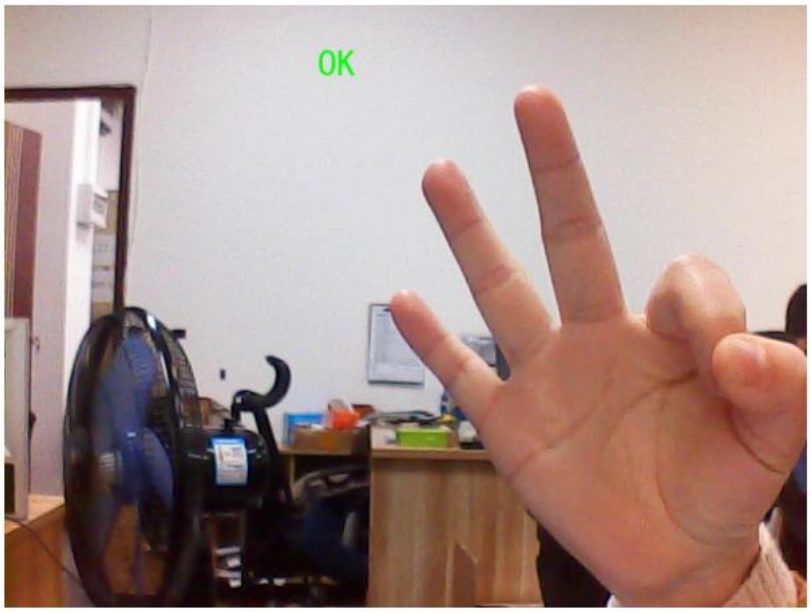
Path: /home/jetson/Dofbot/6.AI_Visual/1.gesture_action.ipynb

Tip:

The gesture recognition used in this example is a service based on Baidu API, which is used 50,000 times a day for free. Only for learning. Do not use it for commercial purposes. You need to purchase related services. Our company is not responsible.

1. Experimental phenomena

After running the program we provided, when the DFOBOT detects different gestures, it will complete the corresponding action and Recognition result will be printed. As shown below.




```
    print(" Program closed! ")
    pass

Recognition result:Rock
Recognition result:number5
Recognition result:number5
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
Recognition result:OK
```

Thumb_up	DOFBOT clip completes a movement similar to applause
OK	DOFBOT nodded
Prayer	DOFBOT prayer
Heart	DOFBOT kneels down to greet

Number_5	DOFBOT returns to the initial position
Number_8	DOFBOT scared
Rock	DOFBOT fell down
Thumb_down	DOFBOT makes a horses pose
Congratulation	Horses began to run
Number_7	Pull back from the cliff and stop running

2. About code

```
#bgr8 to jpeg format
import enum
import cv2

def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

```
# Import related modules
import threading
import time
from Arm_Lib import Arm_Device

# Get DOFBOT object
Arm = Arm_Device()
time.sleep(.1)
```

```
# Define the gesture recognition function part
```

```
import cv2
import time
import demjson
import pygame
from aip import AipBodyAnalysis
from aip import AipSpeech
from PIL import Image, ImageDraw, ImageFont
import numpy
import ipywidgets.widgets as widgets

hand={'One':'number1','Two':'number2','Three':'number3','Four':'number4',
        'Five':'number5', 'Six':'number6','Seven':'number7',
        'Eight':'number8','Nine':'number9','Fist':'fist','Ok':'OK',
        'Prayer':'prayer','Congratulation':'congratulation','Honour':'honour',
        'Heart_single':'heart','Thumb_up':'thumb_up','Thumb_down':'Diss',
        'ILY':'i love you','Palm_up':'palm_up','Heart_1':'Heart_1',
        'Heart_2':'Heart_1','Heart_3':'Heart_3','Rock':'Rock','Face':'face'}

# Using your key and ID
```

```
""" APPID AK SK """
APP_ID = '18550528'
API_KEY = 'K6PWqtiUTKYK1fYaz13O8E3i'
SECRET_KEY = 'IDBUII1j6srF1XVNDX32I2WpuwBWczzK'

client = AipBodyAnalysis(APP_ID, API_KEY, SECRET_KEY)

g_camera = cv2.VideoCapture(0)
g_camera.set(3, 640)
g_camera.set(4, 480)
g_camera.set(5, 30)    #Set frame
g_camera.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter.fourcc('M', 'J', 'P', 'G'))
g_camera.set(cv2.CAP_PROP_BRIGHTNESS, 40) #set brightness -64 - 64    0.0
g_camera.set(cv2.CAP_PROP_CONTRAST, 50) #set contrast -64 - 64    2.0
g_camera.set(cv2.CAP_PROP_EXPOSURE, 156) #set exposure 1.0 - 5000    156.0
ret, frame = g_camera.read()
```

```
# Define camera widget
image_widget = widgets.Image(format='jpeg', width=600, height=500)    # Define camera widget
display(image_widget)
image_widget.value = bgr8_to_jpeg(frame)
```

```
# Define display Chinese text
def cv2ImgAddText(img, text, left, top, textColor=(0, 255, 0), textSize=20):
    if (isinstance(img, numpy.ndarray)):
        img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

    draw = ImageDraw.Draw(img)

    fontStyle = ImageFont.truetype(
        "simhei.ttf", textSize, encoding="utf-8")

    draw.text((left, top), text, textColor, font=fontStyle)

    return cv2.cvtColor(numpy.asarray(img), cv2.COLOR_RGB2BGR)
```

```
look_at = [90, 164, 18, 0, 90, 90]
p_Prayer = [90, 90, 0, 180, 90, 180]
p_Thumb_up = [90, 90, 90, 90, 90, 180]
p_Heart_single = [90, 0, 180, 0, 90, 30]

p_Eight = [90, 180, 18, 0, 90, 90]

p_Congratulation = [90, 131, 52, 0, 90, 180]
p_Rock = [90, 0, 90, 180, 90, 0]
p_fist = [90, 90, 0, 0, 90, 0]
```

```
p_horse_1 = [90, 7, 153, 19, 0, 126]
p_horse_2 = [90, 5, 176, 0, 0, 180]
p_horse_3 = [90, 62, 158, 0, 0, 0]

global running
running = 0
```

```
# Define control DOFBOT function, control No.1-No.5 servo，  p=[S1,S2,S3,S4,S5]
def arm_move_6(p, s_time = 500):
    for i in range(6):
        id = i + 1
        Arm.Arm_serial_servo_write(id, p[i], s_time)
        time.sleep(.01)
    time.sleep(s_time/1000)

# Define hourse movment
def horse_running():
    Arm.Arm_serial_servo_write(6, 150, 300)
    time.sleep(.3)
    Arm.Arm_serial_servo_write(6, 180, 300)
    time.sleep(.3)
```

```
global g_state_arm
g_state_arm = 0
def ctrl_arm_move(index):
    global running
    if index == "Prayer":
        arm_move_6(p_Prayer, 1000)
        time.sleep(1.5)
        arm_move_6(look_at, 1000)
        time.sleep(1)
    elif index == "Thumb_up":
        s_time = 500
        Arm.Arm_serial_servo_write(6, 180, s_time)
        time.sleep(s_time/1000)
        Arm.Arm_serial_servo_write(6, 90, s_time)
        time.sleep(s_time/1000)
        Arm.Arm_serial_servo_write(6, 180, s_time)
        time.sleep(s_time/1000)
        Arm.Arm_serial_servo_write(6, 90, s_time)
        time.sleep(s_time/1000)
    elif index == "Ok":
        s_time = 300
        Arm.Arm_serial_servo_write(4, 10, s_time)
        time.sleep(s_time/1000)
```

```
        Arm.Arm_serial_servo_write(4, 0, s_time)
        time.sleep(s_time/1000)
        Arm.Arm_serial_servo_write(4, 10, s_time)
        time.sleep(s_time/1000)
        Arm.Arm_serial_servo_write(4, 0, s_time)
        time.sleep(s_time/1000)
    elif index == "Heart_single":
        arm_move_6([90, 90, 90, 90, 90, 90], 800)
        time.sleep(.1)
        arm_move_6(p_Heart_single, 1000)
        time.sleep(1)
    elif index == "Five":
        arm_move_6(look_at, 1000)
        time.sleep(.5)
    elif index == "Eight":
        s_time = 300
        arm_move_6(p_Eight, 0)
        time.sleep(1)
        Arm.Arm_serial_servo_write(2, 165, s_time)
        time.sleep(s_time/1000)
    elif index == "Rock": #rock
        Arm.Arm_serial_servo_write6_array(p_Rock, 1300)
        time.sleep(3)
        Arm.Arm_serial_servo_write6_array(look_at, 1000)
        time.sleep(1)
    elif index == "Thumb_down":
        Arm.Arm_serial_servo_write6_array(p_horse_1, 1300)
        time.sleep(1)
    elif index == "Congratulation":
        Arm.Arm_serial_servo_write6_array(p_horse_2, 1000)
        time.sleep(1)
        running = 1
        while running == 1:
            horse_running()
    elif index == "Seven":
        Arm.Arm_Buzzer_On(8)      #Buzzer whistle 5s
        Arm.Arm_serial_servo_write6_array(p_horse_3, 1000)
        time.sleep(2)
        Arm.Arm_serial_servo_write6_array(look_at, 1000)
        time.sleep(1)


    global g_state_arm
```

```
    g_state_arm = 0
arm_move_6(look_at, 1000)
time.sleep(1)
def start_move_arm(index):
    global g_state_arm
    global running
    if g_state_arm == 0:
        closeTid = threading.Thread(target = ctrl_arm_move, args = [index])
        closeTid.setDaemon(True)
        closeTid.start()
        g_state_arm = 1

    if running == 1 and index == "Seven":
            running = 0
# Main process
try:
    Arm.Arm_Buzzer_On(1)
    s_time = 300
    Arm.Arm_serial_servo_write(4, 10, s_time)
    time.sleep(s_time/1000)
    Arm.Arm_serial_servo_write(4, 0, s_time)
    time.sleep(s_time/1000)
    Arm.Arm_serial_servo_write(4, 10, s_time)
    time.sleep(s_time/1000)
    Arm.Arm_serial_servo_write(4, 0, s_time)
    time.sleep(s_time/1000)

    while True:
        """1.Take picture """
        ret, frame = g_camera.read()

        #image = get_file_content('./image.jpg')

        """ 2.call gesture function"""
        raw = str(client.gesture(image_widget.value))
        text = demjson.decode(raw)
        try:
            res = text['result'][0]['classname']
        except:
#               print('nothing' )
#               img = cv2ImgAddText(frame, "unrecognized", 250, 30, (0, 0 , 255), 30)
            img = frame
        else:
```

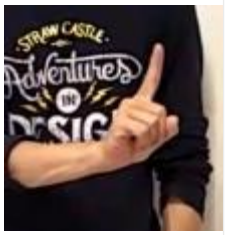
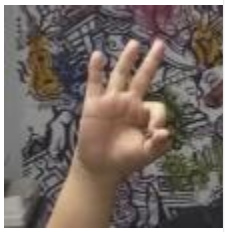
```
#            print('Recognition result:' + hand[res])
#            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
        if res == 'Prayer':
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == 'Thumb_up':
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == 'Ok':
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == 'Heart_single':
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == 'Five':
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == "Eight":
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)

        elif res == "Rock": # rock
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == "Congratulation":
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == "Seven":
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
        elif res == "Thumb_down":
            print('Recognition result:' + hand[res])
            img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
            start_move_arm(res)
```

```
                else:
                        img = frame

            image_widget.value = bgr8_to_jpeg(img)
except KeyboardInterrupt:
        print(" Program closed! ")
        pass
```

3. 23 types of gestures supported can be recognized.

Serial number	Gestures name	Sample
1	number_1	
2	number_5	
3	fist	
4	ok	

Serial number	Gestures name	Sample
5	pray	
6	congratulation	
7	honour	
8	heart_single	
9	thumb_up	
10	thumb_down	

Serial number	Gestures name	Sample
11	i_love_you	
12	palm_up	
13	heart_1	
14	heart_2	
15	heart_3	
16	number_2	

Serial number	Gestures name	Sample
17	number_3	
18	number_4	
19	number_6	
20	number_7	
21	number_8	
22	number_9	

Serial number	Gestures name	Sample
23	rock	

API function.

```
from aip import AipBodyAnalysis

""" YOUR APPID AK SK """
APP_ID = 'Your App ID'
API_KEY = 'Your Api Key'
SECRET_KEY = 'Your Secret Key'

client = AipBodyAnalysis(APP_ID, API_KEY, SECRET_KEY)

""" Read the pictures """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" Call gesture recognition """
Res = client.gesture(image);
```

Gesture recognition returns data parameter details.

Field	Whether the choice	Types	Explanation
result_num	Yes	int	number of results
result	Yes	object[]	Detected targets, gestures, faces
+classname	No	string	Target category. 23 types of gestures, other, face
+top	No	int	Coordinates on the target box
+width	No	int	Width of the target box

Field	Whether the choice	Types	Explanation
+left	No	int	Leftmost coordinates of the target box
+height	No	int	Height of the target box
+probability	No	float	The probability that the target belongs to the class
log_id	No	int64	Unique log id for problem location
image	No	string	Image data, urlencode after base64 encoding, requires that the size after base64 encoding and urlencode does not exceed 4M. The base64 encoding of the picture does not include the picture header, such as (data: image / jpg; base64,), supports picture formats: jpg, bmp, png, the shortest side is at least 50px, the longest side is at most 4096px

Gesture recognition returns an example.


```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [{
      "probability": 0.9844077229499817,
      "top": 20,
      "height": 156,
      "classname": "Face",
      "width": 116,
      "left": 173
  },
  {
      "probability": 0.4679304957389832,
      "top": 157,
      "height": 106,
      "classname": "Heart_2",
      "width": 177,
      "left": 183
  }]
}
```