

8. Object detection

8. Object detection

- 1、Code Function Description
- 2、FCOS
- 3、YOLO
- 4、MobileNet_SSD
- 5、EfficientNet_Det

1、Code Function Description

RDK X3 provides multiple object detection algorithm models, and after the program starts, MIPI cameras can be used to achieve object detection.

You can view the recognition results in a PC browser.

2、FCOS

FCOS is an open-source Onnx model for Horizon, trained using the COCO dataset, and supports 80 types of object detection, including humans, animals, fruits, and transportation vehicles.

Application scenario: FCOS was released in 2019 and is a single stage object detection algorithm that can achieve functions such as pedestrian detection and vehicle detection. It is mainly used in fields such as autonomous driving and smart home.

After SSH connects the car. Input following command:

```
ros2 launch dnn_node_example dnn_node_example.launch.py  
config_file:=config/fcosworkconfig.json image_width:=480 image_height:=272
```

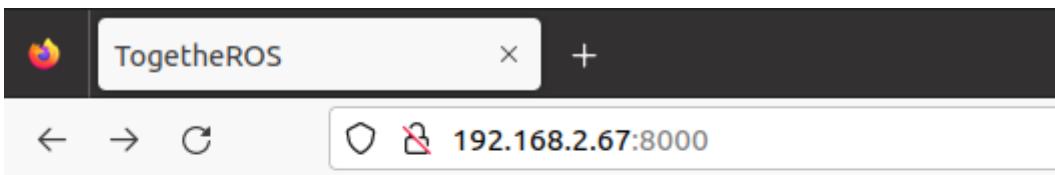
在PC端打开浏览器（注意电脑与旭日派网络须在同一个局域网下），输入网址：小车IP:8000，比如，我的小车IP是192.168.2.67，在虚拟机端的浏览器输入网址，

Open the browser on the PC side (computer and the RDK-X3 must be on the same local area network).

Input the website address: robot car's IP: 8000.

For example, my car IP is 192.168.2.67, and enter the website address in the browser on the virtual machine side,

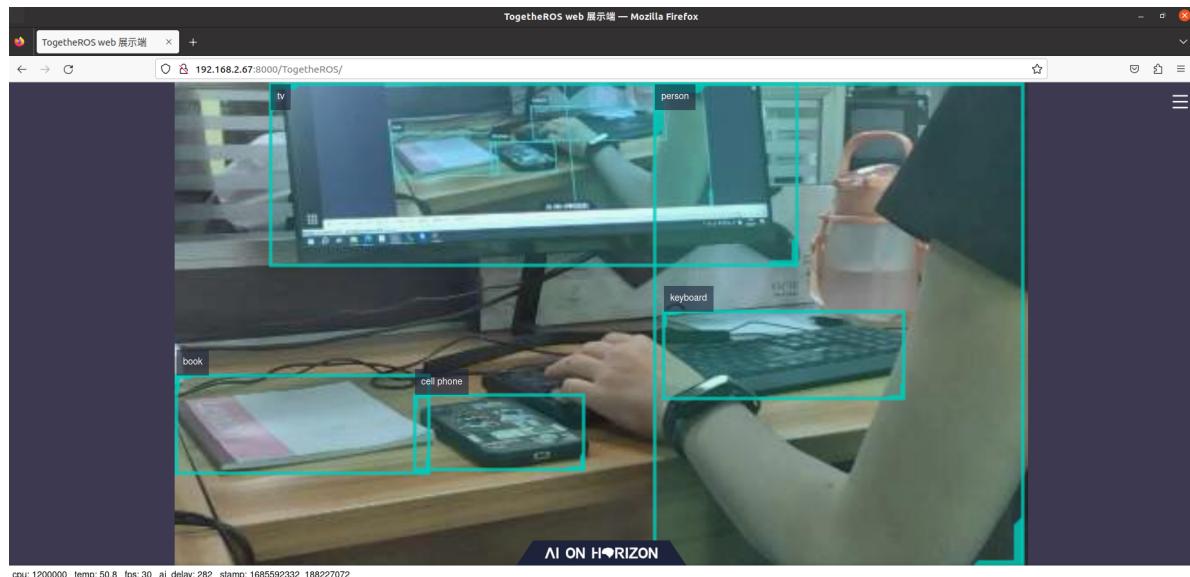
```
192.168.2.67:8000
```



TogetheROS Web 展示端

- Web 展示端

Click to enter the web side, and the display screen is as follows.



3、YOLO

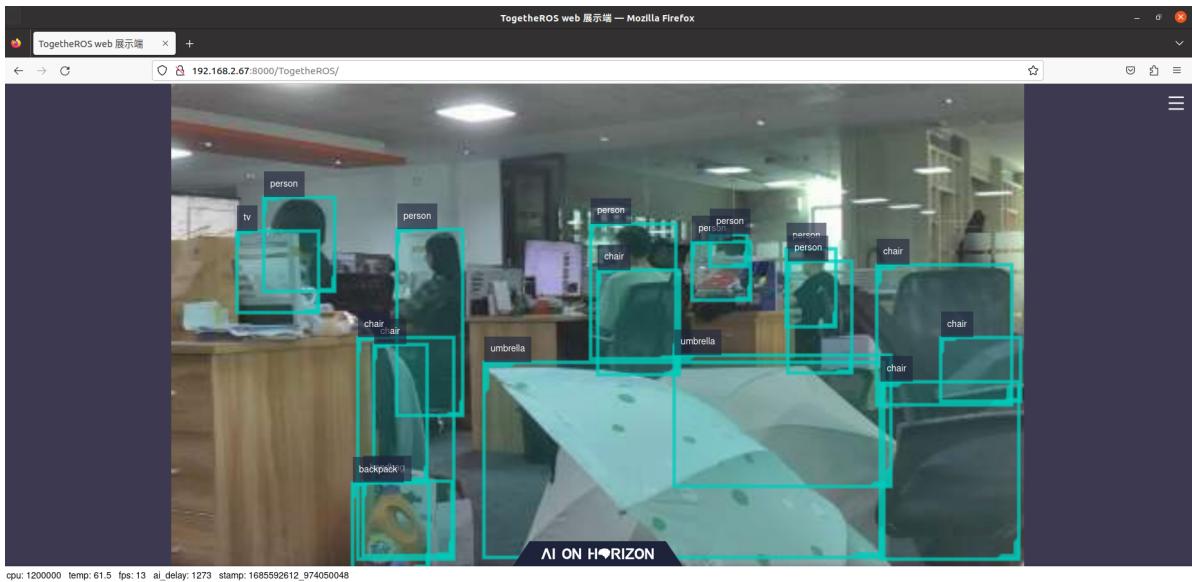
The YOLO object detection algorithm example uses images as input, uses BPU for algorithm inference, and publishes algorithm msgs containing target categories and detection boxes. Currently, it supports three versions: yolov2, yolov3, and yolov5. The model is trained using the COCO dataset and supports a total of 80 types of object detection, including humans, animals, fruits, and transportation vehicles.

Application scenario: YOLO series, as a representative algorithm in single stage object detection, has the advantages of fast speed and good generalization. It can achieve functions such as garbage recognition and vehicle detection, and is mainly used in fields such as autonomous driving and smart home.

After SSH connects the car. Input following command:

```
ros2 launch dnn_node_example dnn_node_example.launch.py  
config_file:=config/yolov2workconfig.json image_width:=480 image_height:=272
```

We can see the display screen on the web display end as follows.



4、MobileNet_SSD

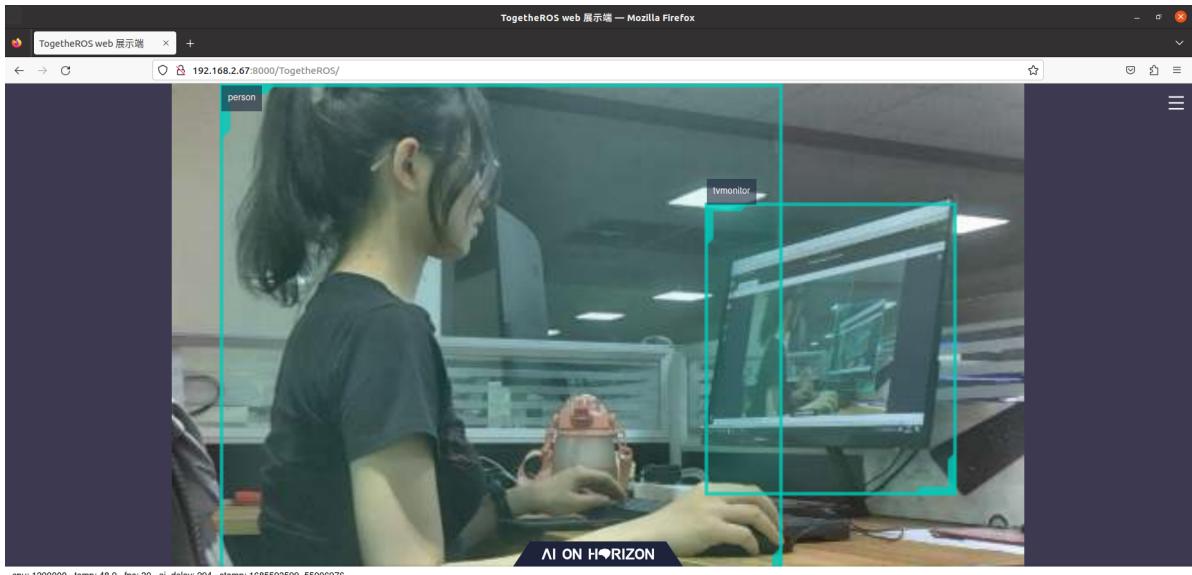
Mobilenet_SSD is derived from <<https://github.com/chuanqij305/MobileNet-SSD>> The caffe model obtained was trained using the VOC dataset, and supported 20 types of object detection, including humans, animals, fruits, and transportation vehicles.

Application scenario: MobileNet_SSD is a target detection algorithm based on MobileNet, which has the advantages of fast speed and easy deployment. It can achieve functions such as object detection and garbage recognition, and is mainly used in fields such as autonomous driving and smart home.

After SSH connects the car. Input following command:

```
ros2 launch dnn_node_example dnn_node_example.launch.py  
config_file:=config/mobilenet_ssd_workconfig.json image_width:=480  
image_height:=272
```

We can see the display screen on the web display end as follows.



5、EfficientNet_Det

EfficientNet_Det is from <<https://github.com/HorizonRobotics-Platform/ModelZoo/tree/master/EfficientDet>> The obtained Onnx model is trained using the COCO dataset, and supports a total of 80 types of object detection, including humans, animals, fruits, vehicles, etc.

Application scenario: EfficientNet_Det can achieve vehicle detection and other functions, mainly applied in fields such as autonomous driving and smart home.

After SSH connects the car. Input following command:

```
ros2 launch dnn_node_example dnn_node_example.launch.py  
config_file:=config/efficient_det_workconfig.json image_width:=480  
image_height:=272
```

We can see the display screen on the web display end as follows.

