7. Face detection

7.1 Gameplay introduction

This course mainly uses the CSI camera of the RDK-X3 board to obtain the camera picture, detect the face and frame the face.

7.2 Core Content analysis

Load face Haar feature data.

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

The image is passed into the face haar parser to extract the face position information.

```
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)
```

After opening the start switch, determine whether there is a face in the returned image, and if so, frame the face.

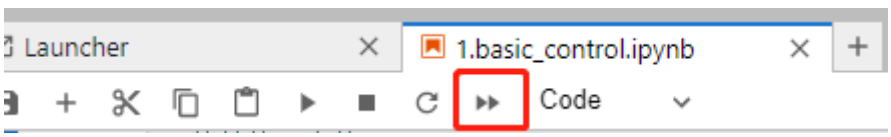
```
state, (face_x,face_y,face_w,face_h) = face_detect(frame)
if state != None:
    cv2.rectangle(frame, (face_x, face_y), (face_x+face_w, face_y+face_h),
(0,255,255), 1)
```

7.3 Gameplay operations

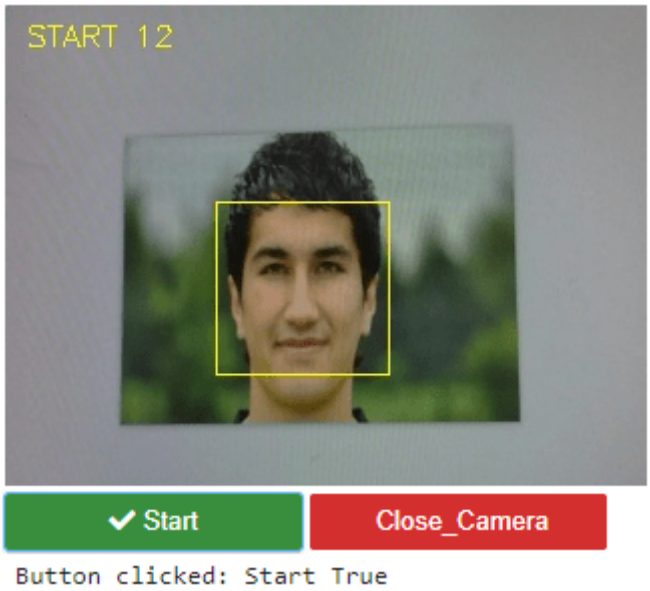
Open the jupyterLab client and find the code path:

```
/root/sunriseRobot/Samples/2_AI/07_face_detect/face_detect.ipynb
```

Click Run All Cells, and then drag to the bottom to see the generated controls.



Click the Start button below to start the face detection function. When a face appears in the camera screen, the face will be framed.



Finally click the Close_Camera button to close the camera.