

# 一、问题定义

**单类别多目标追踪：**追踪多个目标（多辆车）的大小和位置

**应用场景：** *监控视角下的车辆跟踪*

**策略：** Deepsort + YOLOv5 two stages的结构，采用 **detection + track**，可根据实际项目中的跟踪效果分别对 detection部分（YOLO）和 track 部分（deepsort）采取优化手段，本项目仅对YOLO部分优化。

## 二、实现过程

### 2.1 数据集下载

UA-DETRAC 数据集包括在中国北京和天津的24个不同地点使用Cannon EOS 550D相机拍摄的10小时视频，并且在数据采集时是在监控视角下拍摄的。

官网：<http://detrac-db.rit.albany.edu/>

官网下载太慢，可在 <https://aistudio.baidu.com/aistudio/datasetdetail/101880> 下载（仅使用 train 部分）

### 2.2 数据集处理 —— 转为VOC2007格式

转换格式原因：

原数据集格式不可被Yolov5封装的训练代码训练，而VOC格式可以

原数据格式：

xml文件内存有一段视频的信息，需要转为一个xml文件对应一张图片。

步骤：

- 提取每张图片的voc格式的xml —— *DETRAC\_xmlParser.py*
- 复制图片到相应目录 —— *voc\_data\_migrate.py*
- 生成 trainval.txt,test.txt,train.txt,val.txt —— *ImageSets\_Convert.py*
- xml转为txt —— *voc\_label.py*

数据集文件目录：

```
VOC2007/
  Annotations/
    MVI_20011__img00001.xml
  images/
    MVI_20011__img00001.jpg
  ImageSets/
    Main/
      test.txt
      train.txt
```

```
trainval.txt
val.txt
labels/
  MVI_20011__img00001.txt
test.txt
train.txt
val.txt
```

主要文件目录结构:

```
Yolov5_DeepSort_Pytorch/
  deep_sort_pytorch/
  yolov5/
    data/
      voc2007/
    train.py
    test.py
    track.py
```

## 2.3 环境配置

- pytorch 1.8.0
- torchvision 0.9.0
- python 3.7
- cuda 11.4.94
- anaconda

版本对应: [https://blog.csdn.net/jorg\\_zhao/article/details/106883420](https://blog.csdn.net/jorg_zhao/article/details/106883420) ;

torch 离线包安装: [https://download.pytorch.org/whl/torch\\_stable.html](https://download.pytorch.org/whl/torch_stable.html) (需要安装torch和torchvision)

## 2.4 修改配置文件

***data/my\_config.yaml***

```
# PASCAL VOC dataset http://host.robots.ox.ac.uk/pascal/VOC/
# Train command: python train.py --data voc.yaml
# Default dataset location is next to YOLOv5:
#   /parent_folder
#   /VOC
#   /yolov5

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

***models/yolov5s.yaml***

## 2.5 训练 + 测试 + 结果

```

Epoch: 0/1
  Class  Images  Labels  P      R      mAP@0.5  mAP@0.5: 95: 100%
  all    9851    71463   0.928   0.932   0.974     0.717

Epoch: 1/1
  Class  Images  Labels  P      R      mAP@0.5  mAP@0.5: 95: 100%
  all    9851    71463   0.945   0.948   0.984     0.769

2 epochs completed in 2.494 hours.

Optimizer stripped from runs/train/exp13/weights/last.pt, 14.4MB
Optimizer stripped from runs/train/exp13/weights/best.pt, 14.4MB

Fusing layers...
Model Summary: 224 layers, 7053910 parameters, 0 gradients
test: Scanning 'data\VOC2007\test.cache' images and labels... 16417 found, 0 missing, 0 empty, 0 corrupted: 100%
  Class  Images  Labels  P      R      mAP@0.5  mAP@0.5: 95: 100%
  all    16417  119131  0.945   0.947   0.985     0.77

Speed: 6.1/1.1/7.2 ms inference/TMS/total per 640x640 image at batch-size 32

```

```
# 模型部分
--yolo_weights          #yolo模型权重地址
--deep_sort_weights     #deep_sort权重地址

# I/O部分
--source                #input地址，0为webcam
--output                #output文件夹建立
--img-size              #图片尺寸（单位：像素）
--conf-thres            #目标置信度阈值
```

```
--iou-thres      #非极大值抑制（NMS）的IOU阈值
--fourcc         #output video codec
--device         #cuda device
--show-vid       #展示跟踪结果 video
--save-vid       #保存跟踪结果 video
--save-txt       #保存跟踪结果 MOT compliant result

# class部分
--classes        #类别过滤器，可单目标也可多目标
--agnostic-nms   #前后景检测结果
--augment        #augmented inference
--evaluate       #augmented inference
--config-deepsort #deep_sort初始化文件地址
```

## 代码示例

```
python track.py \
  --source test.mp4 \
  --classes 2 \
  --show-vid
```