Part 7 About Yolo

- 핑크봇과 통신 및 연결하기 위한 세팅작업 안내

Part 6 - 목차

1. 라이브러리 설치
2. 코드 Module로 만들기
3. 코드 실행

라이브러리 설치

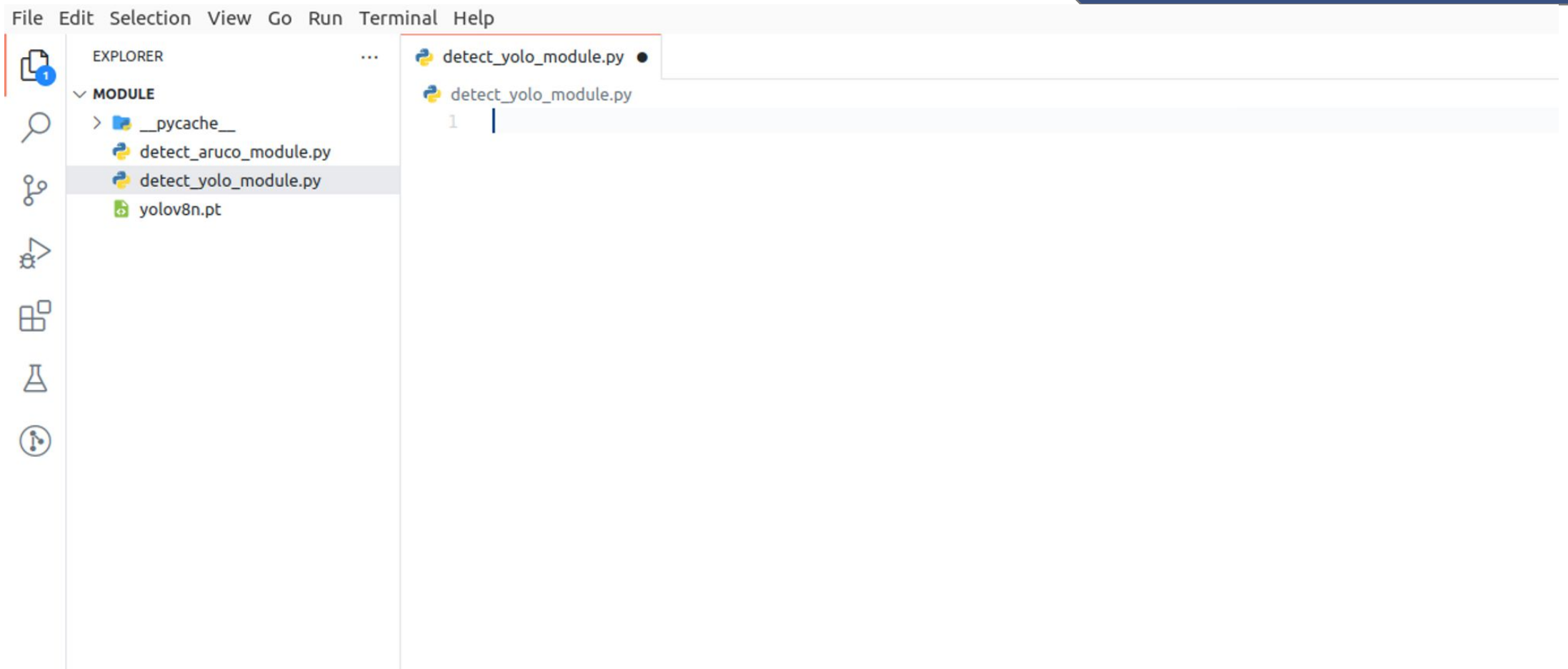
- 해당 내용은 Ubuntu 22.04 기준입니다.

```
pip install torch torchvision
pip install numpy
pip install opencv-python
pip install matplotlib
pip install ultralytics
```

전북경진대회

코드 Module로 만들기

각자의 workspace에 만들기



전북경진대회

코드 Module로 만들기

필요한 모듈 불러오기

```
1   import rclpy
2   from rclpy.node import Node
3   from sensor_msgs.msg import Image
4   from sensor_msgs.msg import CompressedImage
5   from std_msgs.msg import Int32MultiArray
6   from cv_bridge import CvBridge
7   import numpy as np
8   import cv2
9
10  from ultralytics import YOLO
11
```

전북경진대회

코드 Module로 만들기

클래스 생성 후 이미지 받는 subscriber 만들기

```
11
12  class ObjectDetect(Node):
13      def __init__(self):
14          super().__init__('object_detect')
15
16          # 사용할 파라미터를 불러온다
17          self.declare_parameter('rgb_topic', 'image_raw/compressed')
18          self.declare_parameter('conf', 0.5)
19          self.declare_parameter("device", "cpu")
20          # 파라미터의 값을 저장한다
21          self.rgb_topic = self.get_parameter('rgb_topic').value
22          self.conf = self.get_parameter('conf').value
23          self.device = self.get_parameter('device').value
24
25          self.colcor_subscriber = self.create_subscription(
26              CompressedImage,
27              self.rgb_topic,
28              self.color_image_callback,
29              10
30          )
```

코드 Module로 만들기

detect 정보 publisher 만들기

```
31
32          self.detect_publisher = self.create_publisher(
33              Int32MultiArray,
34              '/detect_object',
35              10
36          )
```

전북경진대회

코드 Module로 만들기

CvBridge와 yolo custom 모델 불러오기

```
37
38        self.cv_bridge = CvBridge()
39
40        # 학습한 YOLO 모델 불러오기
41        self.model = YOLO('custom_data_best.pt')
42
```

전북경진대회

코드 Module로 만들기

사람을 인식하고 좌표를 publish하는 callback 함수 만들기

```
42
43  def color_image_callback(self, msg):
44      color_image = self.cv_bridge.compressed_imgmsg_to_cv2(msg, desired_encoding='bgr8')
45
46      results = self.model.predict(color_image, conf=self.conf, device=self.device) #detect
47      result = results[0].boxes.data.cpu().numpy()
48      objects = result[result[:,-1] == 0]
49
50      if len(objects) > 0:
51          # tmp = np.reshape(objects, -1) #1차원 array로 만들기
52          msg = Int32MultiArray()
53          # tmp = tmp.astype(int).tolist()
54          # msg.data = tmp
55          msg.data = [1]
56          self.detect_publisher.publish(msg) #/object_detect으로 퍼블리시
57      else:
58          msg = Int32MultiArray()
59          msg.data = [0]
60          self.detect_publisher.publish(msg)
61
62      id = 0
63      for row in objects:
64          id += 1
65          x1, y1, x2, y2, conf, cls = row
66          cx = int(x1 + x2) // 2
67          cy = int(y1 + y2) // 2
68          center = (cx, cy)
69
70          cv2.rectangle(color_image, (int(x1), int(y1)), (int(x2), int(y2)), (0,255, 0), 1)
71          cv2.circle(color_image, center, 3, (0, 0, 255), -1)
72          cv2.putText(color_image, "object " + str(id), (int(x1), int(y1)),
73          cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
74
75      cv2.imshow("image", color_image)
76      cv2.imshow("image", cv2.resize(color_image, (800, 600)))
77      cv2.waitKey(1)
78
```

전북경진대회

코드 Module로 만들기

마지막으로 노드 실행 함수 만들기

```
//
78  def main(args=None):
79      rclpy.init(args=args)
80      node = ObjectDetect()
81      rclpy.spin(node)
82      rclpy.shutdown()
83
84  if __name__ == '__main__':
85      main()
```

전북경진대회

코드 실행

직접 코드를 실행해 보자

- [원격으로 접속해서] 로봇기동

```
ros2 launch minibot_bringup bringup_robot.launch.py
```

- [원격으로 접속해서] 카메라 실행

```
ros2 run v4l2_camera v4l2_camera_node --ros-args -p image_size:="[400,256]"
```

위의 2가지 작동한 다음

- [원격 접속 아님] 위의 2번에서 생성한 .py파일 실행

```
python3 detect_yolo_module.py
```

전북경진대회

코드 실행

직접 코드를 실행해 보자

dongu@raspdu: ~ 77x10

```
[ros2_control_node-2] [INFO] [1707148087.909242185] [minibot_io_controller]:
state_interface_configuration...
[ros2_control_node-2] [INFO] [1707148087.966863406] [minibot_io_controller]:
on_activate...
[ros2_control_node-2] [INFO] [1707148087.967866628] [minibot_io_controller]:
command_interface_configuration...
[ros2-6] Sucessfully loaded controller minibot_io_controller into state activ
e
[INFO] [ros2-6]: process has finished cleanly [pid 2081]
```

dongu@raspdu: ~ 77x10

```
[WARN] [1707148084.437242866] [v4l2_camera]: Image encoding not the same as r
equested output, performing possibly slow conversion: yuv422_yuy2 => rgb8
[INFO] [1707148084.490198922] [v4l2_camera]: using default calibration URL
[INFO] [1707148084.490553785] [v4l2_camera]: camera calibration URL: file:///
home/dongu/.ros/camera_info/mmal_service_16.1.yaml
[ERROR] [1707148084.491398269] [camera_calibration_parsers]: Unable to open c
amera calibration file [/home/dongu/.ros/camera_info/mmal_service_16.1.yaml]
[WARN] [1707148084.491731892] [v4l2_camera]: Camera calibration file /home/do
ngu/.ros/camera_info/mmal_service_16.1.yaml not found
```

seok@seok-B550M-AORUS-PRO-P: ~/dev_ws/module 84x22

```
seok@seok-B550M-AORUS-PRO-P:~/dev_ws/module$ python3 detect_yolo_module.py

0: 416x640 (no detections), 37.9ms
Speed: 1.7ms preprocess, 37.9ms inference, 0.4ms postprocess per image at shape (1,
3, 416, 640)
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to
run on Wayland anyway.

0: 416x640 (no detections), 33.2ms
Speed: 1.0ms preprocess, 33.2ms inference, 0.3ms postprocess per image at shape (1,
3, 416, 640)

0: 416x640 (no detections), 31.0ms
Speed: 0.9ms preprocess, 31.0ms inference, 0.3ms postprocess per image at shape (1,
3, 416, 640)

0: 416x640 (no detections), 31.7ms
Speed: 1.4ms preprocess, 31.7ms inference, 0.3ms postprocess per image at shape (1,
3, 416, 640)

0: 416x640 (no detections), 30.3ms
```

전북경진대회

코드 실행

- [원격 접속 아님] Local에서 확인

```
ros2 topic list
```

ros2 topic 확인

- detect_object 확인

```
seok@seok-B550M-AORUS-PRO-P: ~ 84x22
---
^Cseok@seok-B550M-AORUS-PRO-P:~$ ros2 topic list
/base_controller/cmd_vel_out
/base_controller/cmd_vel_unstamped
/base_controller/odom
/base_controller/transition_event
/camera_info
/detect_object
/dynamic_joint_states
/image_raw
/image_raw/compressed
/image_raw/compressedDepth
/image_raw/theora
/joint_state_broadcaster/transition_event
/joint_states
/minibot_io_controller/enable_motor
/minibot_io_controller/range
/minibot_io_controller/robot_state
/minibot_io_controller/set_lamp
/minibot_io_controller/transition_event
/parameter_events
/robot_description
```

코드 실행

ros2 topic 받기

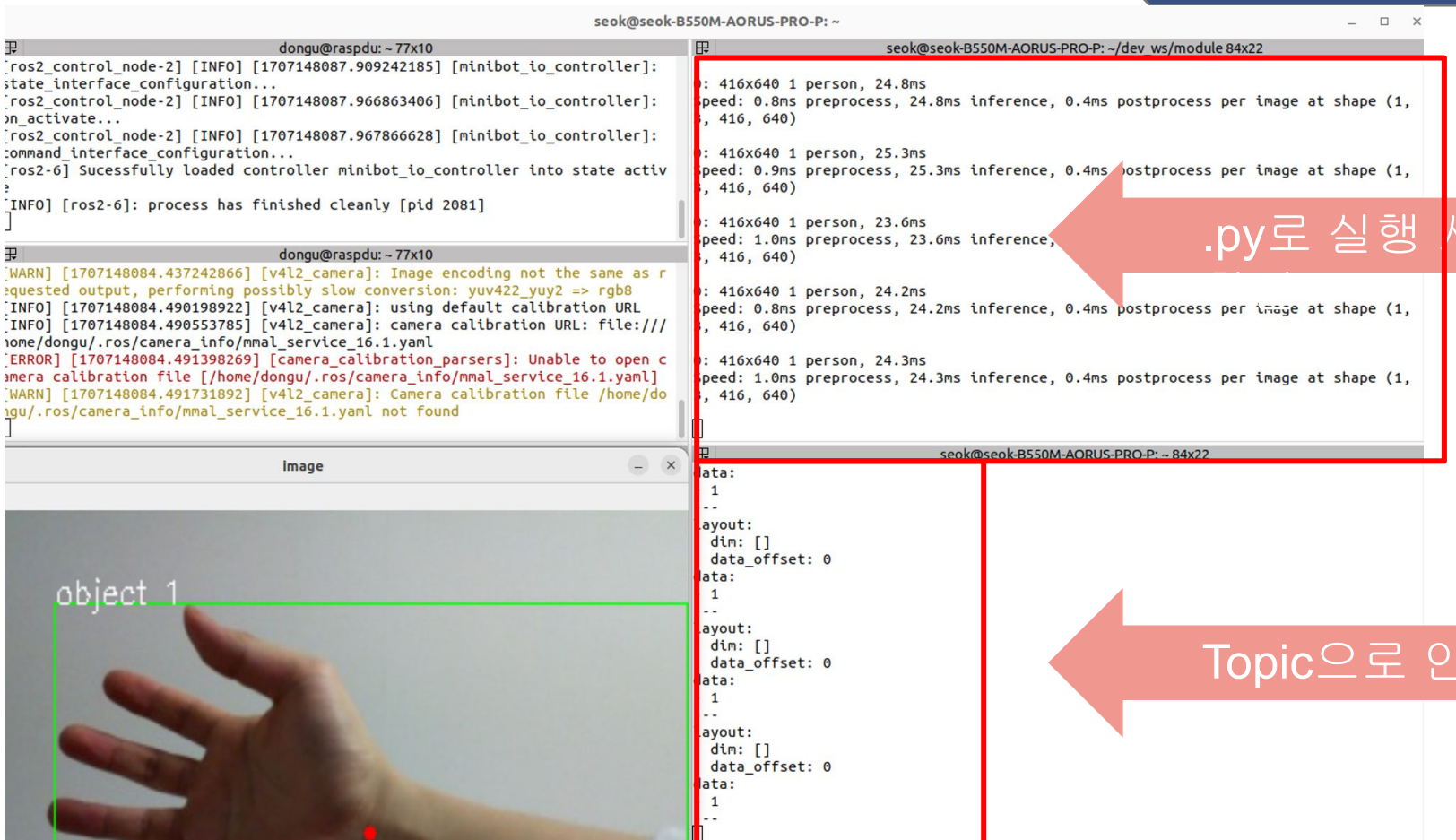
- [원격 접속 아님]

```
ros2 topic echo /detect_object
```

전북경진대회

코드 실행

YOLO로 Object 인식하기



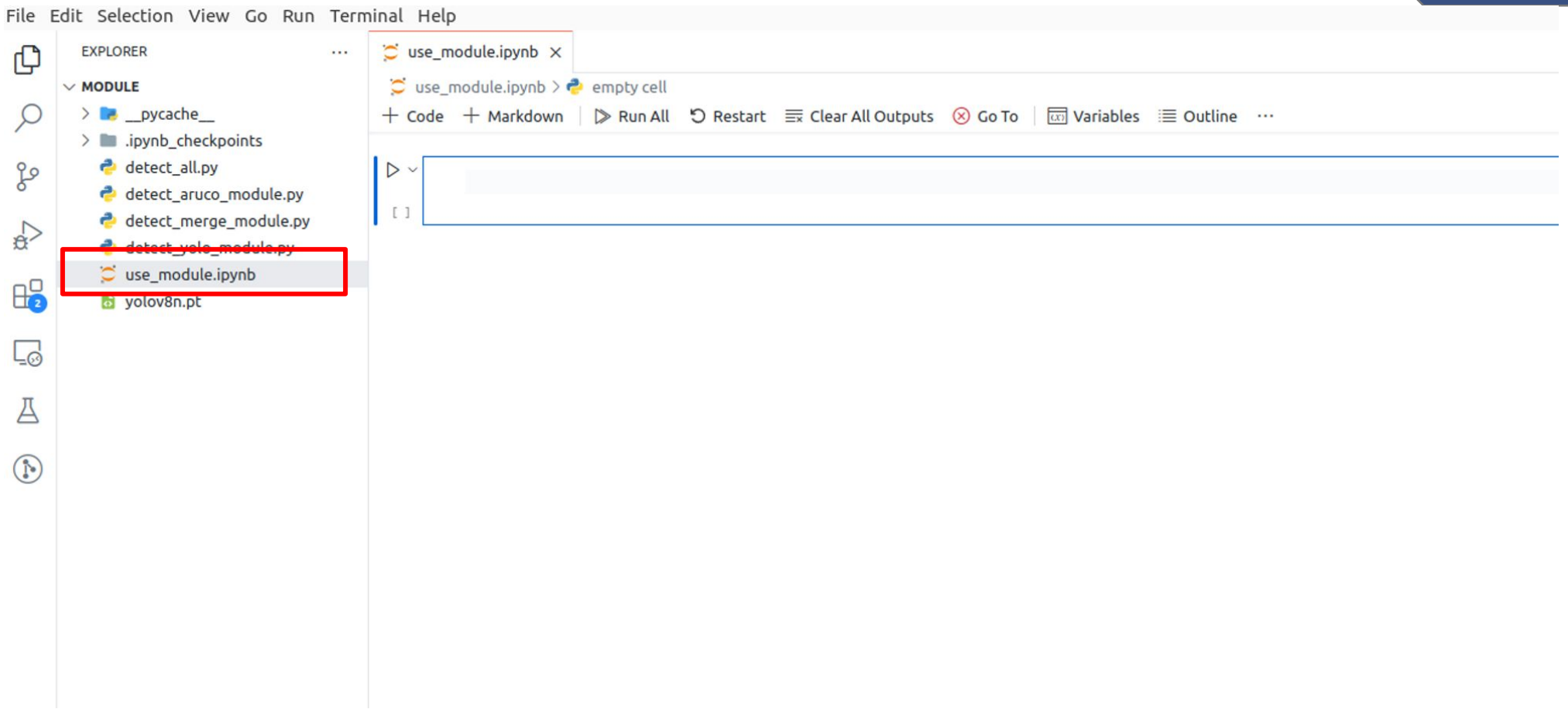
.py로 실행 시킨 화면으로

Topic으로 인식 확인

전북경진대회

.ipynb로 실행

name.ipynb



전북경진대회

.ipynb로 실행

ROS_DOMAIN_ID 설정하기

```
!echo $ROS_DOMAIN_ID
```

```
import os

# 원하는 Domain ID로 설정
domain_id = 7274  # 예시로 7274를 사용하였습니다. 원하는 값으로 변경해주세요.

# ROS_DOMAIN_ID 환경 변수 설정
os.environ['ROS_DOMAIN_ID'] = str(domain_id)
```

다른 팀과 겹치면 안된다

```
!echo $ROS_DOMAIN_ID
```

```
7274
```

전북경진대회

.ipynb로 실행

YOLO 모듈 불러와서 실행하기

```
import rclpy
from rclpy.node import Node
from std_msgs.msg import Int32, Int32MultiArray
from sensor_msgs.msg import Image, CompressedImage
from rclpy.duration import Duration
from cv_bridge import CvBridge, CvBridgeError

from detect_yolo_module import ObjectDetect

def main():
    rclpy.init()

    yolo_detect = ObjectDetect()
    rclpy.spin(yolo_detect)

    rclpy.shutdown()

if __name__ == "__main__":
    main()
```

25.7s

```
0: 416x640 (no detections), 33.8ms
Speed: 2.4ms preprocess, 33.8ms inference, 0.4ms postprocess per image at shape (1, 3, 416, 640)

0: 416x640 (no detections), 30.3ms
Speed: 0.9ms preprocess, 30.3ms inference, 0.4ms postprocess per image at shape (1, 3, 416, 640)

0: 416x640 (no detections), 27.1ms
Speed: 1.0ms preprocess, 27.1ms inference, 0.3ms postprocess per image at shape (1, 3, 416, 640)

0: 416x640 (no detections), 30.1ms
Speed: 0.9ms preprocess, 30.1ms inference, 0.3ms postprocess per image at shape (1, 3, 416, 640)

0: 416x640 (no detections), 26.9ms
```