

5. Garbage sorting

Since the garbage model training and simple garbage identification have been completed before, garbage identification and robotic arm sorting of garbage are now carried out.

How to play:

- 1) Make sure the large program is closed before running the program.
- 2) Start a command line window as the anti-decryption server and keep running.
`cd ~/dofbot_ws/ # Enter the workspace`
`catkin_make # Compile`
`source devel/setup.bash # Update system environment`
`roslaunch dofbot_info dofbot_server.launch # Start the server node`
- 3) Single garbage sorting gameplay path (no need to perform the previous step):
`dofbot_ws/src/dofbot_garbage_yolov5/single garbage sorting gameplay.ipynb`
- 4) Garbage sorting gameplay path: `dofbot_ws/src/dofbot_garbage_yolov5/garbage sorting gameplay.ipynb`

1. 1 Single garbage sorting gameplay

- Import header files

```
#!/usr/bin/env python
# coding: utf-8
import Arm_Lib
import cv2 as cv
import threading
from time import sleep
import ipywidgets as widgets
from IPython.display import display
from single_garbage_identify import single_garbage_identify
```

- Create instance, initialize parameters

```
# Create a target instance
single_garbage = single_garbage_identify()
# Initialization mode
model = "General"
```

- Initialize the robot arm position

```
import Arm_Lib
arm = Arm_Lib.Arm_Device()
joints_0 = [90, 135, 0, 0, 90, 30]
arm.Arm_serial_servo_write6_array(joints_0, 1000)
```

- Create controls

```

button_layout = widgets.Layout(width='320px', height='60px',
align_self='center')
output = widgets.Output()
# Exit
exit_button = widgets.Button(description='Exit', button_style='danger',
layout=button_layout)
imgbox = widgets.Image(format='jpg', height=480, width=640,
layout=widgets.Layout(align_self='center'))
controls_box = widgets.VBox([imgbox, exit_button],
layout=widgets.Layout(align_self='center'))

```

- Exit program button callback

```

def exit_button_callback(value):
    global model
    model = 'Exit'
    with output: print(model)
exit_button.on_click(exit_button_callback)

```

- main program

```

def camera():
    # Turn on the camera
    capture = cv.VideoCapture(0)
    # Loop execution when the camera is opened normally
    while capture.isOpened():
        try:
            # Read every frame of the camera
            _, img = capture.read()
            # Uniform image size
            img = cv.resize(img, (640, 480))
            img = single_garbage.single_garbage_run(img)
            if model == 'Exit':
                cv.destroyAllWindows()
                capture.release()
                break
            imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
        except KeyboardInterrupt: capture.release()

```

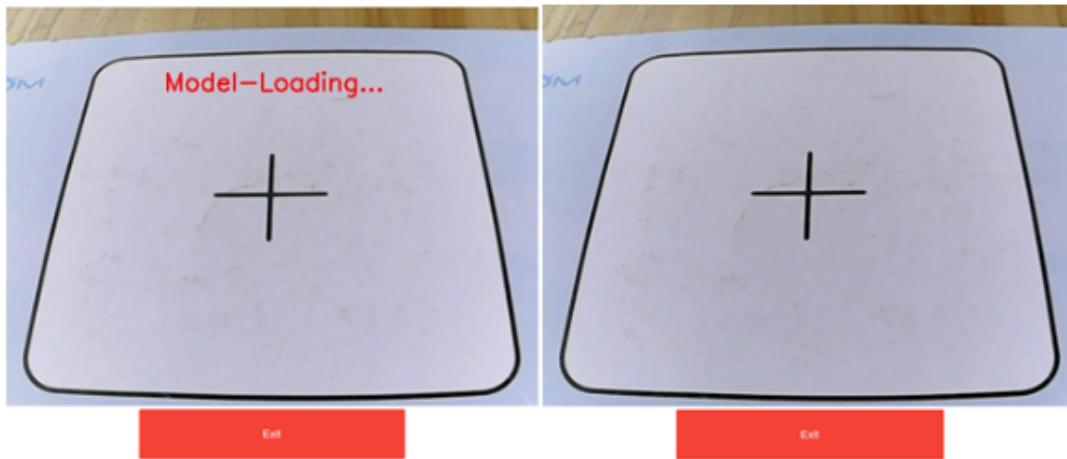
- start up

```

display(controls_box,output)
threading.Thread(target=camera, ).start()

```

When starting the above code block, you have to wait for about 2-3 seconds for the model to be loaded. Place the box with garbage on it facing the robot arm and place it in the center of the box. The robot arm will continue to accurately identify it 10 times and sort it automatically.

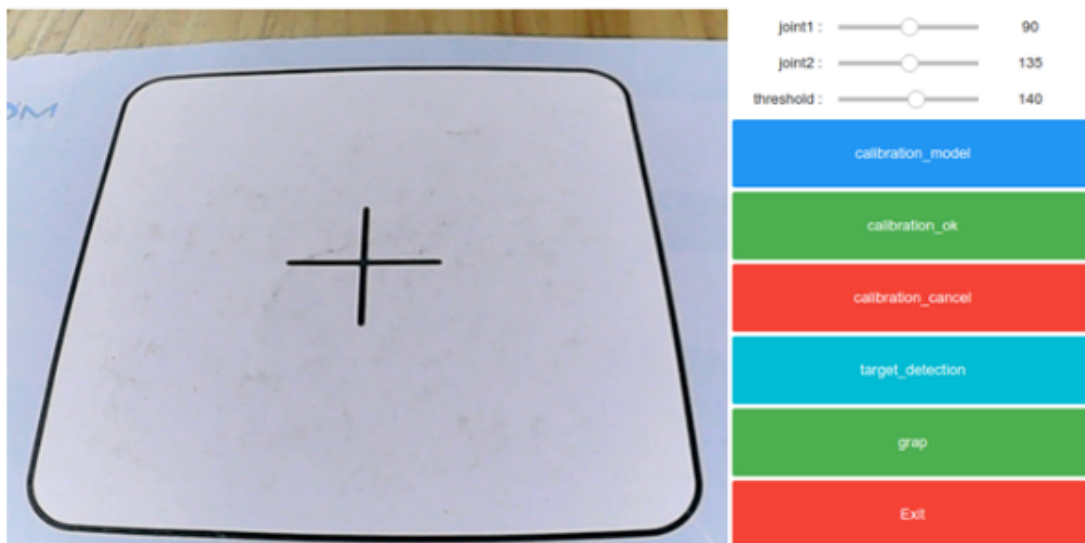


1.2 Garbage disorder sorting gameplay

Note: Before playing with garbage sorting, start a command line window and run it as an anti-decomposition server.

```
cd ~/dofbot_ws/ # Enter workspace
catkin_make     # Compiling
source devel/setup.bash # Update system environment
roslaunch dofbot_info dofbot_server.launch # Start the server node
```

Its operation process and sorting principle are similar to color sorting.



- First calibrate the frame. For specific calibration methods, please see the introduction in the [Visual Positioning] chapter.
- After the calibration is completed, click the [target_detection] button to load the model for identification. This process is time-consuming, so please wait patiently.
- When a block is recognized, click the [grap] button to sort and grab it.
- When the game is over, please click the [Exit] button to exit the program.

For detailed code, please view `dofbot_ws/src/dofbot_garbage_yolov5/garbage sorting gameplay.ipynb`

