

本周主要来回换了几种 base net 的预训练模型在 UCFSports 上测试一帧的效果，因为实验过程中也在排查一些 bug，数据增强的一些参数和方法修改和完善了几次，所以相对刚开始的训练点数有一部分提升。

base net	预训练	roi 方式	map	选用帧数
vgg16	ACT-Detector	Align	0.724	1
vgg16	pytorch	Align	0.7668	1
resnet 50	pytorch	Align	0.6851	1

相对于之前的训练，为了防止过拟合，修改了 expand 的参数以及增加了随机的平移，以及对于正负样本的选择做了一些限定，我觉得一定程度上提高了 frame-map。

不过从以上实验结果看出，map 的结果有点玄学。前面两个 vgg16 在一帧上的实验结果就已经碾压了论文 multi-region two-stream R-CNN 里面的 frame-map，因为之前经常在排查 bug，所以基本上每修复一个就可以获得一点点提升。现在 UCF101 这个大一点的数据集上进行验证，暂时还没有跑出来，看看能不能减少玄学问题。相比较而言，resnet 如果 freeze 前面几层之后的拟合能力很差，一般情况下要把所有的全部放开才能达到比较好的拟合效果，但是点数不涨。。

另外这两天在 GitHub 上面下载多个已经训练好的 faster rcnn 模型，想仅测试 rpn 在 UCFSports 和 UCF101 上面的召回率究竟怎样，但是在实验中发现其召回率并不理想，map 更是感人。由于作者的训练过程中 anchor 的设置参数以及数据的处理方式与我之前的不同，而代码中只给了一种参数，多种模型的参数不知道。今天折腾了一天修改了很多细节地方，resnet 网络模型的原作者就改了一个很小的地方。。。但是可能自己在实现的过程中还有其他的问题，所以还需要花时间来踩坑。下面需要拿两张通用的测试图跑一下原来作者的代码然后对比排查问题。。

现在采用了两个模型修修改改了之后，在 UCFSports 上面最好的结果是（显然不太对。。。还需要花时间去找问题，应该是还有哪里没有注意到）：

数据集	训练好的模型	rpn recall
UCFSports	coco scale600	0.5012
UCFSports	voc2007 scale600	0.6707