

CNN系列 二 (Datasets代码, python)

于 2021-02-05 22:22:02 发布 131 收藏

版权

深度学习 python

系列 二 (Datasets代码, python)

继承Dataset, 自己主要实现__len__和__getitem__模块;

1, 主要目的是定义数据地址, 将数据初始化, __init__模块主要包含图像,xml文件, train_txt和test_txt文件的路径

```
__init__(self, voc_root, transforms, txt_name: str = "train.txt"):
    self.root = os.path.join(voc_root, "VOCdevkit", "VOC2012")
    self.img_root = os.path.join(self.root, "JPEGImages")
    self.annotations_root = os.path.join(self.root, "Annotations")

    # read train.txt or val.txt file
    path = os.path.join(self.root, "ImageSets", "Main", txt_name)
    if not os.path.exists(path):
        raise ValueError("not found {} file.".format(txt_name))

    with open(path) as read:
        self.xml_list = [os.path.join(self.annotations_root, line.strip() + ".xml")
                        for line in read.readlines()]

    # check file
    if len(self.xml_list) > 0:
        if not os.path.exists(self.xml_list[0]):
            raise ValueError("not found {} file.".format(self.xml_list[0]))

    # load class dict
    json_file = open(os.path.join(self.root, "pascal_voc_classes.json"), 'r')
    self.class_dict = json.load(json_file)
    except Exception as e:
        print(e)
        exit(-1)

    self.transforms = transforms
```

2, 主要目的是返回数据长度, xml文件包含了全部的数据, 直接返回长度就好了

```
__len__(self):
    return len(self.xml_list)
```

__getitem__模块, 主要将图像数据与标注好的target信息返回, 此处需要实现parse_xml_to_dic方法, 主要目的是以字典形式存储起来

xml 文件拿出来, 以etree的形式去读取xml文件中的所有数据

```
__getitem__(self, idx):
    # load xml
    path = self.xml_list[idx]
    with open(path) as fid:
        xml_str = fid.read()
    root = etree.fromstring(xml_str)
    # parse xml to dict
    data = self.parse_xml_to_dic(root)["annotation"]
    path = os.path.join(self.img_root, data["filename"])
    img = Image.open(path)
    if img.format != "JPEG":
        raise ValueError("Image format not JPEG")

    # load bounding box
    bboxes = []
    labels = []
    for obj in root.findall("object"):
        xmin = float(obj["bndbox"]["xmin"])
```



一直小老虎

关注

```

xmax = float(obj["bndbox"]["xmax"])
ymin = float(obj["bndbox"]["ymin"])
ymax = float(obj["bndbox"]["ymax"])
boxes.append([xmin, ymin, xmax, ymax])
labels.append(self.class_dict[obj["name"]])
iscrowd.append(int(obj["difficult"]))
nvert everything into a torch.Tensor
s = torch.as_tensor(boxes, dtype=torch.float32)
ls = torch.as_tensor(labels, dtype=torch.int64)
owd = torch.as_tensor(iscrowd, dtype=torch.int64)
e_id = torch.tensor([idx])
    = (boxes[:, 3] - boxes[:, 1]) * (boxes[:, 2] - boxes[:, 0])

```

```

et = {}
et["boxes"] = boxes
et["labels"] = labels
et["image_id"] = image_id
et["area"] = area
et["iscrowd"] = iscrowd

```

```

elf.transforms is not None:
image, target = self.transforms(image, target)

```

```

rn image, target

```

to_dic方法, 将xml文件的信息以字典的形式返回

```

e_xml_to_dict(self, xml):
en(xml) == 0: # 遍历到底层, 直接返回tag对应的信息
return {xml.tag: xml.text}

lt = {}
child in xml:
child_result = self.parse_xml_to_dict(child) # 递归遍历标签信息
if child.tag != 'object':
    result[child.tag] = child_result[child.tag]
else:
    if child.tag not in result: # 因为object可能有多个, 所以需要放入列表里
        result[child.tag] = []
    result[child.tag].append(child_result[child.tag])
rn {xml.tag: result}

```

```

ch.utils.data import Dataset
s
orch
son
    import Image
l import etree

```

```

C2012DataSet(Dataset):
__init__(self, voc_root, transforms, txt_name:str = "train.txt"):
self.root = os.path.join(voc_root,"VOCdevkit", "VOC2012")
self.img_root = os.path.join(self.root, "JPEGImages")
self.annotation_root = os.path.join(self.root, "Annotations")

txt_path = os.path.join(self.root, "ImageSets", "Main", txt_name)
assert os.path.exists(txt_path), "not found {} file.".format(txt_name)
with open(txt_path) as read:
    self.xml_list = [os.path.join(self.annotation_root, line.split() + '.xml')
        for line in read.readlines()]
assert len(self.xml_list) > 0,"in '{}' file does not found any information.".format(txt_path)
for xml_path in self.xml_list:
    assert os.path.exists(xml_path), "not found '{}' file".format(xml_path)
try:
    json_file = open('./pascal_voc_class.json', 'r')
    self.class_dict = json.load(json_file)
except Exception as e:
    print(e)
    exit(-1)

```



一直小老虎

关注

```

self.transforms = transforms
C2012DataSet(Dataset):
    取解析PASCAL VOC2012数据集"""

    __init__(self, voc_root, transforms, txt_name: str = "train.txt"):
        self.root = os.path.join(voc_root, "VOCdevkit", "VOC2012")
        self.img_root = os.path.join(self.root, "JPEGImages")
        self.annotations_root = os.path.join(self.root, "Annotations")

    # read train.txt or val.txt file
    txt_path = os.path.join(self.root, "ImageSets", "Main", txt_name)
    assert os.path.exists(txt_path), "not found {} file.".format(txt_name)

    with open(txt_path) as read:
        self.xml_list = [os.path.join(self.annotations_root, line.strip() + ".xml")
                        for line in read.readlines()]

    # check file
    assert len(self.xml_list) > 0, "in '{}' file does not find any information.".format(txt_path)
    for xml_path in self.xml_list:
        assert os.path.exists(xml_path), "not found '{}' file.".format(xml_path)

    # read class_indict
    try:
        json_file = open('./pascal_voc_classes.json', 'r')
        self.class_dict = json.load(json_file)
    except Exception as e:
        print(e)
        exit(-1)

    self.transforms = transforms

    __len__(self):
    return len(self.xml_list)

    __getitem__(self, idx):
    # read xml
    xml_path = self.xml_list[idx]
    with open(xml_path) as fid:
        xml_str = fid.read()
    xml = etree.fromstring(xml_str)
    data = self.parse_xml_to_dict(xml)["annotation"]
    img_path = os.path.join(self.img_root, data["filename"])
    image = Image.open(img_path)
    if image.format != "JPEG":
        raise ValueError("Image format not JPEG")
    boxes = []
    labels = []
    iscrowd = []
    for obj in data["object"]:
        xmin = float(obj["bndbox"]["xmin"])
        xmax = float(obj["bndbox"]["xmax"])
        ymin = float(obj["bndbox"]["ymin"])
        ymax = float(obj["bndbox"]["ymax"])
        boxes.append([xmin, ymin, xmax, ymax])
        labels.append(self.class_dict[obj["name"]])
        iscrowd.append(int(obj["difficult"]))
    # convert everything into a torch.Tensor
    boxes = torch.as_tensor(boxes, dtype=torch.float32)
    labels = torch.as_tensor(labels, dtype=torch.int64)
    iscrowd = torch.as_tensor(iscrowd, dtype=torch.int64)
    image_id = torch.tensor([idx])
    area = (boxes[:, 3] - boxes[:, 1]) * (boxes[:, 2] - boxes[:, 0])

    target = {}
    target["boxes"] = boxes
    target["labels"] = labels
    target["image_id"] = image_id
    target["area"] = area
    target["iscrowd"] = iscrowd

```



一直小老虎

关注

```
if self.transforms is not None:
    image, target = self.transforms(image, target)

return image, target

def get_height_and_width(self, idx):
    # read xml
    xml_path = self.xml_list[idx]
    with open(xml_path) as fid:
        xml_str = fid.read()
    xml = etree.fromstring(xml_str)
    data = self.parse_xml_to_dict(xml)["annotation"]
    data_height = int(data["size"]["height"])
    data_width = int(data["size"]["width"])
    return data_height, data_width

def parse_xml_to_dict(self, xml):
    """
    将xml文件解析成字典形式, 参考tensorflow的recursive_parse_xml_to_dict
    Args:
        xml: xml tree obtained by parsing XML file contents using lxml.etree

    Returns:
        Python dictionary holding XML contents.
    """

    if len(xml) == 0: # 遍历到底层, 直接返回tag对应的信息
        return {xml.tag: xml.text}

    result = {}
    for child in xml:
        child_result = self.parse_xml_to_dict(child) # 递归遍历标签信息
        if child.tag != 'object':
            result[child.tag] = child_result[child.tag]
        else:
            if child.tag not in result: # 因为object可能有多个, 所以需要放入列表里
                result[child.tag] = []
            result[child.tag].append(child_result[child.tag])
    return {xml.tag: result}
```

与50位技术专家面对面

20年技术见证, 附赠技术全景图

>

实战案例】虎年画虎：绘制老虎Python代码	01-26
案例】虎年画虎：绘制老虎Python代码	
只可爱的小老虎，基于turtle画的老虎 最新发布	01-25
可爱的小老虎，祝你虎年大吉！	
代码理解详细(Python)_laxehr的博客	3-9
是用ImageNet的模型M0来Finetuning RPN网络得到模型M1。以训练为例,这里的args参数都在脚本 experiments/sc...	
RCNN代码理解(Python)_lanyuxuan100的博客	3-12
是用ImageNet的模型M0来Finetuning RPN网络得到模型M1。以训练为例,这里的args参数都在脚本 experiments/sc...	
用Pascal VOC2007+2012联合训练	Ly_bupt的博客 3659
数据集 wget http://host.robots.ox.ac.uk/pascal/VOC/voc2007/VOCtrainval_06-Nov-2007.tar wget http://host.robots.o...	
训练自己的数据集遇到问题总结	南洲.的博客 430
odule' object has no attribute 'text_format' 解决办法： 打开py-faster-rcnn/lib/fast_rcnn/train.py增加一行import googl...	

5——Faster R-CNN2(代码实现)_xiaotig的...

tensorflowstffromdetection.models.backbonesimportresnetfromdel

一直小老虎

关注

代码_深度学习目标检测系列:faster RCNN实... 3-17
CNN系列算法基本原理基础上,使用keras实现faster RCNN算法,在细胞检测任务上表现优异,可动手操作一下。目标检...

(Pytorch版本) 代码及理论笔记 weixin_43490422的博客 423
p主霹雳吧啦Wz的视频讲解,对Faster Rcnncnn代码的流程以伪代码的形式描述出来,同时中间插入了PASCAL VOC20...

python的笔记 qq_42404495的博客 103
| hello word #!/usr/bin/python #coding=utf-8 print("hello word") 规则 标识符由字母、数字、下划线组成,但不能以数...

库turtle画一只老虎,祝您新年虎虎生威,大吉大利 weixin_43790276的博客 9133
le画一只老虎,祝您新年虎虎生威,大吉大利! 虎年到了,祝大家新年虎虎生威! 大吉大利! 去年我用Python画了...

之路 qq_42404495的博客 102
第1天(只是想记录一下学习python的过程让自己通过写文章的方式记录下来) 安装python环境 安装python环境 ti...

用的代码 (持续更新) 进击的菜鸟 482
2012数据集代码 from torch.utils.data import Dataset import os import torch import json from PIL import Image from ...

处理之FastRCNN (pytorch) 代码解读之my_dataset.py com_fang_bean的博客 401
rc_root, transforms, train_set=True):-》 voc_root训练集所在根目录, transforms预处理方法, train_set boolean变量 ...

IN开始学python 篇外卷datasets (二) 白色字体的博客 355
第3章使用一些额外的程序,类等等,为了使文章整体内容不过于分散,这里作为篇外卷学习。文件地址: \py-faster-rcnn...

代码理解 (Python) weixin_30572613的博客 344
cool.net/kb/Python/201611/209696.html#原文地址 第一步,准备从train_faster_rcnn_alt_opt.py入: 初始化参数: a...

ar_rcnn代码的使用 (python+caffe版本) Lin_xiaoyi的博客 4882
rcnn代码链接: https://pan.baidu.com/s/1eS8JclY 密码: mqrh 论文作者源码下载: git clone --recursive https://github.co...

_TF代码解读2: datasets/factory.py 一呆飞仙的博客 1892
----- # Fast R-CNN # Copyright (c) 2015 Microsoft # Licensed under The MIT License [see LIC...

olo V1 逐行逐句解读 qq1483661204的博客 9415
1 逐行逐句解读 博主最近研究完faster-rcnn,读完每一行代码,那叫一个算爽,现在开始读tensorflow版的Yolo V...

方式,生成训练train.txt和vaild.txt baidu_40840693的博客 534
://github.com/yangninghua/deeplearning_backbone # coding:utf-8 """ 将cifar10的数据_batch_12345 转换成 png格...

门到精通——VOC 2012数据读取(pytorch) swpucwf的博客 70
1 import Dataset import os import torch import json from PIL import Image from lxml import etree class VOC2012Dat...

自己的数据集之问题解决篇 qq_37874136的博客 8447
tu + Tesla m40 24GB + cuda10.2 + anaconda 感谢小程的热心帮助~ 1、ValueError: setting an array element with a ...

2012 (Faster RCNN) 平凡简单的执着 1036
iter RCNN的基本原理和思路,原文地址为知乎 Faster RCNN基本结构 经过R-CNN和Fast RCNN的积淀, Ross B. Gi...

“相关推荐”对你有帮助么?

非常没帮助 没帮助 一般 有帮助 非常有帮助

©2022 CSDN 皮肤主题: 大白 设计师: CSDN官方博客 返回首页

门 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00
11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心
络110报警服务 中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新乐知网络技术有限公司 版权与免责声明 版权申诉
出版物许可证 营业执照



一直小老虎
码龄3年 暂无认证

8	111万+	137万+	3324	
原创	周排名	总排名	访问	等级
91	6	3	4	5
积分	粉丝	获赞	评论	收藏



一直小老虎

关注

私信

关注

搜博主文章



热门文章

- DeepGlint AI编程练习赛——匹配先验框(详解, python) 639
- DeepGlint AI编程练习赛——对抗性攻击(详解, python) 587
- DeepGlint AI编程练习赛 - 智慧星球武器 (详解, python) 546
- 百度感知算法实习生面试之路 (已录用) 542
- 深度学习-目标检测 (评价指标) 228

最新评论

- DeepGlint AI编程练习赛——匹配先验框(... Extreme_ways: 没有去重吧?
- DeepGlint AI编程练习赛——匹配先验框(... hopeless_life: 多少分能过笔试啊
- 百度感知算法实习生面试之路 (已录用) DLANDML: 祝贺大佬, 我一直在关注你
- Faster-RCNN系列 二 (Datasets代码, p... 盼盼编程: 学到了, 收藏一波~欢迎回访一起交流!

您愿意向朋友推荐“博客详情页”吗?



强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

- 百度感知算法实习生面试之路 (已录用)
- Faster-RCNN系列 三 (Framework代码, python)
- 深度学习-目标检测 (评价指标)

2021年 8篇



一直小老虎

关注