Face Tracking

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.

1. Function Overview

Based on the face positioning function, combined with the robotic arm to achieve face tracking functionality.

Code path:

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

2. Code Block Design

- Import header files

```
import cv2 as cv
import threading
from time import sleep
import ipywidgets as widgets
from IPython.display import display
from face_position import Face_Position
from dofbot_utils.robot_controller import Robot_Controller
from dofbot_utils.fps import FPS
from dofbot_utils.dofbot_config import *
```

- Create instances, initialize parameters

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

face = Face_Position()
model = 'General'
```

- Create widgets

```
button_layout = widgets.Layout(width='250px', height='50px',
align_self='center')
output = widgets.Output()
# 退出控件 exit button
exit_button = widgets.Button(description='Exit', button_style='danger',
layout=button_layout)
# 图像控件 Image widget
imgbox = widgets.Image(format='jpg', height=480, width=640,
layout=widgets.Layout(align_self='center'))
# 空间布局 spatial distribution
controls_box = widgets.VBox([imgbox, exit_button],
layout=widgets.Layout(align_self='center'))
# ['auto', 'flex-start', 'flex-end', 'center', 'baseline', 'stretch', 'inherit',
'initial', 'un
```

- Exit program

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

- Main program

```
def camera():
    global 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)
    while capture.isOpened():
        try:
            _, img = capture.read()
            fps.update_fps()
            img, pos = follow.follow_function(img)
            if model == 'Exit':
                cv.destroyAllWindows()
                capture.release()
                break
            fps.show_fps(img)
            imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
        except KeyboardInterrupt:capture.release()
```

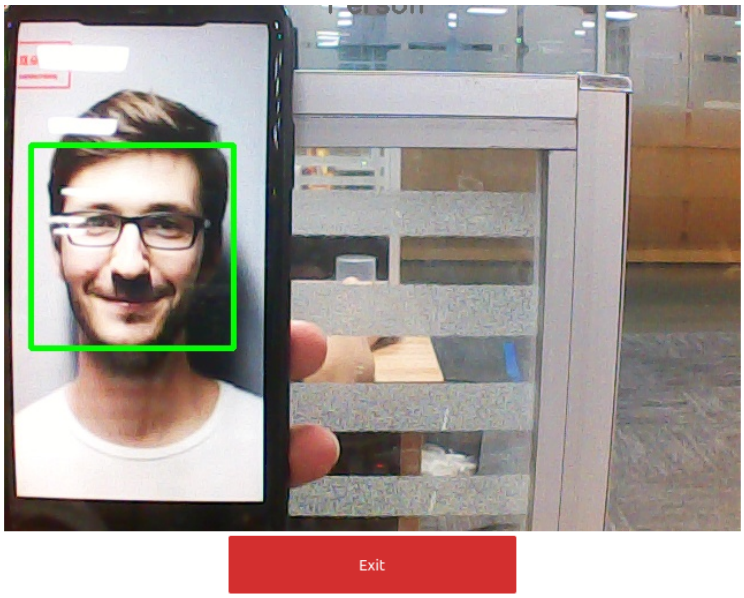
- Startup

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

3. Run the Program

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

You can see the camera display. At this time, place a face in the camera view, and the robotic arm will follow the face movement. Note that when moving the face, the speed cannot be too fast, otherwise the robotic arm may not be able to keep up due to moving too fast.



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