

YOLOV4 操作手册及其应用场景

目录

一、制作 VOC 数据集	1
1. 图片数据准备	1
2. 标注数据	1
3. 创建 VOC 文件夹	1
二、训练准备.....	1
1. 生成 TXT 文件	1
2. CONVERT.PY	3
3. 修改运行 VOC_ANNOTATION 代码	3
4. 运行 K-MEANS	3
三、训练.....	4
四、识别以及结果	4
五、应用场景.....	4

一、制作 VOC 数据集

1.图片数据准备

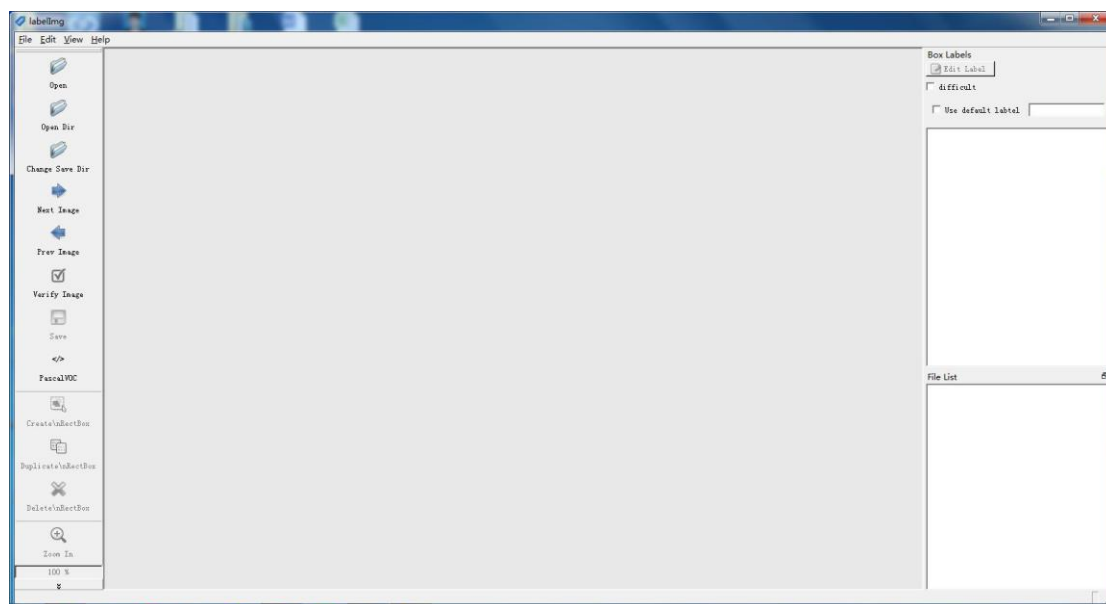
图片数据的获取可以自己用高清摄像机采集视频，然后对其进行抽帧，获得图片数据集[视频>>图片](#)；因为 xml 文件序号要与图片编号一一对应，这里[文件重命名](#)可以用代码完成序号重命名（00001, 00002 等）

0001.xml 0002.xml 0003.xml 0004.xml 0005.xml 0006.xml

2.标注数据

这里用的是比较常用的 **labelImg** 标注工具，每标注一张图片都会生成一个 xml 文件。**labelImg** 可以去浏览器下载，这里给个我当时下载的连接 [labelImg 下载](#)
labelImg 使用方法大家直接搜一下就可以了，很简单也很好用的工具，页面大概

是 这 样



3.创建 VOC 文件夹

新建文件夹命名 **VOC2007**，并且在 **VOC2007** 下新建 **Annotations**，**ImageSets** 和 **JPEGImages** 三个文件夹，在 **ImageSets** 文件夹下再建一个 **Main** 文件夹。**Annotations** 里面放已经标注好的 xml 文件，**JPEGImages** 里面放自己的图片数据（序号 00001, 00002, 00003...）

二、训练准备

1.生成 TXT 文件

在 **VOC2007** 文件夹里新建 **test.py**，运行代码将会在 **Main** 文件夹里生成 **train.txt**, **val.txt**, **test.txt** 和 **trainval.txt** 四个文件。代码如下：

```
import os
import random
```

```
trainval_percent = 0.1
train_percent = 0.9
xmlfilepath = 'Annotations'
txtsavepath = 'ImageSets/Main'
total_xml = os.listdir(xmlfilepath)

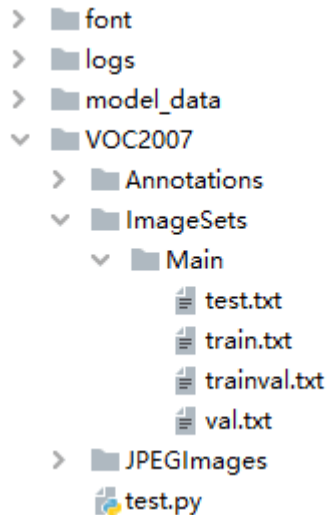
num = len(total_xml)
list = range(num)
tv = int(num * trainval_percent)
tr = int(tv * train_percent)
trainval = random.sample(list, tv)
train = random.sample(trainval, tr)

ftrainval = open('ImageSets/Main/trainval.txt', 'w')
ftest = open('ImageSets/Main/test.txt', 'w')
ftrain = open('ImageSets/Main/train.txt', 'w')
fval = open('ImageSets/Main/val.txt', 'w')

for i in list:
    name = total_xml[i][:-4] + '\n'
    if i in trainval:
        ftrainval.write(name)
        if i in train:
            ftest.write(name)
        else:
            fval.write(name)
    else:
        ftrain.write(name)

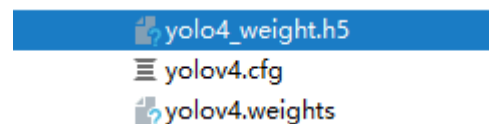
ftrainval.close()
ftrain.close()
fval.close()
ftest.close()
```

做完这一步，VOC2007 目录如下



2.CONVERT.PY

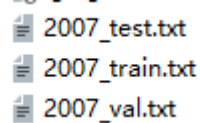
运行 **convert.py** 代码将预训练权重转化为 keras 用的 **h5** 文件



3.修改运行 VOC_ANNOTATION 代码

```
75 sets=[('2007', 'train'), ('2007', 'val'), ('2007', 'test')]
76
77 classes = ["people"]
```

改一下第六行，然后运行运行 **voc_annotation** 代码，生成 **2007_train.txt**, **2007_test.txt**, **2007_val.txt**



4.运行 K-MEANS

生成 **anchors** 大小，然后用这个数据替换 **yolo4_anchor.txt** 的数据。

```
31, 32, 43, 41, 58, 55, 68, 105, 83, 77, 114, 132, 179, 169, 255, 271, 420
```

5. 对 TRAIN.PY 的参数进行修改

因为可以修改的东西比较多，这里就不详细说明了，先把路径，文件输入正确，其他问题不大。

```
18     annotation_path = '2007_train.txt'
19     log_dir = 'logs/000/'
20     classes_path = 'model_data/voc_classes.txt'
21     anchors_path = 'model_data/yolo4_anchors.txt'
```

做到这一步就基本可以训练啦。

三、训练

训练的命令：

```
python3 train.py
```

四、识别以及结果

（一）识别命令：

```
python3 predict.py
```

（二）识别结果图



图 1 列车入口识别一



图 2 列车入口识别

五、应用场景

站点客流量记录是通过乘客刷卡记录的，然而对于列车每个入口的检测却无法检测，有了 yolov4 通过站点摄像头对列车每个入口进行检测，这样就可以对列车每个入口的客流量进行统计，防止如上图 2 所示列车有些入口客流量多，有些入口却没多少人甚至没人在那等候，也可以像天网一样，跟踪罪犯。