10.Garbage stacking

1.Garbage stacking instructions

The file path for the position calibration of the robotic arm is~/jetcobotw_s/src/jetcobot_comlor_identify/scripts/XYT_config. txt.

After calibration, restart the program and click on the calibration mode to automatically read the file information, reducing the need for repeated calibration actions.

Garbage stacking is the addition of the function of stacking a second layer of garbage building blocks on the basis of garbage sorting. Due to the limited range of motion of the robotic arm, a maximum of two layers can be stacked.

2.About code

Code path：~/jetcobot_ws/src/jetcobot_garbage_yolov5/Garbage_stacking.ipynb

~/jetcobot_ws/src/jetcobot_utils/src/jetcobot_utils/grasp_controller.py

Due to the possible deviation in the camera's recognition of the block gripping position, it is necessary to increase the deviation parameter to adjust the deviation value of the robotic arm's grasp in the recognition area.

The type corresponding to garbage sorting and garbage stacking is "garbage", so it is necessary to change the offset parameter below type=="garbage".

The X offset controls the forward and backward offsets, while the Y offset controls the left and right offsets.

```
# Get the XY offset according to the task type
    def grasp_get_offset_xy(self, task, type):
        offset_x = -0.012
        offset_y = 0.0005
        if type == "garbage":
            offset_x = -0.012
            offset_y = 0.002
        elif type == "apriltag":
            offset_x = -0.012
            offset_y = 0.0005
        elif type == "color":
            offset_x = -0.012
            offset_y = 0.0005
        return offset_x, offset_y
```

The coordinate value of the garbage placement area.

If the placement location coordinate is inaccurate, you can modify this coordinate value appropriately.

```
    # Recyclable garbage location
    def goRecyclablePose(self, layer=1):
        layer = self.limit_garbage_layer(layer)
        coords = [-50, -230, 110 + int(layer-1)*40, -180, -2, -43]
        self.go_coords(coords, 3)

    # Hazardous garbage location
    def goHazardousWastePose(self, layer=1):
        layer = self.limit_garbage_layer(layer)
        coords = [20, -230, 110 + int(layer-1)*40, -180, -2, -43]
        self.go_coords(coords, 3)

    # Kitchen garbage location
    def goFoodWastePose(self, layer=1):
        layer = self.limit_garbage_layer(layer)
        coords = [80, -230, 110 + int(layer-1)*40, -180, -2, -43]
        self.go_coords(coords, 3)

    # Other garbage locations
    def goResidualWastePose(self, layer=1):
        layer = self.limit_garbage_layer(layer)
        coords = [145, -230, 120 + int(layer-1)*40, -180, -2, -43]
        self.go_coords(coords, 3)
```

3. Start code

Start roscore

- If you are using Jetson Orin NX/Jetson Orin Nano board. You need to enter the Docker environment using the following command.
- Then, run roscore

```
sh ~/start_docker.sh
roscore
```

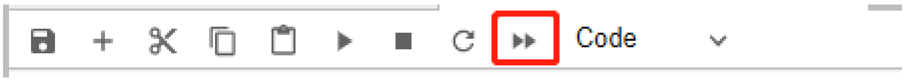
- If you are using Jetson Nano board. You need to enter the following command directly.

```
roscore
```

Start the program

Open the jupyterlab webpage and find the corresponding .ipynb program file.

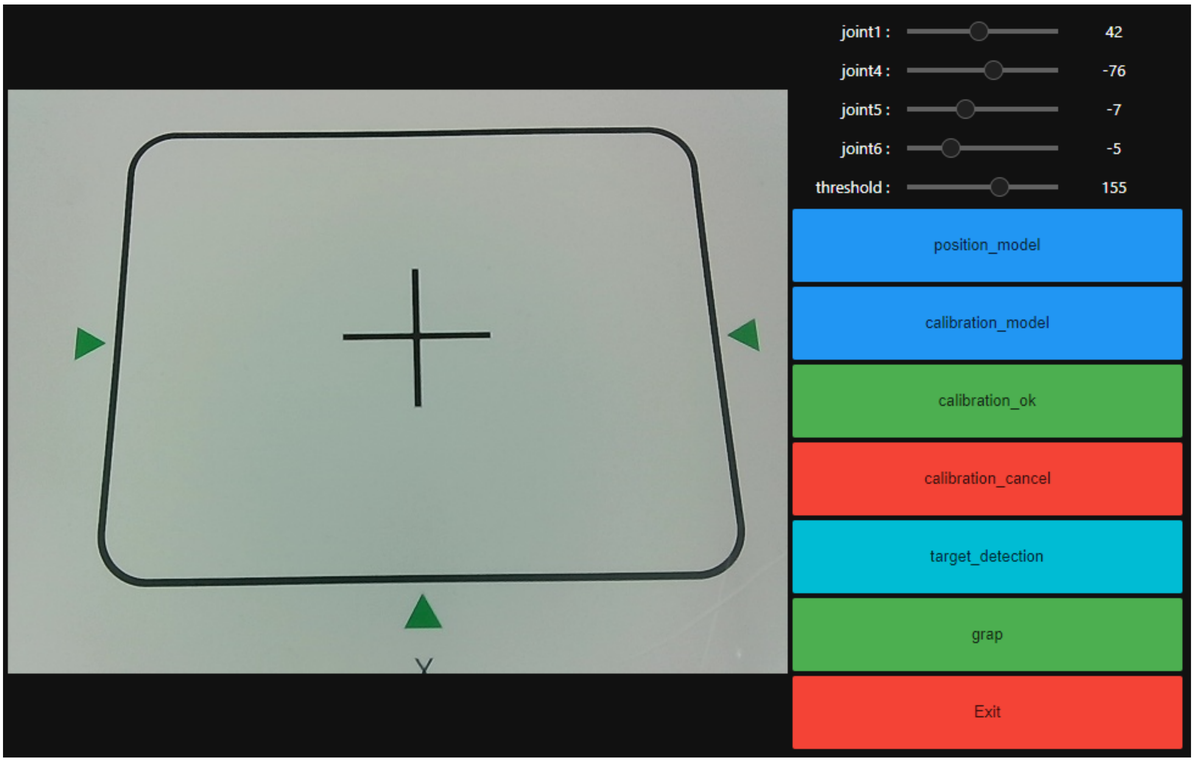
Click the Run button to run the entire code.



4. Experimental operation and results

After the program runs, the Jupyterlab webpage will display the control button.

Camera image on the left side, functions of the related buttons on the right side.



joint1 : 42

joint4 : -76

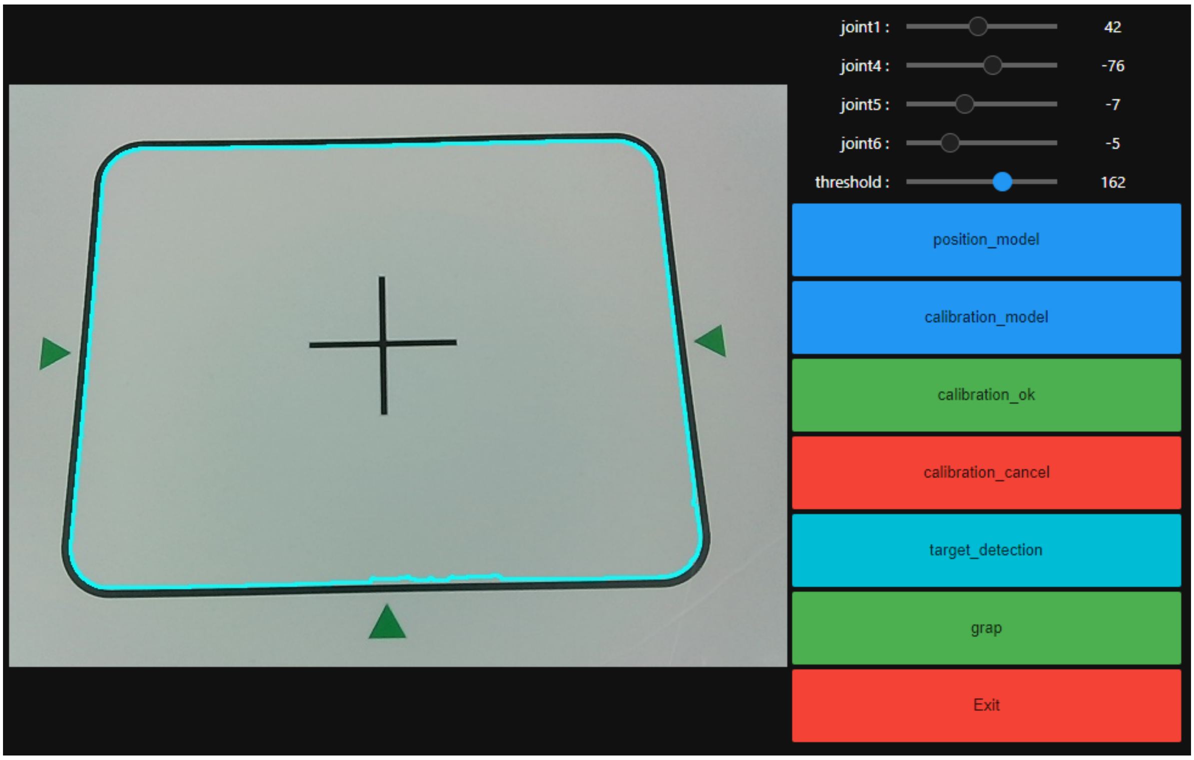
joint5 : -7

joint6 : -5

threshold : 155

Click the 【position_model】 button, drag the joint angle above, update the position of the robotic arm, and make the recognition area in the middle of the entire image.

Then, click 【calibration_model】 to enter the calibration mode, and adjust the robotic arm joint slider and threshold slider above to make the displayed blue line overlap with the black line of the recognition area.



joint1 : 42

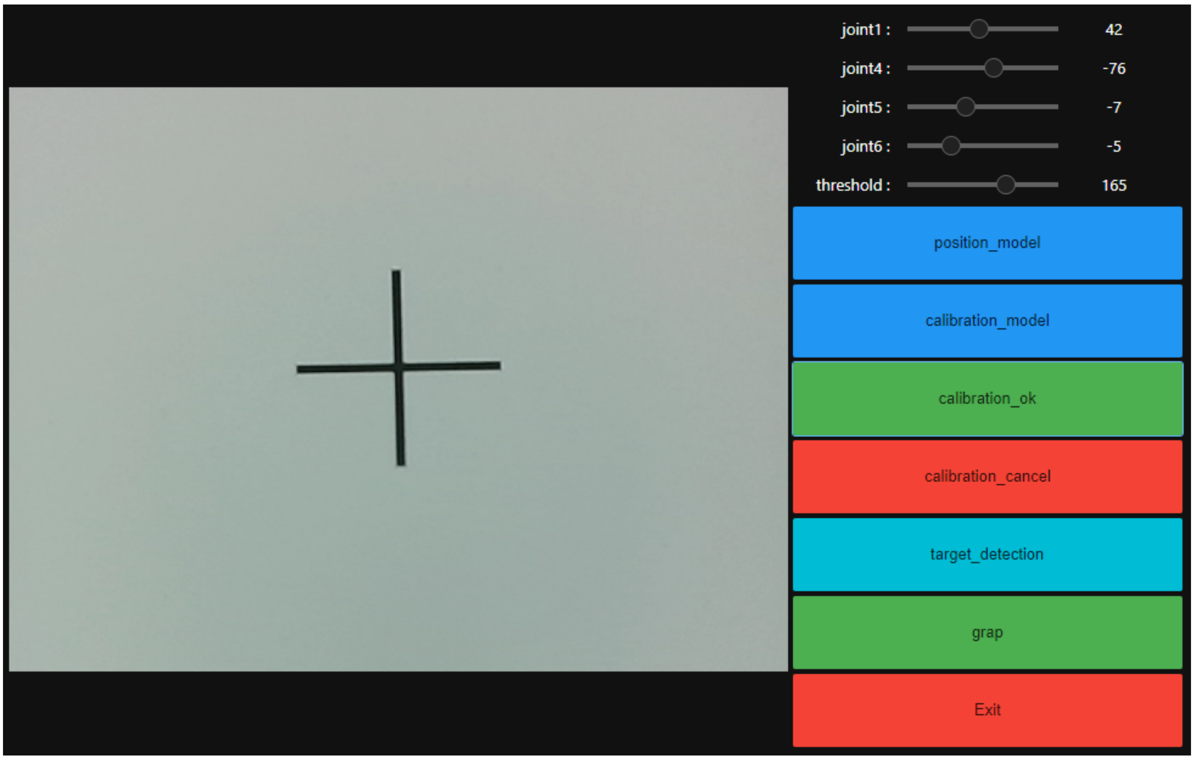
joint4 : -76

joint5 : -7

joint6 : -5

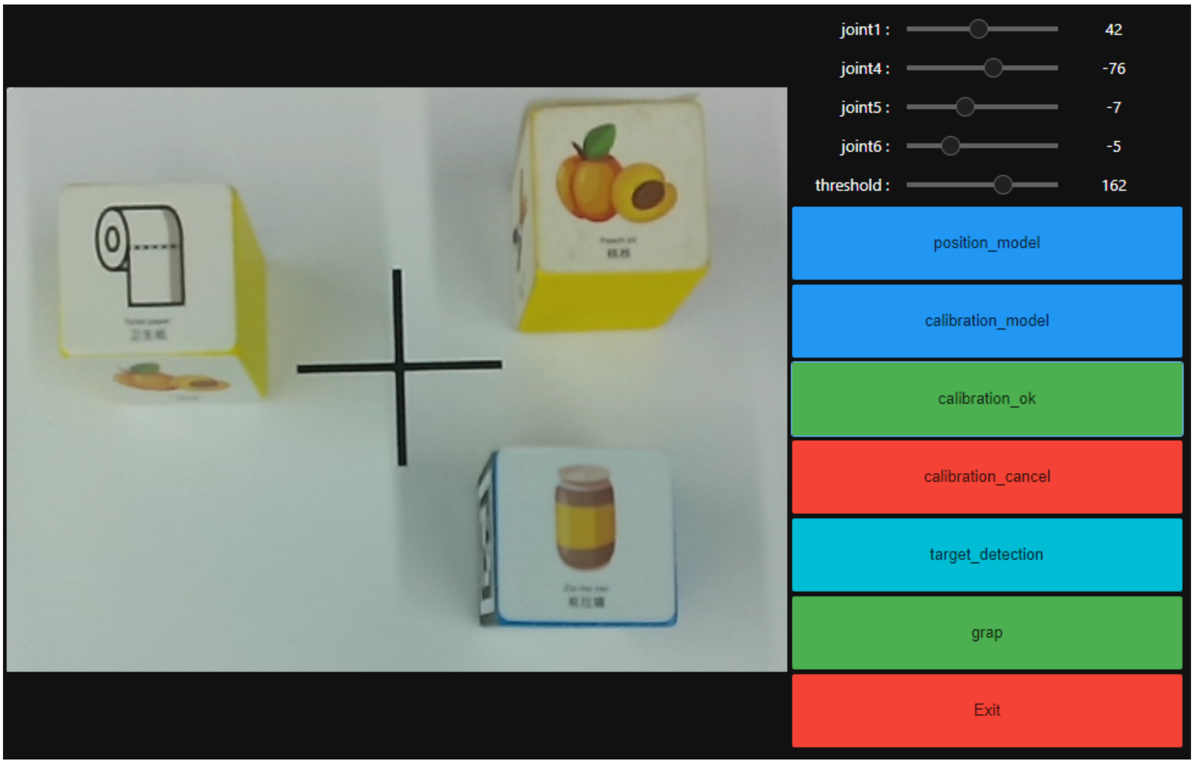
threshold : 162

Click 【calibration_ok】 again to calibrate OK. The camera screen will switch to the recognition area perspective.



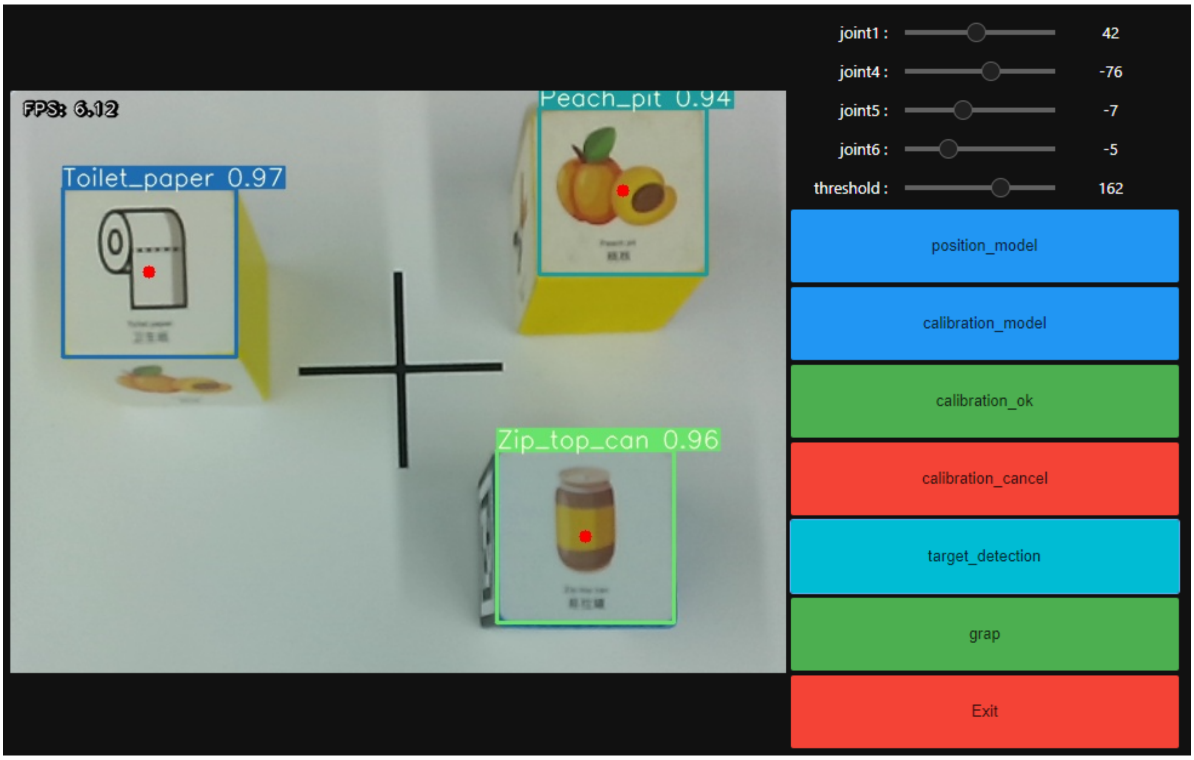
Place the block with the garbage category facing up into the recognition area.

Note: The view of the garbage image from the camera screen must be positive, not reversed.



Then click 【target_detection】 and wait for the model file to be loaded.

After the model file is loaded, start identifying garbage names.



joint1 : 42

joint4 : -76

joint5 : -7

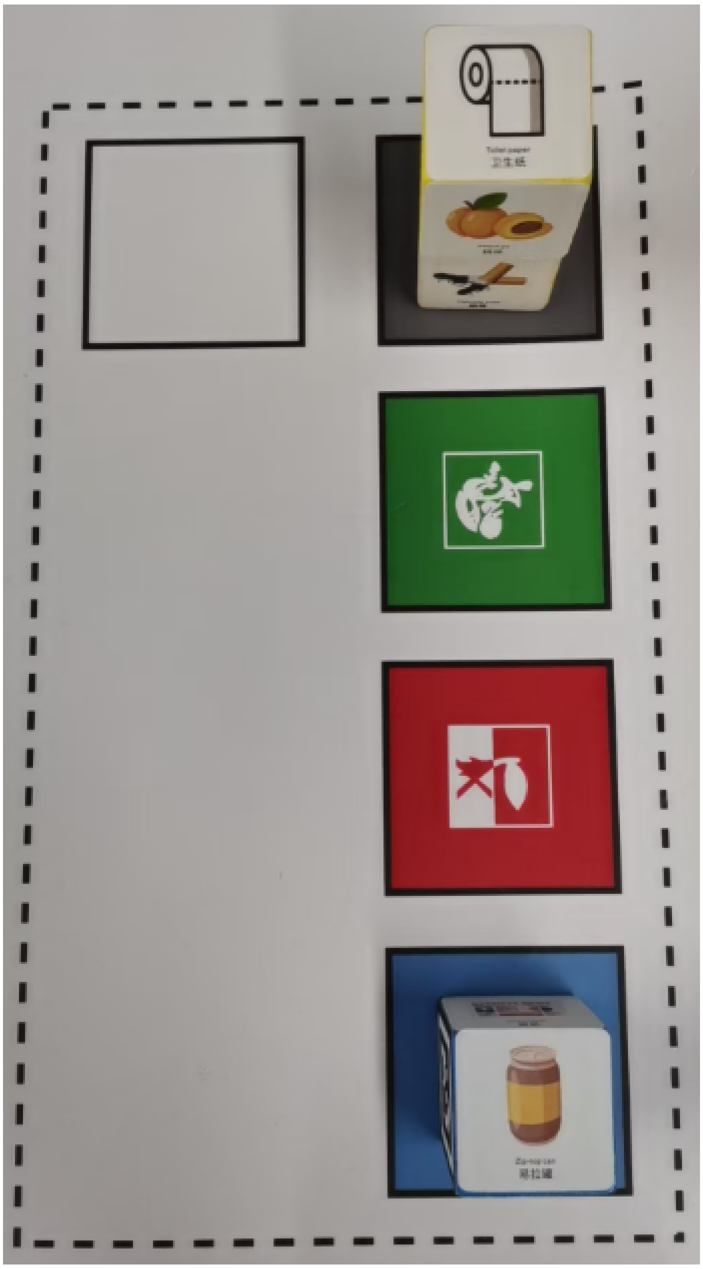
joint6 : -5

threshold : 162

Then, click the 【grap】 button to start sorting and stacking.

The system will identify the garbage name and grab the building blocks to the corresponding garbage area according to the category.

If there are two garbage of the same category, the same type of garbage will be stacked.



If you need to exit the program, please click the 【Exit】 button.