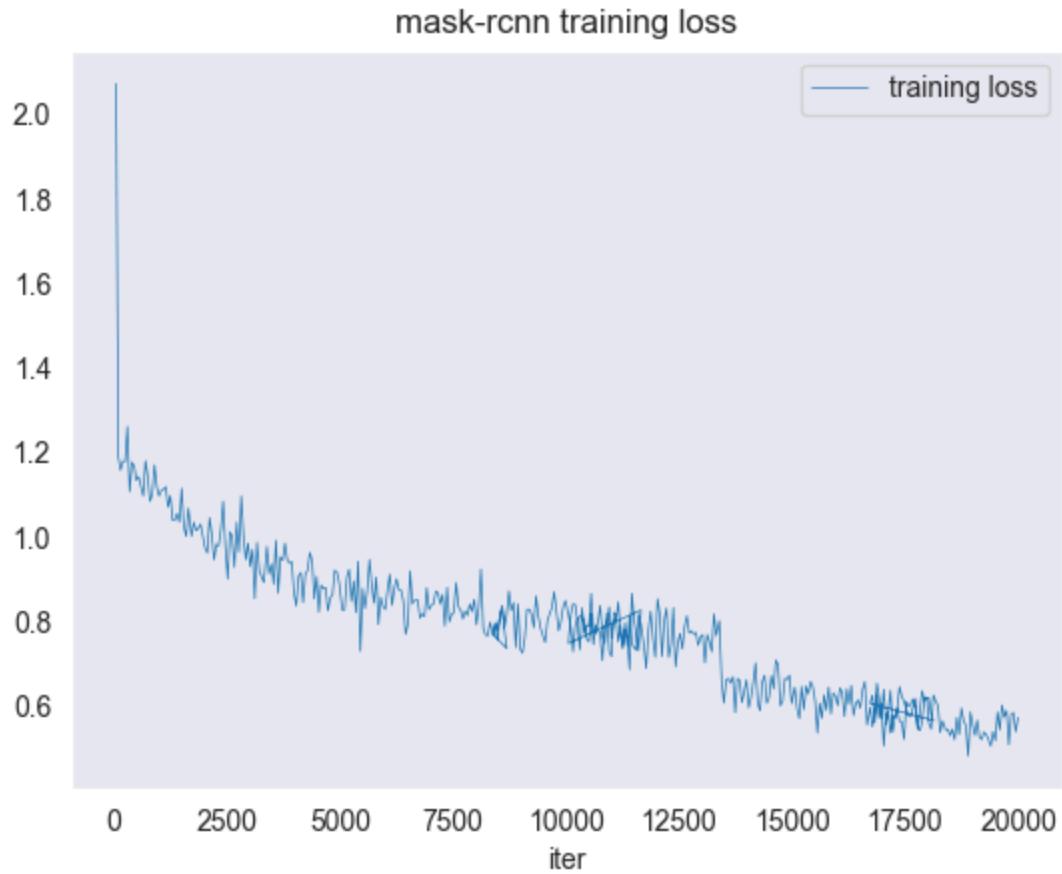
比较结果，可以发现：

- Sparse R-CNN 在整体精度 (mAP) 和严格定位 (mAP75) 上显著领先，但 Mask R-CNN 在低阈值 (mAP50) 下表现更好
- Mask R-CNN 在中小目标检测上优势显著，但 Sparse R-CNN 在大目标上表现更优。
- Sparse R-CNN 的端到端效率更高，适合实时性要求高的场景。

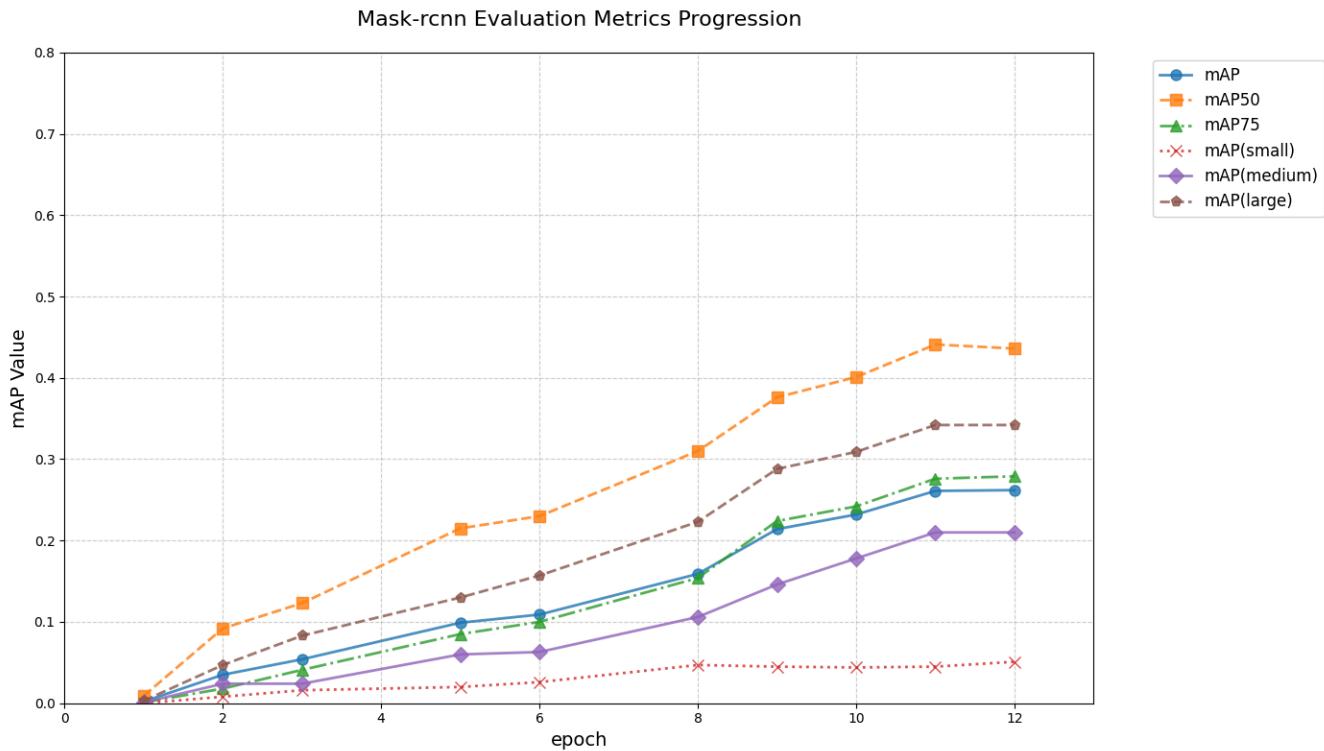
实例分割

框架中的sparse-rcnn模型仅支持目标检测，因此实例分割任务中只训练了mask-rcnn模型。

训练集loss曲线



验证集mAP曲线



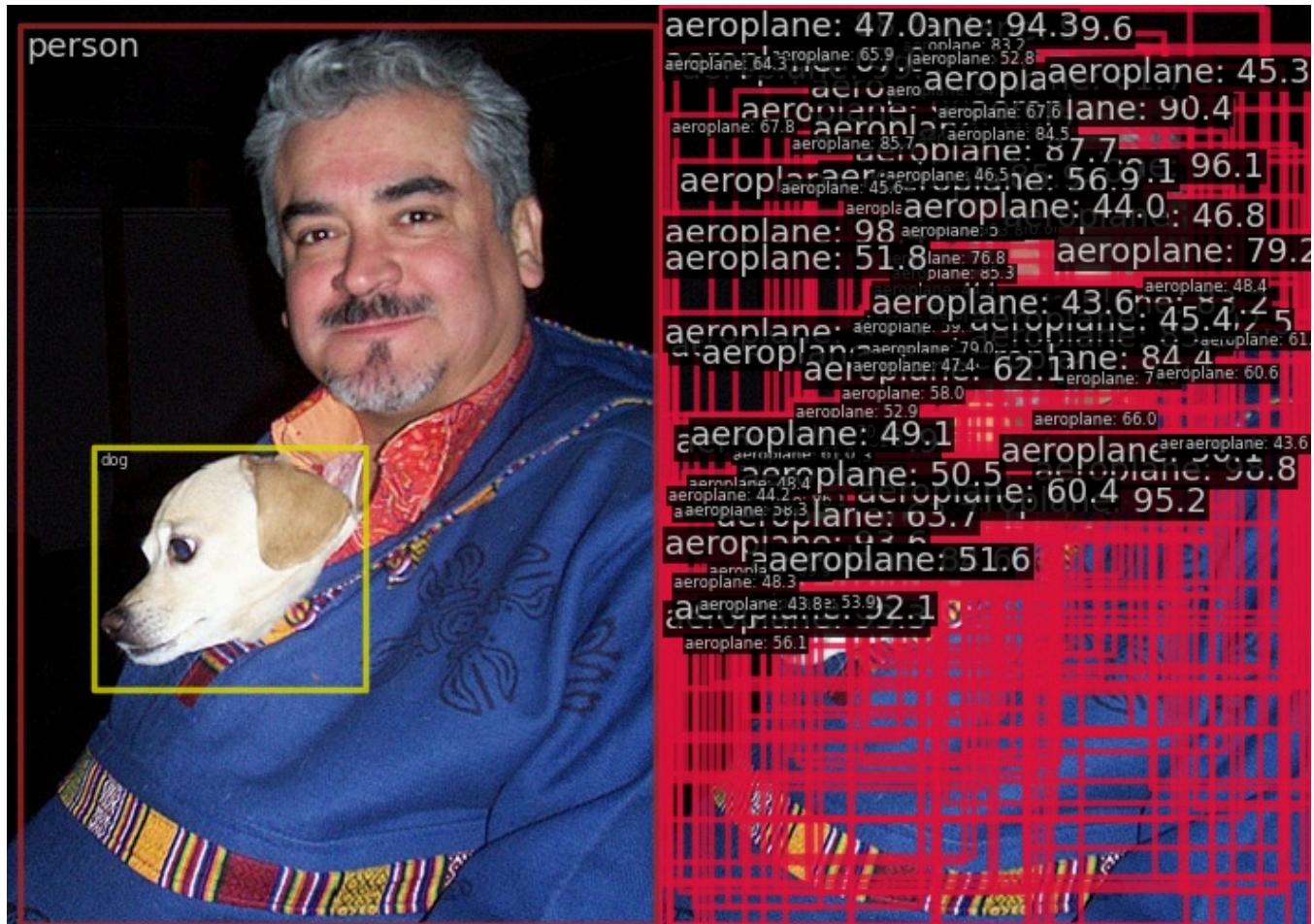
测试集各项指标

bbox_mAP	bbox_mAP_50	bbox_mAP_75	bbox_mAP_s	bbox_mAP_m	bbox_mAP_l
0.495	0.816	0.574	0.435	0.598	0.486

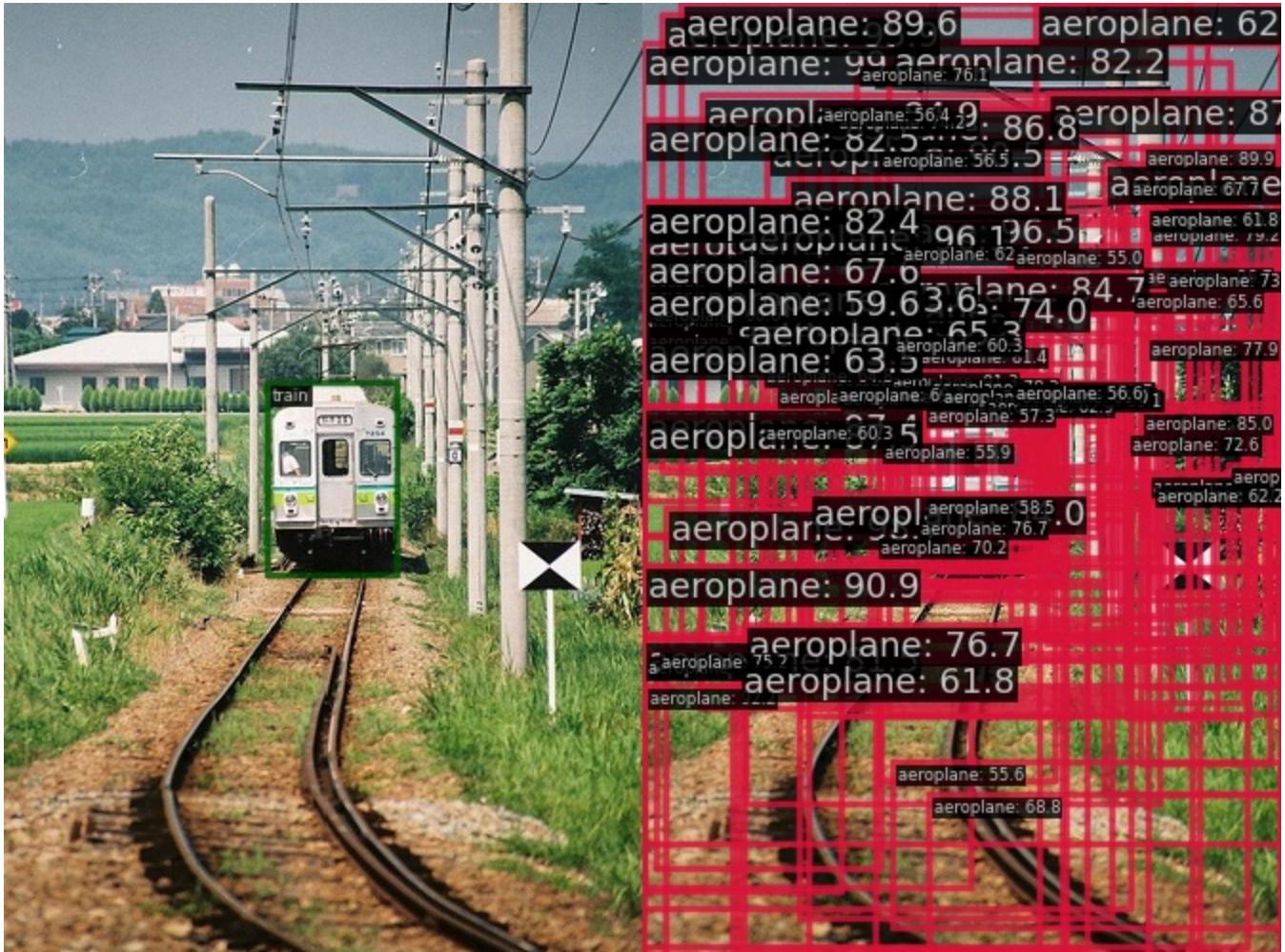
由于数据量不足，导致指标较低，后续可以增加数据量进行训练。值得注意的是，发现这里使用较小数据量的情况下，反而目标检测的性能提升了，说明之前对于目标检测模型的训练还未达到最佳，可以继续增大epoch数以达到更好的效果。

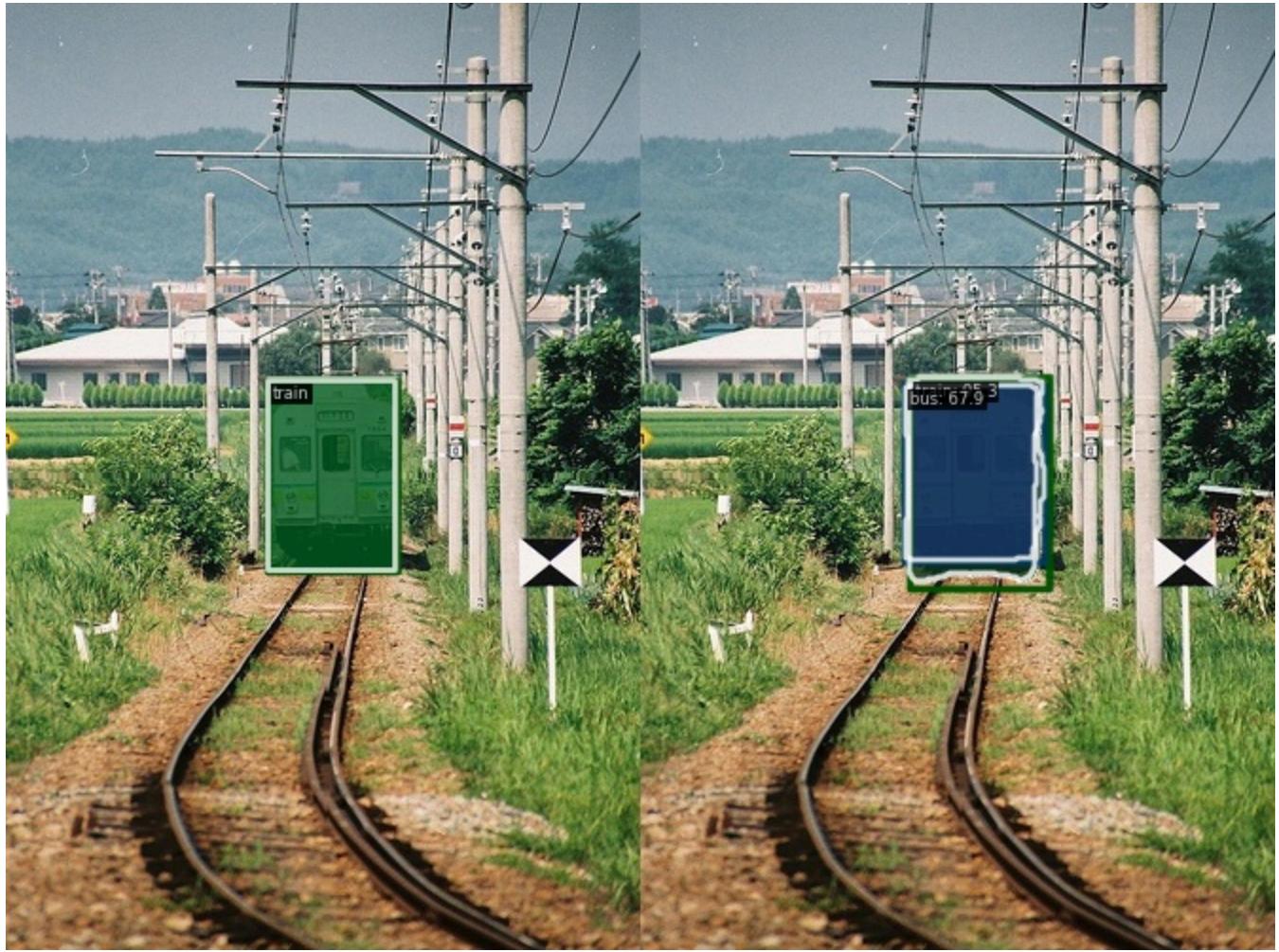
可视化结果

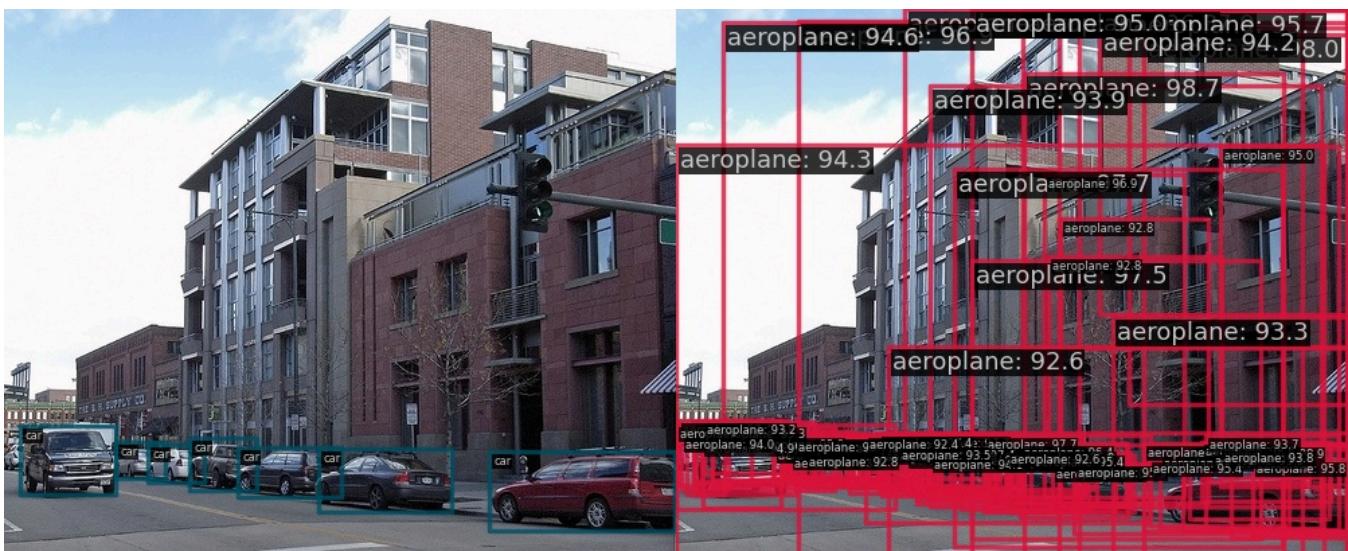
mask-rcnn第一阶段proposal box与最终预测结果对比(忽略proposal box上带的标注)











mask-rcnn与sparse-rcnn目标检测在测试集上的结果对比

上面为mask-rcnn可视化结果，下面为sparse-rcnn可视化结果,这里mask-rcnn中的mask是没有意义的，在处理时segmentation字段都是通过检测框来得到的。

测试集图片000004

在这组图片中，发现mask-rcnn检测出了更多物体，这是由于mask-rcnn依赖于rpn产生的候选框，候选框数量更多，覆盖到的物体也就更多

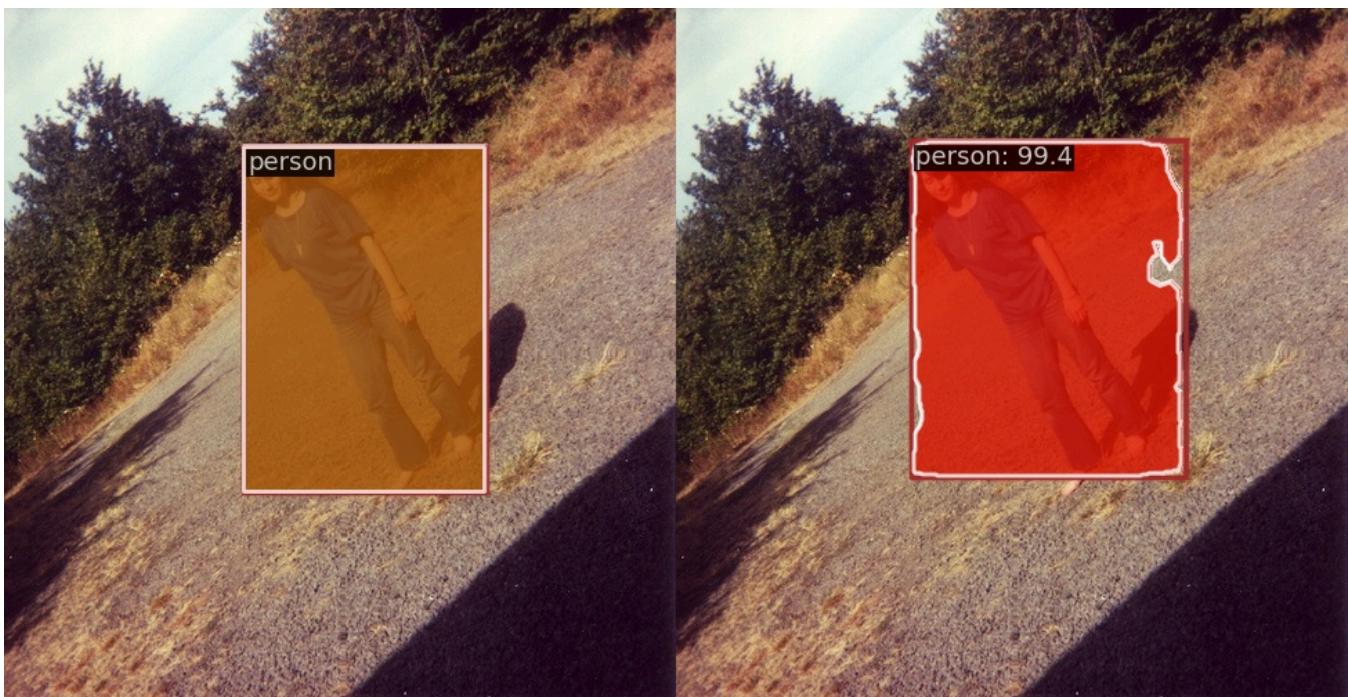


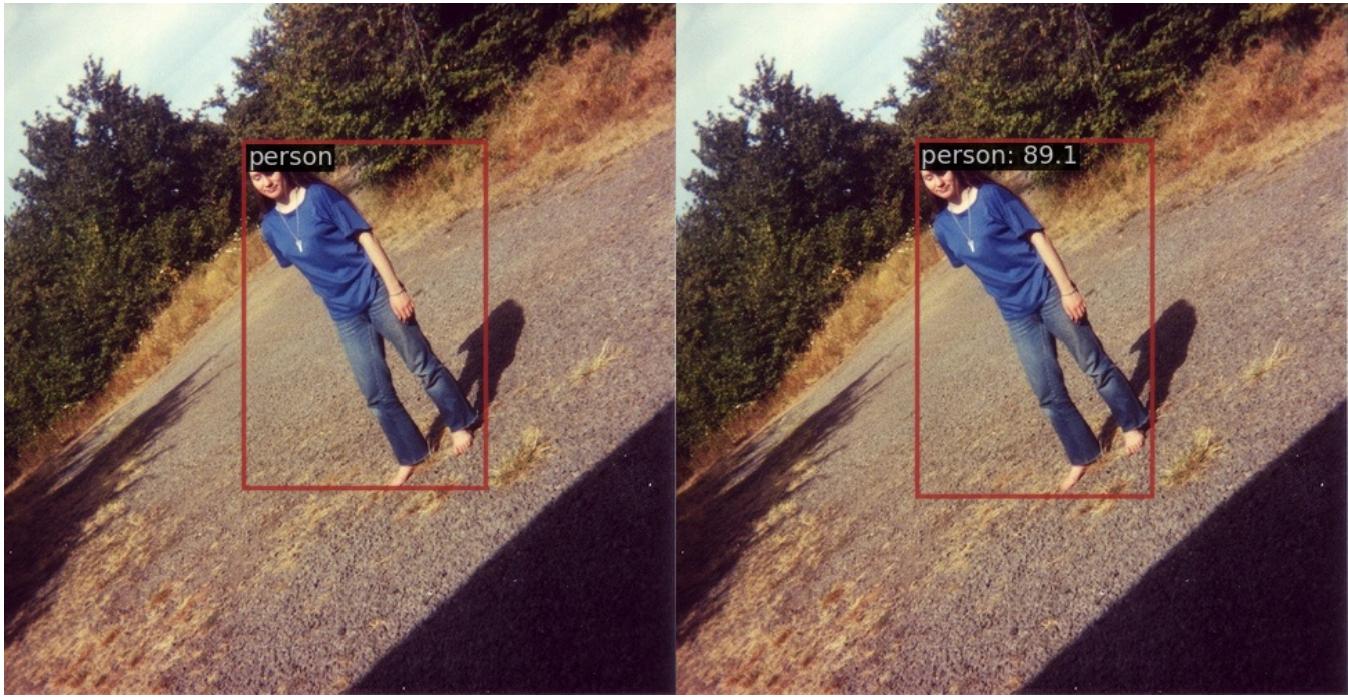
测试集图片000010



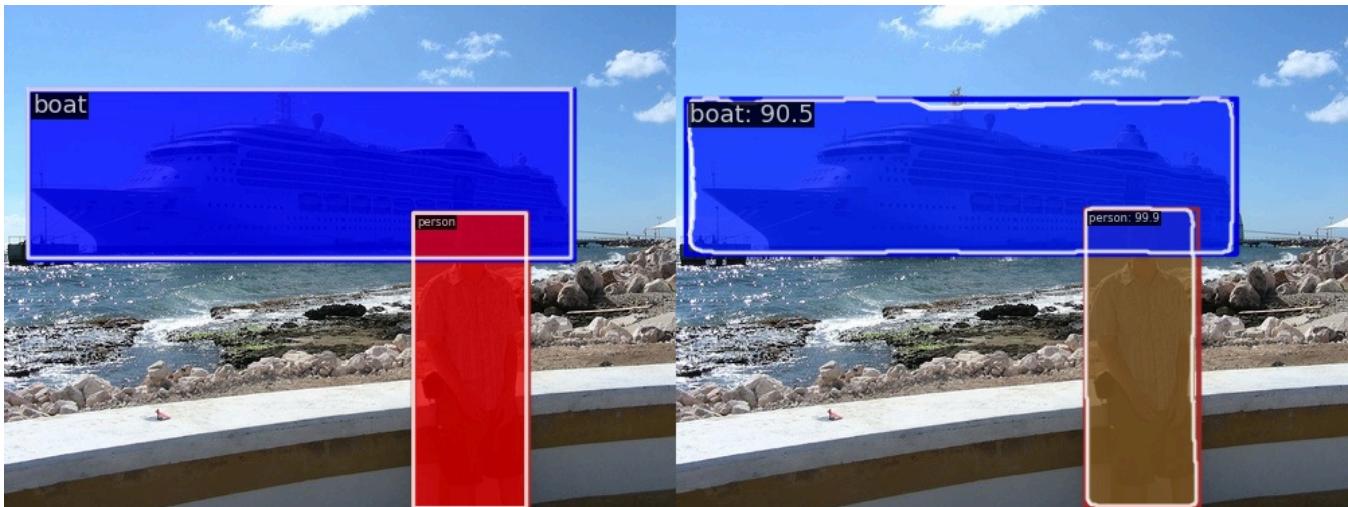


测试集图片000027





测试集图片000069





mask-rcnn实例分割可视化结果

由于训练集数量不足，仅有1673张图片，并且部分png mask转化的质量不高，实例分割的效果不是很好，下面挑选了一些较好的结果作为展示

测试集图片000243



测试集图片001704



测试集图片003131



测试集图片003286



自寻图片可视化结果

这部分选择了coco验证集中的图片

mask-rcnn与sparse-rcnn目标检测结果对比

图片1

两个模型都较好地检测出了目标





图片2

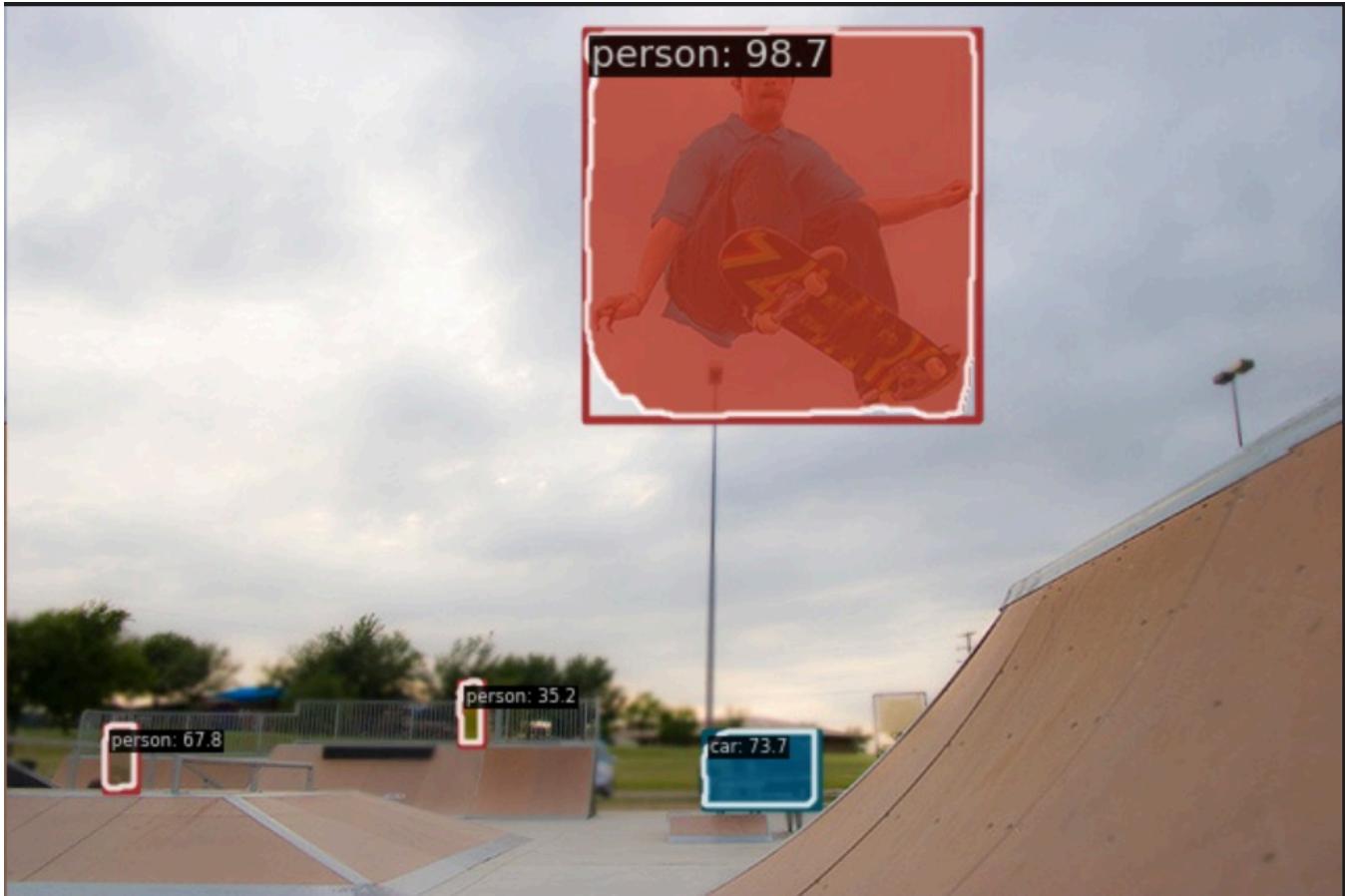
都较好地检测出了目标，相对来说，sparse-rcnn的检测框略微精确一些

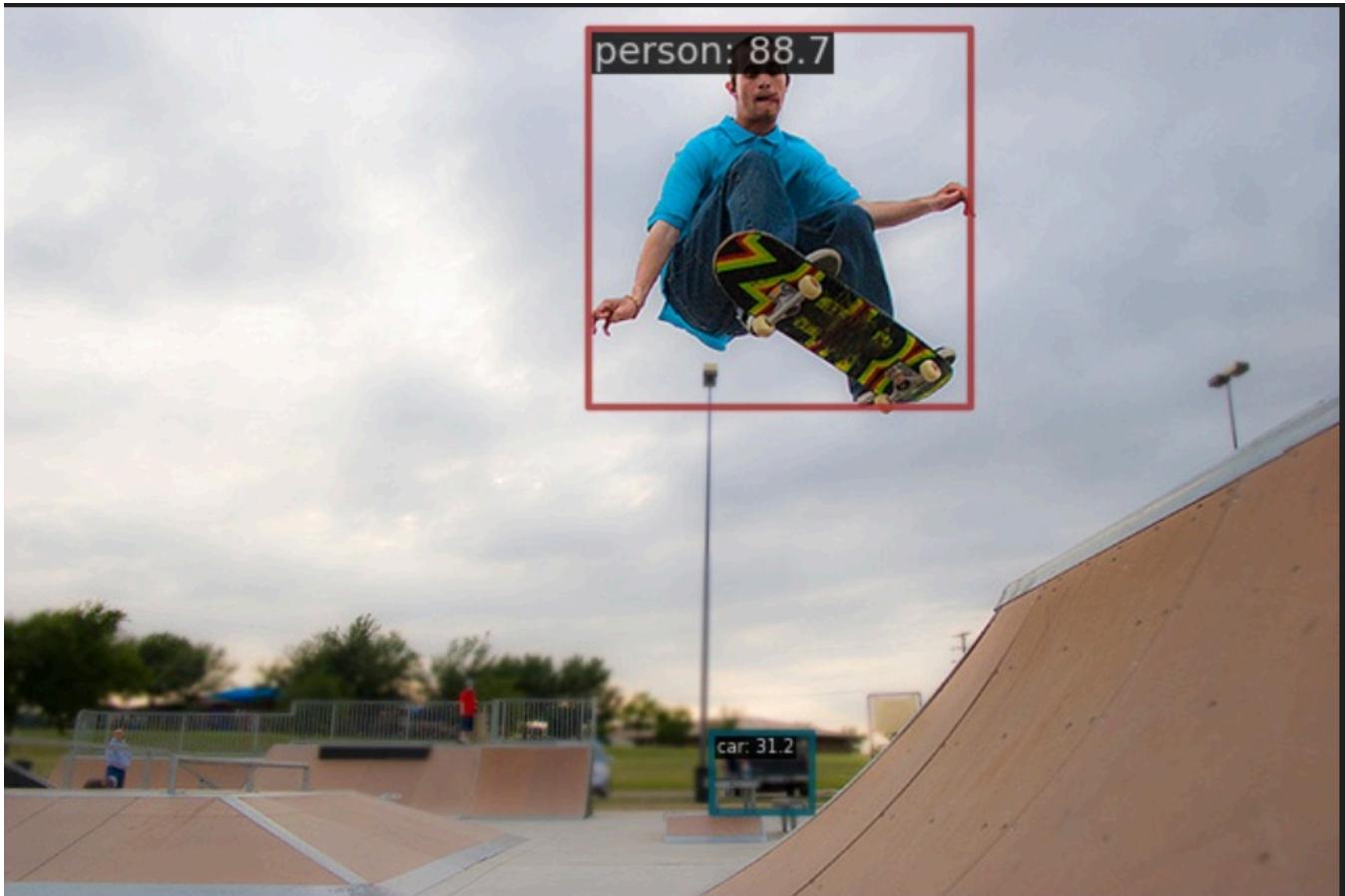




图片3

mask-rcnn相比于sparse-rcnn检测出了更多的目标，这是由于mask-rcnn依赖region proposal，RPN生成的候选框可以覆盖更多目标，从这张图片来看，mask-rcnn的效果要好于sparse-rcnn

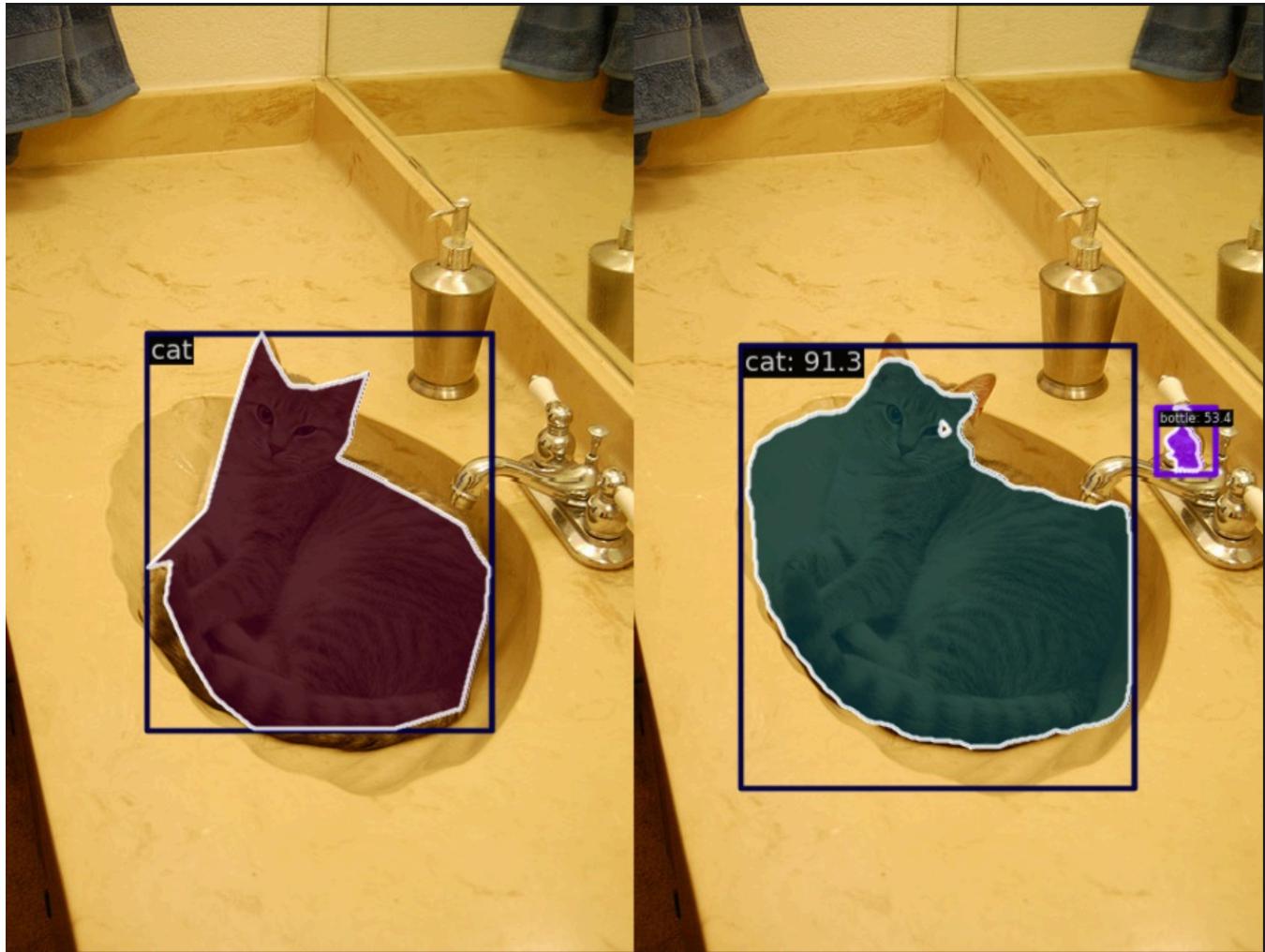




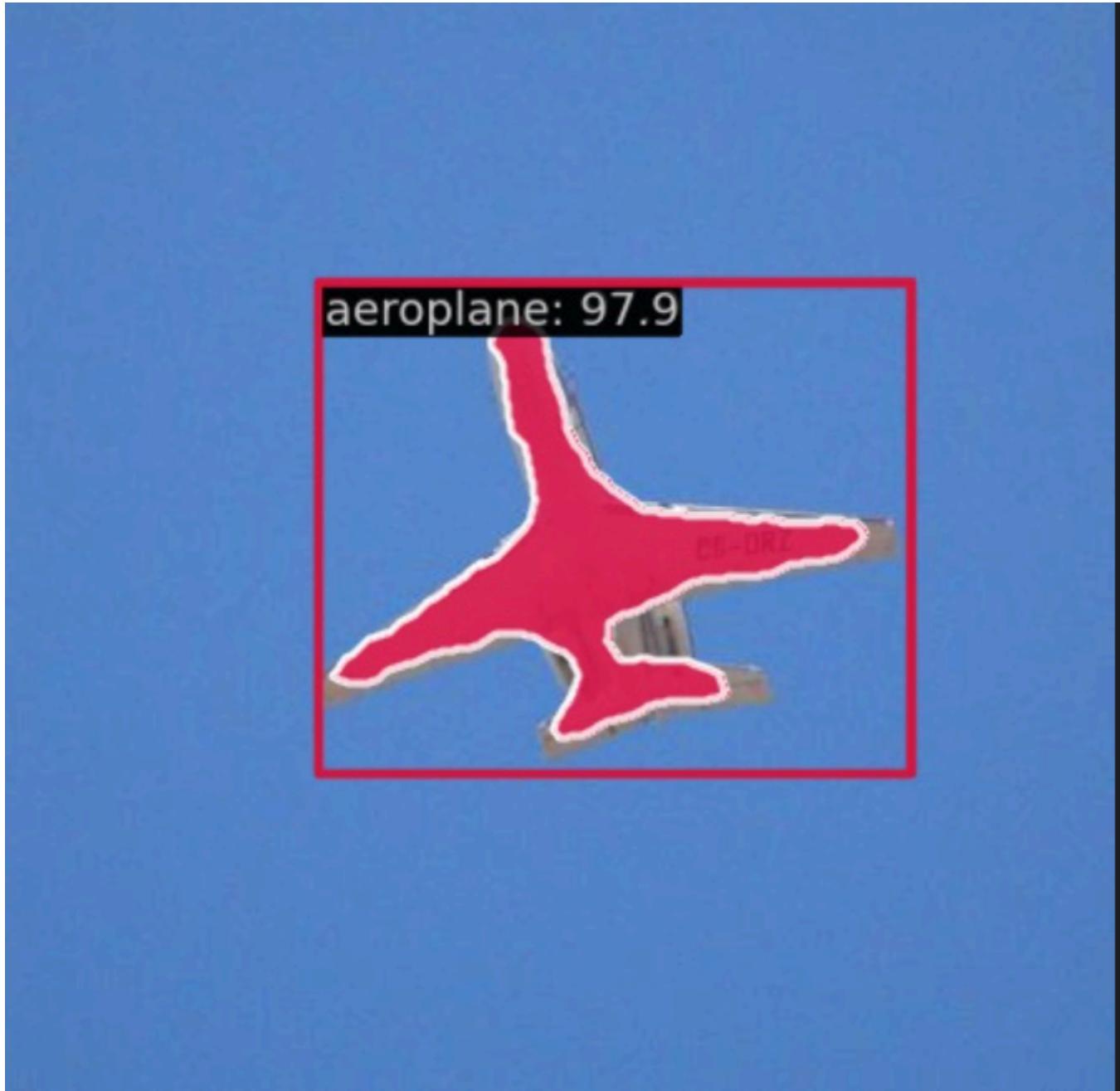
mask-rcnn实例分割结果

由于训练数据不足，整体效果不佳。

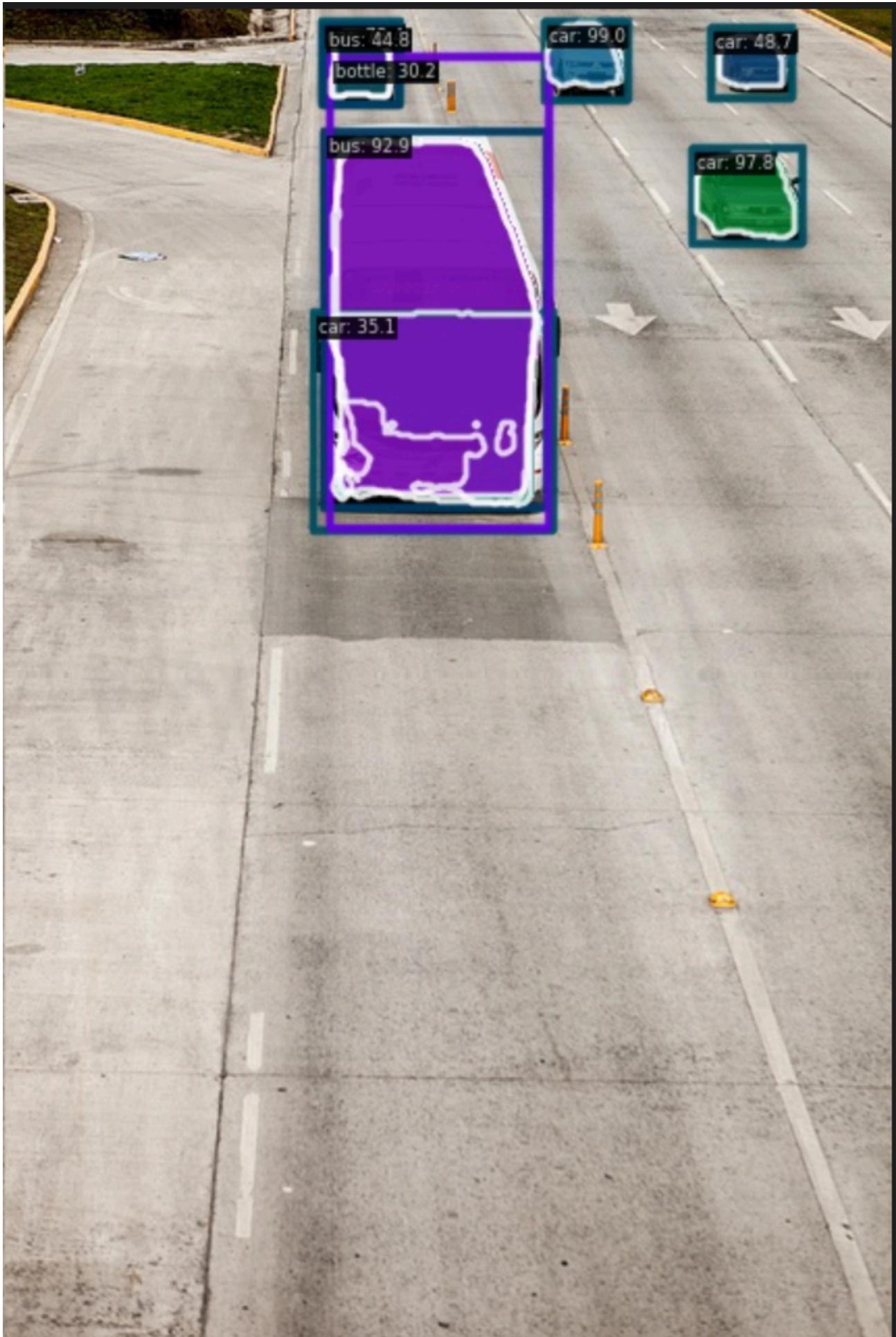
图片1



图片2



图片3



实验结论

本实验中，通过学习mmdetection框架，使用了mask-rcnn模型来进行目标检测和实例分割任务，使用sparse-rcnn模型来进行目标检测任务，并且通过可视化来展示模型预测的结果。