

YOLOv5项目实战-多目标检测：足球和梅西

课程演示环境：Ubuntu16.04

1 准备自己的数据集

1) 下载项目文件

从百度网盘下载到yolov5目录下并解压

百度网盘下载链接：

链接：<https://pan.baidu.com/s/1p1HS0gpWZy55dShj3ihLRQ> 提取码：0sao

- VOCdevkit_bm.tar.gz
- testfiles.tar.gz
- prepare_data.py

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 yolov5 / VOCdevkit / VOC2007

VOC2007下面建立两个文件夹：Annotations和JPEGImages

JPEGImages放所有的训练和测试图片；Annotations放所有的xml标记文件

3) 生成训练集和测试集文件

yolov5路径下执行python脚本：

```
python prepare_data.py
```

注意：classes=["ball","messi"]要根据自己的数据集类别做相应的修改

在VOCdevkit / VOC2007目录下可以看到生成了文件夹YOLOLabels；

在VOCdevkit目录下生成了images和labels文件夹；

在yolov5下生成了两个文件yolov5_train.txt和yolov5_val.txt。

- YOLOLabels下的文件是images文件夹下每一个图像的yolo格式的标注文件，这是由annotations的xml标注文件转换来的
- images文件夹下有train和val文件夹，分别放置训练集和测试集图片；labels文件夹有train和val文件夹，分别放置训练集和测试集标签（yolo格式）
- yolov5_train.txt和yolov5_val.txt分别给出了训练图片文件和测试图片文件的列表，含有每个图片的路径和文件名。

2 修改配置文件

1) 新建文件data/voc-bm.yaml

可以复制data/voc.yaml再根据自己情况的修改；可以重新命名如：data/voc-bm.yaml

然后，修改配置参数

```
# download command/URL (optional)
#download: bash data/scripts/get_voc.sh

# train and val data as 1) directory: path/images/, 2) file: path/images.txt, or
# 3) list: [path1/images/, path2/images/]
train: ../yolov5/VOCdevkit/images/train/
val: ../yolov5/VOCdevkit/images/val/

# number of classes
nc: 2

# class names
names: ['ball', 'messi']
```

2) 新建文件models/yolov5x-bm.yaml

可以复制models/yolov5x.yaml再根据自己情况的修改；可以重新命名如：models/yolov5x-ball.yaml

然后，修改配置参数

```
# parameters
nc: 2 # number of classes
```

3 训练自己的数据集

1) 训练命令

yolov5路径下执行：

```
python train.py --data data/voc-bm.yaml --cfg models/yolov5x-bm.yaml --weights
weights/yolov5x.pt --batch-size 6 --epochs 100
```

注意：显存溢出时，可减小batch-size

2) 训练过程可视化：

yolov5路径下执行：

```
tensorboard --logdir=./runs
```

3) 训练结果的查看

查看runs目录下的文件

6 测试训练出的网络模型

1) 测试图片

yolov5路径下执行：

```
python detect.py --source ./testfiles/img4.jpg --weights  
runs/exp0/weights/best.pt --conf 0.4
```

2) 测试视频

yolov5路径下执行:

```
python detect.py --source ./testfiles/messi.mp4 --weights  
runs/exp0/weights/best.pt --conf 0.4
```

3) 性能统计

yolov5路径下执行:

```
python test.py --data data/voc-bm.yaml --weights runs/exp0/weights/best.pt --  
batch-size 6
```