

# Swin Transformer实战-Ubuntu

课程演示环境: ubuntu, cuda 10.2

## 1 软件安装

### 1) 安装Anaconda

Anaconda 是一个用于科学计算的 Python 发行版, 支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

1. 先去官方地址下载好对应的安装包

下载地址:<https://www.anaconda.com/download/#linux>

2. 然后安装anaconda

```
bash ~/Downloads/Anaconda3-2021.05-Linux-x86_64.sh
```

anaconda会自动将环境变量添加到PATH里面, 如果后面你发现输入conda提示没有该命令, 那么你需要执行命令 `source ~/.bashrc` 更新环境变量, 就可以正常使用了。

如果发现这样还是没用, 那么需要添加环境变量。

编辑`~/.bashrc` 文件, 在最后面加上

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

注意: 路径应改为自己机器上的路径

保存退出后执行: `source ~/.bashrc`

再次输入 `conda list` 测试看看, 应该没有问题。

#### 添加Anaconda国内镜像配置

清华TUNA提供了 Anaconda 仓库的镜像, 运行以下三个命令:

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

### 2) 安装pytorch

首先创建一个anaconda虚拟环境, 环境名字可自己确定, 这里本人使用mypytorch作为环境名:

```
conda create -n mypytorch python=3.8
```

安装成功后激活mypytorch环境:

```
conda activate mypytorch
```

在所创建的pytorch环境下安装pytorch版本, 执行命令:

```
conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
```

注意：10.2处应为cuda的安装版本号

编辑~/.bashrc 文件，设置使用mypytorch环境下的python3.8

```
alias python='/home/bai/anaconda3/envs/mypytorch/bin/python3.8'
```

注意：python路径应改为自己机器上的路径

保存退出后执行：`source ~/.bashrc`

该命令将自动回到base环境，再执行 `conda activate mypytorch` 到pytorch环境。

## 2 Swin Transformer项目安装

### 1) 安装mmdcv

克隆mmdcv项目到本地

```
git clone -b v1.3.1 https://github.com/open-mmlab/mmdcv.git
```

```
cd mmdcv
MMCV_WITH_OPS=1 pip install -e .
```

检查

```
pip list
```

### 2) 安装mmdetection

参考：[https://mmdetection.readthedocs.io/zh\\_CN/v2.19.1/get\\_started.html](https://mmdetection.readthedocs.io/zh_CN/v2.19.1/get_started.html)

```
git clone -b v2.11.0 https://github.com/open-mmlab/mmdetection.git
cd mmdetection
pip install -r requirements/build.txt
pip install -v -e . # or "python setup.py develop"
```

### 3) 安装apex

```
git clone https://github.com/NVIDIA/apex
```

```
cd apex
```

```
python setup.py install
```

注意：cuda的版本应和cudatoolkit一致

## 4) 安装Swin-Transformer-Object-Detection

```
git clone https://github.com/SwinTransformer/Swin-Transformer-Object-Detection.git
```

执行

```
cd Swin-Transformer-Object-Detection
python setup.py develop
```

下载预训练权重文件

下载mask\_rcnn\_swin\_tiny\_patch4\_window7\_1x.pth权重文件，并放置在Swin-Transformer-Object-Detection文件夹下

**百度网盘下载链接：**

链接: [https://pan.baidu.com/s/14K\\_5UZLG9sPO\\_ZEe2-g1iQ](https://pan.baidu.com/s/14K_5UZLG9sPO_ZEe2-g1iQ)  
提取码: 2iuv

**测试命令：**

```
python demo/image_demo.py demo/demo.jpg
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
mask_rcnn_swin_tiny_patch4_window7_1x.pth
```

## 3 图像标注工具labelme的安装与使用

### 1) 安装图像标注工具labelme

Ubuntu下的安装：

```
pip install pyqt5
```

```
pip install labelme
```

如果安装过程中提示缺少某个包，可再安装上，如：

```
pip install pyyaml
```

如上述安装方法不能成功，使用下面的命令安装：

```
pip install git+https://github.com/wkentaro/labelme.git
```

## 2) 使用labelme进行图像标注

执行:

```
labelme
```

标注后生成json文件

课程pothole项目案例的数据集为1280\*720的图片, 136张用于训练, 16张用于测试。

课程中的数据集有5个类别: "car", "dashedline", "midlane", "pothole", "rightlane"

数据集图像文件放置在~/mydataset目录下

## 3) 图像标注后的数据转换

从百度网盘下载

- mydataset.tar.gz (下载并解压)

链接: [https://pan.baidu.com/s/14K\\_5UZLG9sPO\\_ZEe2-g1iQ](https://pan.baidu.com/s/14K_5UZLG9sPO_ZEe2-g1iQ)

提取码: 2iuv

把labelme标注的json数据格式转换成COCO数据格式的。

```
cd ~/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) 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Swin-Transformer-Object-Detection根目录下, 创建目录结构如下:

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

└─ train2017

└─ val2017

其中:

train2017放置roadscene\_train中的图片

val2017放置roadscene\_val中的图片

roadscene\_train.json改名为instances\_train2017.json

roadscene\_val.json改名为instances\_val2017.json

## 4 修改配置文件

---

1) 修改changemaskrcnn.py中num\_class并执行

产生新的权重文件

2) 修改 `configs\_base\_models\mask_rcnn_swin_fpn.py` 中num\_classes, 共两处

3) 修改 `configs\_base\_default_runtime.py` 中interval, load\_from

4) 修改 `configs\swin\mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py` 中的 max\_epochs, lr\_config

5) 修改 `configs\_base\_datasets\coco_instance.py` 中的

```
samples_per_gpu=2,
```

```
workers_per_gpu=0,
```

6) 修改mmdet\datasets\coco.py中的CLASSES

## 5 训练自己的数据集

---

从百度网盘下载文件

testfiles.zip (下载到Swin-Transformer-Object-Detection目录下并解压)

### 1) 训练命令

在Swin-Transformer-Object-Detection路径下执行:

```
python tools/train.py
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
```

### 2) 训练结果的查看

查看Swin-Transformer-Object-Detection/work\_dirs目录下的文件

## 6 测试训练出的网络模型

---

## 1) 测试图片

```
python demo/image_demo.py testfiles/img_val001.jpg
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/latest.pth
```

## 2) 测试视频

```
python demo/video_demo.py testfiles/drive.mp4
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/latest.pth --show
```

## 3) 性能统计

```
python tools/test.py configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/latest.pth --eval bbox segm
```

## 4) 日志分析

```
python tools/analysis_tools/analyze_logs.py plot_curve
work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/20211226_093405.log.json
```

(注意要指定自己的log.json文件)