

Turtlebot4 YOLO

운영체제의 실제
안인규 (Inkyu An)



Turtlebot4 – Sensor Data

- RPLIDAR A1M8
 - Topic name: /scan (type: sensor_msgs/msg/LaserScan)
- OAK-D-Pro
 - Topic names:



Input of YOLOv8

Topic name	Message type	Description
/turtlebot4/oakd/rgb/preview/image_raw	sensor_msgs/msg/Image	컬러(빨강-녹색-파랑) 이미지 스트림.
/turtlebot4/oakd/rgb/preview/image_raw/compressed	sensor_msgs/msg/CompressedImage	압축된 컬러 이미지 스트림.
/turtlebot4/oakd/rgb/preview/camera_info	sensor_msgs/msg/CameraInfo	카메라 내부 파라미터(초점거리, 왜곡 계수 등) 정보.
/turtlebot4/oakd/stereo/depth	sensor_msgs/msg/Image	깊이(depth) 이미지 스트림: 각 픽셀에 depth value가 담긴 이미지 형태.

YOLO

- **Ultralytics**: AI 및 Computer Vision 분야의 선두적인 소프트웨어 회사
- YOLO model
 - YOLO? Image나 Video stream에서 실시간으로 사물 (예: 사람, 자동차, 고양이, 표지판 등)의 위치를 찾아내고 식별하는 기술
- YOLOv8
 - Detect: 사물의 위치를 bounding box로 찾기
 - Segment: 사물의 픽셀 단위 영역을 구분 (Instance segmentation)
 - Classify: 이미지 전체가 무엇인지 분류
 - Pose: 사람의 관절 위치 등 Keypoint 찾기

Install YOLO 1

- Make a workspace:

```
$ mkdir -p ~/ros2_ws/src
```

- Create a virtual environment & activate it

```
$ cd ~/ros2_ws  
$ python3 -m venv venv  
$ source venv/bin/activate
```

- Make “colcon” ignode “venv”:

```
$ touch venv/COLCON_IGNORE
```

Install YOLO 2

- **Yolo_ros:** https://github.com/mgonzs13/yolo_ros
 - Install requirements:

```
$ cd ~\ros2_ws/src  
$ git clone https://github.com/mgonzs13/yolo_ros.git  
$ source venv/bin/activate  
$ pip3 install -r yolo_ros/requirements.txt
```

Install YOLO 3

- Make the build system utilize the venv python:

```
$ source venv/bin/activate  
$ source /opt/ros/jazzy/setup.bash # 배포판에 맞게 변경  
$ export AMENT_PYTHON_EXECUTABLE="$(which python)"  
$ export PYTHON_EXECUTABLE="$(which python)"
```

- Build the packages

```
$ cd ~/ros2_ws  
$ rosdep update  
$ rosdep install --from-paths src --ignore-src -r -y --rosdistro jazzy  
$ colcon build
```

- Check the entrypoint:

```
$ head -n1 install/*/lib/*
```

Execute YOLOv8

- Execute YOLOv8

```
$ ros2 launch yolo_bringup yolo.launch.py  
input_image_topic:=/oakd/rgb/preview/image_raw
```

- Diagnose Turtlebot4

```
$ ros2 launch turtlebot4_viz view_diagnostics.launch.py
```

