

# Object Detector Using YoloV7

## 安装

不使用GPU忽略第三行

```
1  conda create -n yolov7 python=3.7
2  conda activate yolov7
3  conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch
4  python -m pip install -r requirements.txt
```

## 使用

### • 终端

#### - 实时检测（按q停止）

```
1  python Detector.py
```

#### - 寻找物体（检测到指定物体停止）

```
1  python Detector.py --find bottle --range 1.0
```

#### - 只检测指定物体（传入名称用“,”隔开）

```
1  python Detector.py --classes bottle, person
```

#### - 其他可选参数

```

1  --weights .pt模型路径
2  --imgsz 图片大小(默认640)
3  --conf-thres 置信度阈值(默认0.25)
4  --iou-thres iou阈值(默认0.45)
5  --range 画面中心多大范围内的检测结果被采用
6  --device 使用的设备(azure kinect或电脑摄像头,使用kinect传入k4a)
7  --mode realtime(实时检测,按q退出) / find(检测到指定物品在画面范围内退出) / other(检测到任意物体在范围内退出)
8  --find mode为find时还需传入需要寻找的物体名称
9  --classes: 哪些物体可以被检测到,字符串名称间用','分割(例:传"bottle,person"则只会检测到bottle和person)
10 --nosave 不保存图片

```

## • 代码调用

### - example

```

1  from Detector import Detector
2
3  class Tester:
4      def __init__(self):
5          """
6          实例化Detector可传参数:
7          weights: .pt模型路径
8          imgsz: 图片大小
9          conf_thres: 置信度阈值
10         iou_thres: iou阈值
11         """
12         self.yolo = Detector()
13
14     def test(self):
15         """
16         调用detect方法可传参数:
17         device: 使用的设备(azure kinect/电脑摄像头)
18         mode: "realtime"(实时检测,按q退出) / "find"(检测到指定物品在画面范围内退出)
19         / other(检测到任意物体在范围内退出)
20         range: 画面中心多大范围内的检测结果被采用
21         nosave: 是否保存图片到本地
22         find: 需要寻找的物体名称
23         classes: 哪些物体可以被检测到,字符串名称间用','分割(例:传"bottle,person"则只会检测到bottle 和person)
24         运行返回:List of YoloResult(每个YoloResult代表一个物品)
25
26         YoloResult数据类型
27         YoloResult.name: 标签名称
28         YoloResult.box: 矩形框左上右下xyxy坐标
29         YoloResult.x: 矩形框中心点x坐标
30         YoloResult.y: 矩形框中心点y坐标
31         YoloResult.conf: 置信度

```

```
31         YoloResult.distance: 中心点深度值（暂未接入）
32         """
33         results = self.yolo.detect()
34         for item in results:
35             print(item)
36
37     def main():
38         tester = Tester()
39         tester.test()
40
41     if __name__ == '__main__':
42         main()
```