

2. Color recognition

1. Experimental purpose

Drive the car to do color recognition

2. Experimental path source code

Enter the car system, end the car program, enter "ip (ip is the car's ip): 8888" in the browser, enter the password "yahboom"



Password:

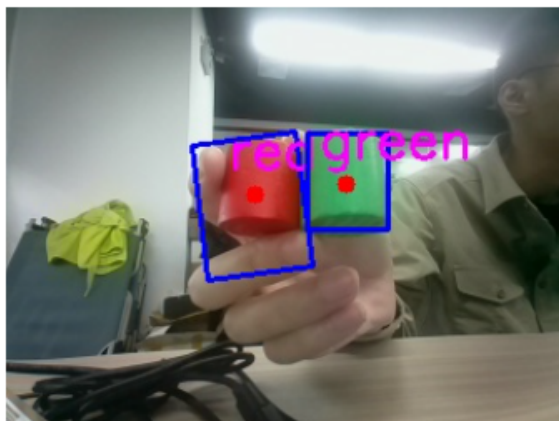
Then log in

Enter the path of **Rider-pi_class/5.AI Visual Recognition Course/2. Color recognition** and run **color_detection.ipynb**.

3. Experimental phenomenon

After running the source code, the car can recognize **red, yellow, blue, and green** at the same time. When a certain color is recognized, the RGB light will also light up the corresponding color. **The car cannot be in a stopped state (that is, not in a standing balance state), otherwise it cannot move.**

```
[11]: thread1 = threading.Thread(target=TEST)
      thread1.daemon = True
      thread1.start()
```



4. Analysis of main source code parameters

```
def TEST():
```

```

detected_colors = [] # Used to store the detected color names
unique_colorstemp=None
global color_hsv
display(image_widget)
while True:
    ret, frame = image.read() #usb camera usb camera
    frame, binary, hsvname=update_hsv.get_contours(frame,color_hsv)
    unique_colors = list(dict.fromkeys(hsvname))
    # Determine how to split the string based on the length of the list
    num_colors = len(unique_colors)
    # If there are 4 elements in the list
    if num_colors == 4:
        first_line = ', '.join(unique_colors[:2])
        second_line = ', '.join(unique_colors[2:])
    elif num_colors == 3:
        first_line = ', '.join(unique_colors[:2])
        second_line = unique_colors[2]
    elif num_colors == 2:
        first_line = ', '.join(unique_colors)
        second_line = ""
    elif num_colors == 1:
        first_line = unique_colors[0]
        second_line = ""
    else:
        first_line = ""
        second_line = ""
    if(unique_colors!=unique_colorstemp):
        light_leds(unique_colors)
        unique_colorstemp=unique_colors.copy()

    # Real-time image data transmission for display
    image_widget.value = bgr8_to_jpeg(frame)

    # Display on the LCD screen of the car
    b,g,r = cv2.split(frame)
    img = cv2.merge((r,g,b))
    imgok = Image.fromarray(img)
    mydisplay.ShowImage(imgok)

```

From the code, we can get: by turning on the camera of the car, we can do the corresponding color recognition; then the recognized color, RGB light corresponding color. And the result is displayed on the computer screen and the car screen.