5. Face tracking experiment

5.1. Functional introduction

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

Code path: ~/jetcobot_ws/src/jetcobot_face_follow/face_tracking.ipynb

5.2, Code block design

- Import header file

```
import cv2 as cv
import threading
from time import sleep
import ipywidgets as widgets
from IPython.display import display
from face_follow import face_follow
```

- Create instance, initialize parameters

```
# Create instance
follow = face_follow()
# Initialize mode
model = 'General'
```

- Create widget

```
button_layout = widgets.Layout(width='250px', height='50px', align_self='center')
output = widgets.Output()
# Exit widget exit button
exit_button = widgets.Button(description='Exit', button_style='danger',
layout=button_layout)
# Image widget Image widget
imgbox = widgets.Image(format='jpg', height=480, width=640,
layout=widgets.Layout(align_self='center'))
# spatial distribution 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', 'unset']
```

- 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():
    global model
    # 打开摄像头 Open camera
    capture = cv.VideoCapture(0)
    while capture.isOpened():
        try:
            _, img = capture.read()
            img = cv.resize(img, (640, 480))
            img = follow.follow_function(img)
            if model == 'Exit':
                cv.destroyAllWindows()
                capture.release()
                break
            imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
        except KeyboardInterrupt:capture.release()
```

- start

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

5.3. Run the program

Click the Run the entire program button on the jupyterlab toolbar, and then pull it to the bottom.

You can see the camera screen. Now 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.