Color tracking

1. Experimental purpose

Drive the car to do color tracking

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"

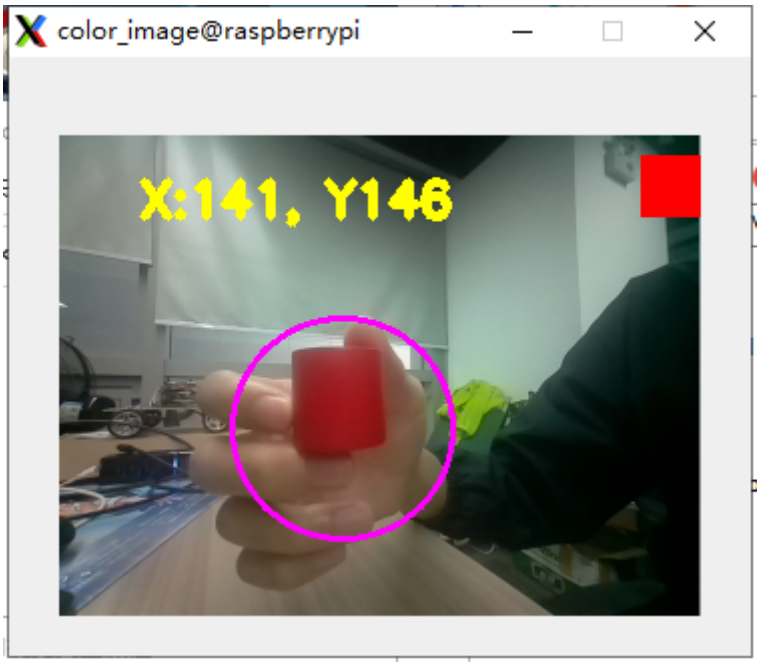
Then log in
Enter the path of **Rider-pi_class/6.AI Visual Interaction Course/1. Color tracking** and run **color.ipynb**.
Or directly enter in the terminal

```
cd /home/pi/Rider-pi_class/6.AI Visual Interaction Course/1. Color tracking
python3 color.py
```

3. Experimental phenomenon

After running the source code, the car can identify one of the four colors **red, yellow, blue and green**. You can press the button in the upper right corner of the car screen to switch the tracking color.
The car cannot be in a stopped state (that is, not in a standing balance state), otherwise it

cannot move.



4. Analysis of main source code parameters

```
#----------------------COMMON INIT----------------------
font = cv2.FONT_HERSHEY_SIMPLEX
cap=cv2.VideoCapture(0)
cap.set(3,320)
cap.set(4,240)
if(not cap.isOpened()):
    print("[camera.py:cam]:can't open this camera")

t_start = time.time()
fps = 0
color_x = 0
color_y = 0
color_radius = 0
while 1:
    ret, frame = cap.read()
    frame_ = cv2.GaussianBlur(frame,(5,5),0)
    hsv = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    mask = cv2.inRange(hsv,color_lower,color_upper)
    mask = cv2.erode(mask,None,iterations=2)
    mask = cv2.dilate(mask,None,iterations=2)
    mask = cv2.GaussianBlur(mask,(3,3),0)
    cnts =
cv2.findContours(mask.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)[-2]
    if g_mode == 1:
        if len(cnts) > 0:
            cnt = max (cnts, key = cv2.contourArea)
            (color_x,color_y),color_radius = cv2.minEnclosingCircle(cnt)
            if color_radius > 10:
                cv2.circle(frame,(int(color_x),int(color_y)),int(color_radius),
(255,0,255),2)
                value_x = color_x - 160
                value_y = color_y - 120
                rider_x=value_x
```

```
                if value_x > 55:
                    value_x = 55
                elif value_x < -55:
                    value_x = -55
                if value_y > 75:
                    value_y = 75
                elif value_y < -75:
                    value_y = -75
                if dog_type=='L' or dog_type=='M':
                    g_car.attitude(['y','p'],[-value_x/15, value_y/15])
                elif dog_type=='R':
                    print(color_y,75+int((190-color_y)/160*40))
                    g_car.rider_height(75+int((190-color_y)/160*40))
                    if rider_x==9999:
                        g_car.rider_turn(0)
                    else:
                        if rider_x > 35:
                            g_car.rider_turn(-20)
                        elif rider_x < -35:
                            g_car.rider_turn(20)
                        else:
                            g_car.rider_turn(0)
        else:
            color_x = 0
            color_y = 0
            rider_x=9999
        cv2.putText(frame, "X:%d, Y%d" % (int(color_x), int(color_y)), (40,40),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 3)
        t_start = time.time()
        fps = 0
    else:
        fps = fps + 1
        mfps = fps / (time.time() - t_start)
        cv2.putText(frame, "FPS " + str(int(mfps)), (40,40),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 3)



    b,g,r = cv2.split(frame)
    img = cv2.merge((r,g,b))
    if mode==1:
        cv2.rectangle(img, (290, 10), (320, 40), red, -1)
    elif mode==2:
        cv2.rectangle(img, (290, 10), (320, 40), green, -1)
    elif mode==3:
        cv2.rectangle(img, (290, 10), (320, 40), blue, -1)
    elif mode==4:
        cv2.rectangle(img, (290, 10), (320, 40), yellow, -1)
    imgok = Image.fromarray(img)
    display.ShowImage(imgok)

    r,g,b = cv2.split(img)
    frame1 = cv2.merge((b,g,r))
    cv2.imshow("color_image", frame1) #同时显示在屏幕上Displayed on screen at the
same time
```

```
    # if (cv2.waitKey(1)) == ord('q'):
    #     break
    if (cv2.waitKey(1)) == ord('f'): #切换颜色 Switch Color
        change_color()
    if button.press_b():
        break
    if button.press_d():
        change_color()

cap.release()
```

From the source code analysis, we can get: the car will track the color according to the camera. If the color goes up, the car will stand up; if the color goes down, the car will squat; if the color goes left or right, the car will also go left or right.