

# Voice Recognition Target Tracking

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

Voice control robotic arm to track color blocks and faces.

## 2. Startup and Operation

### 2.1. Operation

Open the terminal and enter the following command to start:

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

### 2.2. Operation Steps

After the program runs, the voice module will broadcast "here", then say "Hello, yahboom" to the voice module. The voice module will reply "here" to indicate successful wake-up. Then say "Face following/Yellow following/Red following/Green following/Blue following" to the voice module. The voice module will reply and broadcast "ok", then the robotic arm will start tracking the target. If you want to stop tracking, say "Stop following" to the voice module. The robotic arm will stop tracking the target and the voice module will reply and broadcast "ok". Note: If colors cannot be recognized or the target color cannot be tracked correctly, 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_target_follow.py`

```
# Import color tracking and face tracking libraries, the source code locations of
these libraries are at /home/jetson/dofbot_pro/dofbot_color_follow/scripts and
/home/jetson/dofbot_pro/dofbot_face_follow/scripts respectively
from color_follow import color_follow
from face_follow import Face_Follow

# Create color tracking object and human tracking object
follow = color_follow()
follow2 = Face_Follow()

# Initialize mode, 'General' means no tracking, 'color_follow' means color
tracking mode, 'follow2' means human tracking mode
model = 'General'

# Get voice recognition result
result = mySpeech.speech_read()
# According to voice recognition result, modify model and let voice broadcast
module broadcast corresponding audio file
if result == 73:
    choose_color = 'red'
    model = 'color_follow'
    mySpeech.void_write(71)
    time.sleep(0.1)
elif result == 72:
    choose_color = 'yellow'
    model = 'color_follow'
    mySpeech.void_write(71)
    time.sleep(0.1)
elif result == 74:
    choose_color = 'green'
    model = 'color_follow'
    mySpeech.void_write(71)
    time.sleep(0.1)
elif result == 75:
    choose_color = 'blue'
    model = 'color_follow'
    mySpeech.void_write(71)
    time.sleep(0.1)
elif result == 71:
    model = 'follow2'
    mySpeech.void_write(71)
    time.sleep(0.1)
elif result == 76:
    model = 'General'
    mySpeech.void_write(76)
    time.sleep(0.1)

# According to the value of mode, choose to execute color tracking or face
tracking, call corresponding functions and pass required values respectively
if model == 'color_follow':
    img = follow.follow_function(img, color_hsv[choose_color])
    # Add text
```

```
cv.putText(img, choose_color, (int(img.shape[0] / 2), 50),  
cv.FONT_HERSHEY_SIMPLEX, 2, color[random.randint(0, 254)], 2)  
if model == 'follow2':  
    img, pos = follow2.follow_function(img)
```