Gesture stacking

1. Gesture recognition instructions

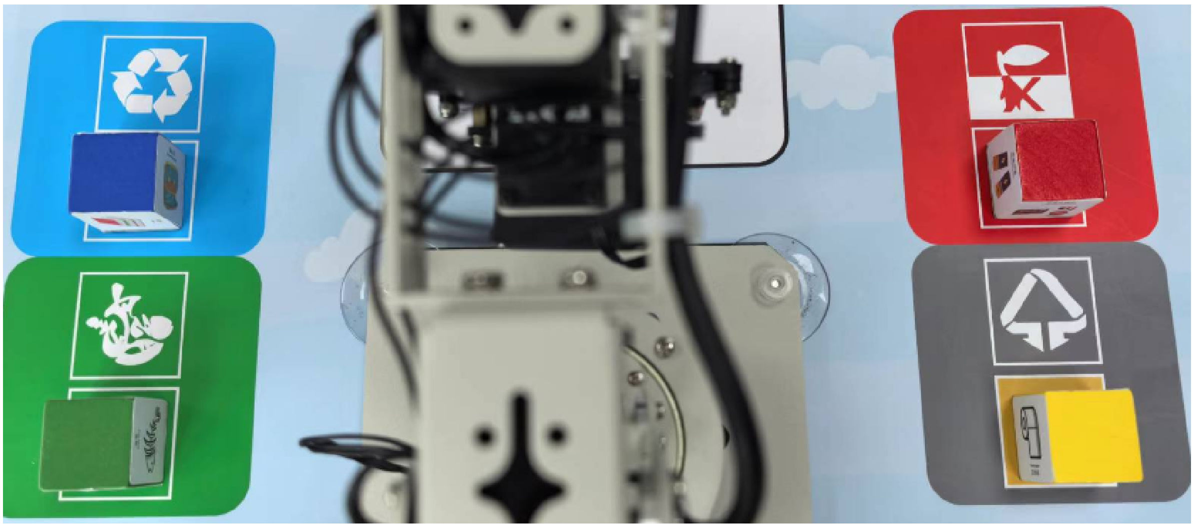
The gesture recognition stacking blocks function is based on gesture recognition gripping, and the placement mode is modified to stacking, and gesture 5 is modified to push down the blocks.

The gesture recognition grab blocks function is mainly combined with the mediapipe gesture recognition function, the robot arm and the map gameplay. A total of five gestures are recognized, gestures 1-5, where gestures 1-4 represent the gripping positions 1-4 on the map, and the blocks are stacked in order. Gesture 5 performs the function of pushing down the blocks and clears the recognition record.

Note: Before starting the program, please follow the [Assembly and Assembly Tutorial] -> [Install Map] tutorial, and operate after the map is correctly installed.

2. Experimental placement

Place the blocks at positions numbered 1-4.



3. Code block design

- Import header file

```
import cv2 as cv
import threading
import time
import ipywidgets as widgets
from IPython.display import display
from gesture_stacking import Gesture_Stacking
from dofbot_utils.fps import FPS
from dofbot_utils.robot_controller import Robot_Controller
```

- Create an instance and initialize parameters

```
robot = Robot_Controller()
robot.move_init_pose()
fps = FPS()

gesture = Gesture_Stacking()
model = 'General'
```

- Create controls

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

exit_button.on_click(exit_button_Callback)
```

- Switch Mode

```
def target_detection_Callback(value):
    global model,debug_pos
    model = 'Detection'
    with output: print(model)
    debug_pos = True
def grap_Callback(value):
    global model
    model = 'Grap'
    with output: print(model)
def exit_button_Callback(value):
    global model
    model = 'Exit'
    with output: print(model)
target_detection.on_click(target_detection_Callback)
grap.on_click(grap_Callback)
exit_button.on_click(exit_button_Callback)
```

- Main program

```
def camera():
    global model, gesture
    # 打开摄像头 Open camera
    capture = cv.VideoCapture(0)
    capture.set(3, 640)
    capture.set(4, 480)
    # Be executed in loop when the camera is opened normally
    # 当摄像头正常打开的情况下循环执行
    while capture.isOpened():
        try:
            _, img = capture.read()
            fps.update_fps()
            gesture.process(img)
            if model == 'Exit':
                capture.release()
                del gesture
                break
            fps.show_fps(img)
            imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
            time.sleep(0.001)
        except Exception as e:
            print("program end")
            print(e)
            capture.release()
```

- Start Display

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

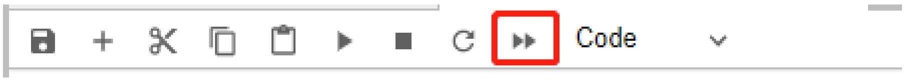
4. Start the program

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

Code path:

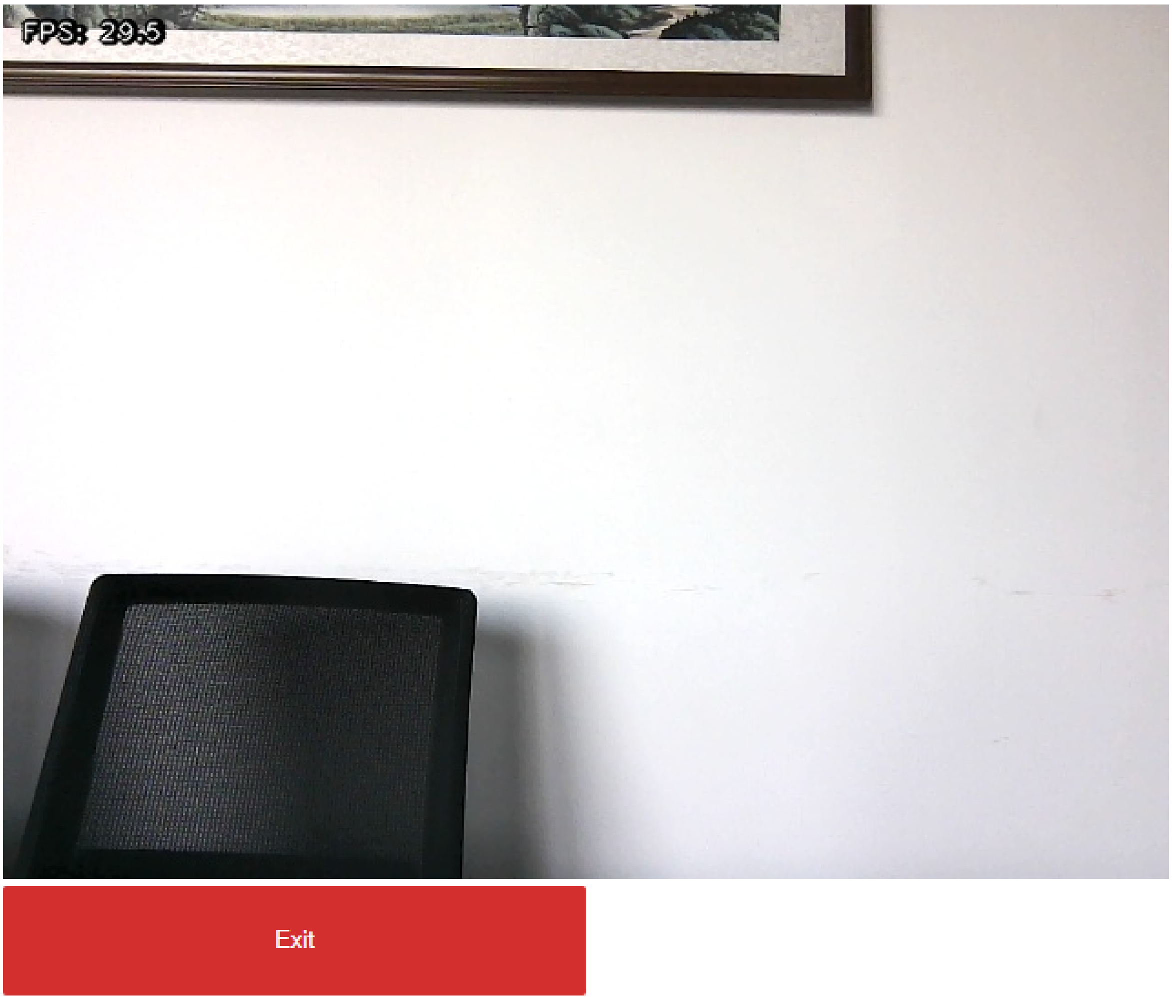
```
~/dofbot_ws/src/dofbot_gesture/scripts/Gesture_Stacking.ipynb
```

Then click Run all commands.



5. Experimental effect

After the program runs, slide to the bottom, and the jupyterlab webpage will display the camera screen and the functions of the related buttons.

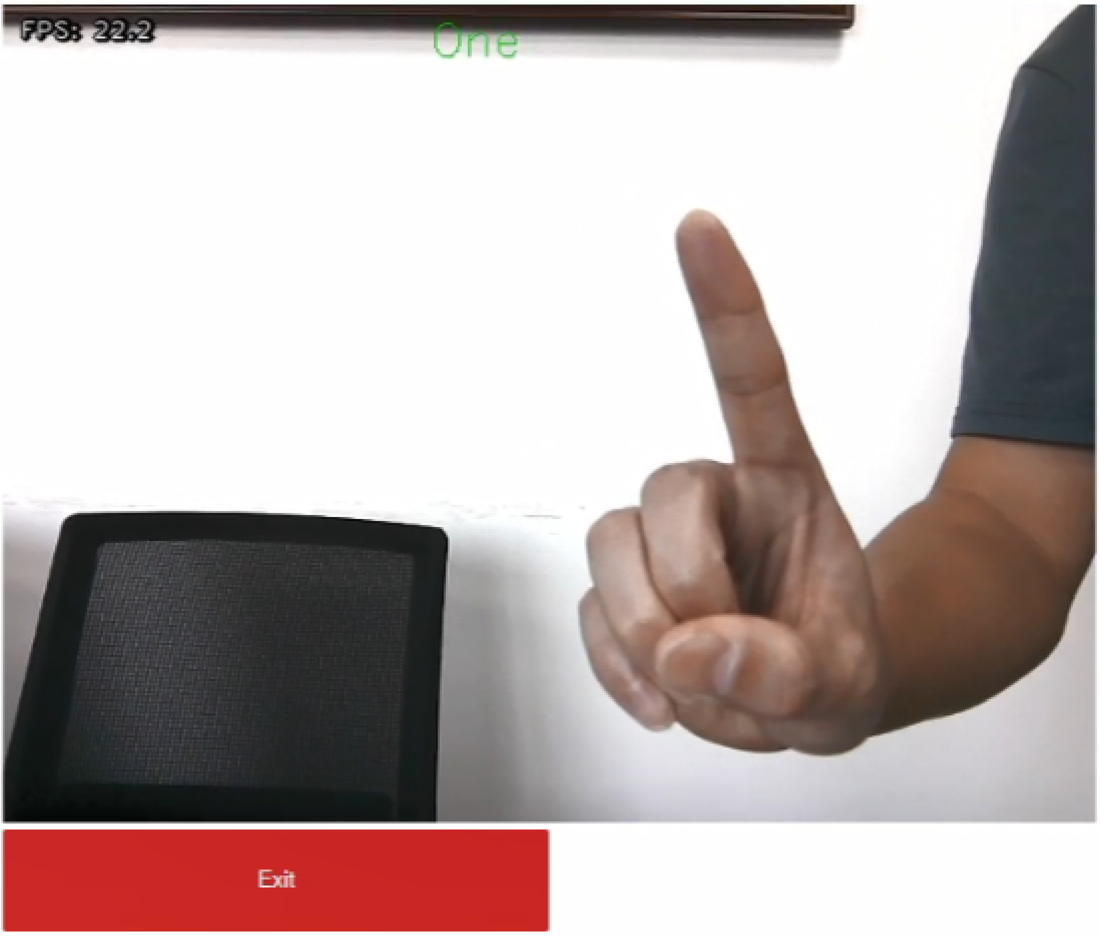


If the set gesture is recognized, the robot arm will perform the corresponding action. When the robot arm recognizes a number for the first time, it goes to the corresponding number position to pick up the building block and put it on the first layer. When it recognizes the gesture number for the second time, it goes to the corresponding number position to pick up the building block and put it on the second layer. When it recognizes the gesture number for the third time, it goes to the corresponding number position to pick up the building block and put it on the third layer. When it recognizes the gesture number for the fourth time, it goes to the corresponding number position to pick up the building block and put it on the fourth layer. Each number can only be recognized once, and the action will not be executed if it is recognized multiple times. When the number five is recognized, the robot arm pushes down the building block, clears the record, and starts again.

The gestures and actions in this example correspond to the following:

Gesture	**Function**
Gesture 1	Grab the building block at position 1 and stack it
Gesture 2	Grab the building block at position 2 and stack it
Gesture 3	Grab the building block at position 3 and stack it
Gesture 4	Grab the building block at position 4 and stack it
Gesture 5	Push down the building block and clear the record

As shown in the figure below:



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