Garbage sorting

1. Garbage sorting instructions

The path for saving the file of the robot arm position calibration is ~/jetcobot_ws/src/jetcobot_color_identify/scripts/XYT_config.txt. After calibration, restart the program and click the calibration mode to automatically read the file information to reduce repeated calibration actions.

2. Important code explanation

Code path: ~/jetcobot_ws/src/jetcobot_garbage_yolov11/Garbage_sorting.ipynb

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

Since the camera may have deviations in the gripping position of the building block, it is necessary to increase the deviation parameter to adjust the deviation value of the robot arm to the recognition area. The type corresponding to garbage sorting and garbage stacking is "garbage", so the offset parameter under type == "garbage" needs to be changed. The X offset controls the front-to-back offset, and the Y offset controls the left-to-right offset.

```python
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 coordinates of the garbage area. If the coordinates of the placement location are inaccurate, you can modify them appropriately.

```python
# 可回收垃圾位置
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)

# 有害垃圾位置
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)

# 厨余垃圾位置
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)

    # 其他垃圾位置
    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)
```

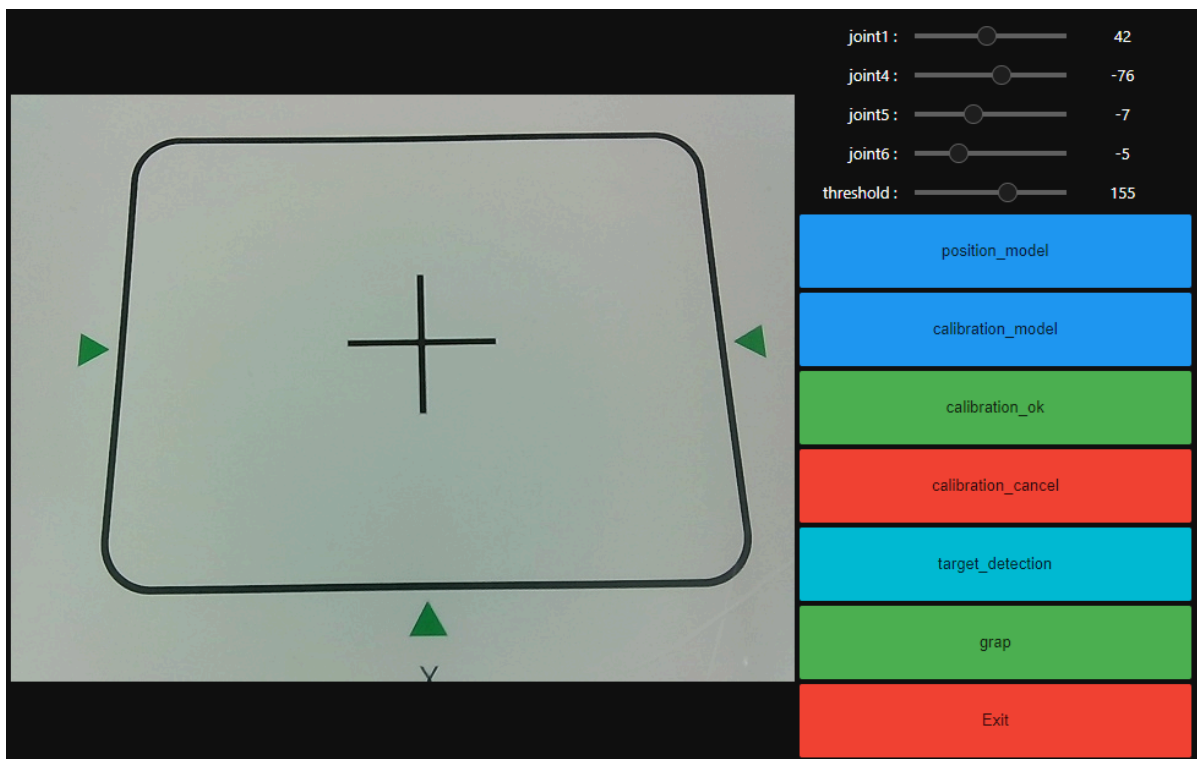# 3. Start the program

Start the program

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

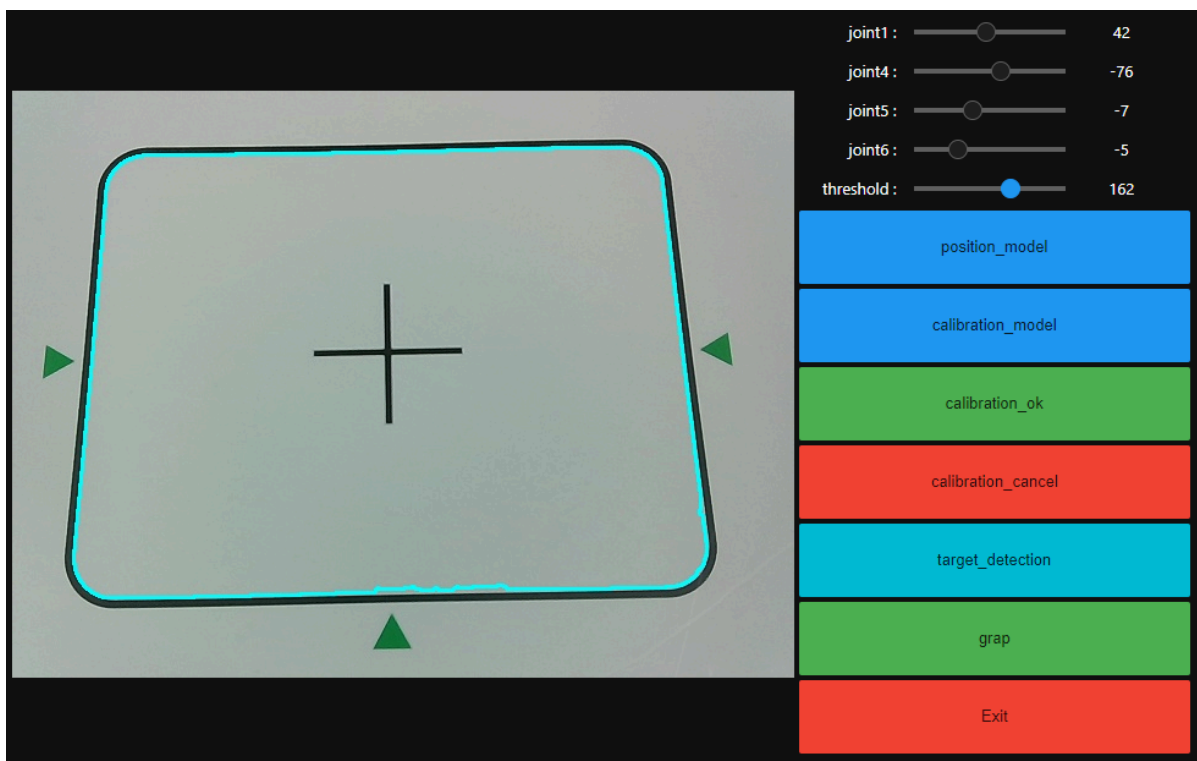Then click Run All Commands.

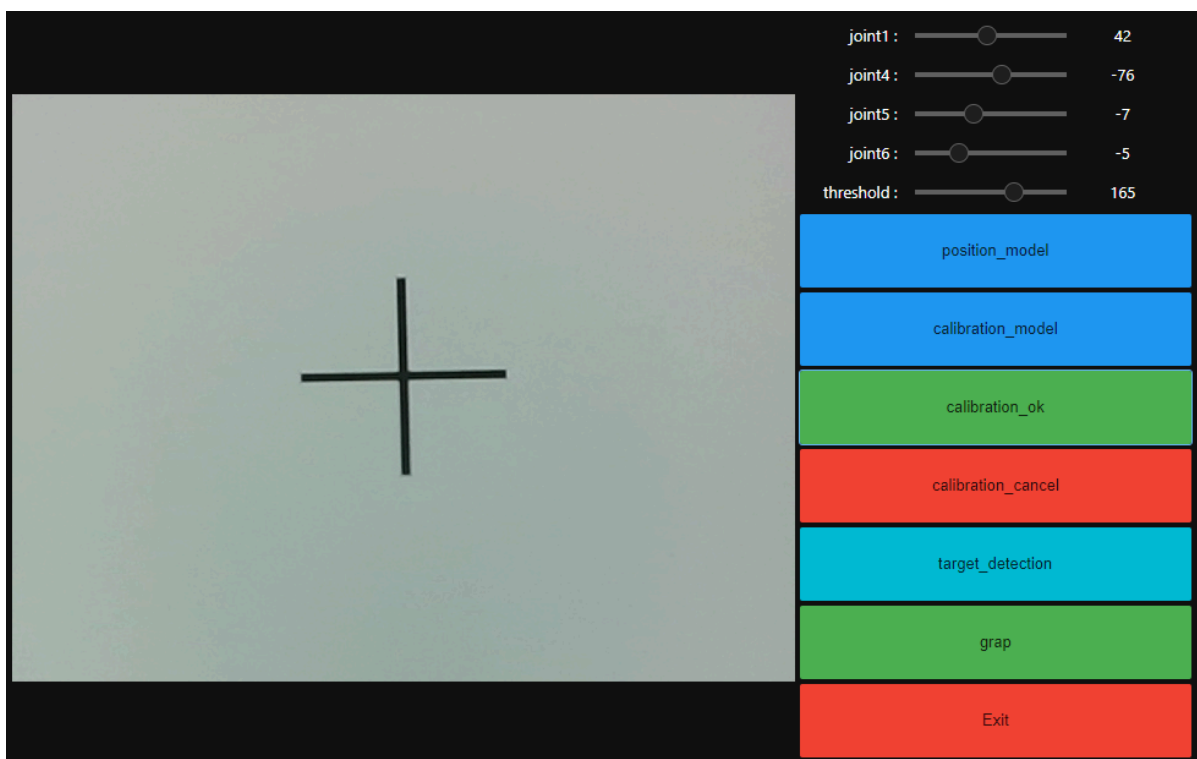
4. Experimental operation and effect

After the program is running, the jupyterlab webpage will display the control, the camera screen on the left, and the functions of the relevant buttons on the right.

Click the [position_model] button, drag the joint angle above, update the position of the robot 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 upper robot arm joint slider and threshold slider to overlap the displayed blue line with the black line of the recognition area.

joint1 :	42
joint4 :	-76
joint5 :	-7
joint6 :	-5
threshold :	162

position_model

calibration_model

calibration_ok

calibration_cancel

target_detection

grap

Exit

Click [calibration_ok] to calibrate OK, and the camera screen will switch to the recognition area view.

joint1 :	42
joint4 :	-76
joint5 :	-7
joint6 :	-5
threshold :	165

position_model

calibration_model

calibration_ok

calibration_cancel

target_detection

grap

Exit

Place the building block with the garbage category facing up in 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 recognizing the garbage name.
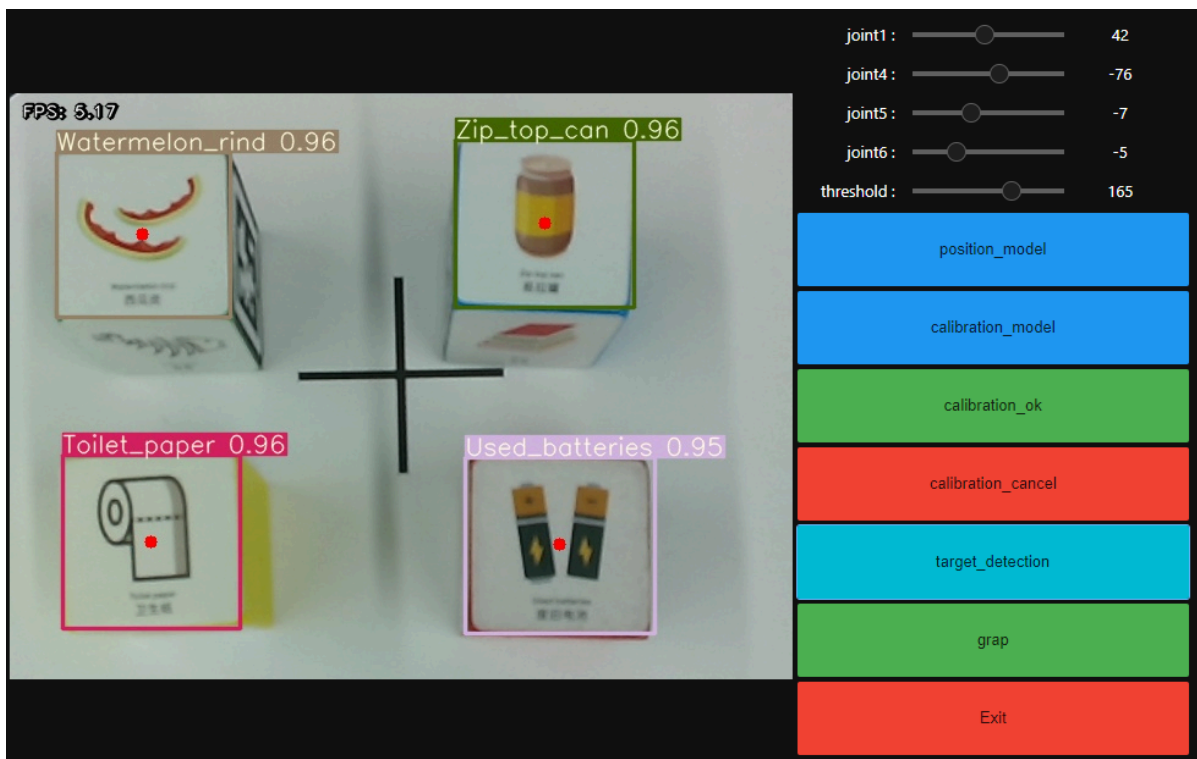
Then click the [grap] button to start sorting. The system will identify the garbage name and grab the building blocks into the corresponding garbage area according to the category.

If you need to exit the program, please click the [Exit] button.