一、主要工作

CornerNet-Squeeze的测试与训练。

二、标注文件的制作

数据集采用Yolov4的数据集，标注文件采用.json格式。获得.json标注文件的方法：用labelImg软件制作xml的标注文件，然后运用格式转换的程序转成json。目前YOLOv4的标注文件是.txt格式，等训练流程跑通后再想办法转换。

制作的数据集应该具有以下目录结构：

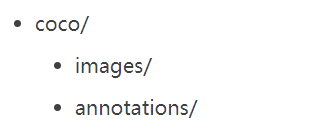


图1：数据集文件夹目录结构

json格式特点：所有训练集放在一个标注文件当中。

目前我所制作的数据集格式如下：

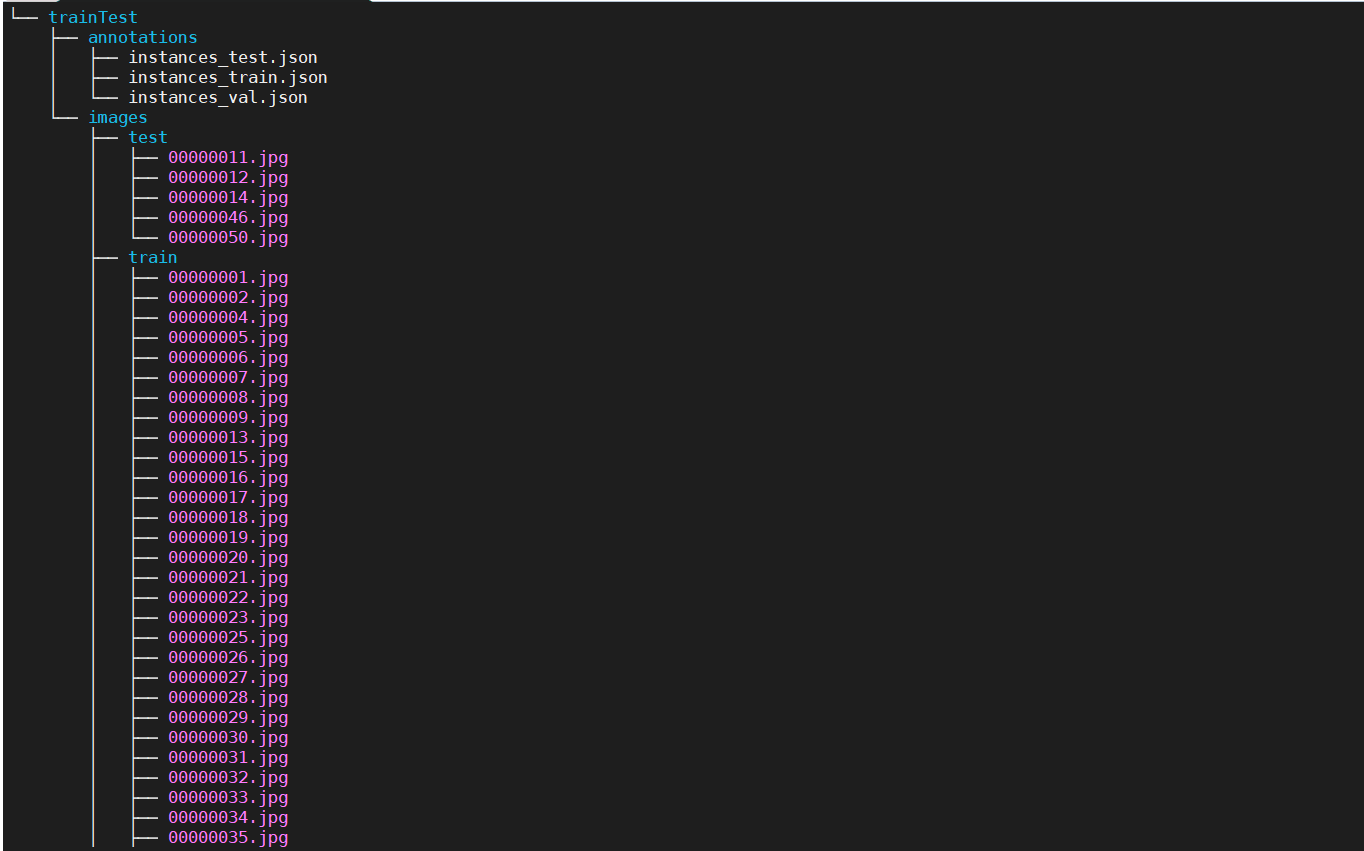


图2：我的数据集

三、修改源码中相关参数

1.参照CornerNet-Lite/core/dbs下的coco.py写一个自己的数据读取接口：

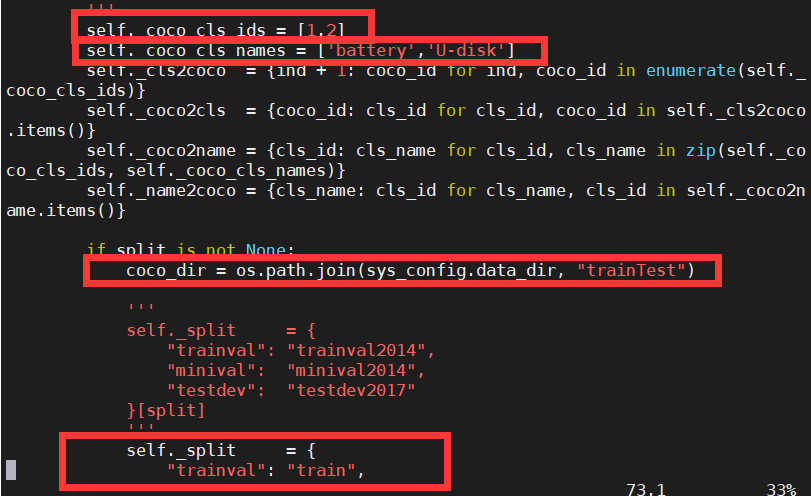


图3：参照coco.py写自己的数据读取接口（trainTest.py）

主要修改的参数：类别总数，类别名称，数据集根目录名称，训练集、测试集、验证集文件夹名称。

2.修改CornerNet-Lite/core/dbs下的\_\_init\_\_.py，指向trainTest.py中定义的数据接口类：

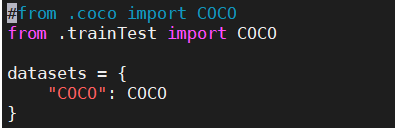


图4：修改\_\_init\_\_.py文件

3.修改CornerNet-Lite/config中的配置文件CornerNet\_Squeeze.json：

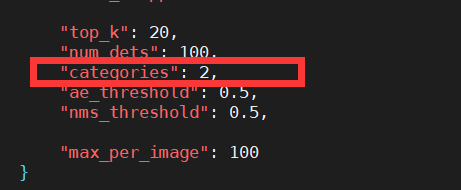


图5：修改CornerNet\_Squeeze.json文件

主要是修改识别的类别数，以及GUP中每个batch处理图片的个数，其他的相关参数也可根据需要修改。

4.修改CornerNet-Lite/core/dbs/detection.py文件：

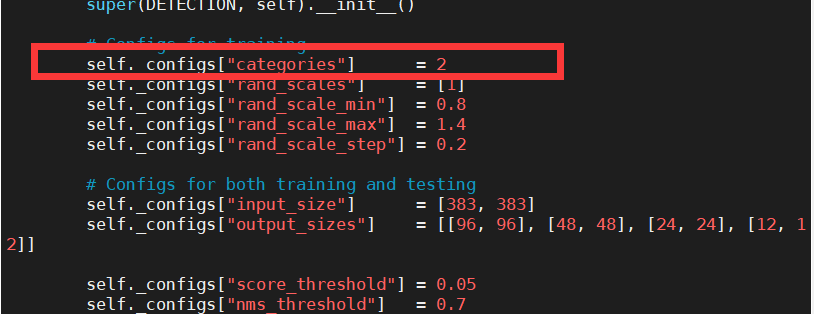


图6：修改detection.py文件

这里也主要是修改类别数。

5.修改CornerNet-Lite/core/models/CornerNet\_Squeeze.py文件：

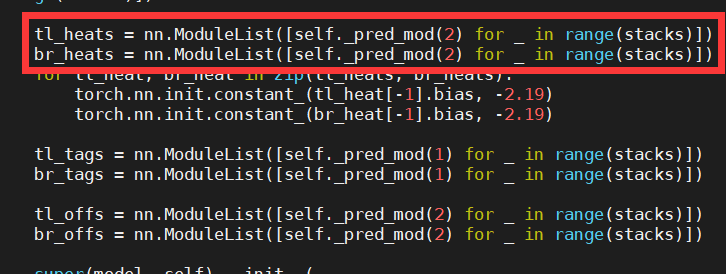


图7：修改CornerNet\_Squeeze.py文件

这里主要也是修改类别数。

四、训练中遇到的问题

训练代码：

cd ~/CornerNet-Lite

python train.py CornerNet\_Squeeze

1.ImportError: /mnt/disk/home1/cjp/anaconda/envs/CornerNet\_Lite/lib/python3.7/site-packages/kiwisolver.cpython-37m-x86\_64-linux-gnu.so : symbol \_ZTVNSt7\_\_cxx1118basic\_stringstreamIcSt11char\_traitsIcESaIcEEE, version GLIBCXX\_3.4.21 not defined in file libstdc++.so.6 with link time reference

解决方法：输入：

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/mnt/disk/home1/cjp/anaconda/lib

将相关的库导入本地路径

2.遇到ValueError: numpy.ufunc size changed, may indicate binary incompatibility或python RuntimeError:cannot join current thread的报错

解决方法：将相关的python中相关的库升级：

pip install --upgrade numpy

pip install --upgrade tqdm

3.出现RuntimeError: CUDA out of memory

解决方法：修改CornerNet-Lite/configs下的CornerNet\_Squeeze.json文件中的batch\_size和chunk\_sizes参数，更改GPU中每个batch处理图片的个数

4.出现RuntimeError: Dimension out of range (expected to be in range of [-2, 1], but got 2)

解决方法：尚未解决

分析：

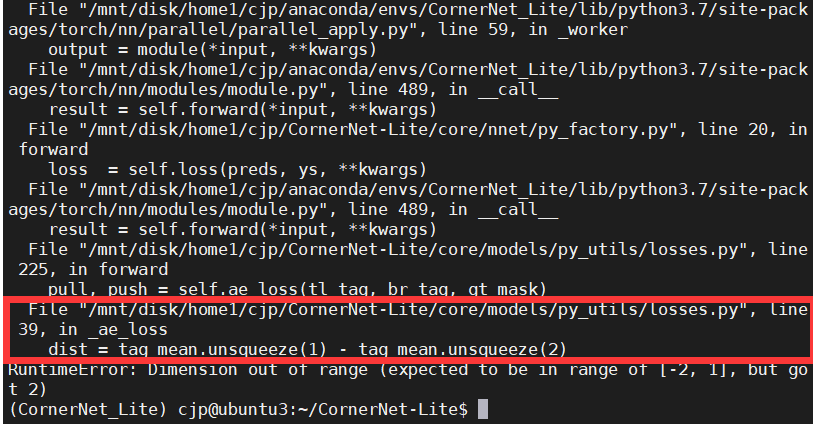


图8：RuntimeError报错

因此我进入到目标文件下查看了具体报错的位置：

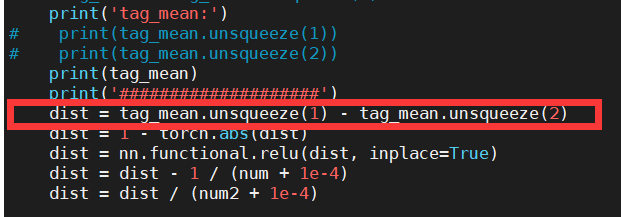


图9：losses.py文件中报错的位置

其中unsqueeze()函数表示在数组的某一维增加一个维度，增加了print(tag\_mean)后打印出来的数据如下：

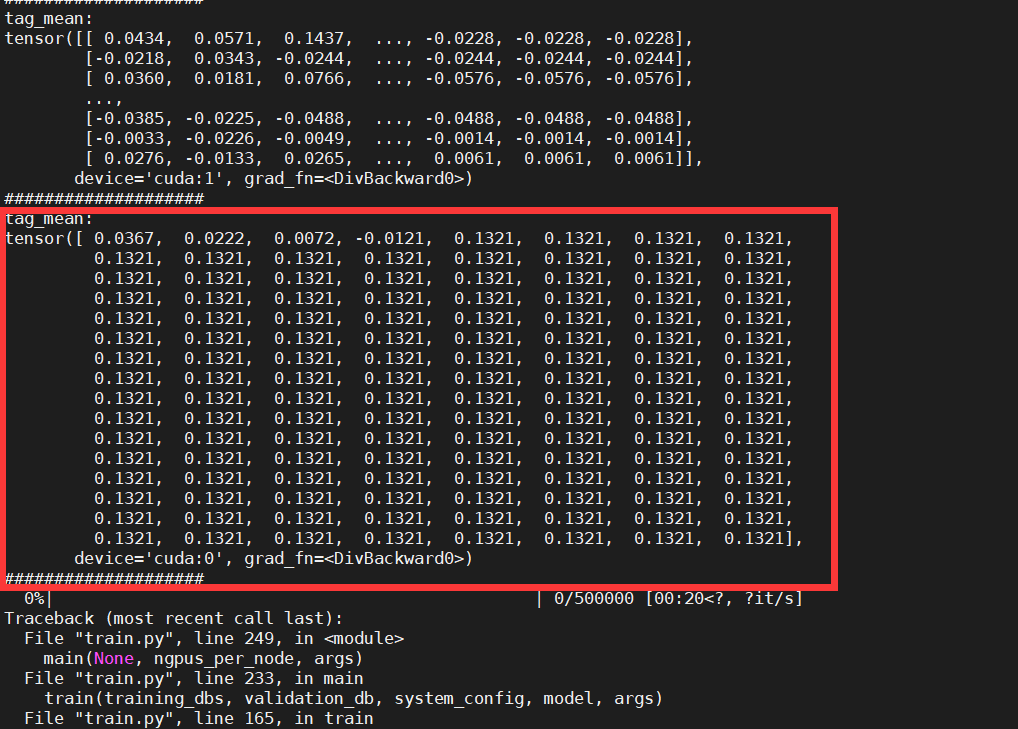


图10：打印出来的信息

发现当tag\_mean变量是二维数组的时候并没有报错，而当tag\_mean是一维数组的时候就报错了。具体为什么tag\_mean会变成一维数组目前还不太清楚，分析可能是由于图片大小的原因，或者是源码中有的参数需要更改。