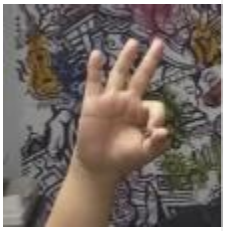
2.1.9 Gesture recognition

The computing power of the Raspberry Pi is limited, so we directly use the 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.

23 types of gestures supported can be recognized.

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

Serial number	Gestures name	Sample
6	congratulation	
7	honour	
8	heart_single	
9	thumb_up	
10	thumb_down	
11	i_love_you	

Serial number	Gestures name	Sample
12	palm_up	
13	heart_1	
14	heart_2	
15	heart_3	
16	number_2	
17	number_3	

Serial number	Gestures name	Sample
18	number_4	
19	number_6	
20	number_7	
21	number_8	
22	number_9	
23	rock	

In addition to gestures recognizing. If a face is detected in the image, it will return to the position of the face frame at the same time.
First, we need to connect to the Raspberry Pi remotely and enable the camera.
Then, we need to install Baidu's API library by sudo pip3 install baidu-aip command

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
+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

Field	Whether the choice	Types	Explanation
			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
   }]
}
```


Code path:

/home/pi/Yahboom_Project/3.AI_Visual_course/09.Gesture recognition.ipynb

```
#bgr8 to jpeg format
```

```
import enum
import cv2

def bgr8_to_jpeg(value, quality=75):
        return bytes(cv2.imencode('.jpg', value)[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':'one','Five':'five','Fist':'fist','Ok':'OK',
          'Prayer':'pray','Congratulation':'congratulation','Honour':'honour',
          'Heart_single':'heart_single','Thumb_up':'thumb_up','Thumb_down':'thumb_
          down','ILY':'i_love_you','Palm_up':'palm_up','Heart_1':'heart_1',
          'Heart_2':'heart_2','Heart_3':'heart_3','Two':'two',
          'Three':'three','Four':'four','Six':'six','Seven':'seven',
          'Eight':'eight','Nine':'nine','Rock':'rock','Face':'face'}

# Change the key below to your own key
""" Analysis of body APPID AK SK """
APP_ID = '18550528'
API_KEY = 'K6PWqtiUTKYK1fYaz13O8E3i'
SECRET_KEY = 'IDBUII1j6srF1XVNDX32I2WpuwBWczzK'

#camera = PiCamera()
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, 120)    #Set the frame rate
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 the camera display component
```

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

```
# Define conversion display Chinese function
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))
    # Create an object that can draw on a given image
    draw = ImageDraw.Draw(img)
    # Font format
    fontStyle = ImageFont.truetype(
        "simhei.ttf", textSize, encoding="utf-8")
    # Draw text
    draw.text((left, top), text, textColor, font=fontStyle)
    # Convert back to OpenCV format
    return cv2.cvtColor(numpy.asarray(img), cv2.COLOR_RGB2BGR)
```

```
# Main process
while True:
    """1.take a picture """
    retval, frame = g_camera.read()

    ret, frame = g_camera.read()

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

    """ 2.Call gesture recognition """
    raw = str(client.gesture(image_widget.value))
    text = demjson.decode(raw)
    try:
        res = text['result'][0]['classname']
    except:
        print('Result: nothing' )
        # 1 dst 2 verbal content 3 coordinate 4 5 frnot size 6 color 7 thickness 8 line
type
#       cv2.putText(frame, 'unidentification', (250,30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,200), 2, cv2.LINE_AA) #Only display english
        img = cv2ImgAddText(frame, "unidentification", 250, 30, (0, 0 , 255), 30)
    else:
        print('Result:' + hand[res])
        #cv2.putText(frame,  hand[res],  (250,30),  cv2.FONT_HERSHEY_SIMPLEX,  1,
(0,0,0), 2, cv2.LINE_AA)
        img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
```

```
image_widget.value = bgr8_to_jpeg(img)
```

After running the above program, the Raspberry Pi interface will print out the data as shown below.

```
Result:face
Result:face
Result:nothing
Result:face
Result:face
Result:face
Result:nothing
Result:face
Result:1
Result:nothing
Result:nothing
Result:1
Result:1
Result:1
Result:1
Result:nothing
Result:1
Result:1
Result:nothing
Result:2
Result:2
Result:2
Result:2
```