

# 学生实验报告

学号	1120192933	学院	计算机学院
姓名	李桐	专业	人工智能

## mmdetection 操作实验

### 1 实验目的

- (1) 理解 mmdetection 框架的特点。
- (2) 掌握 mmdetection 的使用方法。

### 2 实验原理

- (1) mmdetection 操作
- (2) pytorch 安装
- (3) mmcv 安装

### 3 实验条件与环境

要求	名称	版本要求	备注
编程语言	python	3.6 以上	
开发环境	dsw	无要求	
第三方工具包/库/插件	opencv-python	4.5 以上	
第三方工具包/库/插件	tqdm	4.32	
第三方工具包	pytorch	1.0 以上	

/库/插件			
第三方工具包	mmdetection	1.2	
/库/插件			
其他工具	无	无要求	
硬件环境	台式机、笔记本均可	无要求	

## 4 实验步骤及操作

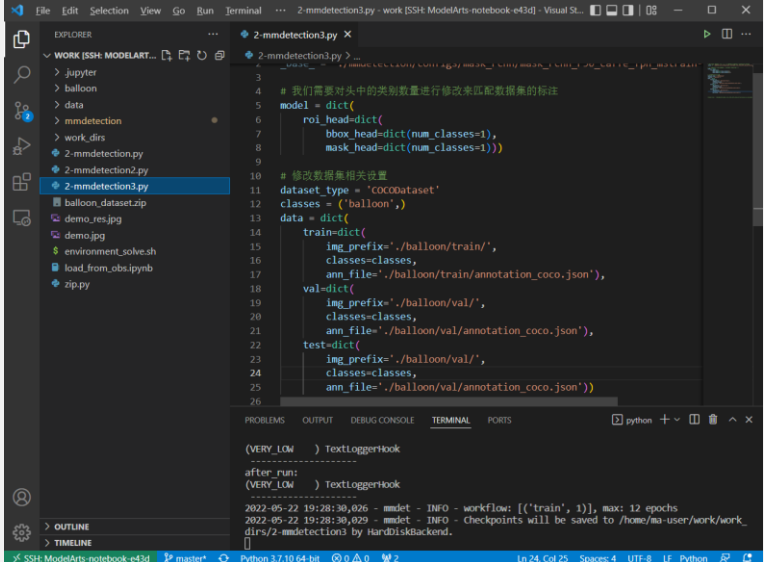
步骤序号	1
步骤名称	下载 checkpoint
步骤描述	通过 wget 下载需要用到的预训练好的模型
代码及讲解	<p>这里没有使用代码，直接在网上找到了这个预训练模型，然后下载即可，地址为：</p> <p><a href="https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth">https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_coco/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth</a></p>

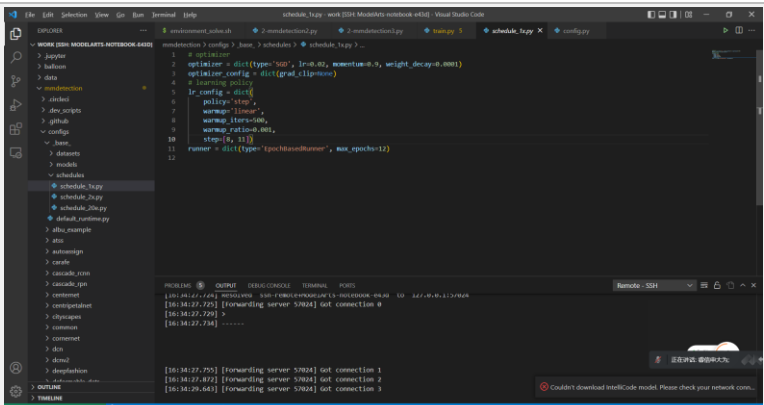
步骤序号	2
步骤名称	运行 demo
步骤描述	构建 demo 模型，并测试。
代码及讲解	<pre> 1 # mmdetection.py 2 from mmdet.apis import init_detector, inference_detector 3 4 def demo_mmdet(): 5     # mmdetection 的安装目录 6     base_dir = os.path.join('C:\Program Files\Third Party Lib\mmdetection') 7     config_file = os.path.join(base_dir, 'configs/faster_rcnn/faster_rcnn_r50_fpn_1x_coco.py') 8     # download the checkpoint from model zoo and put it in 'checkpoints/' 9     # url: https://download.openmmlab.com/mmdetection/v2.0/faster_rcnn/faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth 10    checkpoint_file = os.path.join('checkpoints', 'faster_rcnn_r50_fpn_1x_coco_20200130-047c8118.pth') 11 12    # 根据配置文件和 checkpoint 文件构建模型 13    model = init_detector(config_file, checkpoint_file, device='cpu') 14 15    # 测试单张图片并展示结果 16    img = 'demo.jpg' # 或者 img = mmcv.imread(img). 这样图片只会被读一次 17    result = inference_detector(model, img) 18    # 在一个新的窗口中展示结果可视化 19    model.show_result(img, result, out_file='demo_res.jpg', show=True) 20 21 22 # 在标准数据集上测试 23 24 if __name__ == '__main__': 25     demo_mmdet() 26 </pre> <p>根据配置文件和 checkpoint 文件构建模型，随便再照一张图片进行测试即可。</p>

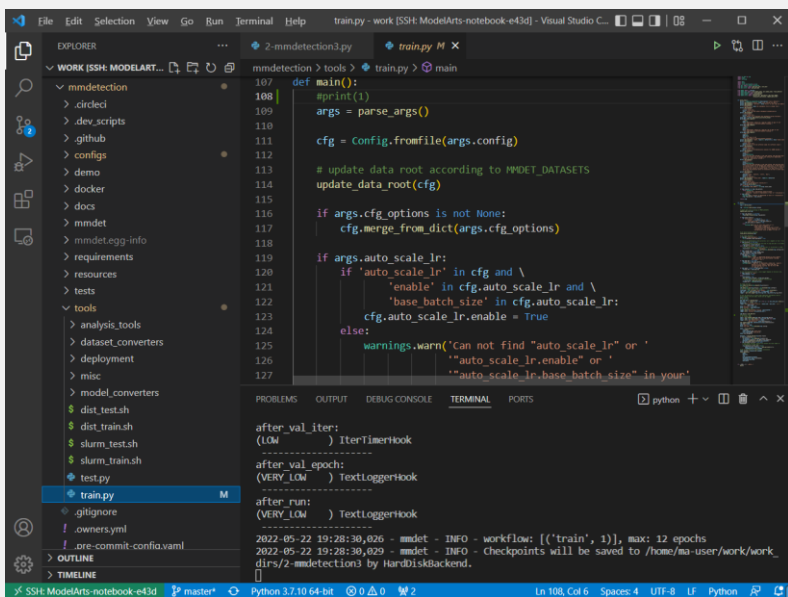
步骤序号	3
步骤名称	下载数据集
步骤描述	下载气球数据集，为之后的转换模型类型做准备。
代码及讲解	不需要代码，直接在在 <a href="https://github.com/matterport/Ma">https://github.com/matterport/Ma</a>

sk\_RCNN/releases 下载即可。

步骤序号	4
步骤名称	构建数据集
步骤描述	将数据集的组织形式，修改为 coco 数据集的形式。
代码及讲解	<pre> height=height, width=width))  bboxes = [] labels = [] masks = [] for _, obj in v['regions'].items():     assert not obj['region_attributes']     obj = obj['shape_attributes']     px = obj['all_points_x']     py = obj['all_points_y']     poly = [(x + 0.5, y + 0.5) for x, y in zip(px, py)]     poly = [p for x in poly for p in x]      x_min, y_min, x_max, y_max = (         min(px), min(py), max(px), max(py))      data_anno = dict(         image_id=idx,         id=obj_count,         category_id=0,         bbox=[x_min, y_min, x_max - x_min, y_max - y_min],         area=(x_max - x_min) * (y_max - y_min),         segmentation=[poly],         iscrowd=0)     annotations.append(data_anno)     obj_count += 1  coco_format_json = dict(     images=images,     annotations=annotations,     categories=[{'id': 0, 'name': 'balloon'}]) </pre> <p>其实就是根据原来的数据集的格式进行读取，然后再重新保存为 coco 数据集的形式。</p>

步骤序号	5
步骤名称	修改模型参数
步骤描述	修改模型参数，适应电脑配置与实验。
代码及讲解	 <pre> # 我们需要对头中的类别数量进行修改来匹配数据集的标注 model = dict(     roi_head=dict(         bbox_head=dict(num_classes=1),         mask_head=dict(num_classes=1)))  # 修改数据集相关设置 dataset_type = 'COCOdataset' classes = ('balloon',) data = dict(     train=dict(         img_prefix='./balloon/train/',         classes=classes,         ann_files='./balloon/train/annotation_coco.json'),     val=dict(         img_prefix='./balloon/val/',         classes=classes,         ann_files='./balloon/val/annotation_coco.json'),     test=dict(         img_prefix='./balloon/val/',         classes=classes,         ann_files='./balloon/val/annotation_coco.json')) </pre> <p>先继承默认的配置，然后再根据实验的配置进行修改。</p>

	 <p>最后还可以在 schedule 中修改模型参数。</p>
--	--

步骤序号	6
步骤名称	模型训练
步骤描述	训练模型
代码及讲解	 <p>调用 mmdetection 文件夹下 tools 文件夹中的 train 文件，在终端输入，python mmdetection/tools/train.py 2-mmdetection3.py，即运行模型、同时指定了第 5 步的设置作为模型的配置。</p>

步骤序号	7
步骤名称	模型预测
步骤描述	用图片或者测试集进行预测。
代码及讲解	<p>用图片进行预测，可以这样:python mmdetection/demo/image_demo.py balloon/val/410488422_5f8991f26e_b.jpg 2-mmdetection3.py work_dirs/2-mmdetection3/latest.pth -out-file</p>

= '2-3.jpg' 依次指定运行的模型、预测的图片、配置文件、模型的位置以为输出的图片的位置。

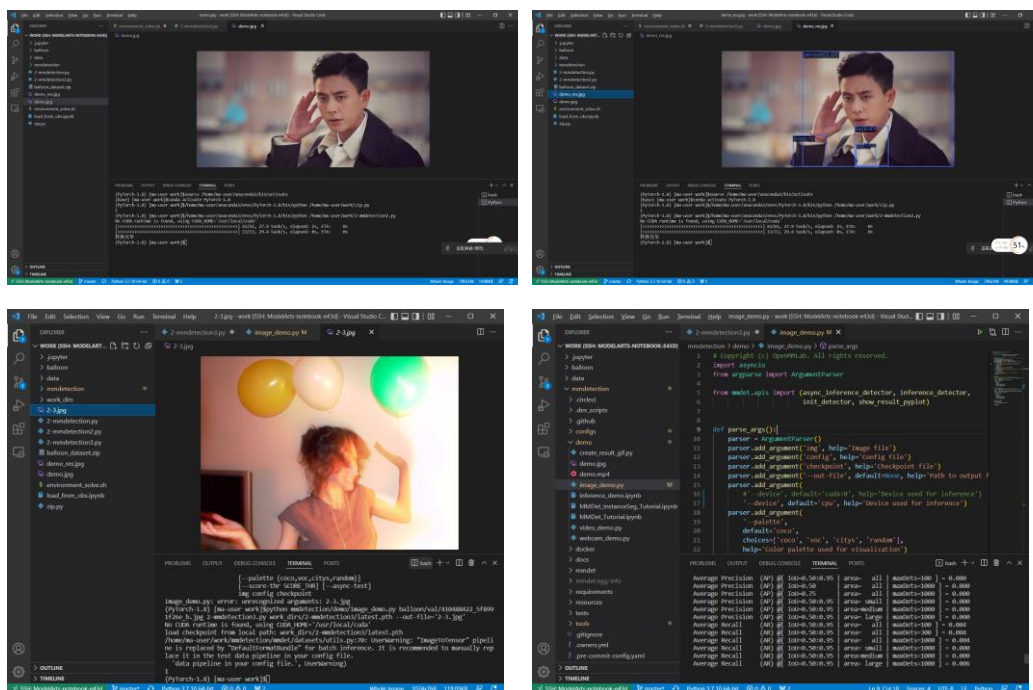
用测试集进行预测，可以这样: `python mmdetection/tools/test.py 2-mmdetection3.py work_dirs/2-mmdetection3/latest.pth --eval bbox segm`

## 5 实验结果及分析讨论

### (1) 最终结果的具体结果（文字说明）

框架安装成功，同时模型能够运行成功、顺利预测。

### (2) 最终结果界面截图（界面截图）



### (3) 最终结果的说明（注意事项或提醒）

第一行是 demo 模型的原图和结果。第二行是后面自己数据集(气球数据集)的结果以及测试集结果。

同时遇到了这样的问题:

```
module._apply(fn)
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/torch/nn/modules/module.py", line 409, in _apply
    param_applied = fn(param)
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/torch/nn/modules/module.py", line 671, in convert
    return t.to(device, dtype if t.is_floating_point() or t.is_complex() else None, non_blocking)
File "/home/ma-user/anaconda3/envs/PyTorch-1.8/lib/python3.7/site-packages/torch/cuda/__init__.py", line 170, in lazy_init
    torch._C._cuda_init()
RuntimeError: Found no NVIDIA driver on your system. Please check that you have an NVIDIA GPU and installed a driver from http://www.nvidia.com/Download/index.aspx
(PyTorch-1.8) [ma-user work]$
```

这个问题是在气球数据集上用单张图片测试的时候遇到的，开的环境是 cpu 的，但是默认的配置是 cuda 上，所以会出现问题。去修改 image\_demo.py 即可。

#### (4) 最终结果的解读与讨论

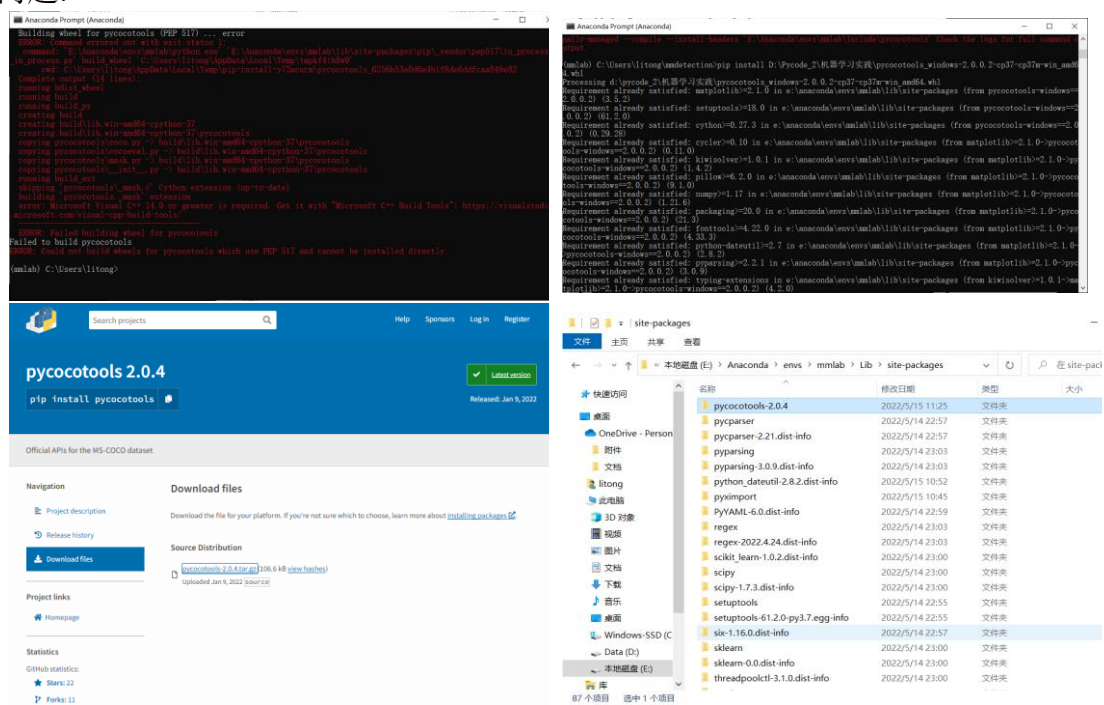
结果很正常，可以跑，没啥问题。

第一行的 demo 结果预测得还蛮差得哈哈哈哈哈，这这个图片是在浏览器里面随便拿了一个图片。把手腕预测成了被子，领子预测成了领结，但是就露出来一点点得领结居然能成功预测，也挺神奇的。

第二行的是在气球数据集上，无论是单张图片还是测试集的效果都不好。应该是模型模型训练不到位。我后来发现默认的配置是只训练 12 个 epoch，效果不好也是情理之中。

## 6 收获与体会

这个实验其实首先安装了 mmdetection。首先在 annaconda 建立了一个虚拟环境，安装了 3.7 的 python，之前安装了个什么库，然后把我的环境弄坏了，从那之后安装新库尽量弄个新环境。然后下载了 1.5 的 pytorch，这些都是按照老师给的实验环境来的。但是可能是 mmdeteciton 不支持低版本了，所以 mmcv==0.5.9 一直安装不上。最后直接 pip install mmcv，自己安装了一个能安装上的版本。一开始实验的时候 demo 运行不了，后来又安装了 mmdet。结果遇到这样的问题：



看了好多解决方法，也没解决成功。比如一个方法是：pip install pycocotools-windows -i https://pypi.tuna.tsinghua.edu.cn/simple，不过还是显示没有 pycocotools，之后上了清华镜像，下载了轮子，然后直接安装，但是显示已经安装.....最后采用了最粗暴的方法，直接把源文件下载下来，解压之后复制文件到虚拟环境的包文件夹里面。

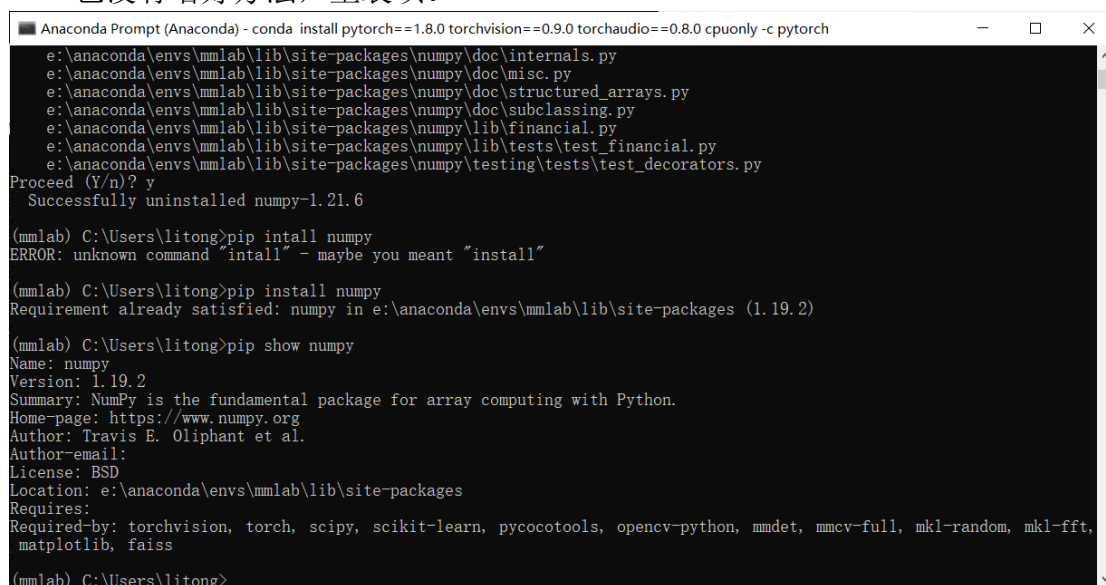


结果运行 demo 又遇到了新问题。不知道是那一步安装的时候出现了问题，好气啊。

```
E:\Anaconda\envs\mmlab\python.exe D:/Pycode_2/机器学习实践/天池3/2-mmdetection.py
Traceback (most recent call last):
  File "D:/Pycode_2/机器学习实践/天池3/2-mmdetection.py", line 2, in <module>
    from mmdet.apis import init_detector, inference_detector
  File "c:\users\litong\mmdetection\mmdet\apis\_init_.py", line 2, in <module>
    from .inference import (async_inference_detector, inference_detector,
  File "c:\users\litong\mmdetection\mmdet\apis\inference.py", line 8, in <module>
    from mmdcv.ops import RoIPool
  File "E:\Anaconda\envs\mmlab\lib\site-packages\mmdcv\ops\_init_.py", line 2, in <module>
    from .active_rotated_filter import active_rotated_filter
  File "E:\Anaconda\envs\mmlab\lib\site-packages\mmdcv\ops\active_rotated_filter.py", line 10, in <module>
    ['active_rotated_filter_forward', 'active_rotated_filter_backward'])
  File "E:\Anaconda\envs\mmlab\lib\site-packages\mmdcv\utils\ext_loader.py", line 13, in load_ext
    ext = importlib.import_module('mmdcv.' + name)
  File "E:\Anaconda\envs\mmlab\lib\importlib\_init_.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
ImportError: DLL load failed: 找不到指定的模块。
```

进程已结束,退出代码1

也没有啥好办法，重装呗。



```
Anaconda Prompt (Anaconda) - conda install pytorch==1.8.0 torchvision==0.9.0 torchaudio==0.8.0 cpuonly -c pytorch
e:\anaconda\envs\mmlab\lib\site-packages\numpy\doc\internals.py
e:\anaconda\envs\mmlab\lib\site-packages\numpy\doc\misc.py
e:\anaconda\envs\mmlab\lib\site-packages\numpy\doc\structured_arrays.py
e:\anaconda\envs\mmlab\lib\site-packages\numpy\doc\subclassing.py
e:\anaconda\envs\mmlab\lib\site-packages\numpy\lib\financial.py
e:\anaconda\envs\mmlab\lib\site-packages\numpy\lib\tests\test_financial.py
e:\anaconda\envs\mmlab\lib\site-packages\numpy\testing\tests\test_decorators.py
Proceed (Y/n)? y
Successfully uninstalled numpy-1.21.6

(mmlab) C:\Users\litong>pip intall numpy
ERROR: unknown command "intall" - maybe you meant "install"

(mmlab) C:\Users\litong>pip install numpy
Requirement already satisfied: numpy in e:\anaconda\envs\mmlab\lib\site-packages (1.19.2)

(mmlab) C:\Users\litong>pip show numpy
Name: numpy
Version: 1.19.2
Summary: NumPy is the fundamental package for array computing with Python.
Home-page: https://www.numpy.org
Author: Travis E. Oliphant et al.
Author-email:
License: BSD
Location: e:\anaconda\envs\mmlab\lib\site-packages
Requires:
Required-by: torchvision, torch, scipy, scikit-learn, pycocotools, opencv-python, mmdet, mmdcv-full, mkl-random, mkl-fft, matplotlib, faiss

(mmlab) C:\Users\litong>
```

安了好几天也安不好，烦死了。后来问了一下张至鑫同学，在云平台上弄的。于是我也开始在 smart art 上部署了，确实还是挺方便的。另外感谢一下张至鑫，没有他我可能得继续安装好久！

## 7 备注及其他

无。