1.制作COCO数据集

COCO数据集的标注文件以JSON格式保存，其中COCO的目录结构如下：

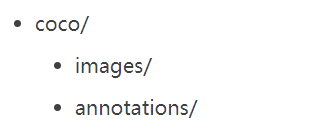


图1：COCO文件夹的目录结构

其中images下面是训练、测试和验证图片，分别放在3个不同的文件夹中。annotations下面保存着训练集、测试集和验证集的标注文件，可以命名为test.json，train.json，val.json。

首先可以用labelImg软件制作xml格式的annotations文件，具体方法可以参考：<https://blog.csdn.net/csdn_zhishui/article/details/84962587#212__119>。然后用xml2json的脚本进行转换，代码可以参考：<https://blog.csdn.net/csdn_zhishui/article/details/95074395?utm_medium=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase>

2.使用训练集进行训练

首先安装MSCOCO API：

在CornerNet-Lite目录下创建文件夹：

mkdir -p data

cd data

git clone <https://github.com/cocodataset/cocoapi>

注意下载后需要将cocoapi改名为coco：

mv cocoapi coco

然后编译：

cd coco/PythonAPI

make install

在数据集制作完成以及上述环境配置好后，需要修改一些配置文件：

cd ~/CornerNet-Lite/core/dbs

参照coco.py构建自己的数据读取接口，并保存在dbs目录下，文件名可自己取，如NJtraffic.py，具体教程可参看<https://zhuanlan.zhihu.com/p/81300373>，以下步骤仍然可以参照此文章。

修改dbs目录下的\_\_init\_\_.py，指向上述NJtraffic.py中定义的数据接口类。

修改CornerNet-Lite/configs下的配置文件CornerNet\_Squeeze.json文件。

修改CornerNet-Lite/core/dbs/detection.py下line10 self.\_configs[“categories”] = 4，这一项是训练的类别数，根据自己的需要进行修改。

修改CornerNet-Lite/core/models/CornerNet\_Squeeze.py的line94和line95，将相关的变量改成自己数据集的类别数。

修改完成后即可开始训练：

cd ~/CornerNet-Lite

python train.py CornerNet\_Squeeze

几天后即可查看结果