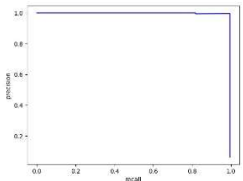
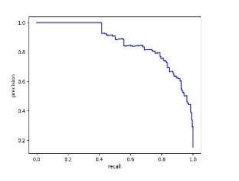
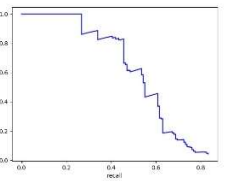
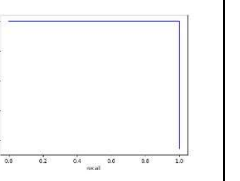
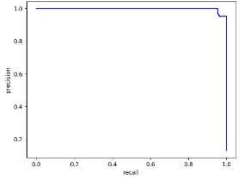
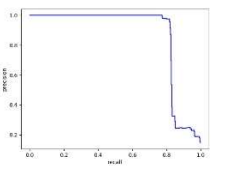
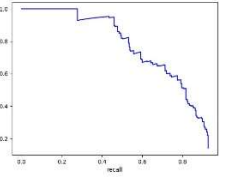
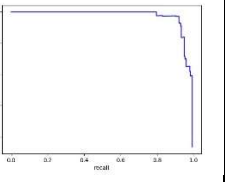
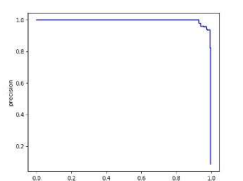
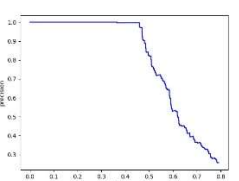


本周主要对 UCFSports 数据集的测试结果进行了分析，代码在 UCF101 上面的训练不仅速度慢而且出现了一些偶然的问题，本来一个 epoch 就很慢，收敛速度也很慢，经常在一个 epoch 快结束的时候 loss 会出现 inf 或者 nan, 断断续续的训练目前在 UCF101 上只拿到了) 0.6154 的 map，离作者提供的 RGB 的网络参数的 map 0.6410 还有不小的差距。因为这个问题出现极具偶然性，经过分析摸索目前已经知道了是在 expand 的时候应该是由于某种巧合导致 ground truth 坐标的 min 和 max 相等，现在已经在问题的前后设置了多个保护现场参数的手段坐等问题再次出现恢复现场。。

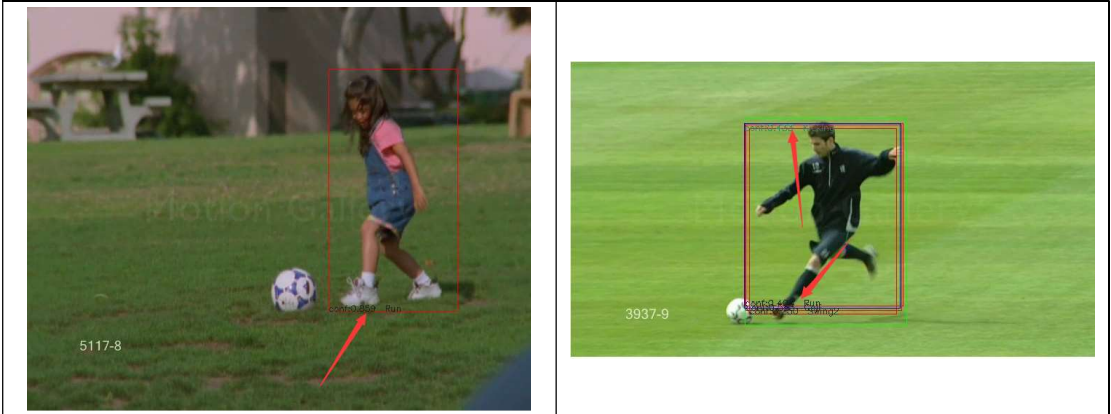
另外针对 UCFSports 里面出现的错误也 YY 了很多，但是经过分析大多不太契合实际，还是要以熟悉现有结构，体会论文训练细节为主。

UCFSports:

Pr 曲线:

			
Diving 0.9946	Golf 0.8631	Kicking 0.5423	Lifting 0.9999
			
Riding 0.9979	Run 0.8674	SkateBoarding 0.7337	Swing1 0.9711
			
Swing2 0.9932	Walk 0.6381		

首先关注效果最差的 Kicking，经过贯彻确认，里面确实是有 walk 的部分，没有被标注出来，网络给了算是比较高的置信度，Kicking 和 Run 以及 walk 非常容易混淆。



在这些情况下左图网络给出了 0.85 的 Run 的置信度，右图 0.404 的 RUN，而 kicking 才 0.433（上图中，红色的框是错误标注的，蓝色的是网络预测正确的框，绿色是 ground truth）

这里有个问题，网络有认出来图片里面的球吗？我应该怎么知道网络有没有认出来图片里面的球？？？如果没有我应该怎么让网络认出来里面的球？

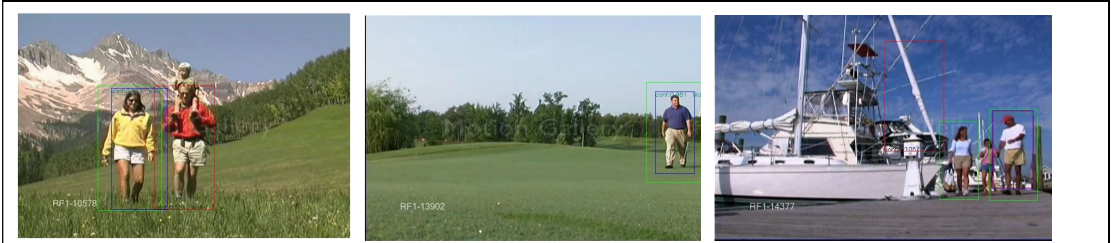


这是 Golf 的部分结果，里面可以看到有很多明显的错例，对于高尔夫，网络似乎并没有注意到 Golf 人手中的东西，其实只要姿势有点像的都会被当成 golf，这和上面 kicking 的错例都是缺乏对事物之间联系的注意力。

不得不承认，前后动作的连贯很重要，但是也必须指出，我们人来判断动作的时候，很多时候是通过周围有什么辅助物体来帮助判断的。不然的话在小范围内很难实现准确判断，如果说只是聚焦于一个人，然后希望通过人的动作，来判断人在做什么。说句实在话如果把关键物体 P 掉，正常人也不好判断这个人在做什么。

所以识别的一个关键性问题就是是否能够在片段中能够找到帮助识别的关键物体。然后根据人的动作来判断。

你会发现这些动作不太好的分类里面，都是有及其相似的部分的。但是你说分类很好的部分其实他们的动作相对单一，但是对于复杂的动作我们必须要有关键物体辅助。网络不能只聚焦于人。



人和背景区分度比较大的时候网络可以给出很好的结果，但是一旦背景复杂而且人和背

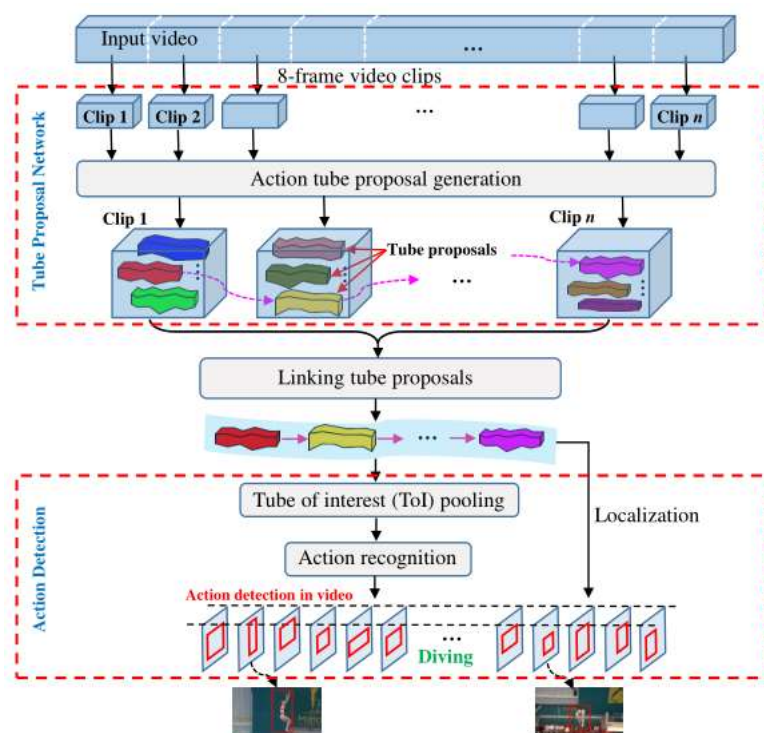
景差距不大的时候，如第三个图就很容易错检或者漏检。（数据集标注也有不完整的地方，比如第一幅图穿红衣服的男人并没有 ground truth 标注）

其实还有一些不切实际的想法，比如时间前后的时序关系，如果是同一个人，那么他在做的事情会有什么不同呢？定位一个人很容易，但是怎么把和人有关的部分都联系起来？比如网络怎么知道这个东西被这个人所控制所掌握？如何建立这个联系？

所以说网络看到了什么样的特征比如一个圆形的球，那么网络如何把这个特征和人相联系？这里我认为，在现在的检测方法中，都是检测到物体，然后就只是把这这个 anchor 的部分的感受野中的信息来对它进行分类，它并没有考虑到事物之间的联系，比如一个球如果飞的比较远，那么这个 anchor 肯定是看不到的，为考虑图中事物之间的联系，我认为是不是可以考虑前面使用一个目标检测网络来检测物体，然后后面的网络来考察这些检测到的物体之间的关系？

找到了一篇基于 C3D 的端到端的人体行为检测论文还是很浅显易懂：(IEEE 2017)

An End-to-end 3D Convolutional Neural Network for Action Detection and Segmentation in Videos 这个里面把 TOI pooling 讲的非常清晰。



作者将该网络称为 T-CNN。

作者将一段视频每 8 帧分为一个切片 (clip)，通过 tube proposal network 为每一个切片生成 tube proposals，因为每一个切片都是独立的 proposals，所以根据重合度将它们连接起来，把连接起来的 proposals 进行 TOI pooling 之后得到统一大小的特征向量，最后使用全连接层对 tube 进行分类。

作者在实施该论文的过程中，为每一个数据集学习 12 个 anchor boxes 来适应不同的数据集。使用了时序跳跃池化来保证 TOI pooling 的过程不会丢失帧间的信息。当捕捉到相关信息的时候，三维卷积三维最大池化的应用不仅仅在空间维度，而且在时间维度上减少视频的尺寸。时间维度上的池化对于识别任务非常重要因为可以减少背景噪声。然而时序会被丢失，这也就意味着如果我们任意改变输入帧的顺序，三维池化之后的特征仍然相同，这对动作定位带来问题。因为动作定位需要依赖 feature cube 来为原始帧图像生成 bounding box。

为此提出时间跳跃池化来保持时序。

作者的总体网络结构非常简单，和复现的论文 Action tubelet detector for spatio-temporal action localization. 的 RGB+FLOW 相比要略微逊色，但是和 ACT-RGB 相比，在一些类别上的 ap 还是有所提升，毕竟他没有用到光流：

Model	Diving	Golf	Kicking	Lifting	Riding	Run	Skate	Swing1	Swing2	Walk	Map
Boarding											
T-CNN	0.8438	0.9079	0.8648	0.9977	1.0000	0.8365	0.6872	0.6575	0.9962	0.8779	0.8670
ACT(only RGB)	0.9946	0.8631	0.5423	0.9999	0.9979	0.8674	0.7337	0.9711	0.9932	0.6381	0.8601

从上表可以看出，3D CNN 的方法在一些易混淆的类别上表现比 ACT 要更加出色。所以从这里看我认为如果可以摸清楚 3D CNN 在这方面对于提高区分度的联系似乎会对提高准确率有一定帮助。

论文中作者在训练的过程中有提到有一点难例挖掘：作者在训练前面 TPN 网络的时候，对于没有 ground truth 部分的 clip，也会送入网络，然后选出打分最高的负样本部分作为难例来训练网络，以此提高网络对无关动作的区分度。

这一点难例挖掘，我认为是我当前的训练中所缺乏的，也是直接指向了我当前正纠结的一点：当前的训练网络对于无关动作很容易混淆。因为在当前的训练中我们其实只取出有 ground truth 的部分，对于那些没有 ground truth 的部分其实并没有进入网络。所以在动作的开始以及无关动作部分对于网络的学习而言是不够的。

那么在当前的这种 SSD 模式下这种难例挖掘也可以考虑加入，因为 TPN 可以认为是一个单独的分类网络，但是在我目前复现的网络中，如果使用没有 gt 部分的视频进行训练其实是没有正样本的，这就无法对 loc 部分进行训练，但是其实可以考虑只训练分类网络而对定位部分不训练。