

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

```
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/1c031n3fokP3cazveC-bjuw>
提取码：uzzk

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. 标注自己的数据集

1) 安装图像标注工具labelImg

克隆labelImg

```
git clone https://github.com/tzutalin/labelImg.git
```

使用Anaconda安装

到labelImg路径下执行命令

```
conda install pyqt=5
pip install lxml
pyrcc5 -o libs/resources.py resources.qrc
python labelImg.py
```

2) 添加自定义类别

修改文件labelImg/data/predefined_classes.txt

```
ball  
messi  
trophy
```

3) 使用labelImg进行图像标注

用labelImg标注生成PASCAL VOC格式的xml标记文件。例如：



width = 1000

height = 654

PASCAL VOC标记文件如下：

```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <annotation>
  <folder>bai</folder>
  <filename>trophy.jpg</filename>
  <path>/home/bai/trophy.jpg</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>1000</width>
    <height>654</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>trophy</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>187</xmin>
      <ymin>21</ymin>
      <xmax>403</xmax>
      <ymax>627</ymax>
    </bndbox>
  </object>
</annotation>
```

4 准备自己的数据集

4 准备自己的数据集

4 准备自己的数据集

1) 下载项目文件

从百度网盘下载

- VOCdevkit_bm.zip (下载到并解压)
- prepare_mydata.py
- pascal_voc.py
- testfiles.zip (下载到Swin-Transformer-Object-Detection目录下并解压)

链接: <https://pan.baidu.com/s/1c03ln3fokP3cazveC-bjuw>
提取码: uzzk

2) 解压建立或自行建立数据集

使用PASCAL VOC数据集的目录结构:

建立文件夹层次为 VOCdevkit / VOC2007

VOC2007下面建立三个文件夹: Annotations, JPEGImages和ImageSets/Main

JPEGImages放所有的数据集图片; Annotations放所有的xml标记文件; ImageSets/Main下存放训练集、验证集、测试集划分文件 (目前为空)

3) 划分训练集和验证集/测试集

执行python脚本:

```
python prepare_mydata.py
```

注意根据自己的数据集修改其中的classes=["ball", "messi"]

ImageSets/Main目录下可以看到生成四个文件

- train.txt给出了训练集图片文件的列表 (不含文件名后缀)
- val.txt给出了验证集图片文件的列表
- test.txt给出了测试集图片文件的列表
- trainval.txt给出了训练集和验证集图片文件的列表

4) PASCAL VOC数据格式转换成COCO数据格式

执行python脚本:

```
python pascal_voc.py VOCdevkit --out-dir mycoco --out-format coco
```

注意根据自己的数据集修改其中的voc_classes = ["ball", "messi"]

然后, 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voc-images/train2017中的图片

val2017放置voc-images/val2017中的图片

voc07_train.json改名为instances_train2017.json

voc07_val.json改名为instances_val2017.json

5 修改配置文件

- 1) 修改changemaskrcnn.py中num_class并执行该脚本, 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_base_, max_epochs, lr
- 5) 修改 configs_base_datasets\coco_detection.py 中的

```
samples_per_gpu=2,  
workers_per_gpu=2,
```

- 6) 修改mmdet\datasets\coco.py中的CLASSES

另外, 如果在训练时出现错误: IndexError: list index out of range

注释掉coco.py中_segmask2json(self, results)函数中的以下语句 (268~271行)

```
#if isinstance(segms[i]['counts'], bytes):  
#    segms[i]['counts'] = segms[i]['counts'].decode()  
#data['segmentation'] = segms[i]  
#segm_json_results.append(data)
```

6 训练自己的数据集

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

7 测试训练出的网络模型

1) 测试图片

```
python demo/image_demo.py testfiles/img1.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 --score-thr 0.5
```

2) 测试视频

```
python demo/video_demo.py testfiles/messi.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 --score-thr 0.5 --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
```

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/20211225_030436.log.json
```