Label Code Positioning

Orin board users can directly open a web page and enter IP address:8888 to access jupyter-lab and run directly. Jetson-Nano board users need to first enter the docker container, then enter the following command in docker:

```
cd
jupyter-lab --allow-root
```

Then open a web page and enter IP address:9999 to access jupyter-lab and run the following program.

Apriltag is a coded marker commonly used in machine vision, which has high recognition rate and reliability, and can be used for various tasks including augmented reality, robotics, and camera calibration. This apriltag label code uses the TAG36H11 format. The factory has already provided corresponding label codes, which are attached to building blocks. You need to take out the building blocks and place them under the camera view for recognition.

Code path:

```
#Jetson-Nano users need to enter the docker container to view
~/dofbot_pro/dofbot_apriltag/scripts/Apriltag_Position.ipynb
```

1. Main Code

- Import header files

```
import cv2 as cv
import threading
import random
from time import sleep
import ipywidgets as widgets
from IPython.display import display
from apriltag_identify import ApriltagIdentify
from dofbot_utils.fps import FPS
from dofbot_utils.robot_controller import Robot_Controller
```

- Create instances, initialize parameters

```
apriltag_Identify = ApriltagIdentify()
model = 'General'
robot = Robot_Controller()
robot.move_look_map()
fps = FPS()
```

- Main process

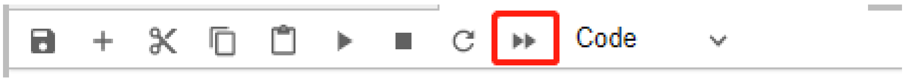
```
def camera():
    global HSV_learning,model
    # 打开摄像头 Open camera
    capture = cv.VideoCapture(0, cv.CAP_V4L2)
    capture.set(3, 640)
    capture.set(4, 480)
    capture.set(5, 30)
    # Be executed in loop when the camera is opened normally
    while capture.isOpened():
        try:
            _, img = capture.read()
            fps.update_fps()
            img, msg = apriltag_Identify.getApriltagPosition(img)
            if len(msg):
                print(msg)
            if model == 'Exit':
                capture.release()
                break
            fps.show_fps(img)
            imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
        except Exception as e:
            print("program end")
            print(e)
            capture.release()
```

- Startup

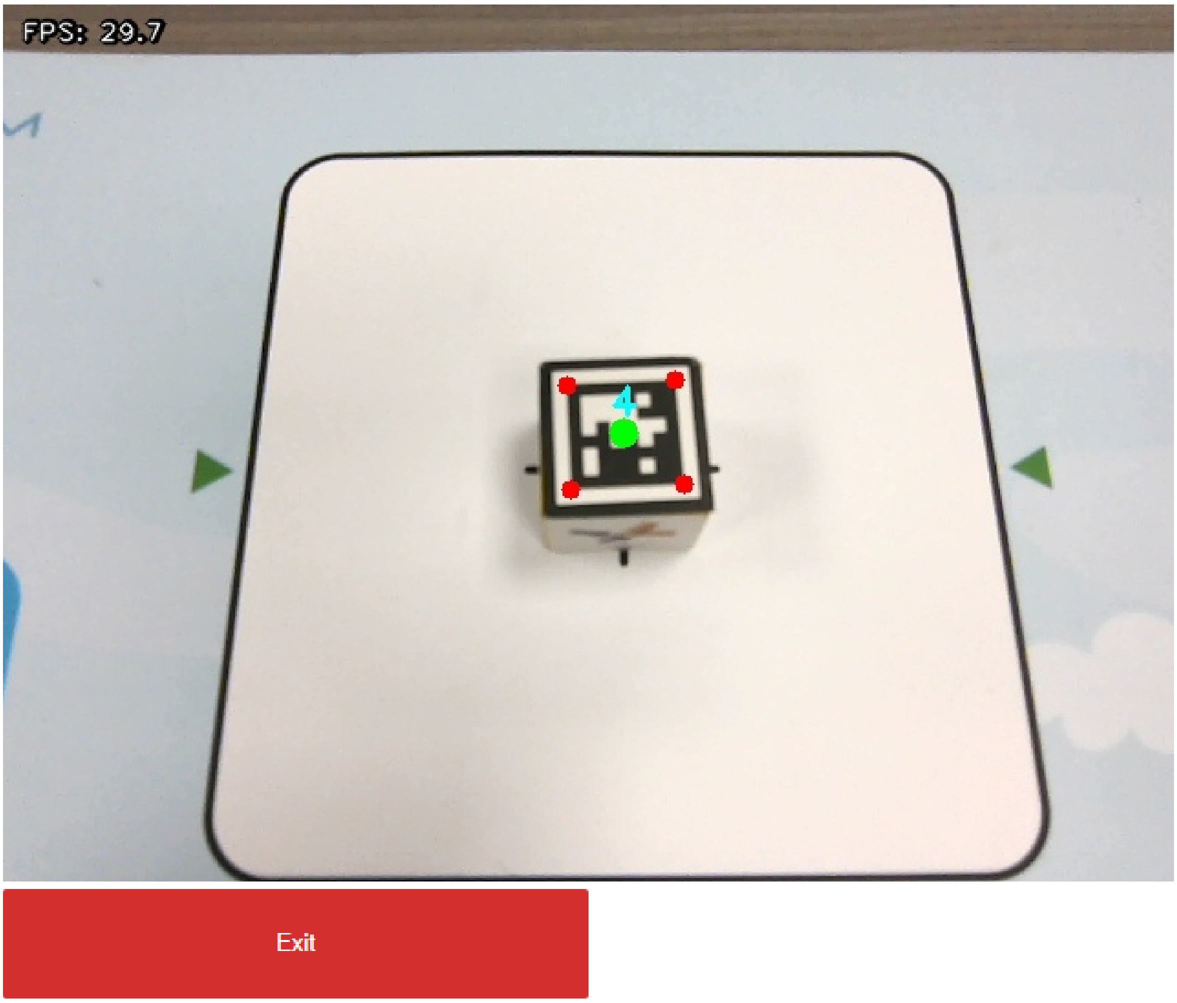
```
display(controls_box,output)
threading.Thread(target=camera, ).start()
```

2. Run the Program

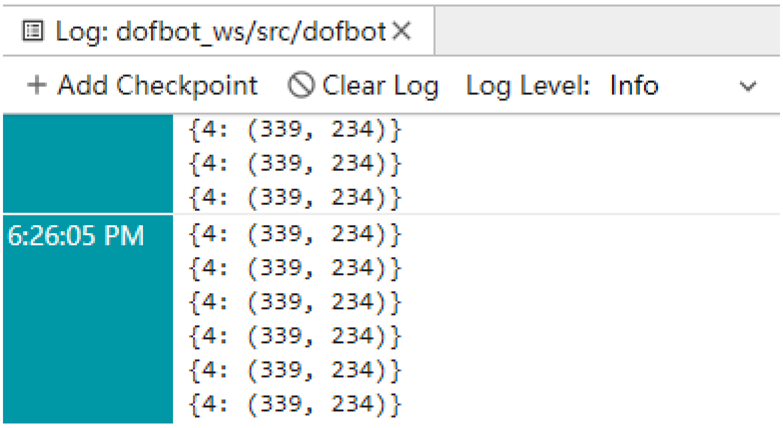
Click the run entire program button on the jupyterlab toolbar, then scroll to the bottom.



After the program starts, place the label code building blocks in the camera view. You can see that the label code is correctly positioned and the label code ID number is written. As you move the label code, the recognition effect will also follow the movement.



Open Log information, select Log Level as Info, you can see the printed label code ID number and position coordinates.



Log Level: Info

```
             {4: (339, 234)}
             {4: (339, 234)}
             {4: (339, 234)}
6:26:05 PM   {4: (339, 234)}
             {4: (339, 234)}
             {4: (339, 234)}
             {4: (339, 234)}
             {4: (339, 234)}
             {4: (339, 234)}
```

If you need to end the program, please click [Exit] to avoid affecting other programs from calling resources.