

# Lesson 6 Facial Recognition

## 1. Program Logic

The trained face model is first zoomed to detect the face. Then the recognized face coordinates are converted to the coordinates before scaling. Judge whether it is the largest face, and frame the recognized face. Then set the gimbal servo to rotate left and right to obtain the face, and call the action group to let the robot perform the recognized feedback.

The source code of the program is located in:

/home/pi/TonyPi/Functions/FaceDetect.py

```

16  # Face Detection
17
18  if sys.version_info.major == 2:
19      print('Please run this program with python3!')
20      sys.exit(0)
21
22  # Threshold
23  conf_threshold = 0.6
24
25  # Model position
26  modelFile = "/home/pi/TonyPi/models/res10_300x300_ssd_iter_140000_fp16.caffemodel"
27  configFile = "/home/pi/TonyPi/models/deploy.prototxt"
28  net = cv2.dnn.readNetFromCaffe(configFile, modelFile)
29
30  servo_data = None
31  def load_config():
32      global servo_data
33
34      servo_data = yaml_handle.get_yaml_data(yaml_handle.servo_file_path)
35
36  load_config()
37
38  servo2_pulse = servo_data['servo2']
39  # Initial position
40  def initMove():
41      Board.setPWMServoPulse(1, 1800, 500)
42      Board.setPWMServoPulse(2, servo2_pulse, 500)
43
44  d_pulse = 10
45  start_greet = False
46  action_finish = True
47  # variable reset
48  def reset():
49      global d_pulse
50      global start_greet
51      global servo2_pulse
52      global action_finish
53

```

## 2. Operation Steps

---

**i** Pay attention to the text format in the input of instructions.

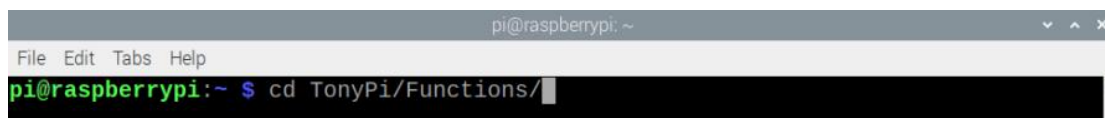
---

1) Turn on robot and connect to Raspberry Pi desktop with VNC.

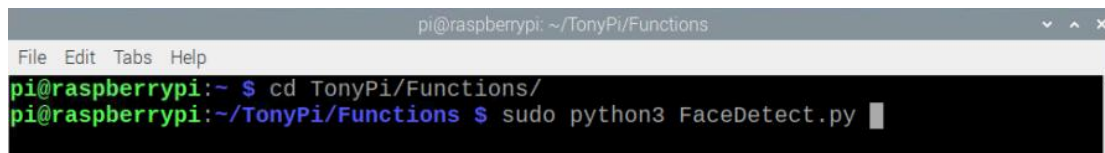
2) Click  or press “Ctrl+Alt+T” to enter the LX terminal.



3) Enter “cd TonyPi/Functions/” command, and then press “Enter” to come to the category of games programmings.



4) Enter “sudo python3 FaceDetect.py”, then press “Enter” to start the game.



5) If you want to exit the game programming, press “Ctrl+C” in the LX terminal interface. If the exit fails, please try it few more times.

## 3. Project Outcome

---

**i** Please do not try the Facial Recognition game under strong light, such as sunlight. Strong light will affect the recognition performance, so it is recommended to play this game indoors. It's better to set the distance between face and camera with 50-100cm.

---

Start the facial recognition function, TonyPi will rotate its head to detect face. It will stop when the face is recognized, and run the greeting actions.

## 4. Function Extension

### 4.1 Modify Feedback Action

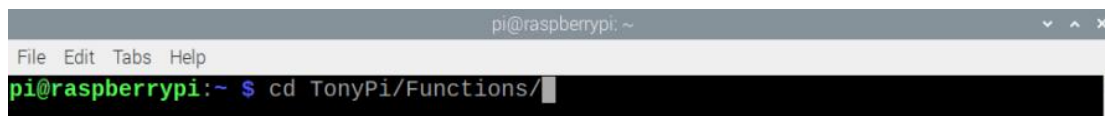
---

The built-in file is located in /home/pi/TonyPi/ActionGroups.

---

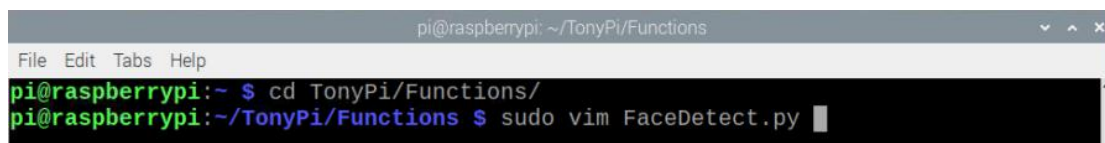
Program default setting is that TonyPi will execute the greeting action when detect the face. The feedback action can be revised to others such as bowing.

Step1: Enter command “cd TonyPi/Functions/” to the directory where the game program is located.



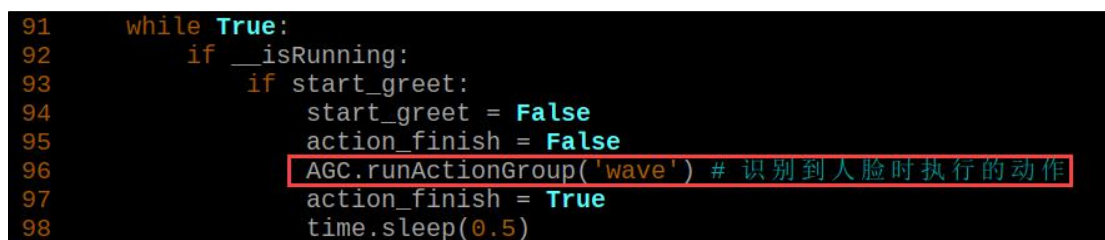
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $ cd TonyPi/Functions/
```

Step2: Enter command “sudo vim FaceDetect.py” to go into the game program through vi editor.



```
pi@raspberrypi: ~/TonyPi/Functions  
File Edit Tabs Help  
pi@raspberrypi:~ $ cd TonyPi/Functions/  
pi@raspberrypi:~/TonyPi/Functions $ sudo vim FaceDetect.py
```

Step3: Input “96” and press “shfit+g” to the line for modification.



```
91     while True:  
92         if __isRunning:  
93             if start_greet:  
94                 start_greet = False  
95                 action_finish = False  
96                 AGC.runActionGroup('wave') # 识别到人脸时执行的动作  
97                 action_finish = True  
98                 time.sleep(0.5)
```

“wave” in the above image is the name of greeting action. If we want to revise the action to bowing, enter “bow” instead of “wave” in the “Action group instruction” in the path /home/pi/TonyPi/ActionGroups

---

The action name can be found in the “Action group instruction”.

---

Step4: Press “i” to enter the editing mode, then modify “wave” to “bow”.

```

91     while True:
92         if __isRunning:
93             if start_greet:
94                 start_greet = False
95                 action_finish = False
96                 AGC.runActionGroup('bow') # 识别到人脸时执行的动作
97                 action_finish = True
98                 time.sleep(0.5)
99             else:
100                 if servo2_pulse > 2000 or servo2_pulse < 1000:
101                     d_pulse = -d_pulse
102
103                     servo2_pulse += d_pulse
104                     Board.setPWMServoPulse(2, servo2_pulse, 50)
105                     time.sleep(0.05)
106                 else:
107                     time.sleep(0.01)
108
-- 插入 --
97, 37
55%

```

Step5: Press “Esc” to enter last line command mode. Input “:wq” to save the file and exit the editor.

## 5. Program Parameter Instruction

### 5.1 Face Detection Parameter

The parameters involved in the process of detection are as follow:

In the process of detection, the image needs to be preprocessed, such as mean reduction, scaling, cropping, exchange channels, to returns a 4-channel blob (“blob” can be simply understood as an N-dimensional array for neural network input ), as the figure shown below:

```

124
125 blob = cv2.dnn.blobFromImage(img_copy, 1, (150, 150), [104, 117, 123], False, False)
126 net.setInput(blob)

```

The first parameter “img\_copy” is the input image.

The second parameter “1” is the image scaling ratio after mean subtraction.

The third parameter “(150, 150)” is the spacial size of the output image and the values here represent w=150 and h=150 respectively.

The fourth parameter “[104, 117, 123]” is the subtraction value of each channel. The image channel order of OpenCV is B, G, R. The value here means the value of the B channel minus 104, the value of the G channel minus 117, and the value of the R channel minus 123.

The fifth parameter “False” is used to judge whether R and B channels are exchanged. It is “False” by default, that is, R and B channels are not exchanged. When the order of mean subtraction is assumed to be R, G, B, the R and B channels need to be exchanged, that is, fill in “True”.

The sixth parameter “False” is used to decide whether to crop the image, the default is “False”, that is, the image will not be cropped, the size will be adjusted directly, and the aspect ratio will be preserved. When the value is “True”, the image is first scaled proportionally, and then cropped from its center according to the size set by parameter 3.

## 5.2 Face Recognition Parameter

The parameters involved in face recognition are as follow:

After a human face is detected, rectangle() function can be used to draw a rectangle to frame the face, as the figure shown below:

```
132 x1 = int(detections[0, 0, i, 3] * img_w)
133 y1 = int(detections[0, 0, i, 4] * img_h)
134 x2 = int(detections[0, 0, i, 5] * img_w)
135 y2 = int(detections[0, 0, i, 6] * img_h)
136 cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2, 8) #将识别到的人脸框出
137 if abs((x1 + x2)/2 - img_w/2) < img_w/4:
```

The first parameter “img” is the input image and it refers to the image of drawn rectangle here.

The second parameter “(x1, y1)” is the coordinates of starting point of the rectangle.

The third parameter “(x2, y2)” is the coordinates of the ending point of the rectangle.

The fourth parameter “(0, 255, 0)” is the color of the rectangle edge and its order is B, G, R. Here is green.

The fifth parameter “2” the width of the rectangle edge and “-1” represents the rectangle is filled with specified color.

The sixth parameter “8” is the type of rectangle edge and here it is 8-connected wire.

## 5.3 Execute Action Parameter

After recognizing human face, the robot will call and run “wave.d6a” action group file in “/home/pi/TonyPi/ActionGroups”, and then perform “wave” action, as the figure shown below:

```
92 if __isRunning:
93     if start_greet:
94         start_greet = False
95         action_finish = False
96         AGC.runActionGroup('wave') # 识别到人脸时执行的动作
97         action_finish = True
98         time.sleep(0.5)
```