Color Recognition with Voice Broadcast

Before running the function, you need to close the App and large programs. For the closing method, refer to [4.Preparation] - [1. Manage APP control services].

Orin board users can directly open the terminal and enter the tutorial commands to run. Jetson-Nano board users need to enter the docker container first, then enter the tutorial commands in the docker to start the program.

1. Function Description

Broadcast the colors recognized by the program through the voice recognition broadcast module.

2. Startup and Operation

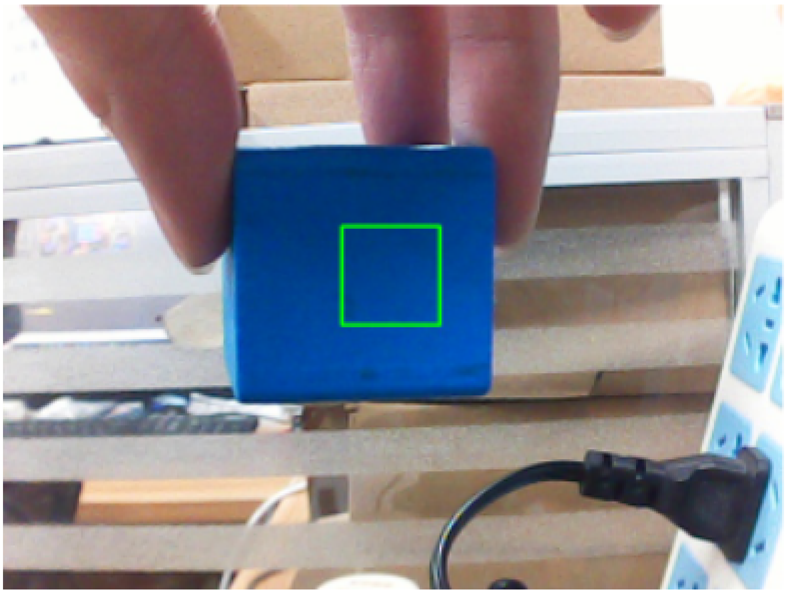
2.1. Startup

Open the terminal and enter the following command to start:

```
python3 ~/dofbot_voice/scripts/voice_color_identify.py
```

2.2. Operation Steps

After the program runs successfully, the program will call the camera to capture the current image. Hold a color block and place the green square in the image on the color block, as shown in the figure below:



Then the terminal will print "get color!", then say "Hello, yahboom" to the voice module. The voice module will reply "here" to indicate successful wake-up. Then say "What color is it?" to the voice module. The voice module will broadcast the current color of the block. If the current recognized block is blue, the voice module will broadcast "This is blue". Note: If colors cannot be recognized, you may need to modify the HSV values in the program or adjust the ambient lighting.

3. Core Code Analysis

Jetson-Nano users need to enter the docker container to view

`Source code path: ~/dofbot_voice/scripts/voice_color_identify.py`

```python
#Get the color of the color block in the current image
def get_color(img):
    H = []
    color_name={}
    img = cv2.resize(img, (640, 480), )
    # Convert color image to HSV
    HSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    # Draw rectangle frame
    cv2.rectangle(img, (280, 180), (360, 260), (0, 255, 0), 2)
    # Take out H, S, V values of each row and column sequentially and put them in the container
    for i in range(280, 360):
        for j in range(180, 260): H.append(HSV[j, i][0])
    # Calculate the maximum and minimum of H, S, V respectively
    H_min = min(H);H_max = max(H)
    # print(H_min,H_max)
    # Judge color
    if H_min >= 0 and H_max <= 10 or H_min >= 156 and H_max <= 180: color_name['name'] = 'red'
    elif H_min >= 21 and H_max <= 28: color_name['name'] = 'yellow'
    elif H_min >= 35 and H_max <= 78: color_name['name'] = 'green'
    elif H_min >= 100 and H_max <= 124: color_name['name'] = 'blue'
    return img, color_name
#Get voice result
result = mySpeech.speech_read()
#If the recognized voice result is 60, broadcast the audio file of the color block result according to the value of color_name['name'].
if result == 60:
    if color_name['name'] == 'yellow':
        mySpeech.void_write(64)
    elif color_name['name'] == 'red':
        mySpeech.void_write(61)
    elif  color_name['name'] == 'green':
        mySpeech.void_write(63)
    elif color_name['name'] == 'blue':
        mySpeech.void_write(62)
```