2018 年秋-第 12 周

本周工作是学完了上周剩下的一些部分，把服务器的操作弄明白了，以及参考一些源码实现 ssn 的 tag proposals 生成部分，其中对于原数据集的帧和光流部分的提取直接参考了 temporal-segment-networks。

主要记录一下一些复习和阅读源码过程中遇到的一些问题：

__init__() 的特殊方法（**构造方法**），该方法在类实例化时会自动调用

```
def __init__(self):
    self.data = []

class Complex:
    def __init__(self, realpart, imagpart):
        self.r = realpart
        self.i = imagpart
x = Complex(3.0, -4.5)
print(x.r, x.i)   # 输出结果：3.0 -4.5
```

类的继承：

```
class DerivedClassName(BaseClassName1):
    <statement-1>
```

若基类中有相同的方法名，而在子类使用时未指定，python 从左至右搜索 即方法在子类中未找到时，从左到右查找基类中是否包含方法。

类的私有方法：

__private_method：两个下划线开头，声明该方法为私有方法，只能在类的内部调用 ，不能在类地外部调用。self.__private_methods。

```
parser = argparse.ArgumentParser(description="Make window file used for detection")
parser.add_argument("subset")
parser.add_argument("modality", choices=['rgb', 'flow'])
parser.add_argument("frame_path")
parser.add_argument("output_file")
parser.add_argument("--overlap", type=float, default=0.7)
parser.add_argument("--max_level", type=int, default=8)
parser.add_argument("--time_step", type=float, default=1)
parser.add_argument("--version",  default="1.2")
parser.add_argument("--avoid",  default=None, type=str)
parser.add_argument("--dataset",  default="activitynet", choices=['thumos14', 'activitynet'])
args = parser.parse_args()
```

import argparse 这个模块与命令行参数相关

```
gt_spans = [[(x.num_label, x.time_span) for x in v.instances] for v in videos]
```

嵌套循环的使用

```
print('average # of proposals: {} at overlap param {}'.format(np.mean(list(map(len, proposal_list))), args.overla
```

Format 的使用

```
out_full_path = os.path.join(out_path, vid_name)
```

os.path：主要用于文件的属性获取

如：

10.os.path.join(path1[, path2[, ...]])
将多个路径组合后返回，第一个绝对路径之前的参数将被忽略。

```
>>> os.path.join('c:\\', 'csv', 'test.csv')
'c:\\csv\\test.csv'
>>> os.path.join('windows\temp', 'c:\\', 'csv', 'test.csv')
'c:\\csv\\test.csv'
>>> os.path.join('/home/aa','/home/aa/bb','/home/aa/bb/c')
'/home/aa/bb/c'
```

```
from multiprocessing import Pool, current_process
```

Multiprocessing 模块：一个使用类似于 threading 模块的 API 支持生成进程的包。主要的例子是该 Pool 对象，它提供了一种方便的方法，可以跨多个输入值并行化函数的执行，跨过程分配输入数据（数据并行）。

multiprocessing.current_process（）返回与 Process 当前进程对应的对象。

class multiprocessing.Pool（[processes [, initializer [, initargs [, maxtasksperchild]]]] ）
一个进程池对象，它控制可以提交作业的工作进程池。

TSN repo：git clone https://github.com/yjxiong/temporal-segment-networks.git
脚本 scripts/extract_optical_flow.sh
参数
SRC_FOLDER 指向放置视频数据集的文件夹
OUT_FOLDER 指向将提取的帧和光学图像放入的根文件夹
NUM_WORKER 指定并行用于流提取的 GPU 数量，必须大于 1
光流提取：bash scripts/extract_optical_flow.sh SRC_FOLDER OUT_FOLDER NUM_WORKER