**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. Experiment preparation

Each block needs to be placed in the corresponding color area on map.

2. Experimental phenomena

After running the program we provided, when the DFOBOT detects different gestures, it will stack blocks and recognition result will be printed. As shown below.
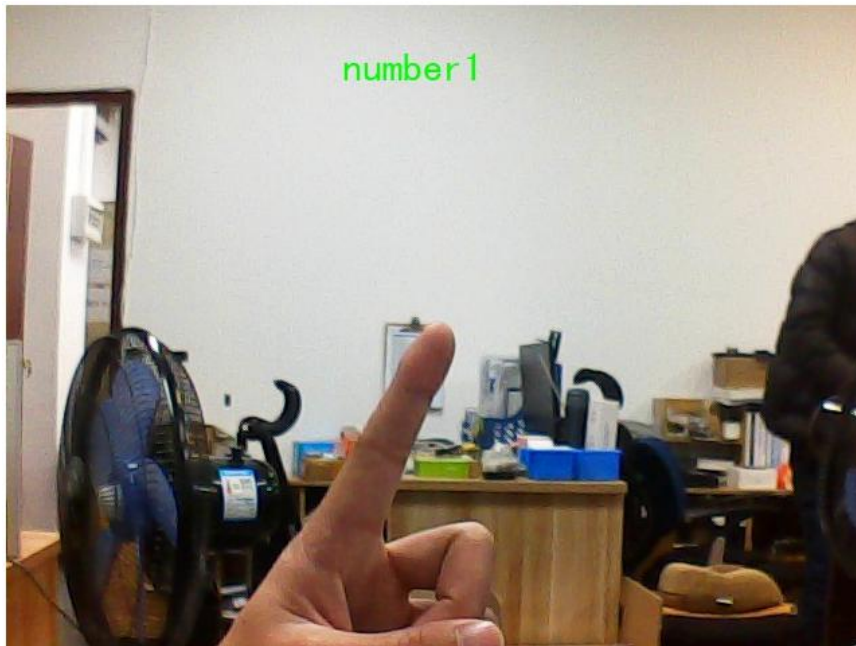
```
except KeyboardInterrupt:
    print(" Program closed! ")
    pass

Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
Recognition result:number1
```

When the DOFBOT recognizes gesture number at the first time, it will grab the block and place it on the first layer. When DOFBOT recognizes the gesture number at the second time, it will grab the block and place it on the second floor. When DOFBOT recognizes the gesture number at the third time, it will grab the block and place it on the third layer. When DOFBOT recognizes the gesture number is recognized at the fourth time, it will grab the block and place it on the third layer.

The number of blocks grabbed by the DOFBOT each time is determined by the recognized number. The number 1 represents yellow, the number 2 represents red, the number 3 represents green, and the number 4 represents yellow.
If this number is recognized again, it will not be grabbed again.
When a fist is recognized, DOFBOT will push down all the blocks, the record is cleared.

Number 1	DOFBOT grabs the block in yellow area
Number 2	DOFBOT grabs the block in red area
Number 3	DOFBOT grabs the block in green area
Number 4	DOFBOT grabs the block in blue area
Fist	DOFBOT push down all the blocks

3. About code
Path: /home/jetson/Dofbot/6.AI_Visual/2.gesture_stack.ipynb

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

def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```
```
#Import library
import threading
from Arm_Lib import Arm_Device
```

```
Arm = Arm_Device()
```

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

```
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 own 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)
```

```python
# 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)
```

```python
# Define control DOFBOT function, control No.1-No.6 servo，p=[S1,S2,S3,S4,S5,S6]
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 control DOFBOT function, control No.1-No.5 servo，p=[S1,S2,S3,S4,S5]
def arm_move(p, s_time = 500):
    for i in range(5):
        id = i + 1
        if id == 5:
            time.sleep(.1)
            Arm.Arm_serial_servo_write(id, p[i], int(s_time*1.2))
        elif id == 1 :
            Arm.Arm_serial_servo_write(id, p[i], int(3*s_time/4))
        else:
            Arm.Arm_serial_servo_write(id, p[i], int(s_time))
        time.sleep(.01)
    time.sleep(s_time/1000)


# enable=1：clip，=0：release
def arm_clamp_block(enable):
    if enable == 0:
        Arm.Arm_serial_servo_write(6, 60, 400)
    else:
        Arm.Arm_serial_servo_write(6, 135, 400)
    time.sleep(.5)
```

```
# Define variable parameters at different locations
look_at = [90, 164, 18, 0, 90, 90]
p_top = [90, 80, 50, 50, 270]


p_Yellow = [65, 22, 64, 56, 270]
p_Red = [118, 19, 66, 56, 270]


p_Green = [136, 66, 20, 29, 270]
p_Blue = [44, 66, 20, 28, 270]


p_layer_4 = [90, 76, 40, 17, 270]
p_layer_3 = [90, 65, 44, 17, 270]
p_layer_2 = [90, 65, 25, 36, 270]
p_layer_1 = [90, 48, 35, 30, 270]



p_push_over_1 = [90, 90, 5, 0, 90, 150]
p_push_over_2 = [90, 90, 0, 50, 90, 150]



#Define the state of the grab block
yellow_grabbed = 0
red_grabbed = 0
green_grabbed = 0
blue_grabbed = 0



#Define the number of gesture recognition
Count_One = 0
Count_Two = 0
Count_Three = 0
Count_Four = 0
Count_Fist = 0
```

```
arm_move_6(look_at, 1000)
time.sleep(1)
```

```
#Define the function corresponding to each number
def number_action(index):
    if index == 1:
        # Grab yellow block
        arm_move(p_top, 1000)
        arm_move(p_Yellow, 1000)
        arm_clamp_block(1)
#           time.sleep(.5)
```

```
            arm_move(p_top, 1000)
        elif index == 2:
            # Grab red block
            arm_move(p_top, 1000)
            arm_move(p_Red, 1000)
            arm_clamp_block(1)
            arm_move(p_top, 1000)
        elif index == 3:
            # Grab green block
            arm_move(p_top, 1000)
            arm_move(p_Green, 1000)
            arm_clamp_block(1)
            arm_move(p_top, 1000)
        elif index == 4:
            # Grab blue block
            arm_move(p_top, 1000)
            arm_move(p_Blue, 1000)
            arm_clamp_block(1)
            arm_move(p_top, 1000)


def put_down_block(layer):

    if layer == 1:
        arm_move(p_layer_1, 1000)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)
    elif layer == 2:
        arm_move(p_layer_2, 1000)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)
    elif layer == 3:
        arm_move(p_layer_3, 1000)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)
    elif layer == 4:
        arm_move(p_layer_4, 1000)
        time.sleep(.1)
        arm_clamp_block(0)
        arm_move_6(look_at, 1000)

# Knock down blocks
def push_over_block():
```

```
        arm_move_6(p_push_over_1, 1000)
        time.sleep(.2)
        arm_move_6(p_push_over_2, 1000)
        time.sleep(.1)
        arm_move_6(look_at, 1000)
        time.sleep(1)
        global g_layer
        g_layer = 0
```

```
global g_state_arm
g_state_arm = 0

global g_layer
g_layer = 0

def ctrl_arm_move(index):
    global g_layer
    g_layer = g_layer + 1
    if g_layer >= 5:
        g_layer = 1
    arm_clamp_block(0)
    if index == 1:
        number_action(index)
        put_down_block(g_layer)
    elif index == 2:
        number_action(index)
        put_down_block(g_layer)
    elif index == 3:
        number_action(index)
        put_down_block(g_layer)
    elif index == 4:
        number_action(index)
        put_down_block(g_layer)
    elif index == 5:
        time.sleep(1)
        push_over_block()


    global g_state_arm
    g_state_arm = 0
```

```
def start_move_arm(index):
    global g_state_arm
    if g_state_arm == 0:
        closeTid = threading.Thread(target = ctrl_arm_move, args = [index])
```

```
            closeTid.setDaemon(True)
            closeTid.start()

            g_state_arm = 1
# 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 a pciture"""
        ret, frame = g_camera.read()

        """ 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 == 'One':
                Count_One = Count_One + 1
                Count_Two = 0
                Count_Three = 0
                Count_Four = 0
                Count_Fist = 0
                if Count_One >= 3:
                    print('Recognition result:' + hand[res])
                    img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
                    if yellow_grabbed == 0:
```

```python
                    start_move_arm(1)
#                       global yellow_grabbed
                    yellow_grabbed = 1
        elif res == 'Two':
            Count_Two = Count_Two + 1
            Count_Three = 0
            Count_Four = 0
            Count_Fist = 0
            if Count_Two >= 3:
                print('Recognition result:' + hand[res])
                img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
                if red_grabbed == 0:
                    start_move_arm(2)
#                       global red_grabbed
                    red_grabbed = 1
        elif res == 'Three':
            Count_Three = Count_Three + 1
            Count_Two = 0
            Count_Four = 0
            Count_Fist = 0
            if Count_Three >= 3:
                print('Recognition result:' + hand[res])
                img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
                if green_grabbed == 0:
                    start_move_arm(3)
#                       global green_grabbed
                    green_grabbed = 1
        elif res == 'Four':
            Count_Four = Count_Four + 1
            Count_Two = 0
            Count_Three = 0
            Count_Fist = 0
            if Count_Four >= 3:
                print('Recognition result:' + hand[res])
                img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
                if blue_grabbed == 0:
                    start_move_arm(4)
#                       global blue_grabbed
                    blue_grabbed = 1
        elif res == 'Fist':
            Count_Fist = Count_Fist + 1
            Count_One = 0
            Count_Two = 0
```

```
                Count_Three = 0
                Count_Four = 0

                if Count_Fist >= 3:
                        print('Recognition result:' + hand[res])
                        img = cv2ImgAddText(frame, hand[res], 250, 30, (0, 255 , 0), 30)
                        start_move_arm(5)
                        yellow_grabbed = 0
                        red_grabbed = 0
                        green_grabbed = 0
                        blue_grabbed = 0
                        Count_Fist = 0
            else:
                    img = frame

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

4. 23 types of gestures supported can be recognized.

Serial number	Gestures name	Sample
1	number_1	
2	number_5	

Serial number	Gestures name	Sample
3	fist	
4	ok	
5	pray	
6	congratulation	
7	honour	
8	heart_single	

Serial number	Gestures name	Sample
9	thumb_up	
10	thumb_down	
11	i_love_you	
12	palm_up	
13	heart_1	
14	heart_2	

Serial number	Gestures name	Sample
15	heart_3	
16	number_2	
17	number_3	
18	number_4	
19	number_6	
20	number_7	

Serial number	Gestures name	Sample
21	number_8	
22	number_9	
23	rock	