

示例——利用yolov8完成目标检测

1. 源码获取

<https://github.com/ultralytics/ultralytics>

2. 环境搭建

安装CUDA、cudnn、python、pytorch等且要求版本对应

3. 环境检测

下载预训练文件 `yolov8n.pt` 和图片 `bus.jpg`，然后命令行输入

```
1 yolo predict model=yolov8n.pt source='ultralytics/data/images/bus.jpg'
```

运行命令行后，在 `runs/predict` 文件夹下就会生成下图，说明环境配置没问题

4. 数据集准备

数据集准备就参考CSDN上YOLOv5、v8是怎么准备数据集的即可。

5. 模型训练/验证/预测

YOLOv8的训练采用命令行的方式，下列为YOLOv8官方给定的训练/预测/验证方式

```
1 yolo task=detect      mode=train      model=yolov8n.pt      args...
2          classify      predict      yolov8n-cls.yaml     args...
3          segment      val      yolov8n-seg.yaml     args...
```

5.1模型训练

打开终端（或pycharm等IDE），进入anaconda虚拟环境，随后进入 `ultralytics` 文件夹下，在终端输入下面指令，即可开始训练。

```
1 yolo task=detect mode=train model=yolov8n.yaml data=dataset.yaml batch= 16 epoch
```

多卡训练

YOLOv8多卡训练仅需把 `device=0` 改为 `device=0,1` 即可

```
1 yolo task=detect mode=train model=yolov8n.pt data=voc.yaml batch=8 epochs=100 de
```

5.2 模型验证

命令行输入如下指令即可开始验证

```
1 yolo task=detect mode=val model=runs/detect/train/weights/best.pt data=dataset.y
```

5.3 模型测试

使用如下命令既可完成对新数据的预测， `source` 需要指定为自己图像的路径或摄像头(0)

```
1 yolo task=detect mode=predict model=runs/detect/train/weights/best.pt source=xxx
```

6. 训练好的模型，完成对赛题视频的检测步骤如下

把赛题视频放在 `ultralytics/video` 下

命令行输入

```
1 yolo predict model=yolov8n.pt source='ultralytics/video/test_traffic.mp4' device
```

开始预测，等待完成即可

```
(yolov8) wsj@wsj:~/ultralytics+bytetrack$ yolo predict model=yolov8n.pt source='ultralytics/video/test_traffic.mp4' device=0,1
Ultralytics YOLOv8.0.215 Python-3.8.18 torch-1.12.1+cu113 CUDA:0 (NVIDIA GeForce RTX 2080 Ti, 10989MiB)
CUDA:1 (NVIDIA GeForce RTX 2080 Ti, 11012MiB)
YOLOv8n summary (fused): 168 layers, 3151904 parameters, 0 gradients, 8.7 GFLOPs

video 1/1 (1/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 5 cars, 2 buss, 1 train, 1 truck, 9.7ms
video 1/1 (2/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 8 cars, 1 bus, 1 train, 2 trucks, 5.7ms
video 1/1 (3/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 7 cars, 1 bus, 1 train, 2 trucks, 5.5ms
video 1/1 (4/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 8 cars, 1 bus, 1 train, 1 truck, 5.5ms
video 1/1 (5/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 7 cars, 1 bus, 1 train, 1 truck, 5.5ms
video 1/1 (6/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 7 cars, 1 bus, 1 train, 2 trucks, 5.5ms
video 1/1 (7/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 8 cars, 1 bus, 1 train, 1 truck, 5.5ms
video 1/1 (8/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 8 cars, 2 buss, 1 train, 1 truck, 5.5ms
video 1/1 (9/942) /home/wsj/ultralytics+bytetrack/ultralytics/video/test_traffic.mp4: 384x640 8 cars, 2 buss, 1 train, 1 truck, 5.5ms
```

结果保存在 `runs/detect/predict` 中，结果显示如下图（视频截图）

