5. Voice control robotic arm garbage identification and sorting

5.1. Function description

Through interaction with the voice module, rosmaster can identify and classify the junk pictures on the prescription blocks and broadcast the recognition and classification results by voice; then the robot arm will pick up the identified blocks, navigate to the set area autonomously, and put them down , and finally returns to the set origin.

5.2. Start

Note: Due to the processing speed of nano master, loading model data and image processing will be a little stuck but it can still run successfully. NX master will be better.

5.2.1. Function package path

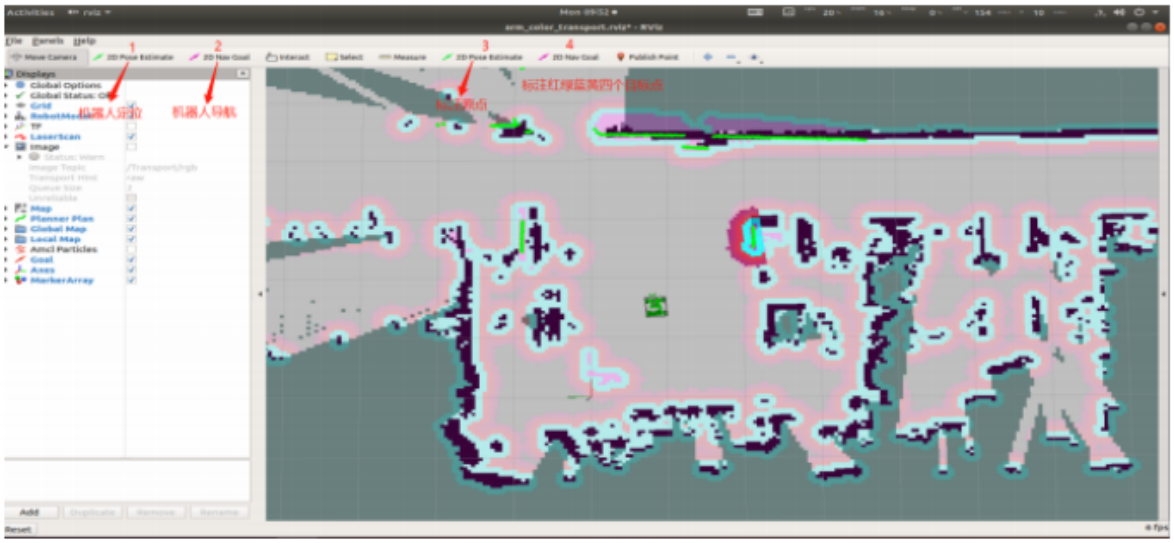
```
~/yahboomcar_ws/src/garbage_identify_yolov5/
```

5.2.2. Start

Note: **Multi-machine communication configuration** needs to be implemented between rosmaster X3Plus and the virtual machine. You can view the content of **"yahboomcar Course\06, Linux Operating System\04. Multi-machine Communication Configuration"** for configuration.

```
#rosmaster X3Plus start
roslaunch yahboomcar_voice_ctrl voice_transport_base.launch
rosrun garbage_identify_yolov5 garbage_identify_yolov5.py
roslaunch yahboomcar_yolov5 yolodetect.launch
#Start virtual machine
roslaunch arm_color_transport transport_rviz.launch
```

Each tool annotation on the virtual machine rviz is as shown in the figure below,



- Step 1

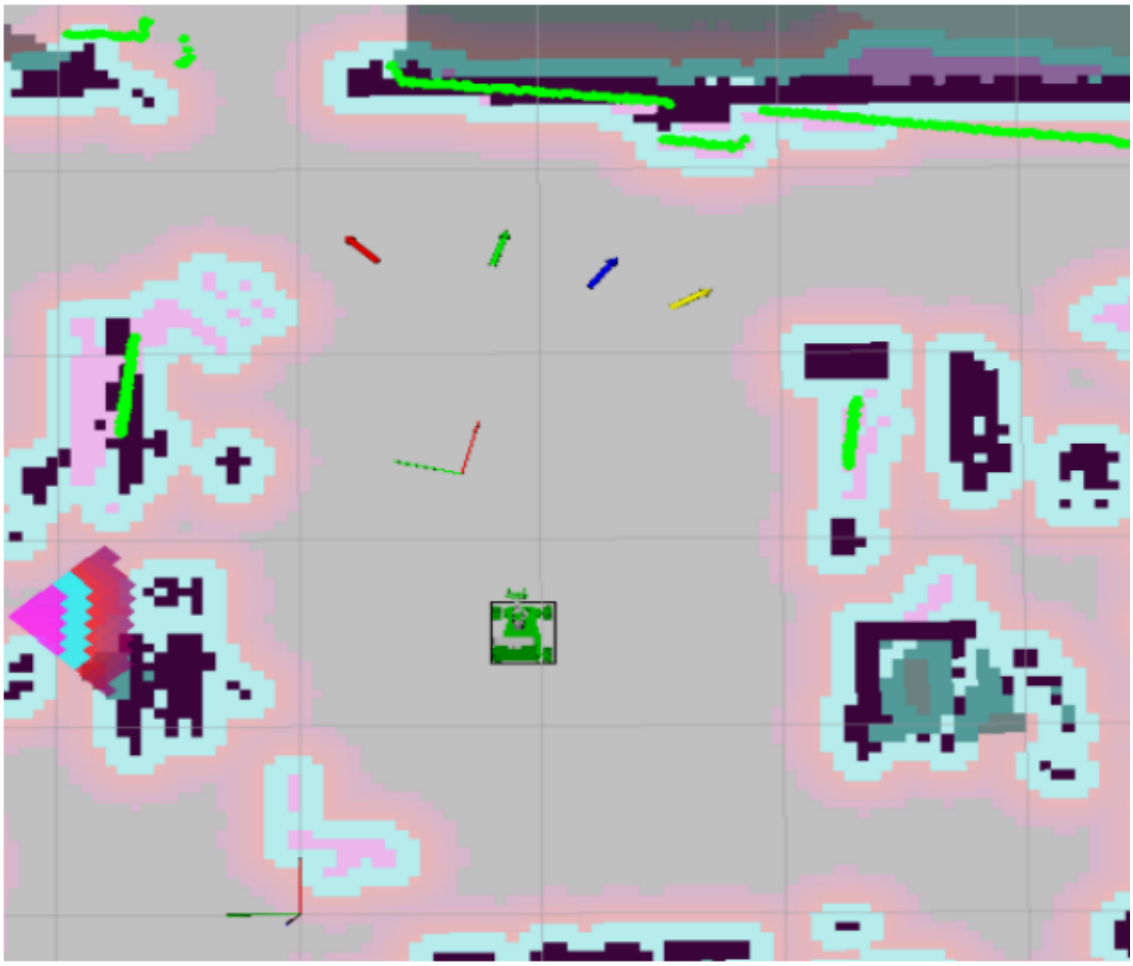
  In rviz, use tool 1 to calibrate and position the rosmaster;

- Step 2

  In rviz, use the 3 tool to mark the origin of the rosmaster. This origin is the position where the car automatically returns after putting down the color block;

- Step 3

  In rviz, use 4 tools to mark red, green, blue and yellow destinations on the map in sequence. After the rosmaster's robotic arm gripper picks up the color block, it will navigate to the destination corresponding to the color of the color block.It is best to mark four points in a row here, and the front and back distance should not be too far apart, otherwise, when planning the path, you may hit the color block.



Note: **Hazardous garbage** corresponds to the **Red** marking point; **Wet garbage** corresponds to the **Green** marking point; **Recyclable garbage** corresponds to the **Blue** marking point; **dry garbage** corresponds to the **yellow** marking point.

- Step 4

  Say "Hello, Xiaoya" to rosmaster. After waking up the voice module, place the color block about 20cm away from the camera on the robotic arm. When the object appears stably in the screen, say to it "What kind of garbage is this?" It will answer the name and type of garbage it recognizes, and after the second beep, hand the block to the gripper of the robotic arm, which will clamp the block. After rosmaster moves back to adjust its attitude, it will autonomously navigate to the previously set location. After arriving, it will put down the block with its paws and announce "Placement completed". Then, after moving back and adjusting the position, it will navigate back to the set origin and wait for the next recognition.

5.2.3. Core code garbage_identify_yolov5.py

- Code path

```
~/yahboomcar_ws/src/garbage_identify_yolov5
```

- Core code analysis

  1）Import the corresponding library file

```
from Speech_Lib import Speech   #Voice recognition library
from garbage_library import GarbageTransport    #Handling navigation library
```

  2）Create object

```
spe = Speech()  #Create speech recognition objects
self.garbage_transbot = GarbageTransport()  #Create a handling navigation
object
```

  4）Garbage_identify_yolov5.py mainly handles the following work:

  - Subscribe to the DetectMsg topic data, receive the recognition results, and execute the process function in the callback function.
  - process：In this function, it mainly determines the value of the DetectMsg topic data (frame_id), which is the name of the garbage. Then, it enters the self.garbage_transbot.process() function. This function takes parameters.The meaning of the parameters is to go to the points marked on rviz.

  5）、garbage_transbot.process() function

  After entering this function, the current mode state will be determined first. If it is the default Grip state, the passed parameter value will be determined. **1 represents the red mark point, 2 represents the green mark point, 3 represents the blue mark point, and 4 Indicates a yellow marker**. For other statuses, you can view the source code of the program. It can be seen that every time the program in this state is executed, the mode state will be changed to the next state expected to be executed.

5.3. Program node communication

```
rosrun rqt_grapf rqt_graph
```