Lesson 3 Color Sorting

Note: The servo on robotic hand has been set with corresponding ID and limited angle before delivery. If the ID and the limited angle change due to other operations, please refer to the tutorial in “5.Appendix/8.Servo Debugging Method” to adjust.

1. Getting Ready

1) This lesson can be started after the robotic hands are assembled. The specific assembly method refers to “Advanced Lesson/ Vision Gripping Lesson/ Lesson 1 Robotic Hands Assembly”.

2) Place the colored block on the surface at the height of 15 cm and the error of height can not exceed 1 cm, otherwise it will affect the performance. You can use the product package box to experience this game directly.

2. Working Principle

The working principle of this this lesson is as follow:

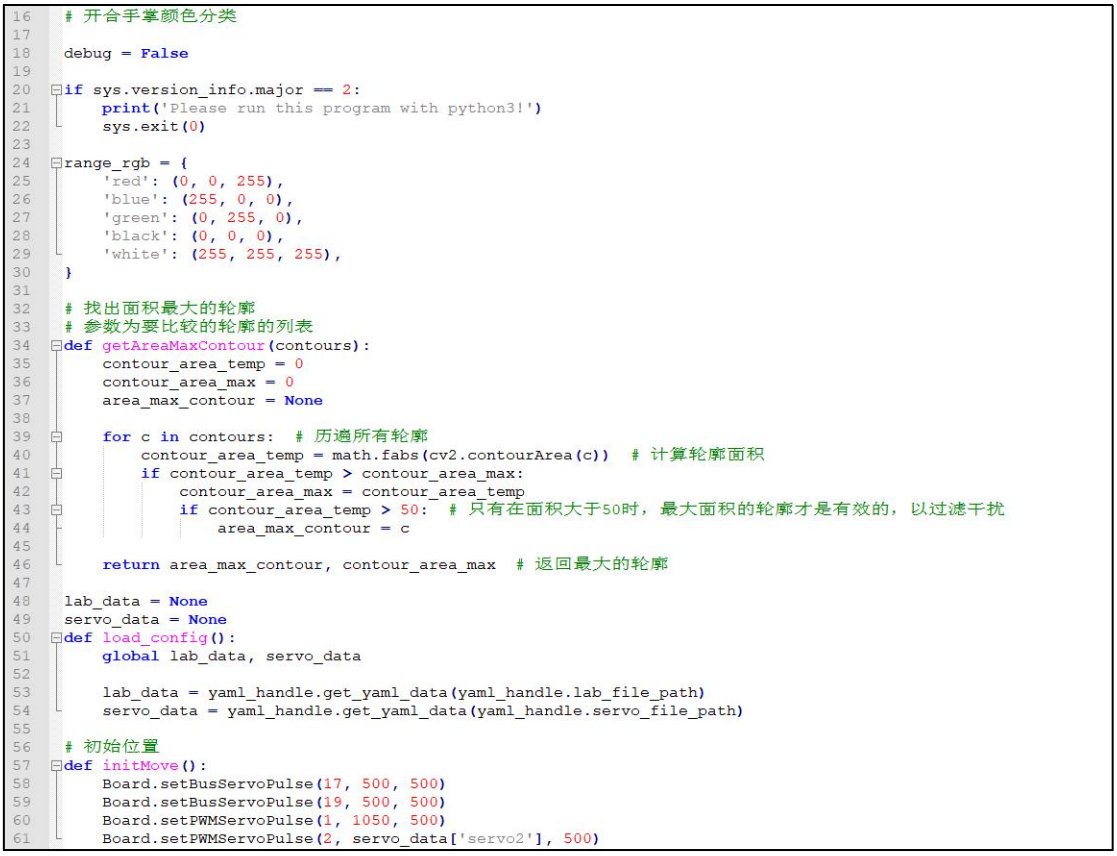
The color is recognized through Lab color space firstly. Then convert RGB color into Lab color space, and proceed with binarization, dilation and erosion to obtain the contour of the target color.

Then, filter out the largest contour among red, green and blue three colors through he traversal function, and circle it, so that the color of the object can be recognized.

At last, judge according to the recognized color. and control the robot to grip and place or shake its head after recognizing the corresponding color object.

The source code of the program is located in:

/home/pi/TonyPi/Extend/Color_classify.py



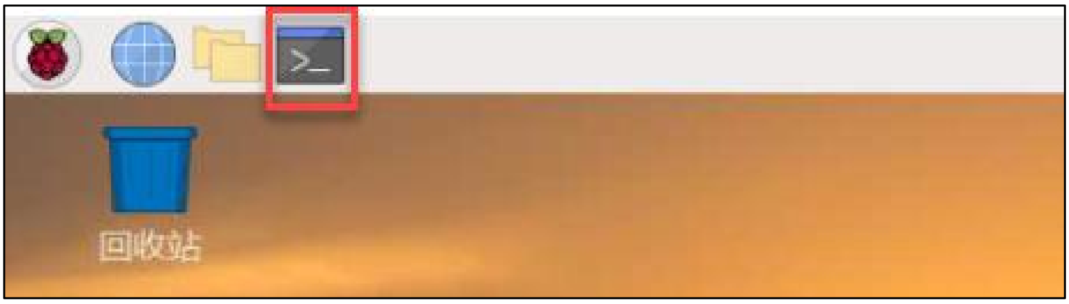
```
# 开合手掌颜色分类

debug = False

if sys.version_info.major == 2:
    print('Please run this program with python3!')
    sys.exit(0)

range_rgb = {
    'red': (0, 0, 255),
    'blue': (255, 0, 0),
    'green': (0, 255, 0),
    'black': (0, 0, 0),
    'white': (255, 255, 255),
}

# 找出面积最大的轮廓
# 参数为要比较的轮廓的列表
def getAreaMaxContour(contours):
    contour_area_temp = 0
    contour_area_max = 0
    area_max_contour = None

    for c in contours:  # 历遍所有轮廓
        contour_area_temp = math.fabs(cv2.contourArea(c))  # 计算轮廓面积
        if contour_area_temp > contour_area_max:
            contour_area_max = contour_area_temp
            if contour_area_temp > 50:  # 只有在面积大于50时，最大面积的轮廓才是有效的，以过滤干扰
                area_max_contour = c

    return area_max_contour, contour_area_max  # 返回最大的轮廓

lab_data = None
servo_data = None
def load_config():
    global lab_data, servo_data

    lab_data = yaml_handle.get_yaml_data(yaml_handle.lab_file_path)
    servo_data = yaml_handle.get_yaml_data(yaml_handle.servo_file_path)

# 初始位置
def initMove():
    Board.setBusServoPulse(17, 500, 500)
    Board.setBusServoPulse(19, 500, 500)
    Board.setPWMServoPulse(1, 1050, 500)
    Board.setPWMServoPulse(2, servo_data['servo2'], 500)
```

3. Operation Step

The entered command must be pay attention to case sensitivity and space.

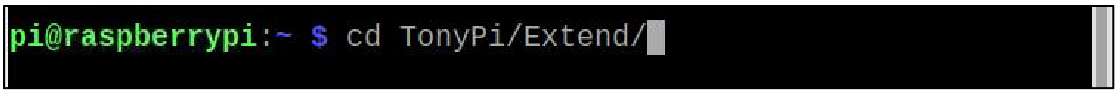
1) Turn on the robot and connect to Raspberry Pi desktop with VNC.

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



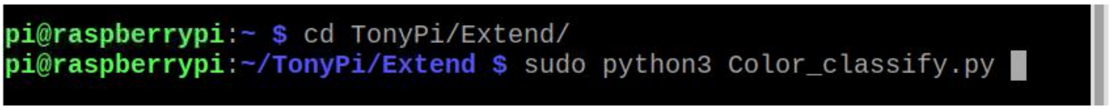
3) Enter “cd TonyPi/Extend/” and press “Enter” to come to the directory of

game programmings.



```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

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



```
pi@raspberrypi:~ $ cd TonyPi/Extend/
pi@raspberrypi:~/TonyPi/Extend $ sudo python3 Color_classify.py
```

5) If want to exit the game, press “Ctrl+C” in the LX terminal. Please try multiple times if fail to exit.

4. Project Outcome

After the program is started, TonyPi Pro will squat first. When the red block is recognized, it will grip the block with right hand and place it to the right. When the blue block is recognized, it will grip the block and place it to the left. When the green block is recognized, it will shake head.

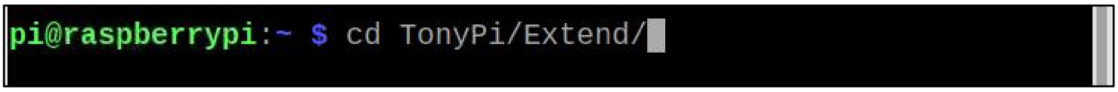
5. Function Extension

5.1 Modify Default Recognized Color

The program of color sorting has three built-in colors by default, which are red, green and blue. When red and blue are recognized, TonyPi Pro will grip the corresponding object. When green is recognized, TonyPi Pro will shake its head.

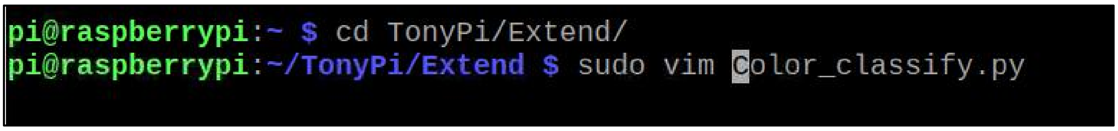
Take the robot nods when recognizing red and grip when recognizing green and blue as example. The specific modification steps are as follow:

1) Enter “cd TonyPi/Extend/” command and press “Enter” to come to the directory of the game programmings.



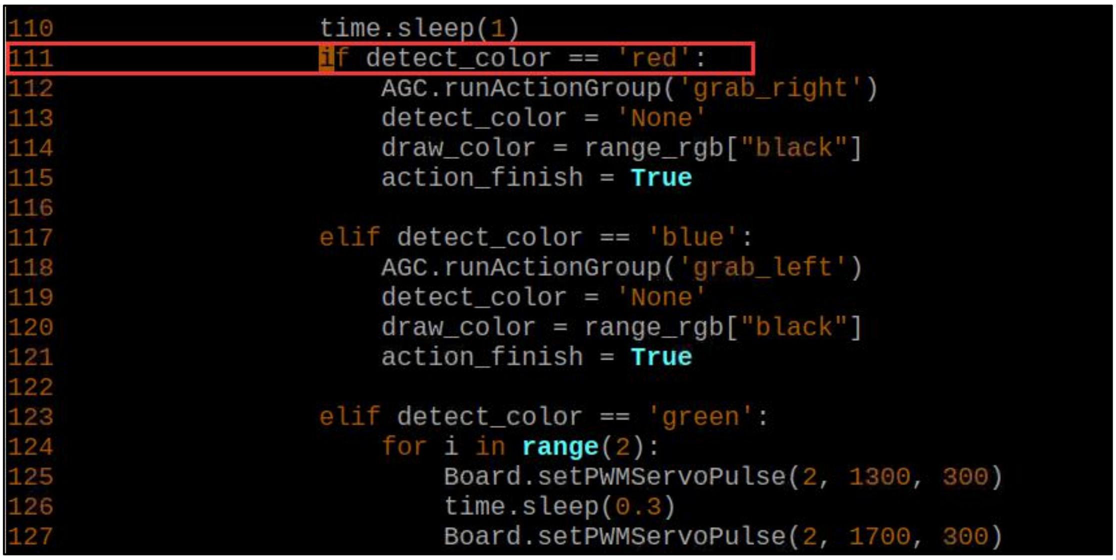
```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

2) Enter “sudo vim Color_classify.py” command, and then press “Enter”.



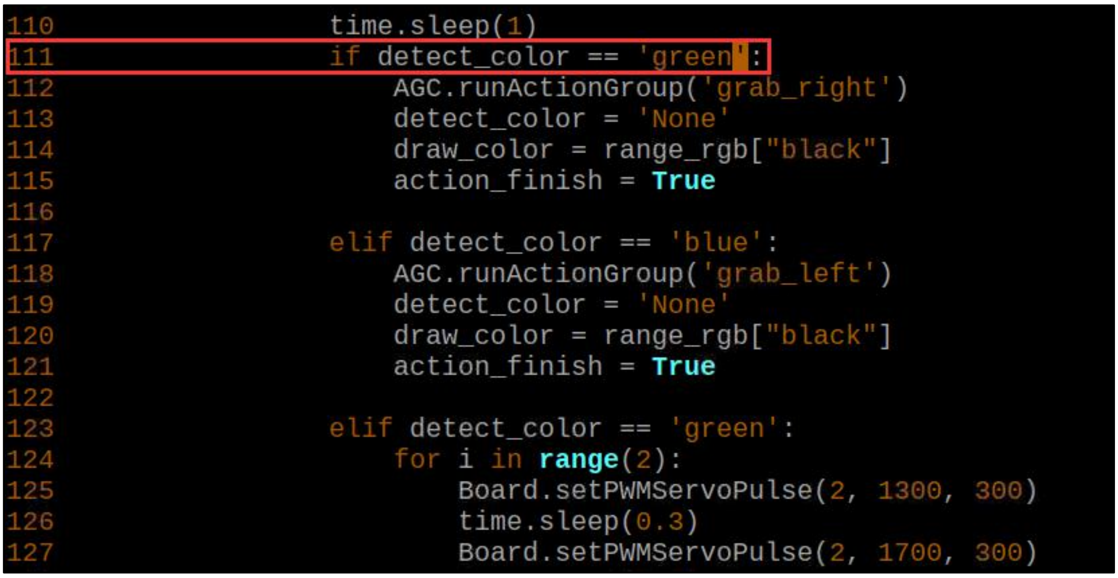
```
pi@raspberrypi:~ $ cd TonyPi/Extend/
pi@raspberrypi:~/TonyPi/Extend $ sudo vim Color_classify.py
```

3) Find the code shown in the figure below:



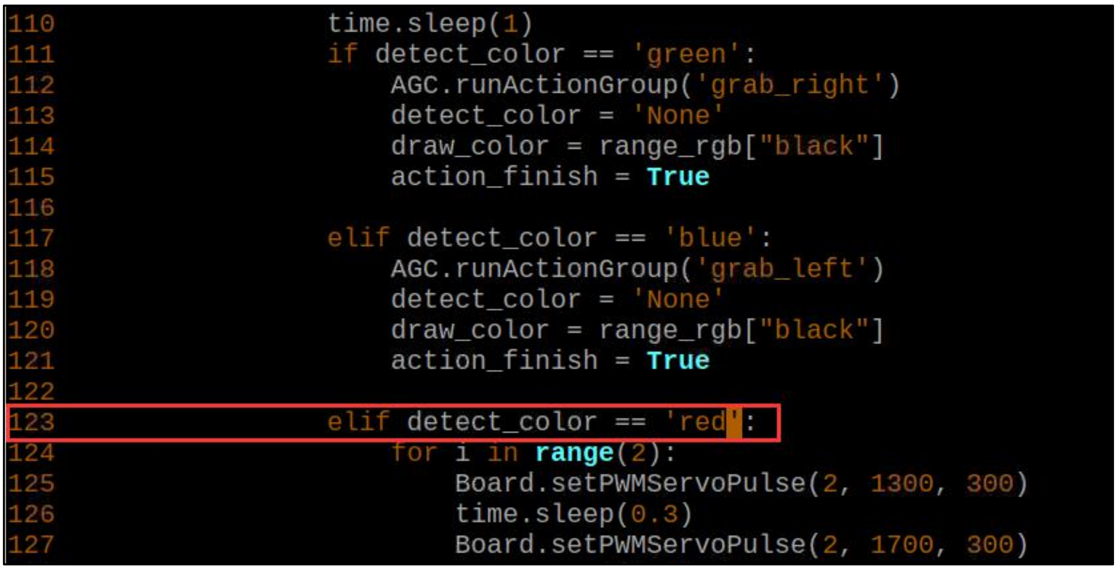
```
                    time.sleep(1)
                    if detect_color == 'red':
                        AGC.runActionGroup('grab_right')
                        detect_color = 'None'
                        draw_color = range_rgb["black"]
                        action_finish = True

                    elif detect_color == 'blue':
                        AGC.runActionGroup('grab_left')
                        detect_color = 'None'
                        draw_color = range_rgb["black"]
                        action_finish = True

                    elif detect_color == 'green':
                        for i in range(2):
                            Board.setPWMServoPulse(2, 1300, 300)
                            time.sleep(0.3)
                            Board.setPWMServoPulse(2, 1700, 300)
```

4) Press “i” to enter the editing mode. Then modify “red” in “if detect_color == ‘red’:” to “green”.



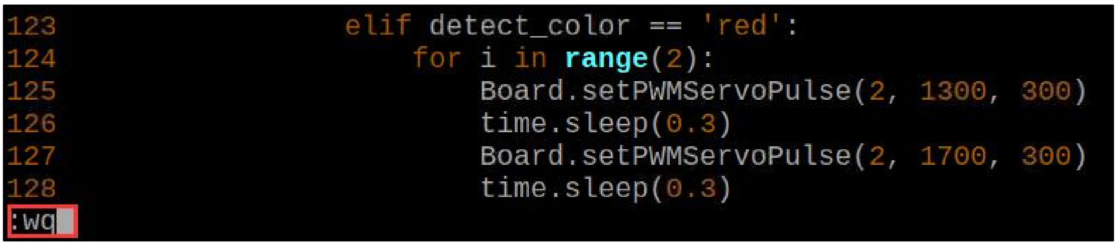
```
110                 time.sleep(1)
111                 if detect_color == 'green':
112                     AGC.runActionGroup('grab_right')
113                     detect_color = 'None'
114                     draw_color = range_rgb["black"]
115                     action_finish = True
116
117                 elif detect_color == 'blue':
118                     AGC.runActionGroup('grab_left')
119                     detect_color = 'None'
120                     draw_color = range_rgb["black"]
121                     action_finish = True
122
123                 elif detect_color == 'green':
124                     for i in range(2):
125                         Board.setPWMServoPulse(2, 1300, 300)
126                         time.sleep(0.3)
127                         Board.setPWMServoPulse(2, 1700, 300)
```

5) Locate line 123, and then modify “green” in “elif detect_color == ‘green’:” to “red”.



```
110                 time.sleep(1)
111                 if detect_color == 'green':
112                     AGC.runActionGroup('grab_right')
113                     detect_color = 'None'
114                     draw_color = range_rgb["black"]
115                     action_finish = True
116
117                 elif detect_color == 'blue':
118                     AGC.runActionGroup('grab_left')
119                     detect_color = 'None'
120                     draw_color = range_rgb["black"]
121                     action_finish = True
122
123                 elif detect_color == 'red':
124                     for i in range(2):
125                         Board.setPWMServoPulse(2, 1300, 300)
126                         time.sleep(0.3)
127                         Board.setPWMServoPulse(2, 1700, 300)
```

6) After modification, press “Esc” and then enter “:wq” (Please note that the colon is in front of wq). Then press “Enter” to save and exit the modified content.



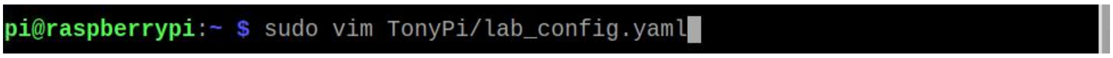
```
123                 elif detect_color == 'red':
124                     for i in range(2):
125                         Board.setPWMServoPulse(2, 1300, 300)
126                         time.sleep(0.3)
127                         Board.setPWMServoPulse(2, 1700, 300)
128                         time.sleep(0.3)
:wq
```

7) After the game is restarted, the robot will shake its head after recognizing the red block. When recognizing the blue or green block, the robot will grip and sort the block.

5.2 Add New Recognized Color

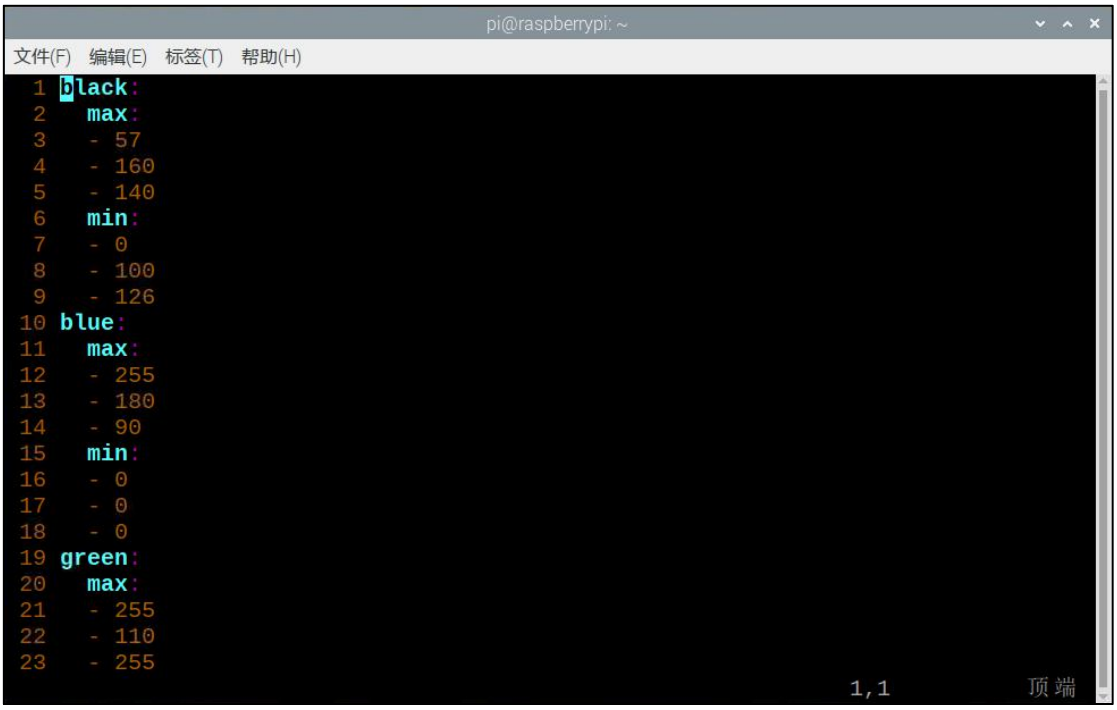
In addition to three built-in recognized colors, you can add other recognized colors. This section takes orange as example and the specific operation steps are as follow:

1) Open VNV and enter command “sudo vim TonyPi/lab_config.yaml” to open Lab color setting file.



```
pi@raspberrypi:~ $ sudo vim TonyPi/lab_config.yaml
```

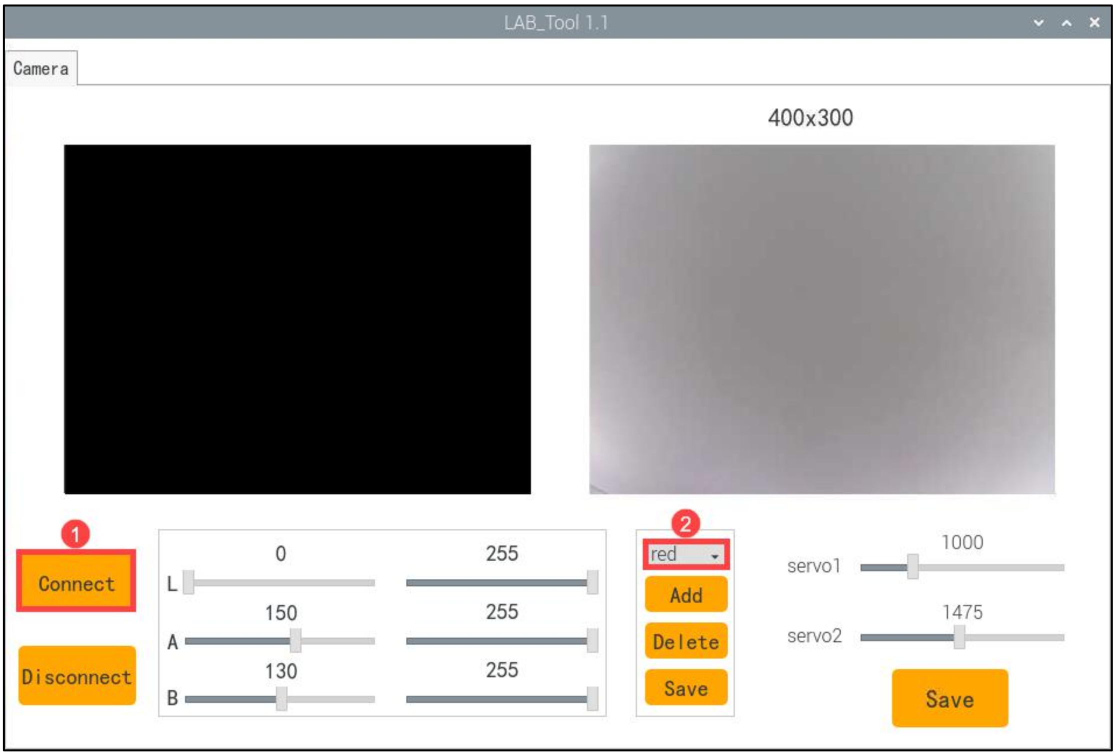
The initial value can be recorded by screenshot or file backup.



```
black:
  max:
  - 57
  - 160
  - 140
  min:
  - 0
  - 100
  - 126
blue:
  max:
  - 255
  - 180
  - 90
  min:
  - 0
  - 0
  - 0
green:
  max:
  - 255
  - 110
  - 255
                                        1,1           顶端
```

2) Click “LAB_Too” on system desktop, and then click “Execute” in the pop-up prompt window.

3) Click “Connect” button in the lower left corner. When the picture transmitted by the camera is displayed in the interface, it means that the connection is successful. Then select “red” in the right frame.



red

L 0 255

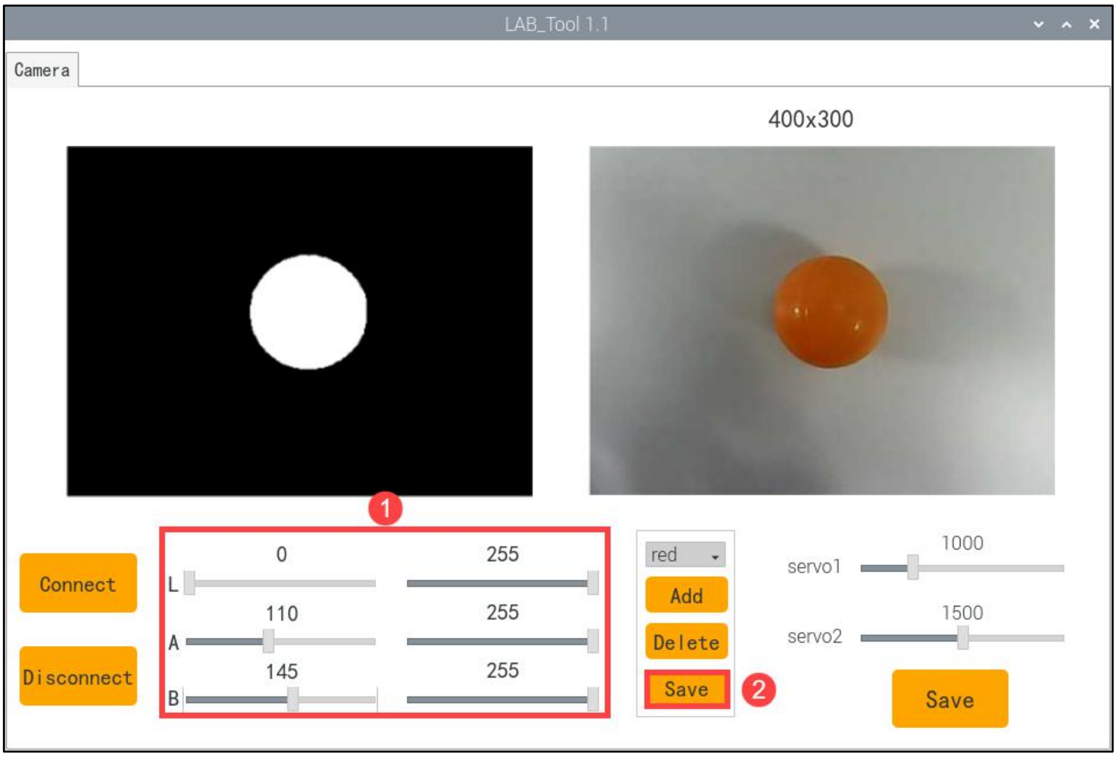
A 150 255

B 130 255

servo1 1000

servo2 1475

4) Point the camera at the color you want to recognize and drag the corresponding sliders of L, A and B until the orange part in left interface becomes white and other colors become black. Then click “save” button behind the “red” to save the modified data.



red

L 0 255

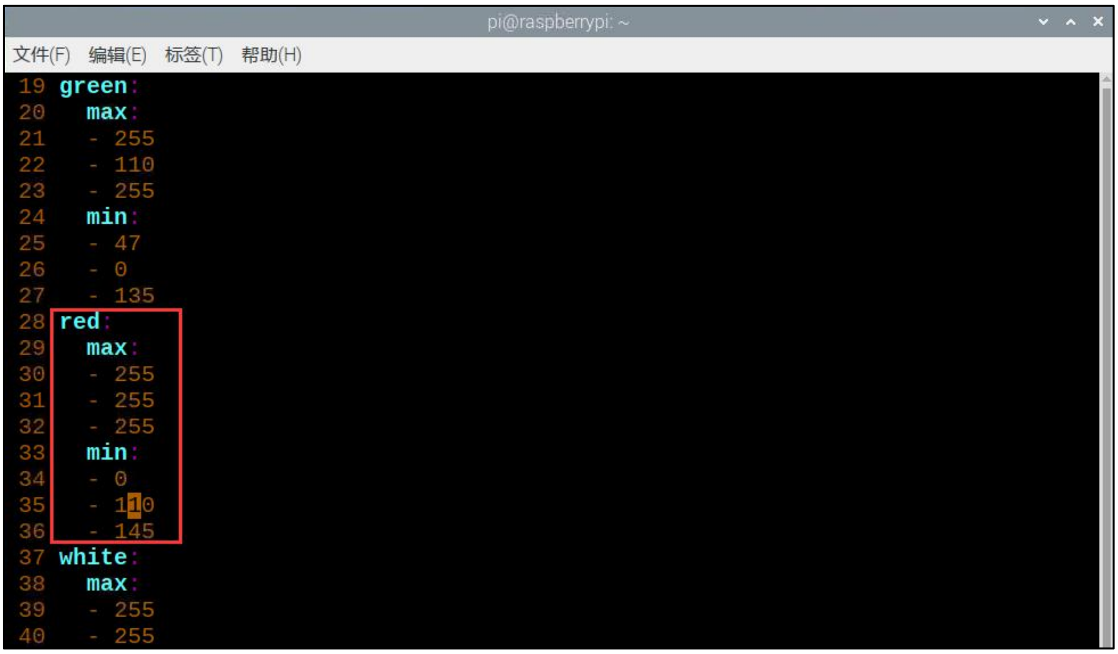
A 110 255

B 145 255

servo1 1000

servo2 1500

5) After modification, enter command “sudo vim TonyPi/lab_config.yaml” to view whether the color setting parameters are modified.



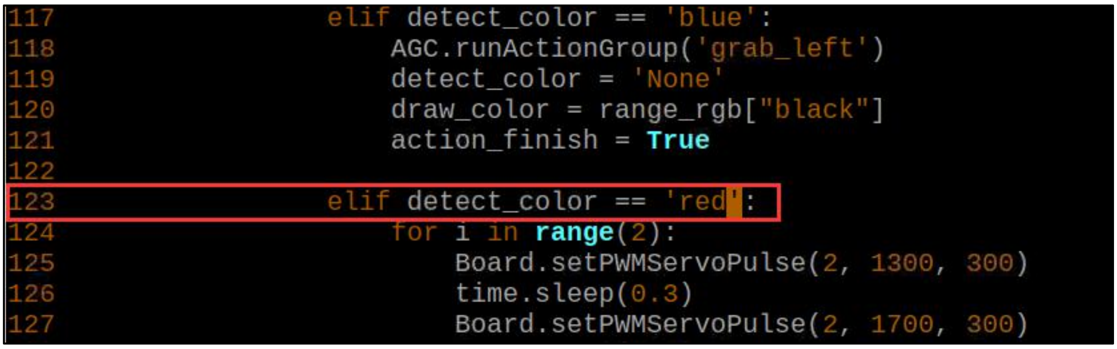
```
green:
  max:
  - 255
  - 110
  - 255
  min:
  - 47
  - 0
  - 135
red:
  max:
  - 255
  - 255
  - 255
  min:
  - 0
  - 110
  - 145
white:
  max:
  - 255
  - 255
```

To avoid any impact on the game, it’s recommended to use the LAB_Tool to modify the value back to the initial value after the modification is completed.

6) Check the data in red frame above. If the edited value was written in the

configuration program, press “Esc” and enter “:wq” to save it and exit.

7) According to the content in “5.1 Modify the Recognized Color”, set the default color that make TonyPi Pro shake its head after recognition as red according to the content in “5.1 Modify the recognized color”.



```
                    elif detect_color == 'blue':
                        AGC.runActionGroup('grab_left')
                        detect_color = 'None'
                        draw_color = range_rgb["black"]
                        action_finish = True

                    elif detect_color == 'red':
                        for i in range(2):
                            Board.setPWMServoPulse(2, 1300, 300)
                            time.sleep(0.3)
                            Board.setPWMServoPulse(2, 1700, 300)
```

8) After the game is started, place the orange object in front of the camera, you can see that TonyPi Pro performs “shake head” action.

9) If want to add other colors as the recognition color, please refer to the same operation steps.