Single Garbage Sorting(Jetson-Nano)

1. Function Description

The single garbage sorting function uses the yolo-v5 model to identify garbage names, then grabs building blocks from the center cross according to the garbage category and places them in the corresponding garbage category.

Note:
1. Before starting the program, please follow the [Assembly and Installation Tutorial] -> [Installing the Map] tutorial to correctly install the map before proceeding.
2. This example runs on the host machine; simply open a web browser and enter the IP address:8888

2. Code Block Design

- Import header files

```
import Arm_Lib
import os
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
from dofbot_utils.fps import FPS
```

- Create instances, initialize parameters

```
single_garbage = single_garbage_identify()
single_garbage.init_robot_joint()
fps = FPS()
model = "General"
```

- Create widgets

```
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'))
```

- Mode switching

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

- Main program

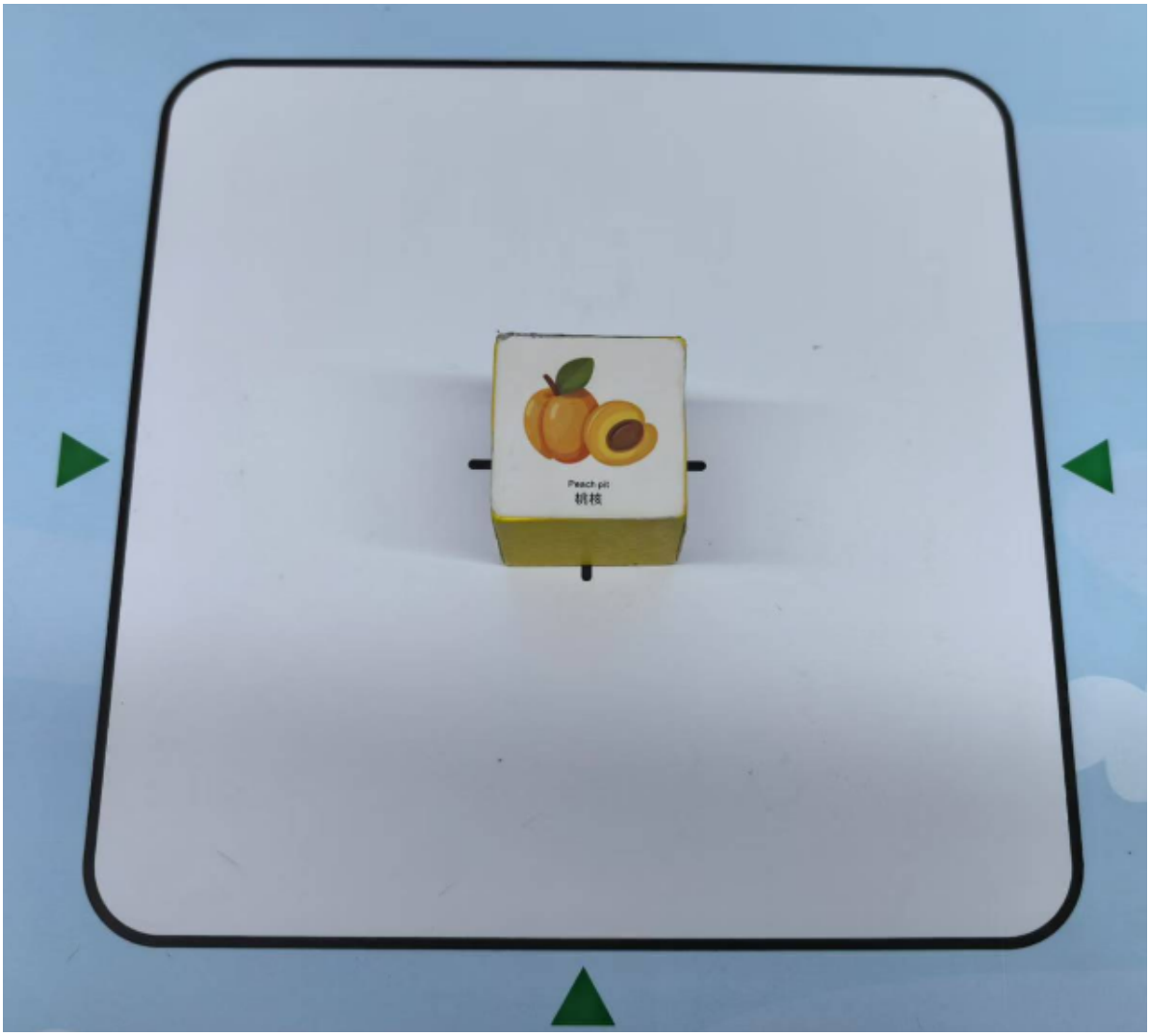
```
def camera():
    # 打开摄像头 Open camera
    capture = cv.VideoCapture(0)
    capture.set(cv.CAP_PROP_FRAME_WIDTH, 640)
    capture.set(cv.CAP_PROP_FRAME_HEIGHT, 480)
    # Execute in loop when camera opens normally
    while capture.isOpened():
        try:
            _, img = capture.read()
            fps.update_fps()
            img = single_garbage.single_garbage_run(img)
            if model == 'Exit':
                cv.destroyAllWindows()
                capture.release()
                break
            fps.show_fps(img)
            imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
        except:
            capture.release()
```

- Startup display

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

3. Experimental Setup

Place building blocks with garbage category labels on the cross, keeping the text direction consistent with the camera direction.



4. Start the Program

Start ROS Node Service

Open the system terminal and enter the following command. If it's already running, there's no need to start it again.

```
sudo systemctl start yahboom_arm.service
```

Start the Program

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

Code path:

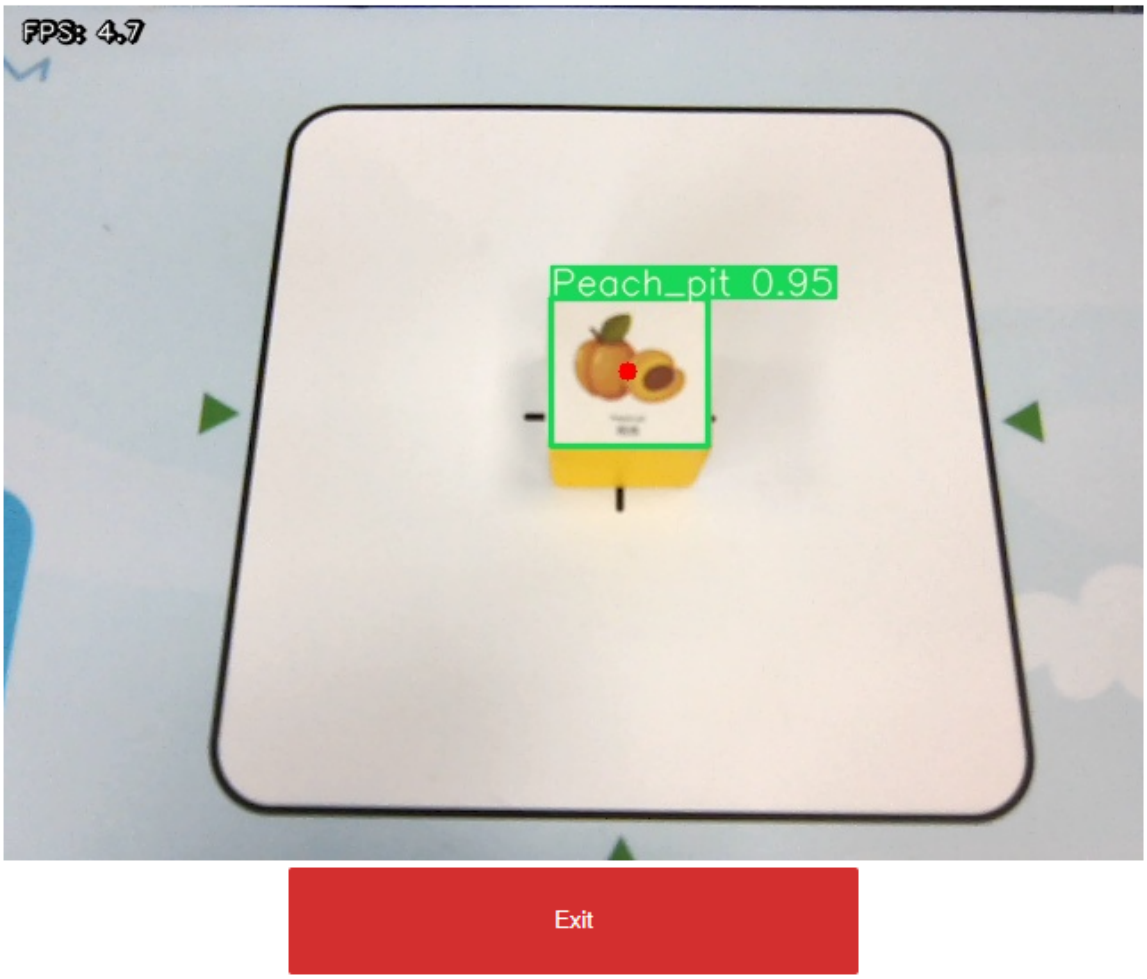
```
dofbot_ws/src/dofbot_garbage_yolov5/Garbage_Sorting_Single.ipynb
```

Click the "Run entire program" button on the jupyterlab toolbar, then scroll to the bottom.

5. Experimental Effects

Place building blocks with garbage category labels on the cross, keeping the text direction consistent with the camera direction. When the camera recognizes garbage, it prints the garbage name and the robotic arm automatically grabs the building block from the center cross and places it in the corresponding garbage category area, then returns to the initial pose.

The following example uses a walnut:



Effect after grabbing and placement is complete

Before performing the next recognition operation, please remove the building block to avoid conflicts during placement.

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

Since garbage recognition requires loading model files and occupies a large amount of memory space, [Exit] only ends the functionality and does not close the model file-related programs. You need to close the current kernels to close the model file programs. Click [Kernel] -> [Shut Down All Kernels] in the menu bar sequentially.