Object Detection

Feature Name: Voice Recognition

Version	1.0.0
Date	27 Oct 2023
Author(s)	Renwei Hu

Overview

The ***yolo_detector*** node under the ***`object_detector`*** package will try to recognise and extract pixel coordinates of objects. The extracted information will be encapsulated into ROS 2 messages and published to a topic named ***`/vision/yolo_object`***.

Prerequisites

- Python 3.x installed
- ROS2 environment
- Install the following packages via pip:

```
ultralytics >= 8.0
```

Usage

1. Setting up the Workspace
   a. Navigate to the ROS2 workspace

   ```
   cd franka_ws
   ```

   b. Build packages

   ```
   colcon build --symlink-install
   ```

   c. Source the setup file

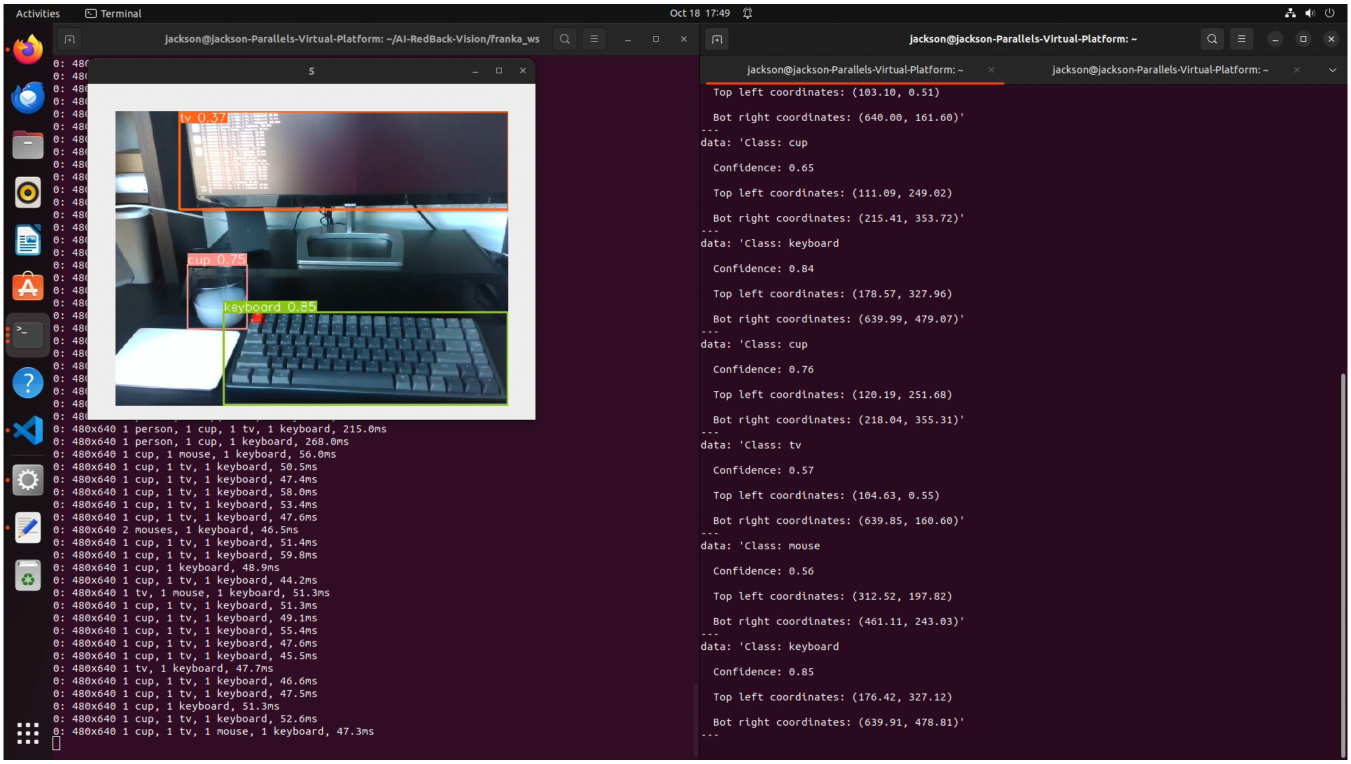
   ```
   source install/setup.bash
   ```

2. Launch the ***yolo_detector*** node
   a. After successfully built, ***`yolo_detector`*** ROS2 node can be launched via the following command:

   ```
   ros2 run object_detector yolo_detector
   ```

3. Check detected objects
   a. A window will show the RGB camera view with **object label**, and **confidence rate** around the boundary of detected objects.
   b. The relevant information extracted from the objects detected can also be checked by inspecting the ***`/vision/yolo_object`*** topic with the command:

   ```
   ros2 topic echo /vision/yolo_object
   ```


Activities
Terminal
Oct 18 17:49
jackson@jackson-Parallels-Virtual-Platform: ~/AI-RedBack-Vision/franka_ws
5
tv 0.37
cup 0.75
keyboard 0.85
jackson@jackson-Parallels-Virtual-Platform: ~

```
0: 480x640 1 person, 1 cup, 1 tv, 1 keyboard, 215.0ms
0: 480x640 1 person, 1 cup, 1 keyboard, 268.0ms
0: 480x640 1 cup, 1 mouse, 1 keyboard, 56.0ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 50.5ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 47.4ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 58.0ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 53.4ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 47.6ms
0: 480x640 2 mouses, 1 keyboard, 46.5ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 51.4ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 59.8ms
0: 480x640 1 cup, 1 keyboard, 48.9ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 44.2ms
0: 480x640 1 tv, 1 mouse, 1 keyboard, 51.3ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 51.3ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 49.1ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 55.4ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 47.6ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 45.5ms
0: 480x640 1 tv, 1 keyboard, 47.7ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 46.6ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 47.5ms
0: 480x640 1 cup, 1 keyboard, 51.3ms
0: 480x640 1 cup, 1 tv, 1 keyboard, 52.6ms
0: 480x640 1 cup, 1 tv, 1 mouse, 1 keyboard, 47.3ms
```

```
  Top left coordinates: (103.10, 0.51)

  Bot right coordinates: (640.00, 161.60)'
---
data: 'Class: cup

  Confidence: 0.65

  Top left coordinates: (111.09, 249.02)

  Bot right coordinates: (215.41, 353.72)'
---
data: 'Class: keyboard

  Confidence: 0.84

  Top left coordinates: (178.57, 327.96)

  Bot right coordinates: (639.99, 479.07)'
---
data: 'Class: cup

  Confidence: 0.76

  Top left coordinates: (120.19, 251.68)

  Bot right coordinates: (218.04, 355.31)'
---
data: 'Class: tv

  Confidence: 0.57

  Top left coordinates: (104.63, 0.55)

  Bot right coordinates: (639.85, 160.60)'
---
data: 'Class: mouse

  Confidence: 0.56

  Top left coordinates: (312.52, 197.82)

  Bot right coordinates: (461.11, 243.03)'
---
data: 'Class: keyboard

  Confidence: 0.85

  Top left coordinates: (176.42, 327.12)

  Bot right coordinates: (639.91, 478.81)'
---
```

Messages Published to Topic

The following command will list all active ROS 2 topics:

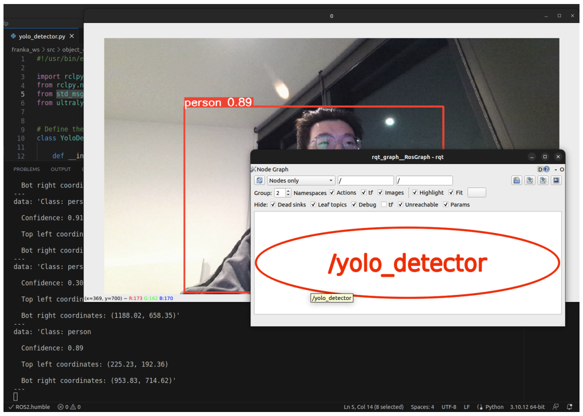
```
$ ros2 topic list

/parameter_events
/rosout
/vision/yolo_object
```

The relevant information extracted from the objects detected by YOLO models will be encapsulated in the form of ***std_msgs.msg.String*** and published to ***/vision/yolo_object*** topic every second (can be configured in code for higher frequency).

The published message will include the following:

- Object class
- Confidence rate
- Top left coordinates (pixel)
- Bottom right coordinates (pixel)

While the YOLO is configured to show the real-time object detection results on screen, we can launch ***rqt_graph*** to verify that the ***yolo_detector*** node is active and publishing messages as expected.


yolo_detector.py
person 0.89
rqt_graph__RosGraph - rqt
Node Graph
Nodes only
Group: 2 Namespaces ☑ Actions ☑ tf ☑ Images ☑ Highlight ☑ Fit
Hide: ☑ Dead sinks ☑ Leaf topics ☑ Debug □ tf ☑ Unreachable ☑ Params
/yolo_detector
/yolo_detector
PROBLEMS OUTPUT
(x=369, y=700) ~ R:173 G:162 B:170
ROS2.humble
Ln 5, Col 14 (8 selected) Spaces: 4 UTF-8 LF Python 3.10.12 64-bit

```
#!/usr/bin/e

import rclpy
from rclpy.n
from std_msg
from ultraly


# Define the
class YoloDe

    def __in
```

```
  Bot right coordi
---
data: 'Class: pers

  Confidence: 0.91

  Top left coordin

  Bot right coordi
---
data: 'Class: pers

  Confidence: 0.30

  Top left coordin

  Bot right coordinates: (1188.02, 658.35)'
---
data: 'Class: person

  Confidence: 0.89

  Top left coordinates: (225.23, 192.36)

  Bot right coordinates: (953.83, 714.62)'
---
```

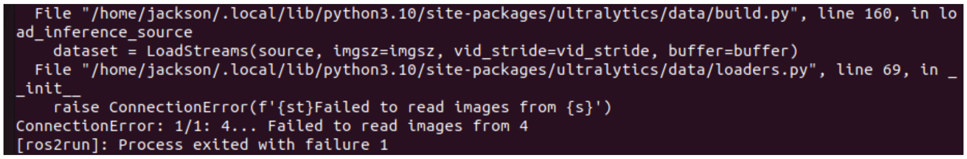
Release Notes

Features:

- Integrate YOLO streaming object detection as a ROS2 node
- Mark detected objects with object labels, confidence rates and boundaries on camera view in real time
- Continuously publish corresponding object information to a ROS2 topic

Known Issues:

1. Can't gracefully shutdown the node when force quit with Ctrl + C
    - Reason: Can't break the streaming object detection properly unless modify source code of YOLO
2. Can't disconnect camera channel properly when using Intel RealSense depth camera as video input and the node was force quit

- 

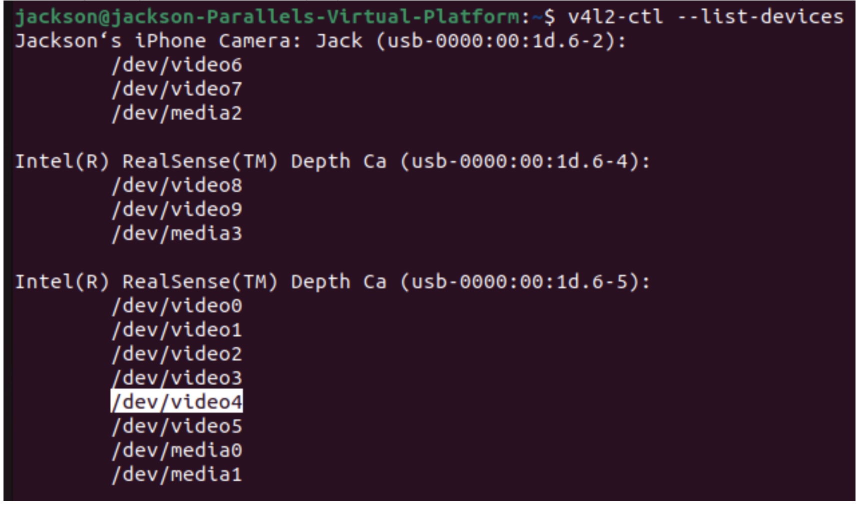
```
  File "/home/jackson/.local/lib/python3.10/site-packages/ultralytics/data/build.py", line 160, in lo
ad_inference_source
    dataset = LoadStreams(source, imgsz=imgsz, vid_stride=vid_stride, buffer=buffer)
  File "/home/jackson/.local/lib/python3.10/site-packages/ultralytics/data/loaders.py", line 69, in _
_init__
    raise ConnectionError(f'{st}Failed to read images from {s}')
ConnectionError: 1/1: 4... Failed to read images from 4
[ros2run]: Process exited with failure 1
```

- Consequence: Raise error "ConnectionError: ... failed to read images from ..."
- Reason: Force quit the node will leave the device file occupied (by YOLO streaming loop), connection to camera can't be re-established
- Workaround: Re-plug the camera connection cable to reset camera connection

Notes:

- Please set the correct device reference
  - First check all active devices for video input using the command:

    ```
    v4l2-ctl --list-devices
    ```

  - Find the correct video input device. If use Intel RealSense depth camera as video input, please find the corresponding RGB frame channel (usually the 5th one as highlighted below, but reference number could be different).
  - 

    ```
    jackson@jackson-Parallels-Virtual-Platform:~$ v4l2-ctl --list-devices
    Jackson's iPhone Camera: Jack (usb-0000:00:1d.6-2):
            /dev/video6
            /dev/video7
            /dev/media2

    Intel(R) RealSense(TM) Depth Ca (usb-0000:00:1d.6-4):
            /dev/video8
            /dev/video9
            /dev/media3

    Intel(R) RealSense(TM) Depth Ca (usb-0000:00:1d.6-5):
            /dev/video0
            /dev/video1
            /dev/video2
            /dev/video3
            /dev/video4
            /dev/video5
            /dev/media0
            /dev/media1
    ```

  - Change the device reference number predefined in ***yolo_detector.py*** file

    ```
    def detect_objects(self):
        # Define streaming source (e.g. camera dev number)
        # May need to try out for different ports
        camera_dev = '4'

        # Return a list of Results objects
        self.results_ = self.yolo_model_(source=camera_dev,
                        show=True,
                        boxes=True,
                        stream=True
                    )
    ```

Future Improvements:

1. Try to gracefully shutdown the node by correctly handling exceptions like force quit.
2. Properly release camera connection before shutdown the node.
3. Add arguments in node launch command to set video input device number, or dynamically find the correct device number in code.
4. Feed image frames subscribed from topics as input for YOLO object detection instead occupy camera channel for object detection only.