

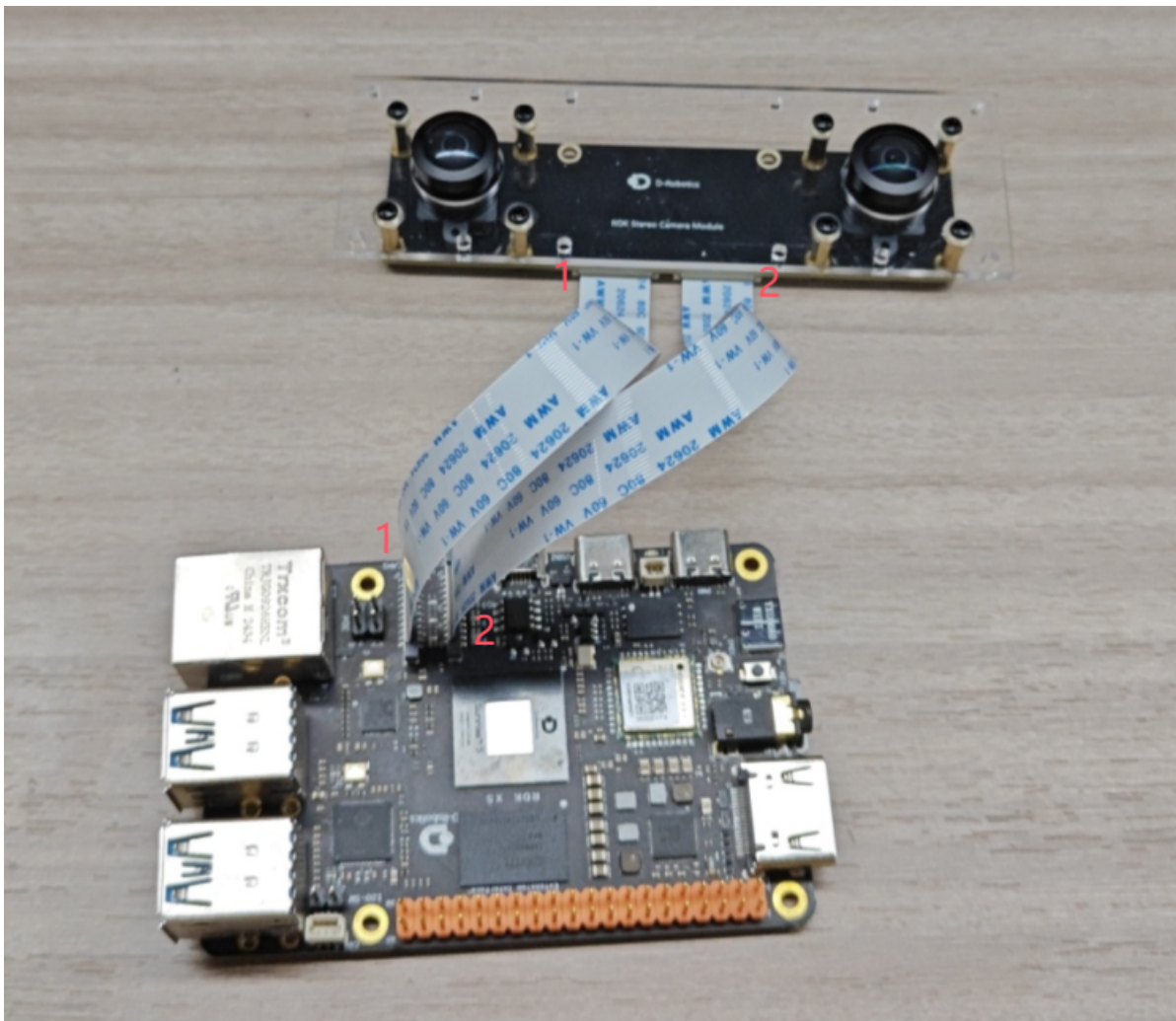
Target detection algorithm—YOLOv5s

Target detection algorithm—YOLOv5s

1. Environment preparation
2. How to run
3. Expected effect

1. Environment preparation

This example is based on the `YOLOv5` model and implements the camera target detection algorithm function. Users can preview the detection results on the display.



- When the development board is powered off, connect the `MIPI` camera to the development board, with the blue side facing the HDMI interface
- Connect the development board and the monitor via an HDMI cable
- Power on the development board and log in via the command line

2. How to run

The sample code is provided in source code form and needs to be compiled and run using the `make` command.

The steps are as follows.

```
sunrise@ubuntu:~$ cd /app/cdev_demo/bpu/src
sunrise@ubuntu:/app/cdev_demo/bpu/src$ sudo make
sunrise@ubuntu:/app/cdev_demo/bpu/src$ cd bin
sunrise@ubuntu:/app/cdev_demo/bpu/src/bin$ sudo ./sample -f
/app/model/basic/yolov5s_672x672_nv12.bin -m 0
```

Parameter configuration:

- -f: Model path
- -m: Model type, default is 0

3. Expected effect

After the program runs correctly, when the terminal stops printing, enter 0 or 1, then press the Enter key, a camera will be started randomly, and the video and algorithm detection rendered images will be output through the `HDMI` interface, and users can preview them through the monitor.

The running log is as follows.

```
sunrise@ubuntu:/app/cdev_demo/bpu/src/bin$ sudo ./sample -f
/app/model/basic/yolov5s_672x672_nv12.bin -m 0
[BPU_PLAT]BPU Platform Version(1.3.6)!
[HBRT] set log level as 0. version = 3.15.49.0
[DNN] Runtime version = 1.23.8_(3.15.49 HBRT)
[A][DNN][packed_model.cpp:247][Model](2000-01-01,19:06:39.821.214) [HorizonRT]
The model builder version = 1.23.5
[W][DNN]bpu_model_info.cpp:491[Version](2000-01-01,19:06:39.876.293) Model:
yolov5s_v2_672x672_bayese_nv12. Inconsistency between the hbrt library version
3.15.49.0 and the model build version 3.15.47.0 detected, in order to ensure
correct model results, it is recommended to use compilation tools and the BPU SDK
from the same OpenExplorer package.
Model info:
model_name: yolov5s_v2_672x672_bayese_nv12Input count: 1input[0]: tensorLayout:
2 tensorType: 1 validShape:(1, 3, 672, 672, ), alignedShape:(1, 3, 672, 672, )
Output count: 3Output[0]: tensorLayout: 0 tensorType: 13 validShape:(1, 84, 84,
255, ), alignedShape:(1, 84, 84, 255, )
Output[1]: tensorLayout: 0 tensorType: 13 validShape:(1, 42, 42, 255, ),
alignedShape:(1, 42, 42, 255, )
Output[2]: tensorLayout: 0 tensorType: 13 validShape:(1, 21, 21, 255, ),
alignedShape:(1, 21, 21, 255, )
[INFO] board_id is 301, not need skip sci1.
Searching camera sensor on device: /proc/device-tree/soc/cam/vcon@0 i2c bus: 6
mipi rx phy: 0
INFO: Found sensor name:ov5647 on mipi rx csi 0, i2c addr 0x36,
config_file:linear_1920x1080_raw10_30fps_2lane.c
2000/01/01 19:06:40.012 !INFO [CamInitParam][0139]Setting VSE channel-0:
input_width:1920, input_height:1080, dst_w:672, dst_h:672
```

```
2000/01/01 19:06:40.013 !INFO [CamInitParam][0139]Setting VSE channel-1:  
input_width:1920, input_height:1080, dst_w:1920, dst_h:1080  
... omission ...
```