Face tracking experiment

1. Functional introduction

Based on the face positioning function, the face tracking function is realized in combination with the robotic arm.

Code path:

```
~/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 an instance and initialize parameters

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

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

- Create controls

```
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 the program

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

- Main program

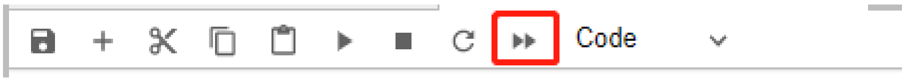
```
def camera():
    global model
    # 打开摄像头 Open camera
    capture = cv.VideoCapture(0, cv.CAP_V4L2)
    capture.set(3, 640)
    capture.set(4, 480)
    capture.set(5, 30)
    while capture.isOpened():
        try:
            _, img = capture.read()
            fps.update_fps()
            img, 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()
```

- Start

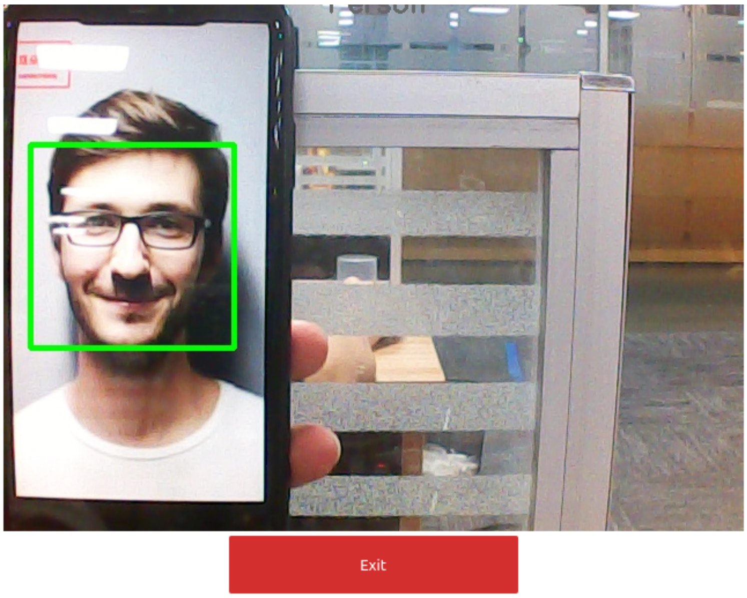
```
display(controls_box,output)
threading.Thread(target=camera, ).start()
```

3. Run the program

Click the Run Entire Program button on the jupyterlab toolbar and then pull it to the bottom.



You can see the camera screen. At this time, put the face into the camera screen, and the robot arm will move with the face. Note that the speed should not be too fast when moving the face, otherwise the robot arm may not be able to keep up because of the fast movement.



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