

项目实战流程

1. 安装Anaconda

1.1 下载和安装Anaconda

访问Anaconda官网 (<https://www.anaconda.com/>) 下载Anaconda3的linux(Python 3.7)版本;

Anaconda安装包也可以到清华镜像站: <http://mirror.tuna.tsinghua.edu.cn/anaconda/archive/>下载。

本人下载后文件存储到

/home/bai/Downloads/Anaconda3-2019.07-Linux-x86_64.sh

然后, 执行

```
cd ~/Downloads/
```

```
bash Anaconda3-2019.07-Linux-x86_64.sh
```

1.2 更改~/.bashrc文件

执行:

```
sudo gedit ~/.bashrc
```

~/.bashrc文件中添加语句

```
export PATH=/home/bai/anaconda3/bin:$PATH
```

```
alias python='/home/bai/anaconda3/envs/maskrcnn/bin/python3.7'
```

添加后保存~/.bashrc文件, 并执行:

```
source ~/.bashrc
```

2. 安装maskrcnn-benchmark项目

<https://github.com/facebookresearch/maskrcnn-benchmark>

2.1 官方建议的安装需求:

- PyTorch 1.0 from a nightly release. It **will not** work with 1.0 nor 1.0.1. Installation instructions can be found in <https://pytorch.org/get-started/locally/>
- torchvision from master
- cocoapi
- yacs
- matplotlib
- GCC >= 4.9
- OpenCV

- CUDA >= 9.0

2.2 逐步安装过程 (Step-by-step installation)

本人对安装过程做了部分更新修改。具体安装步骤如下：

创建虚拟环境：

```
conda create -n maskrcnn
conda activate maskrcnn
```

创建时也可指定Python版本

```
conda create -n maskrcnn python=3.7
```

安装依赖包：

```
# this installs the right pip and dependencies for the fresh python
conda install ipython pip
```

```
# maskrcnn_benchmark and coco api dependencies
pip install ninja yacs cython matplotlib tqdm opencv-python -i
https://pypi.tuna.tsinghua.edu.cn/simple
```

安装Pytorch:

根据自己的cuda版本执行命令安装Pytorch:

本人使用

```
conda install pytorch torchvision cudatoolkit=10.1 -c pytorch
```

安装了pytorch 1.0.0。

如发生HTTP Error

解决方法：添加镜像站到Anaconda

第一步、添加镜像站到Anaconda执行如下命令：

```
conda config --add channels
http://mirror.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
conda config --add channels
http://mirror.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
conda config --set show_channel_urls yes
```

第二步、还可以附加第三方的conda源：

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/menpo/
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/
```

第三步、切记在官网的命令上去除: -c pytorch

```
conda install pytorch torchvision cudatoolkit=10.1
```

注意: 使用清华源安装时, 去掉 -c pytorch, 否则, 不是从清华源下载相应的包。

安装cocoapi:

进入安装cocoapi的目录下, 如/home/bai

```
cd ~
```

```
export INSTALL_DIR=$PWD
```

```
# install pycocotools
cd $INSTALL_DIR
git clone https://github.com/cocodataset/cocoapi.git
cd cocoapi/PythonAPI
make
```

安装apex:

```
cd $INSTALL_DIR
git clone https://github.com/NVIDIA/apex
cd apex
pip install -v --no-cache-dir --global-option="--cpp_ext" --global-option="--cuda_ext" ./
python setup.py install
```

安装maskrcnn benchmark:

```
# install PyTorch Detection
cd $INSTALL_DIR
git clone https://github.com/facebookresearch/maskrcnn-benchmark.git
cd maskrcnn-benchmark
```

```
# the following will install the lib with symbolic links, so that you can modify
# the files if you want and won't need to re-build it
python setup.py build develop
```

```
unset INSTALL_DIR
```

3 官方demo实践

执行

```
conda activate maskrcnn
```

```
cd ~/maskrcnn-benchmark
```

```
jupyter notebook &
```

在Jupyter Notebook中打开/home/bai/maskrcnn-benchmark/demo/Mask_R-CNN_demo.ipynb

执行cell

错误信息: [ModuleNotFoundError: No module named 'maskrcnn_benchmark'](#)

解决办法:

```
import sys
sys.path.append("~/maskrcnn-benchmark")
```

错误信息: ModuleNotFoundError: No module named 'torch'

解决办法: 安装 nb_conda_kernels 包:

```
conda install nb_conda_kernels
```

安装缺少的包, 如:

```
conda install requests
```

```
conda install matplotlib
```

在Jupyter Notebook中使用新的kernel

4 制作自己的数据集

4.1 图像标注工具labelme的安装与使用

```
conda install scikit-image
```

```
pip install labelme -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
pip install pycocotools -i https://pypi.tuna.tsinghua.edu.cn/simple
```

4.2 使用labelme进行图像标注

执行

数据集图像文件放置在/home/bai/mydataset目录下

4.3 图像标注后的数据转换

把labelme标注的json数据格式转换成COCO数据格式的。

```
cd /home/bai/mydataset
```

```
python labelme2cocoAll.py roadscene_train --output roadscene_train.json
```

```
python labelme2cocoAll.py roadscene_val --output roadscene_val.json
```

```
jupyter notebook &
```

在Jupyter Notebook中打开~/mydataset/COCO_Image_Viewer.ipynb

注意：由labelme标注的数据格式转成COCO数据格式后只包含3个字段信息：images, annotations, categories。而原始COCO数据集包含5个字段信息：info, licenses, images, annotations, categories。

4.4 项目数据准备

把转成的COCO数据格式的数据的目录结构准备成COCO目录结构格式。

在maskrcnn-benchmark根目录下面的datasets文件夹下，创建目录结构如下：

```
├── coco
│   ├── annotations
│   ├── train2017
│   └── val2017
```

其中：

```
├── annotations
│   ├── instances_train2017.json
│   └── instances_val2017.json
```

roadscene_train.json改名为instances_train2017.json

roadscene_val.json改名为instances_val2017.json

另外：

roadscene_train目录改名为train2017

roadscene_val目录改名为val2017

5 训练自己的数据集

5.1 配置文件选择与修改

根目录下的configs文件夹里面有很多yaml网络配置文件，这里选择的是e2e_mask_rcnn_R_50_FPN_1x.yaml，更改如下：

拷贝文件并改名为:my_e2e_mask_rcnn_R_50_FPN_1x.yaml, 内容修改如下：

```
.....

WEIGHT: "./weights/my_pretrained_R_50.pth"

.....

DATASETS:
TRAIN: ("coco_2017_train", )
TEST: ("coco_2017_val",)

.....

OUTPUT_DIR: "./weights/"
```

拷贝my_e2e_mask_rcnn_R_50_FPN_1x.yaml为测试配置文件my_test_e2e_mask_rcnn_R_50_FPN_1x.yaml, 内容修改如下：

```
WEIGHT: "./weights/model_final.pth"
```

5.2 预训练权重文件准备

```
python weights/trim_detectron_model.py --pretrained_path
weights/e2e_mask_rcnn_R_50_FPN_1x.pth --save_path weights/my_pretrained_R_50.pth
```

生成的预训练权重文件为weights/my_pretrained_R_50.pth

5.3 修改配置文件和代码后重新编译项目

```
python setup.py build develop
```

5.4 网络训练

在项目目录 maskrcnn-benchmark/下运行：

```
python tools/train_net.py --config-file
configs/my_e2e_mask_rcnn_R_50_FPN_1x.yaml MODEL.ROI_BOX_HEAD.NUM_CLASSES 6
SOLVER.IMS_PER_BATCH 4 SOLVER.BASE_LR 0.001 SOLVER.MAX_ITER 36000 SOLVER.STEPS "
(24000, 32000)" TEST.IMS_PER_BATCH 1 MODEL.RPN.FPN_POST_NMS_TOP_N_TRAIN 2000
```

其中MODEL.ROI_BOX_HEAD.NUM_CLASSES的值根据自己的数据集物体的个数设定为：物体的个数+1

6 网络模型验证

6.1 性能指标统计

```
python tools/test_net.py --config-file configs/my_e2e_mask_rcnn_R_50_FPN_1x.yaml
--ckpt weights/model_final.pth MODEL.ROI_BOX_HEAD.NUM_CLASSES 6
```

configs/my_test_e2e_mask_rcnn_R_50_FPN_1x.yaml文件中指定了测试所用的网络模型

WEIGHT: "/home/bai/maskrcnn-benchmark/weights/model_final.pth"

6.2 demo演示

~/maskrcnn-benchmark/demo/my_demo.ipynb预测演示：其中使用mypredictor.py, 相比predictor.py修改了

```
CATEGORIES = [  
    "__background",  
    "car",  
    "dashedline",  
    "midlane",  
    "pothole",  
    "rightlane",  
]
```

物体分割的对应掩码颜色也做了修改。

另外，maskrcnn-benchmark/maskrcnn_benchmark/config下的defaults.py，更改下分类

_C.MODEL.ROI_BOX_HEAD.NUM_CLASSES = 5+1 #分类数量需要对应更改，默认81

建立maskrcnn-benchmark/configs/my_test_e2e_mask_rcnn_R_50_FPN_1x.yaml文件指定

WEIGHT: "/home/bai/maskrcnn-benchmark/weights/model_final.pth"

6.3 图片测试

```
python demo/seg_image.py --config-file  
configs/my_test_e2e_mask_rcnn_R_50_FPN_1x.yaml --input-file  
datasets/coco/val2017/img_val001.jpg --output-file demo/mypredictions.jpg
```

6.4 视频测试

```
python demo/seg_video.py --config-file  
configs/my_test_e2e_mask_rcnn_R_50_FPN_1x.yaml --input-file demo/drive.mp4
```