YOLOv3目标检测算法

目标检测任务与数据集

@tm9161

L1

目标检测任务与数据集

1. 目标分类与目标检测
2. PASCAL VOC 数据集
3. 制作自己的数据集

实验环境

OS: Ubuntu 20.04
CUDA: 11.1
cuDNN: 8.0.5
TensorFlow-GPU : 2.4.0 (建议TF 2.2.0以上)

编译器: Jupyter Notebook VScode
包管理工具: Anaconda

TensorFlow安装: https://www.bilibili.com/read/cv9003982
*pip安装时候可以 使用 ==2.4.0 指定TF版本

遥感图像分类 1.5 ResNet

3376 2021-10-2

遥感图像分类 1.4 GoogLeNet

1476 2021-7-9

遥感图像分类 1.3 VGGNet

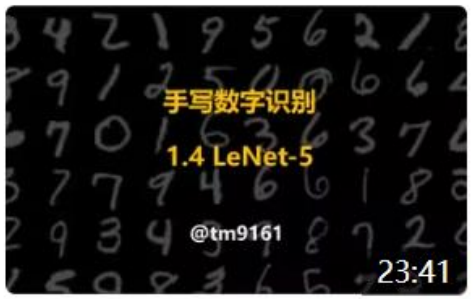
2158 2021-5-19

遥感图像分类 1.2 AlexNet

2287 2021-4-27

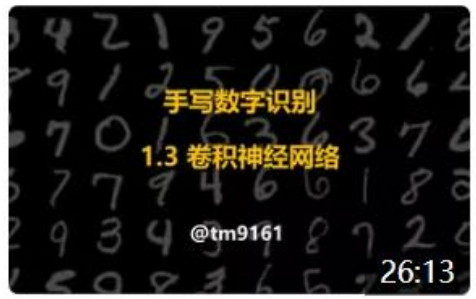
遥感图像分类 1.1 LeNet-5

3899 2021-4-9



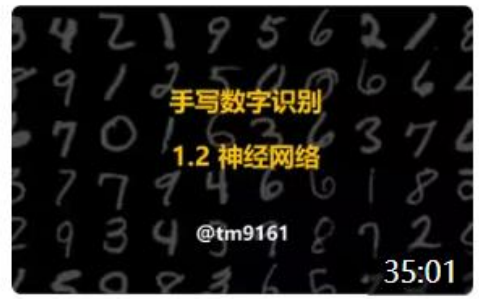
手写数字识别 1.4 LeNet-5

1.1万 2021-3-30



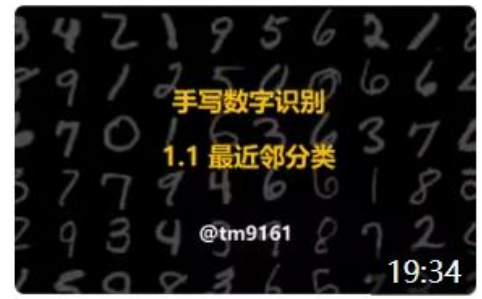
手写数字识别 1.3 卷积神经网络

5444 2021-3-23



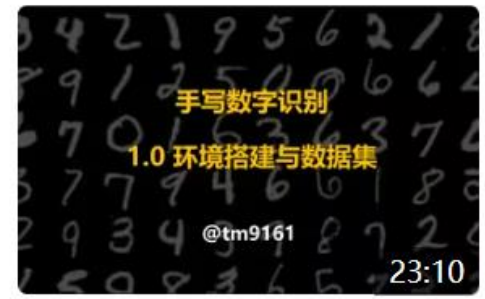
手写数字识别 1.2 神经网络

4030 2021-3-8



手写数字识别 1.1 最近邻分类

5214 2021-2-26



手写数字识别 1.0 环境搭建与数据集

9324 2021-2-22

目标分类任务

输入图片-网络模型-类别

目标检测任务

输入图片-网络模型-类别+**预测框**

基于机器学习的目标检测任务流程

PASCAL VOC 挑战赛

Development Kit

The development kit provided for the VOC challenge 2007 is available. You can:

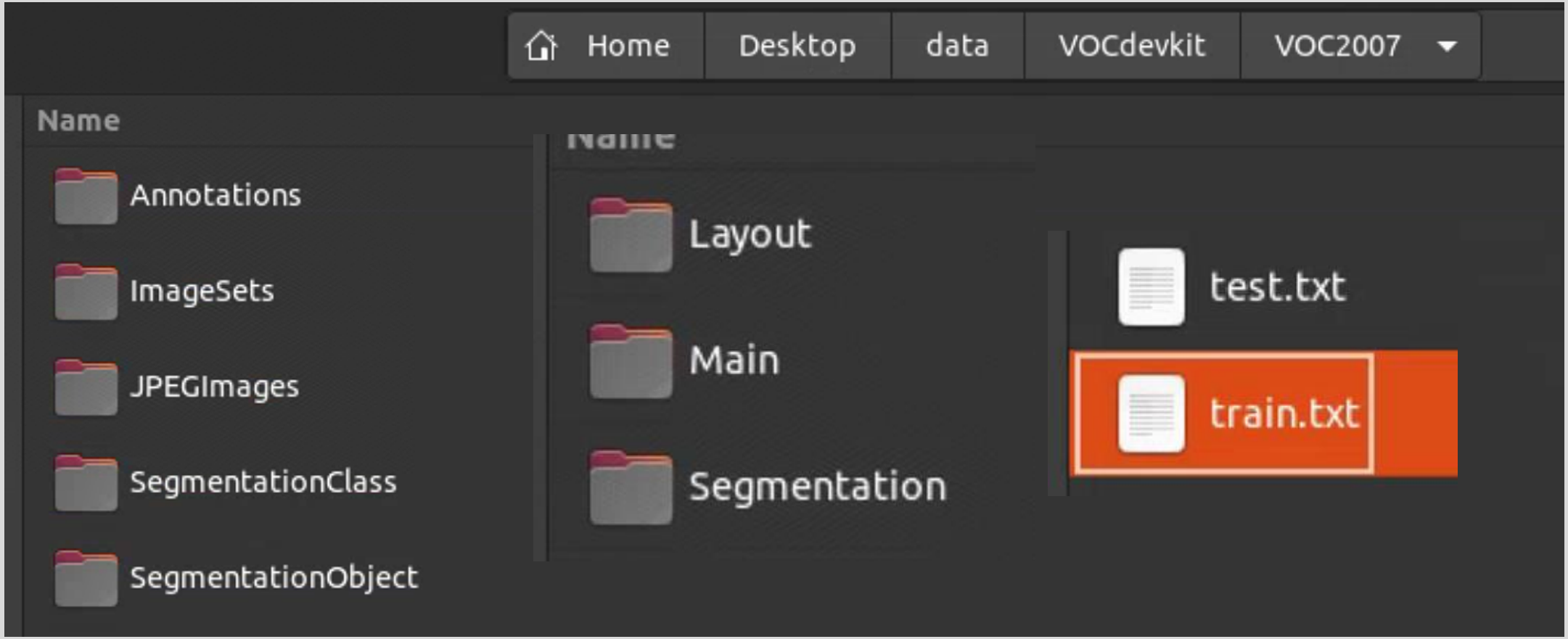
- Download the training/validation data (450MB tar file)
- Download the development kit code and documentation (250KB tar file)
- Download the PDF documentation (120KB PDF)
- Browse the HTML documentation
- View the guidelines used for annotating the database

PASCAL VOC是一个计算机视觉挑战赛，比赛中诞生了众多分类，检测和分割等算法。

VOC 2007 训练和验证数据集中，包含5011张图片，总共包含20个类别。
官网：http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html

VOC 训练数据集


VOCtrainval_06-Nov-2007.tar


Home
Desktop
data
VOCdevkit
VOC2007
Name
Annotations
ImageSets
JPEGImages
SegmentationClass
SegmentationObject
Layout
Main
Segmentation
test.txt
train.txt

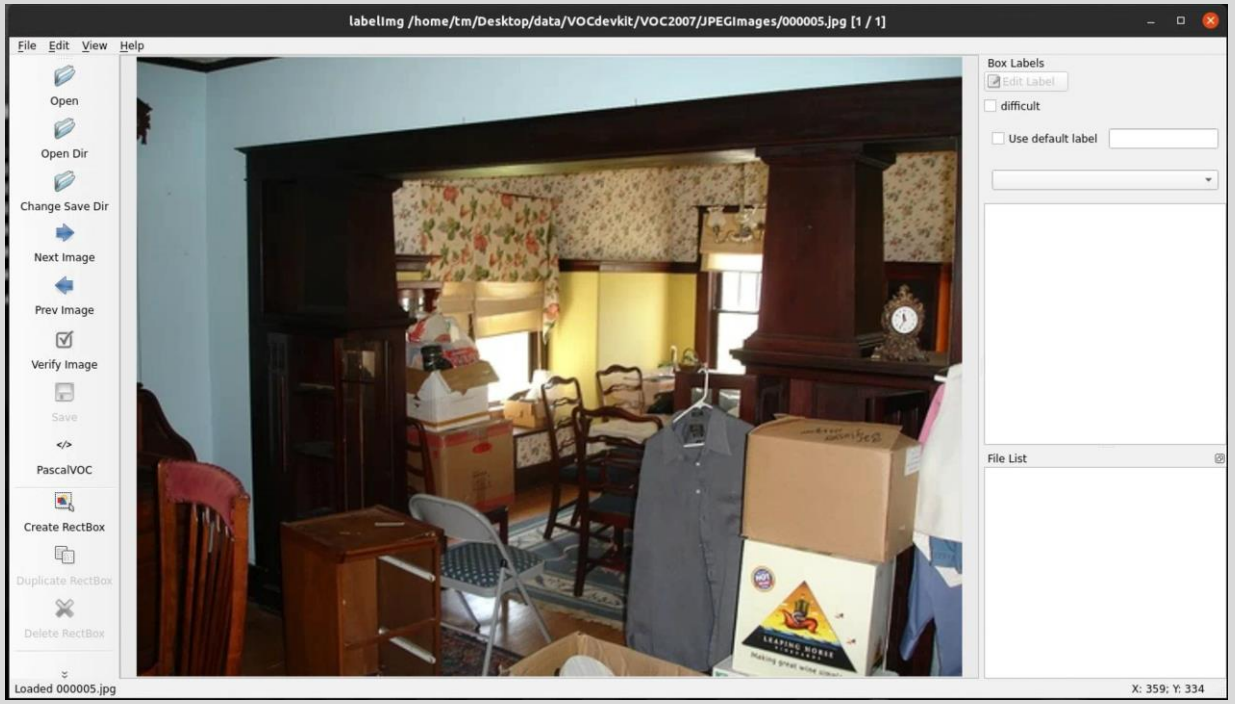
VOC 训练数据集

jpg格式的图片信息

```
<annotation>
  <folder>VOC2007</folder>
  <filename>009161.jpg</filename>
  <source>
    <database>The VOC2007 Database</database>
    <annotation>PASCAL VOC2007</annotation>
    <image>flickr</image>
    <flickrid>331925459</flickrid>
  </source>
  <owner>
    <flickrid>paultownend</flickrid>
    <name>Paul Townend</name>
  </owner>
  <size>
    <width>500</width>
    <height>379</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>bus</name>
    <pose>Left</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>19</xmin>
      <ymin>179</ymin>
      <xmax>423</xmax>
      <ymax>317</ymax>
    </bndbox>
  </object>
  <object>
    <name>person</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>212</xmin>
      <ymin>247</ymin>
      <xmax>262</xmax>
      <ymax>364</ymax>
    </bndbox>
```

xml格式的标签信息（目标位置+类别）

制作自己的数据集



1.收集相关图片

2.使用labelImg标注图片

3.生成包含文件名的txt

4.生成路径+标注信息的txt

LabelImg安装：https://www.bilibili.com/read/cv21375972

生成路径+标注信息

```
import os

xml = os.listdir('VOCdevkit/VOC2007/Annotations')
train = open('VOCdevkit/VOC2007/ImageSets/Main/train.txt', 'w')

for i in range(0,len(xml)):
    name = xml[i][:-4] + '\n'
    train.write(name)

train.close()

#自己制作的数据集需要运行，生成Main目录下的train.txt文件，VOC官方数据集已自带train.txt文件，不需要执行。
#参考：https://blog.csdn.net/tzwsg/article/details/107116132
```

```
# sets=[('2007', 'trainval')]
# classes = ["aeroplane", "bicycle", "bird", "boat", "bottle", "bus", "car", "cat", "chair", "cow", "diningtable", "dog", "horse", "motorb
sets=[('2007', 'train')]
classes = ["airplane"]

#自己制作的数据集需要运行下面两行，VOC官方数据集运行上面两行。
```

```
import xml.etree.ElementTree as ET
from os import getcwd

def convert_annotation(year, image_id, list_file):
    in_file = open('VOCdevkit/VOC%s/Annotations/%s.xml'%(year, image_id))
    tree=ET.parse(in_file)
    root = tree.getroot()

    for obj in root.iter('object'):
        difficult = obj.find('difficult').text
        cls = obj.find('name').text
        if cls not in classes or int(difficult)==1:
            continue
        cls_id = classes.index(cls)
        xmlbox = obj.find('bndbox')
        b = (int(xmlbox.find('xmin').text), int(xmlbox.find('ymin').text), int(xmlbox.find('xmax').text), int(xmlbox.find('ymax').text))
        list_file.write(" " + ",".join([str(a) for a in b]) + ',' + str(cls_id))

wd = getcwd()

for year, image_set in sets:
    image_ids = open('VOCdevkit/VOC%s/ImageSets/Main/%s.txt'%(year, image_set)).read().strip().split()
    list_file = open('%s_%s.txt'%(year, image_set), 'w')
    for image_id in image_ids:
        list_file.write('%s/VOCdevkit/VOC%s/JPEGImages/%s.jpg'%(wd, year, image_id))
        convert_annotation(year, image_id, list_file)
        list_file.write('\n')
    list_file.close()

#生成图片路径+标签信息的txt，用于YOLOv3后续载入数据集使用。
#参考：https://github.com/qqwweee/keras-yolo3/blob/master/voc_annotation.py
```

参考资料：

1. PASCAL VOC 官网
http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html

2. YOLO:labelImg使用+xml文件转为txt文件
https://blog.csdn.net/tzwsg/article/details/107116132

3. qqwweee/keras-yolo3
https://github.com/qqwweee/keras-yolo3

4. AaronJny/tf2-keras-yolo3
https://github.com/AaronJny/tf2-keras-yolo3