

4.Face positioning

The main purpose of color block positioning is to achieve face tracking for advanced functions. The principle is to determine the distance and position information of the face to the camera, and judge by calculating the coordinates of the center point of the face in the camera screen, thereby achieving face positioning. The experimental results show that it will always find the center point of the face to follow the movement and print the center point. **(For face recognition, please see the previous tutorial)**

Code path: dofbot_ws/src/dofbot_face_follow/face positioning.ipynb

1.1 Main code

- Import header files

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

- Create an instance and initialize parameters

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

- Mainly implements functions to print center coordinates.

```
def follow_function(self, img):

    img = cv.resize(img, (640, 480))
    # Copy the original image to avoid interference during processing
    # 复制原始图像,避免处理过程中干扰
    img = img.copy()
    # Convert image to grayscale
    # 将图像转为灰度图
    gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
    # Face detection
    # 检测人脸
    faces = self.faceDetect.detectMultiScale(gray, scaleFactor=1.3,
minNeighbors=5)
    if len(faces) != 0:
        face = self.face_filter(faces)
        # Face filtering
        # 人脸过滤
        (x, y, w, h) = face
        # Draw a rectangle on the original color map
        # 在原彩图上绘制矩形
```

```

        cv.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 4)
        cv.putText(img, 'Person', (280, 30), cv.FONT_HERSHEY_SIMPLEX, 0.8,
(105, 105, 105), 2)
        point_x = x + w / 2
        point_y = y + h / 2

        print("x= ",point_x)
        print("y= ",point_y)

    return img

```

- main process

```

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()

```

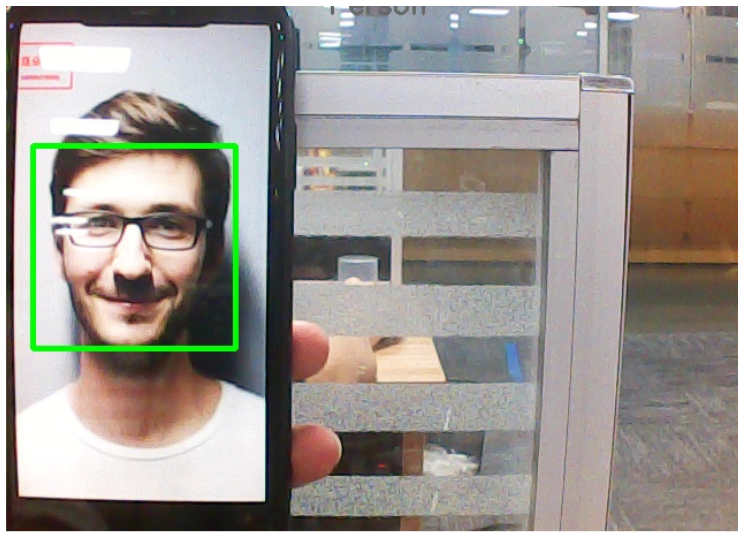
- start up

```

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

```

After the program is started, you can see the printed center coordinates and selected faces.



Exit

x= 80.0
y= 229.0
x= 83.0
y= 228.0
x= 93.0
y= 226.0
x= 100.0
y= 227.0
x= 107.0
y= 226.0