

BEV Perception Algorithm

1.Function Introduction

The BEV perception algorithm is a multi-task model trained on the [nuscnescs](#) dataset using [OpenExplorer](#).

The algorithm takes 6 sets of image data as input: front view, left-front, right-front, rear view, left-rear, and right-rear images. The model outputs the 3D detection boxes for 10 categories of objects, including obstacles, various types of vehicles, traffic signs, and semantic segmentation for lane lines, sidewalks, and road edges.

This example uses local image data as input, performs algorithm inference using the BPU, and publishes the rendered images of the perception results. The results are displayed on the PC browser.

Code Repository: https://github.com/D-Robotics/hobot_bev.git

2.Preparation Work

- The RDK has the Ubuntu 20.04 system image flashed.
- The RDK has TogetherROS.Bot installed successfully.
- Ensure that the PC can access the RDK through the network.

3.Usage Instructions

3.1Using Local Dataset for Injection

Use the local dataset for injection, perform inference, and publish the rendered images of the algorithm's results. These images will be displayed on the PC browser via a WebSocket package.

- **Open the recharge node**

Prepare the Local Dataset for Injection

```
# Download the dataset
cd ~
wget http://archive.d-robotics.cc/TogetherROS/data/nuscenes_bev_val/nuscenes_bev_val.tar.gz

# Extract the dataset
mkdir -p ~/hobot_bev_data
tar -zxvf ~/nuscenes_bev_val.tar.gz -C ~/hobot_bev_data
```

Using Dataset for Injection

```
# Configure the tros.b humble environment
source /opt/tros/humble/setup.bash

if [ -L qat ]; then rm qat; fi
ln -s `ros2 pkg prefix hobot_bev`/lib/hobot_bev/qat/ qat
ln -s ~/hobot_bev_data/nuscenes_bev_val nuscenes_bev_val

# Start the running script
ros2 launch hobot_bev hobot_bev.launch.py
```

• Output Results

The terminal outputs the following information:

```
[INFO] [launch]: All log files can be found below /root/.ros/log/2025-05-08-09-44-40-838952-ubuntu-20037
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [hobot_bev-1]: process started with pid [20040]
[INFO] [websocket-2]: process started with pid [20042]
[hobot_bev-1] [UCP]: log level = 3
[hobot_bev-1] [UCP]: UCP version = 3.3.3
[hobot_bev-1] [VP]: log level = 3
[hobot_bev-1] [DNN]: log level = 3
[hobot_bev-1] [HPL]: log level = 3
[websocket-2] [WARN] [1746668681.078783258] [websocket]:
[websocket-2] Parameter:
[websocket-2] image_topic: /image_jpeg
[websocket-2] image_type: mjpeg
[websocket-2] only_show_image: 1
[websocket-2] output_fps: 0
[websocket-2] [INFO] [1746668681.079077507] [websocket]: websocket using image
mjpeg
[hobot_bev-1] [UCPT]: log level = 6
[hobot_bev-1] [DSP]: log level = 3
[hobot_bev-1] [INFO] [1746668681.182092730] [bev_node]: BevNode init
[hobot_bev-1] [WARN] [1746668681.182327429] [bev_node]:
[hobot_bev-1] topic_name: image_jpeg
[hobot_bev-1] save_image: false
[hobot_bev-1] glog_level: 1
[hobot_bev-1] [WARN] [1746668681.186660916] [ai_wrapper]:
[hobot_bev-1] Set glog level in cmd line with '--glog_level=$num'
[hobot_bev-1] EXAMPLE_SYSTEM = 0, EXAMPLE_REPORT = 1, EXAMPLE_DETAIL = 2,
EXAMPLE_DEBUG = 3
[hobot_bev-1] [BPU][[BPU_MONITOR]] [281473498852256] [INFO]BPULib verison(2, 1, 2)
[0d3f195]!
[hobot_bev-1] [DNN] HBTLEXT_DNN log level:6
[hobot_bev-1] [DNN]: 3.3.3_(4.1.17 HBRT)
[hobot_bev-1] [INFO] [1746668681.944706857] [bev_node]: Get render imgs size: 8,
frame_id: 0, duration ms infer: 12.52, postp: 3.37, prep: 0.00
[hobot_bev-1] [INFO] [1746668681.997575564] [bev_node]: Publish ros compressed
image msg, format: jpeg, topic: image_jpeg
```

To view the images and algorithm rendering results, open a web browser on the PC and enter the following URL (replace `IP` with the RDK's IP address):<http://IP:8000>

