Mask R-CNN

谈金潇

摘要

本文提出了一个概念上简单、灵活和通用的实例分割框架。该方法可以有效地检测图像中的对象，同时为每个实例生成高质量的分割掩码 mask。该方法称为 Mask R-CNN，通过添加一个分支用于预测对象 mask，并与现有的用于边界框识别的分支并行来拓展 Faster R-CNN 得到。Mask R-CNN 训练简单，只给 Faster R-CNN 增加了很小的开销，以 5fps 的速度运行。此外，Mask R-CNN 很容易推广到其他任务，例如，允许我们在相同的框架中估计人体姿态。

关键词：实例分割；物体检测；姿态估计；卷积神经网络

1 引言

目标检测以及语义分割结果的相关工作短时间内取得了改善。作者认为这种进步是由 baseline 带来的，例如 Fast/Faster R-CNN[1-2]和 FCN[3]。这些框架提供了很好的灵活性和鲁棒性，以及效率上的提升。本文的目标是为实例分割开发一个相对可行的框架。

实例分割具有挑战性，因为它需要正确的地检测图像中的所有对象，同时还需要精确地分割每个实例。因此，它结合了对象检测的经典计算机视觉任务中的元素。其中目标是对单个对象进行分类，并使用边界框对每个对象进行定位，而语义分割的目标是在不区分对象实例的情况下将每个像素分类为一组固定的类别。考虑到这一点，人们可能会认为需要一个复杂的方法来实现良好的效果。然而，本文设计了一种更加简单、快速和灵活的系统，超越了本文之前最先进的实例分割结果。

本次复现工作是为了复现该常用的语义分割网络，方便后续对该网络的应用。

2 相关工作

R-CNN。基于区域的 CNN(R-CNN) 方法[4]中边界框目标检测是关注数量可控的候选目标区域，并在每个 RoI 上独立评估卷积网络。R-CNN 扩展了 Fast R-CNN[1]和 Spatial pyramid pooling[5]，以允许使用 RoIPool 关注特征图上的 roi，从而更快以及更加准确。Faster R-CNN[2]通过使用区域建议网络 (RPN) 学习注意力机制改进了这一流程。Faster R-CNN 对许多后续改进具有灵活性和鲁棒性，并且在几个 baseline 测试中是当时的领先框架。

实例分割。在 R-CNN 有效性的驱动下，许多实例分割方法都是基于分段建议。早期的方法采用自下而上的片段。DeepMask[6]和接下来的工作学习提出片段候选，然后通过 Fast R-CNN 进行分类。在这些方法中，分割先于识别，识别速度慢且准确率不高。同样，Dai 等人[7]提出了复杂的多阶段级联，从边界框建议中预测分段建议，然后进行分类。相反，本文的方法是基于掩码和类标签的并行预测，更简单、更灵活。

最近，Li 等人[8]结合了分割建议系统[9]和目标检测系统[10]，实现了“全卷积实例分割”(FCIS)。它们的共同思想是使用完全卷积预测一组位置敏感的输出通道。这些通道同时定位对象类别、边界框和掩码，使系统更加快速。但 FCIS 在重叠实例上表现出系统错误，并创建虚假边缘 (图 1)，这表明它遇

到了实例分割的基本困难的挑战。另一类解决方案中的实例分割是由语义分割的成功驱动的。这些方法从逐像素分类结果 (例如，FCN 的输出) 开始，试图将同一类别的像素分割为不同的实例。与这些方法的分割优先策略相比，Mask R-CNN 基于实例优先策略。

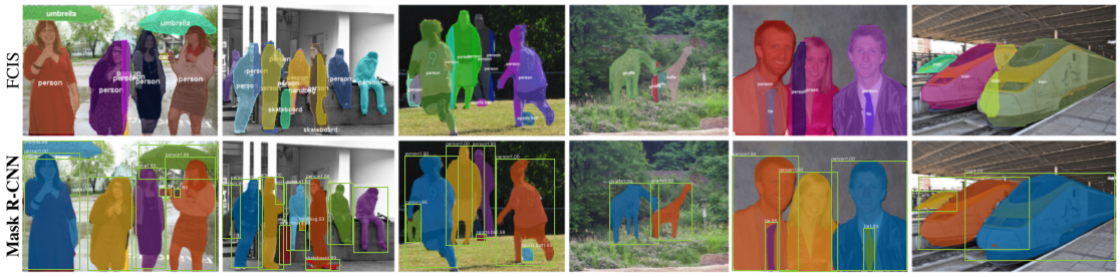


图 1: FCIS++ 与 Mask R-CNN 对比。FCIS 在重叠的实例上出现错误。

3 本文方法

3.1 本文方法概述

Mask R-CNN 在概念上很简单，Faster R-CNN 对每个候选对象有两个输出，一个类标签和一个边界框偏移量; 在此基础上，本文添加了第三个分支来输出对象掩码。因此 Mask R-CNN 是一个自然而直观的想法。但额外的 mask 输出与 class 和 box 输出不同，需要提取对象更精细的空间布局。介绍一下 Mask R-CNN 的关键元素，包括像素到像素的对齐，这是 Fast/Faster R-CNN 的主要缺失部分。

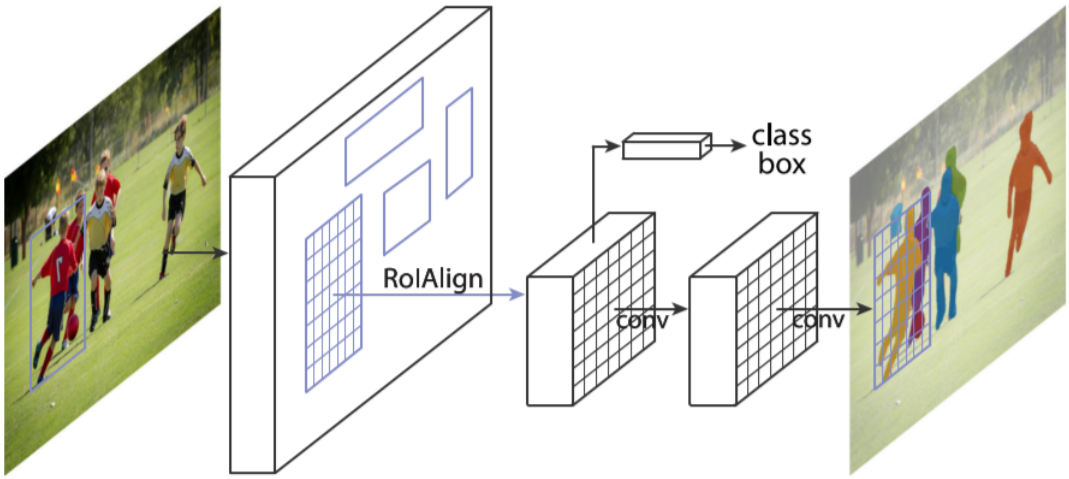


图 2: 用于实例分割的 Mask R-CNN 框架

3.2 Mask R-CNN

首先简要介绍一下 Faster R-CNN[2]。Faster R-CNN 由两个阶段组成。第一个阶段称为区域建议网络 (RPN)，提出候选对象边界框。第二阶段，本质上是 Fast R-CNN[1]，使用 RoIPool 从每个候选框中提取特征，并执行分类和边界框回归。两个阶段使用的特征可以共享，以实现更快的推理。Mask R-CNN 采用相同的两阶段。使用相同的第一阶段 (即 RPN)。在第二阶段中 Mask R-CNN 添加了一个与类别预测和边界框偏移并行的分支来为每个 RoI 输出一个二进制掩码。这与大多数最近的系统形成了对比，在这些系统中，分类依赖于掩码预测。本文方法遵循 Fast R-CNN 的精神，并行应用分类和边界框回归 (这在很大程度上简化了原始 R-CNN[4]的多级管道)

3.3 Mask Representation

与类标签和边界框偏移量这两个分支不可避免地被全连接层折叠成短输出向量不同，提取 mask 的空间结构可以通过卷积提供的像素到像素的对应关系来解决。具体的说，本文通过使用 FCN[3]对每一个 RoI 预测了一个 $m \times m$ 的 mask，这允许 mask 分支中的每一层都维持显式的 $m \times m$ 的空间布局，而不是将其折叠成缺乏空间信息的矢量表示。与之前借助全连接层进行 mask 预测的方法不同，本文的全卷积需要的参数更少，实验表明更准确。这种像素到像素的对应关系要求 RoI 的特征有良好的对齐关系，来更好的保留像素空间对应关系。这种需求也由本文提出的 RoI Align 解决。

3.4 RoIAlign

RoIPool[1]是从 RoI 中提取小特征图 (7×7) 的标准操作。RoIPool 首先将一个浮点数 RoI 量化到特征图的离散粒度，然后将量化后的 RoI 细分为空间 bins，这些 bins 本身也进行了量化，最后聚合每个 bin 覆盖的特征值 (通常使用最大池化)。举个例子，量化是在连续坐标 x 上计算 $[x/16]$，其中 $[\cdot]$ 操作是四舍五入，16 是特征图步长。这些量化操作导致了 RoI 和所提取的特征之间的不对齐。虽然这种不对齐可能不会影响分类结果，但是对最终预测的 mask 会有很大的负面影响。

为了解决这个问题，本文提出了 RoIAlign 层，消除了 RoIPool 的量化误差，将提取的特征与输入正确对齐。本文提出的改变很简单，为了避免对 RoI 或 bins 边界进行量化，不再使用 $[\cdot]$ 操作。并通过双线性插值[11]来计算每个 RoI 的 bin 中四个固定采样位置的输入特征的精确值，并聚合结果 (使用最大值或均值)。实现细节见图 3

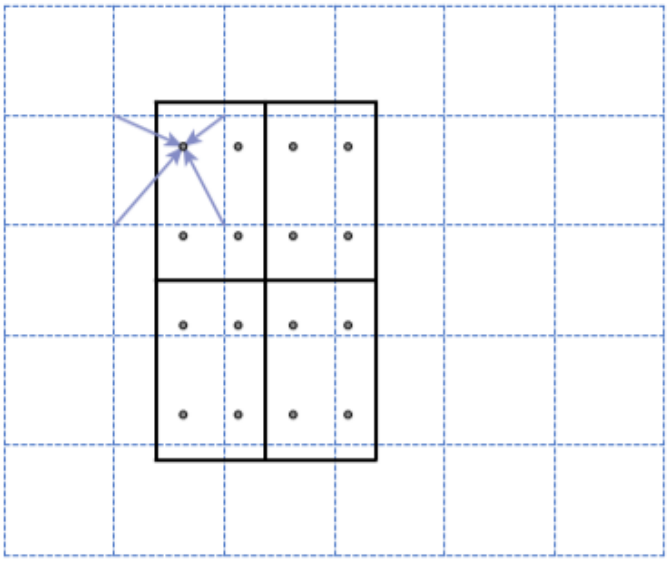


图 3: RoIAlign 的实现。虚线网格是在其上执行 RoIAlign 的特征图，实线表示 RoI，本例中有 2×2 个 bin，圆点表示每个 bin 内的 4 个采样点。每个采样点的值是通过双线性插值从特征图上附近的网格点计算得到的。不对 RoI、bins 或采样点的任何坐标进行量化。

3.5 Network Architecture

本文中区分用于在整个图像上进行特征提取的卷积 backbone 架构以及用于边界框识别 (分类和回归) 和 mask 预测的 network head，并分别应用于每个的 RoI。本文还探索了另一个更有效的 backbone[12]，称为特征金字塔网络 (FPN)。FPN 使用具有横向连接的自顶向下架构，从单尺度输入构建网络内特征金字塔。具有 FPN backbone 结构的 Faster R-CNN 根据特征金字塔的尺度从不同层次提取 RoI 特征，但是该方法的其余部分类似于 ResNet。使用 ResNet-FPN 进行 Mask R-CNN 的特征提取，在精度和速

度上都有所提升。

对于 network head，本文添加了一个完全卷积 mask 预测分支。具体的说，拓展了 ResNet[13]和 FPN[12]论文中的 Faster R-CNN box head。

4 复现细节

4.1 与已有开源代码对比

本篇论文有相关源代码。在整体代码实现后，对论文的整体结构进行思考，本文的主要创新点在并行运行的 mask 分支以及为了解决特征之间不对齐问题而提出的 RoIAlign。经过思考，想要对代码进行改进，只能想到对 mask 分支进行一定的改进。论文中 mask 分支使用类似于 FCN 的结构提取掩码，FCN 是将一般的分类 CNN 的后面几个全连接 FC 层更换为卷积，以得到 2 维的特征图，后接 softmax 层获取每个像素点的分类信息。根据近些年神经网络以及语义分割技术的发展，我认为可以将该分支结构更改为 U-Net 结构，U-Net 是在 FCN 基础上进行改良的，可以更好的得到掩码信息。所以我认为，修改 mask 分支的结构有可能会小幅度优化 mask 结果。但是该想法因时间问题并未成功实现。

4.2 实验环境搭建

我的实验环境基于 Ubuntu 22.04：

- python 3.9.15
- torch 1.13.0
- torchvision 0.14.0
- pycocotools
- Pillow
- lxml
- matplotlib
- numpy
- tqdm

4.3 界面分析与使用说明

- pycocotools 安装

```
pip install cython
git clone https://github.com/pdollar/coco.git
cd coco/PythonAPI
python setup.py build_ext --inplace
python setup.py build_ext install
```

- COCO 数据集

COCO 数据集官网https://cocodataset.org/

主要下载

  - 2017 Train images(训练过程中使用到的所有图像文件)
  - 2017 Val images(验证过程中使用到的所有图像文件)

– 2017 Train/Val annotations(对应训练集和验证集的标注 json 文件)

- 预训练权重下载

Resnet50 预训练权重地址 https://download.pytorch.org/models/resnet50-0676ba61.pth，并在下载后重命名为 resnet50.pth

Mask R-CNN(Resnet50+FPN) 预训练权重地址 https://download.pytorch.org/models/maskrcnn_resnet50_fp
bf2d0c1e.pth, 并在下载后重命名为 maskrcnn_resnet50_fpn_coco.pth

- 指标

训练过程中保存的 det_results.txt(目标检测任务) 和 seg_results.txt(实例分割任务) 是每个 epoch 在验证集上的 COCO 指标，前 12 个值是 COCO 指标，后两个是训练平均损失以及学习率。

- 训练

train.py 训练脚本使用单 GPU，其中- -data-path 设置存放数据集的根目录。

```
python train.py --data-path ./data/coco2017
```

- 预测

predict.py 是预测脚本。需要先将训练好的权重设置在 weights_path(默认为"./save_weights/model_25.pth
已设置), 以及设置正确的 img_path(默认为"./test.jpg")。

最简单的使用方法为选择一张图片放在根目录下并重新命名为 test.jpg，然后运行预测脚本。

```
python predict.py
```

4.4 创新点

在原文中，作者提出的创新点包括 RoIAlign 和一个 mask 分支。其中对于 RoIAlign，没有任何可以改进的想法。而对于 mask 分支，我认为作者提出的类似于 FCN 的结构去求 mask，或许可以改进为 U-Net 结构。

如图 4，FCN[3]将原本 CNN 网络的全连接层换成了卷积层，然后通过逆卷积将特征图扩大到跟原图同样大小的若干张概率图，每张概率图表示这个像素属于该类别的概率值。

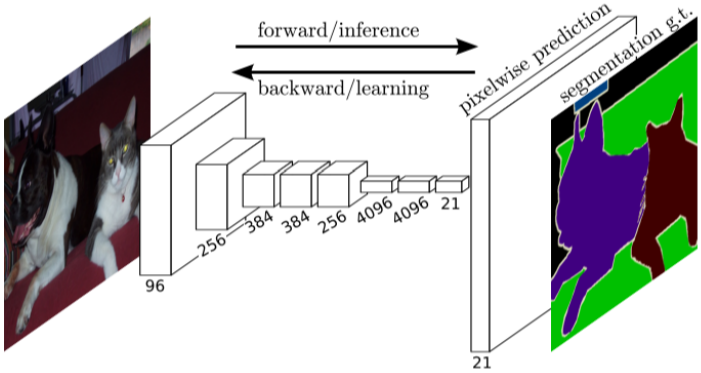


图 4: FCN 架构。

如图 5，U-Net[14]整体网络的思路与 FCN 相同，是 FCN 的改进。U-Net 包括了收缩路径和扩张路径。与 FCN 不同的地方在于，U-Net 采用拼接的方式进行特征融合。

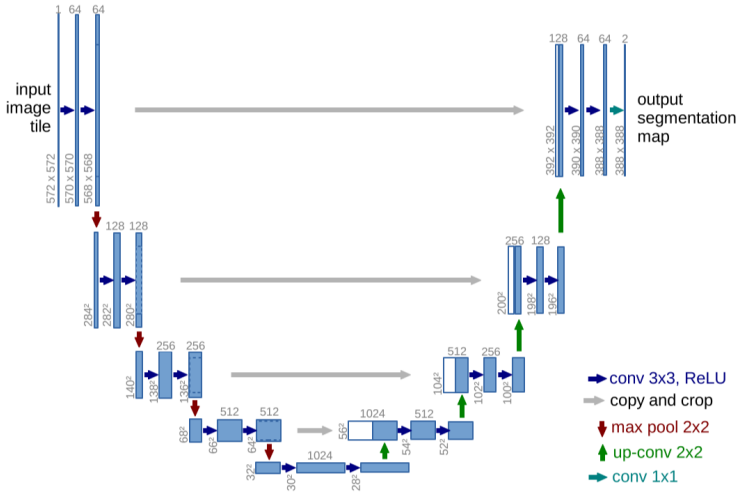
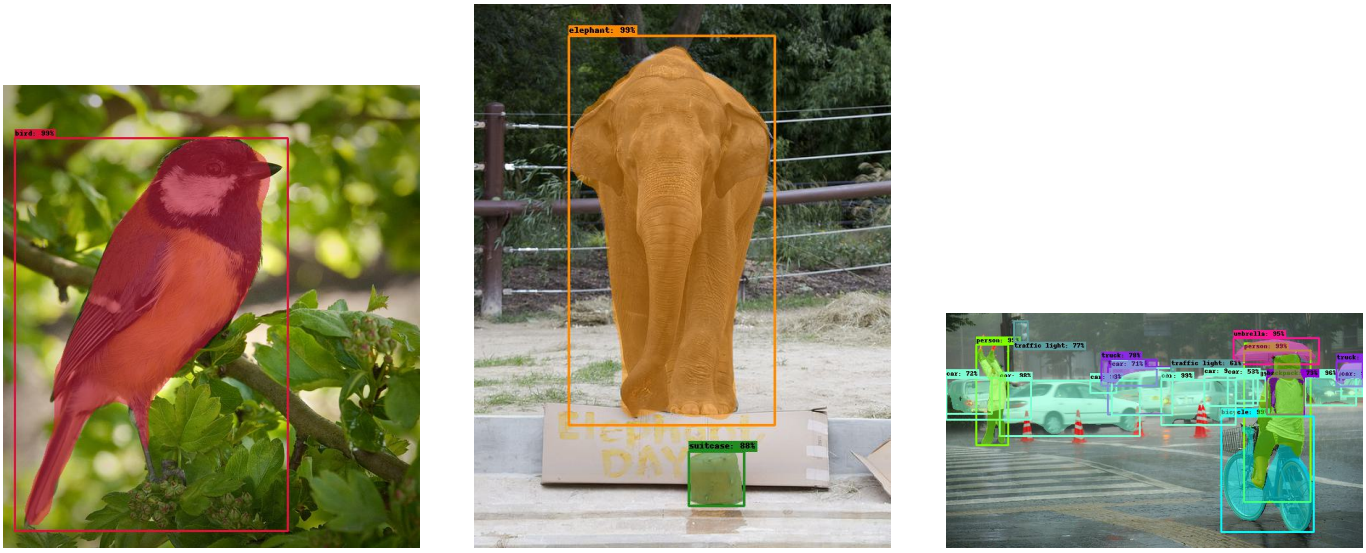


图 5: U-Net 架构。

5 实验结果分析

该部分首先展示了几张 Mask R-CNN 在 COCO 数据集进行实例分割的结果，如图 6。通过肉眼观察，可以看到，分割结果还是比较优秀的。



(a) (b) (c)

图 6: Mask R-CNN 在 COCO 数据集上的一些结果

然后，通过图 7对比，可以看到，此次实验的结果 mAP 为 34.0，与原文同样使用 ResNet-50-FPN 的 Mask R-CNN 的 33.6 基本相同。

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.340
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.552
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.361
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.151
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.369
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.500
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.290
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.449
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.468
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.266
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.509
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.619
```

(a) 实验结果

net-depth-features	AP	AP_{50}	AP_{75}
ResNet-50-C4	30.3	51.2	31.5
ResNet-101-C4	32.7	54.2	34.3
ResNet-50-FPN	33.6	55.2	35.3
ResNet-101-FPN	35.4	57.3	37.5
ResNeXt-101-FPN	**36.7**	**59.5**	**38.9**

(b) 原文精度

图 7: 精度对比

6 总结与展望

通过这门课程，我认为我学会了如何去复现论文，这对我日后的科研学习有很大的帮助。在这次复现工作中，我选择了 Mask R-CNN 这篇论文，也是因为我的方向目前很多人在向语义方向探索，但是 SLAM 的工程量太大因此我选择了 Mask R-CNN 这个很多人在用的框架。在这次复现中，我学会了 Mask R-CNN 的原理和使用方法，这让我在日后需要使用语义信息的时候可以更快的使用 Mask R-CNN 框架。但是我也认为我在此次复现中有很多不足，首先是一些基础知识的缺少让我需要花时间去补充学习，另一方面是第一次做这个工作导致有很多地方思考不足。另外，在原工作基础上进行创新的思路也并没有成功，而且也不能确定使用 U-Net 代替 FCN 结构是否真的能够提升精度。这一切还需要后续进行探索。

参考文献

[1] GIRSHICK R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[2] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[C]//: vol. 28. 2015: 91-99.

[3] LONG J, SHELHAMER E, DARRELL T. Fully Convolutional Networks for Semantic Segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015.

[4] 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.

[5] 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.

[6] O. PINHEIRO P O, COLLOBERT R, DOLLAR P. Learning to Segment Object Candidates[C]// Advances in Neural Information Processing Systems: vol. 28. Curran Associates, Inc., 2015: 1990-1998.

[7] DAI J, HE K, SUN J. Instance-aware semantic segmentation via multi-task network cascades[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 3150-3158.

[8] LI Y, QI H, DAI J, et al. Fully convolutional instance-aware semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2359-2367.

[9] DAI J, HE K, LI Y, et al. Instance-sensitive fully convolutional networks[C]//European conference on computer vision. 2016: 534-549.

[10] DAI J, LI Y, HE K, et al. R-fcn: Object detection via region-based fully convolutional networks[J]. Advances in neural information processing systems, 2016, 29.

[11] JADERBERG M, SIMONYAN K, ZISSERMAN A, et al. Spatial transformer networks[J]. Advances in neural information processing systems, 2015, 28.

[12] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

[13] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[14] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. 2015: 234-241.