4.Face Detection

1. Introduction

This course mainly uses the camera of the motherboard to obtain the camera picture. The OpenCV library imports the face Haar feature classifier to analyze the position of the face in the image and frame the face.

2. Code flow analysis

Initialize the hexapod robot and set the camera pan/tilt angle. The default is S1=90 and S2=30. The initial angle can be modified according to actual needs.

```
from MutoLib import Muto
g_bot = Muto()
g_bot.Gimbal_1_2(90, 30)
```

Use the default haar as a cascade classifier for face detection.

```
face_haar = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
```

For the face detection task, first grayscale the camera image, then pass the image to the Haar face detection feature classifier for processing, detect the face and output the data of the face position, and then draw a box for the face in the image.

```
def face_detect(image):
    gray_img = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = face_haar.detectMultiScale(gray_img, 1.1, 3)
    for face_x,face_y,face_w,face_h in faces:
        return "1", (face_x,face_y,face_w,face_h)
    return None, (0,0,0,0)
```

Start a daemon thread, run the camera recognition task, and display the camera image.

```
def task_processing():
    global g_stop_program, g_start_function

    t_start = time.time()
    m_fps = 0
    fps = 0
    while g_camera.isOpened():
        if g_stop_program:
```

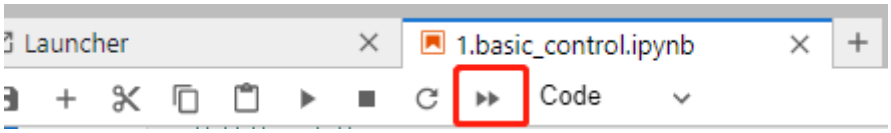
```
            break
        ret, frame = g_camera.read()

        if g_start_function:
            state, (face_x,face_y,face_w,face_h) = face_detect(frame)
            if state != None:
                cv2.rectangle(frame, (face_x, face_y), (face_x+face_w,
face_y+face_h), (0,255,255), 1)
            # time.sleep(.01)

        m_fps = m_fps + 1
        fps = m_fps / (time.time() - t_start)
        if (time.time() - t_start) >= 2:
            m_fps = fps
            t_start = time.time() - 1
        if g_start_function:
            cv2.putText(frame, "START " + str(int(fps)), (10,20),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,255,255), 1)
        else:
            cv2.putText(frame, "FPS " + str(int(fps)), (10,20),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,255,255), 1)

        # 图像传输给显示组件 The image is transmitted to the display component
        image_widget.value = bgr8_to_jpeg(frame)
```

3. Operation

Open the jupyterLab client and find the code path:

```
muto/Samples/AI_Samples/05_face_detect/face_detect.ipynb
```

Click to run all cells, and then scroll to the bottom to see the generated controls.



When the display control is started, it only displays the camera image. You need to click the Start button to turn on the face detection function. At this time, if a face enters the camera detection area, the face will be automatically framed.

Finally click the Close_Camera button to close the camera.