1 Mask R-CNN 的原理及流程

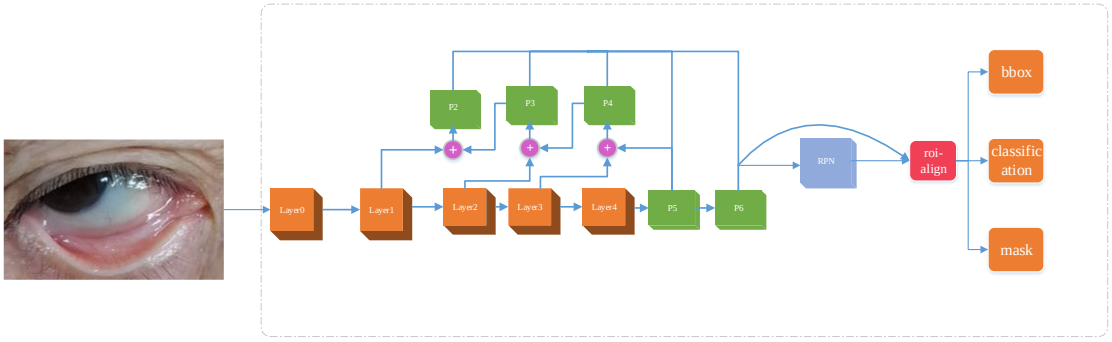


图 1 神经网络框架图

本文采用 FPN+MaskRCNN 神经网络来完成（如图 1），FPN 层的的功能为增加目标的检测能力。本文的 backbone 采用主流的残差网络（resnet），如果考虑到网络的推理速度，可以采用轻量级的网络如果 mobilenet-v2 等仍然可以达到理想的效果。在网络的 RPN 部分，该部分主要是图像的感兴趣区域的提取，并形成目标候选区域。ROI align 层结合 FPN 特征层和 RPN 层形成固定的特征层便于网络计算。

本文的训练流程如下：

（1） 对于一张眼睛图像，进行数据的预处理，例如数据增强以及归一化等操作，形成处理后的图像。

（2） 经预处理后的图像，经网络的 backbone 网络，计算出出特征图。

（3） 根据 anchor 获取多个感兴趣区域（ROI）

（4） 感兴趣区域（ROI）经 RPN 网络，过滤掉候选 ROI

（5） ROI 的输出和 FPN 的输出经 ROI Align，形成固定尺寸的特征图

（6） 对固定尺寸的特征图进行分类，位置的回归以及 mask 的生成。

2、实验环境、实验过程，实验参数

实验环境：本文是在 ubuntu16.04TSL 环境下进行，采用 NVIDIA RTX 2080TI GPU 训练。本实验选用的是 pytorch1.3，python3.7，cuda11.0

实验过程：1）训练集和验证集的采集，本文采集了 1273 张人眼图像，并对眼睑部分进行人工标注。

2）对标注的图像转换为 COCO 数据集格式

3）设置网络结构，包括 backbone+fpn+maskrcnn 的，backbone 采用残差网络，从第二个 block 开始融合 FPN 结构。

4）训练，本文采用单 GPU 进行训练。设置好超参数，观察训练过程中的 loss 等值，不断的调整训练结果。

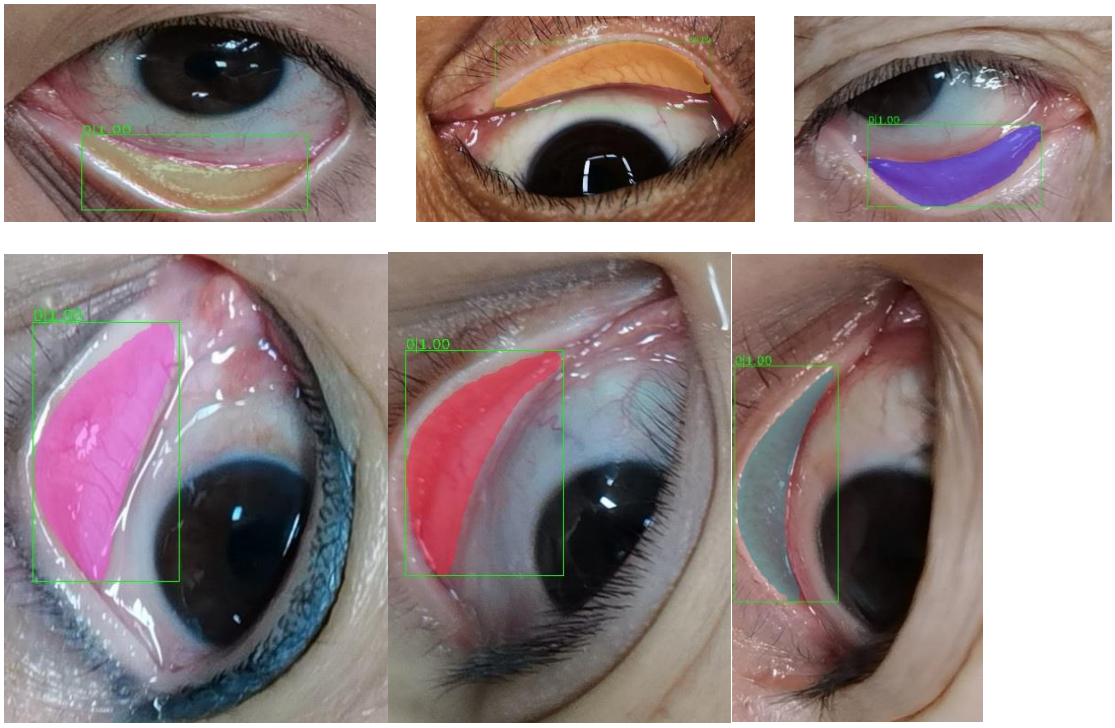
5）对于训练结果进行测试，得出性能指标。

实验参数：

训练集采用多尺度训练，训练尺度为 300 到 600，batchsize=8，测试尺度为 400。采用随机梯度下降（SGD）,学习率 lr=0.01，num_epochs=24，采用 step 的学习率下降策略。在第 16 个 epoch，学习率下降 1 倍，在第 20 个 epoch，学习率再下降 1 倍。

3 实验结果

本文挑选 1151 张眼睛图像用于训练，112 张图像用于测试。测试效果如下：



给结果返回眼睑的类别（这里只有 1 类），位置（x,y,w,h）以及分数和 mask 信息。其中 x,y 表示矩形框的左上角坐标，w,h 表示宽度和高度。
测试性能指标采用 coco 指标

```
DONE (t=0.26s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.672
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.989
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.847
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.672
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.673
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.721
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.721
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.721
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.676
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.726
 Average Recall     (AR) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.996
1 0.8353856802866316
mIOU: 0.8353856802866316
```

从上图可以看出，本文的 mIOU 为 0.835，这里，预测框假设为 pb，真实的眼睑框为 gb，那么 IOU=（pb 与 gb 的交集）/(pb 与 gb 的并集)
对于上面检测结果的解读：

(1)若一个实例是正类并且被预测为正类，即为真正类(True Postive TP)

(2)若一个实例是正类，但是被预测成为负类，即为假负类(False Negative FN)

(3)若一个实例是负类，但是被预测成为正类，即为假正类(False Postive FP)

(4)若一个实例是负类，但是被预测成为负类，即为真负类(True Negative TN

精度 precision=TP/(TP+FP)

召回率 recall=TP/(TP+FN)

Iou=0.50：0.95 是把阈值分为 0.5，0.55，0.6，0.65…0.95 去分别测试，得到平均精度（Average Precision AP）以及召回率(Average Recall AR)。其中，因本实验只有 1 类，所以 AP=mAP，AR=mAP。

area=all,large,medium,small,这是根据检测框的大小分为所有的，大目标，中目标，小目标进行分类统计。

maxDets=1 表示单张图像上只选择概率最大的 1 个目标进行统计。

maxDets=10 表示单张图像上只选择概率最大的 10 个目标进行统计。

maxDets=100 表示单张图像上只选择概率最大的 100 个目标进行统计。