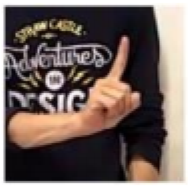
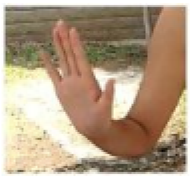
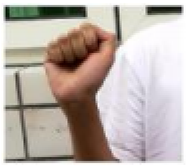
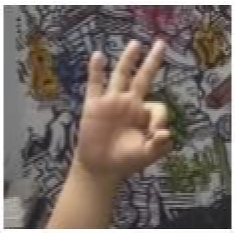
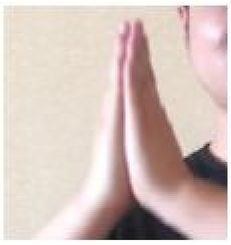
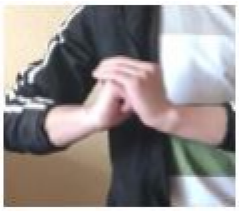
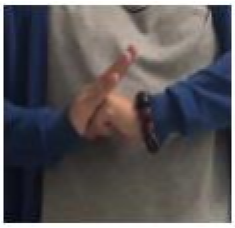
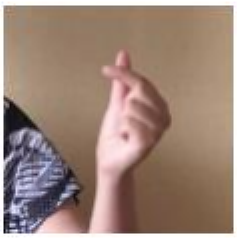
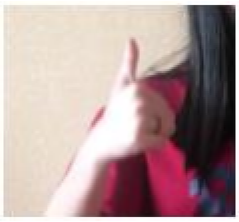
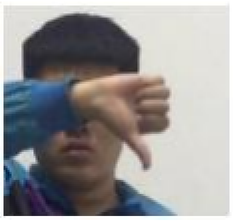
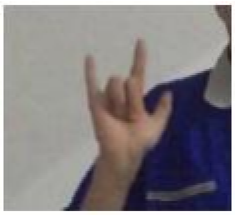
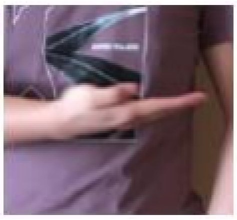
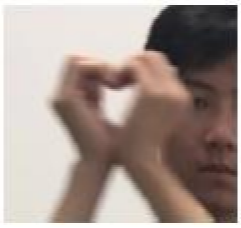
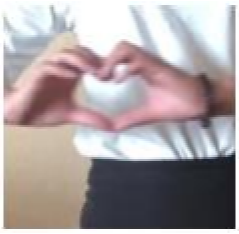
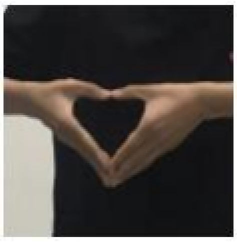
Gesture recognition stacked building blocks

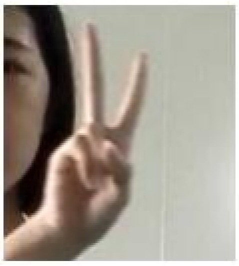
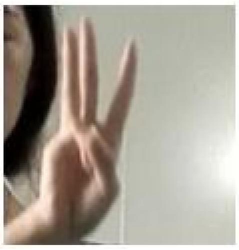
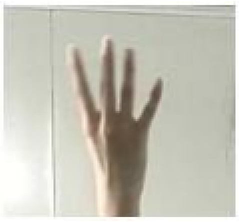
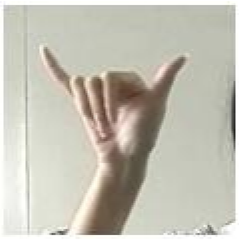
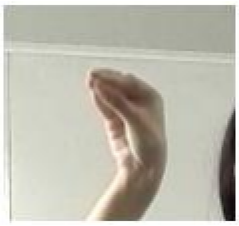
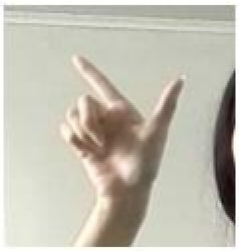
1. Gesture recognition instructions

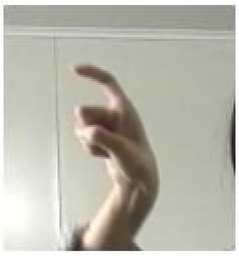
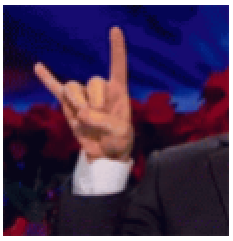
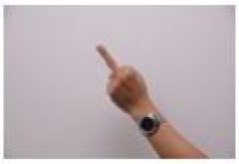
1. The gesture recognition used in this routine is based on the service of Baidu Intelligent Cloud Platform. This service has 50,000 free use opportunities every day. It is only for learning and not for commercial purposes. If you need long-term use, please purchase related services. Our company No responsibility is assumed.
2. For KEY application instructions and API documentation, please see the course content in the previous section.
3. Gestures supported by gesture recognition and example images

序号	手势名称	classname	示例图
1	数字 1（原食指）	One	
2	数字 5（原掌心向前）	Five	
3	拳头	Fist	

4	OK	OK	
5	祈祷	Prayer	
6	作揖	Congratulation	
7	作别	Honour	
8	单手比心	Heart_single	
9	点赞	Thumb_up	

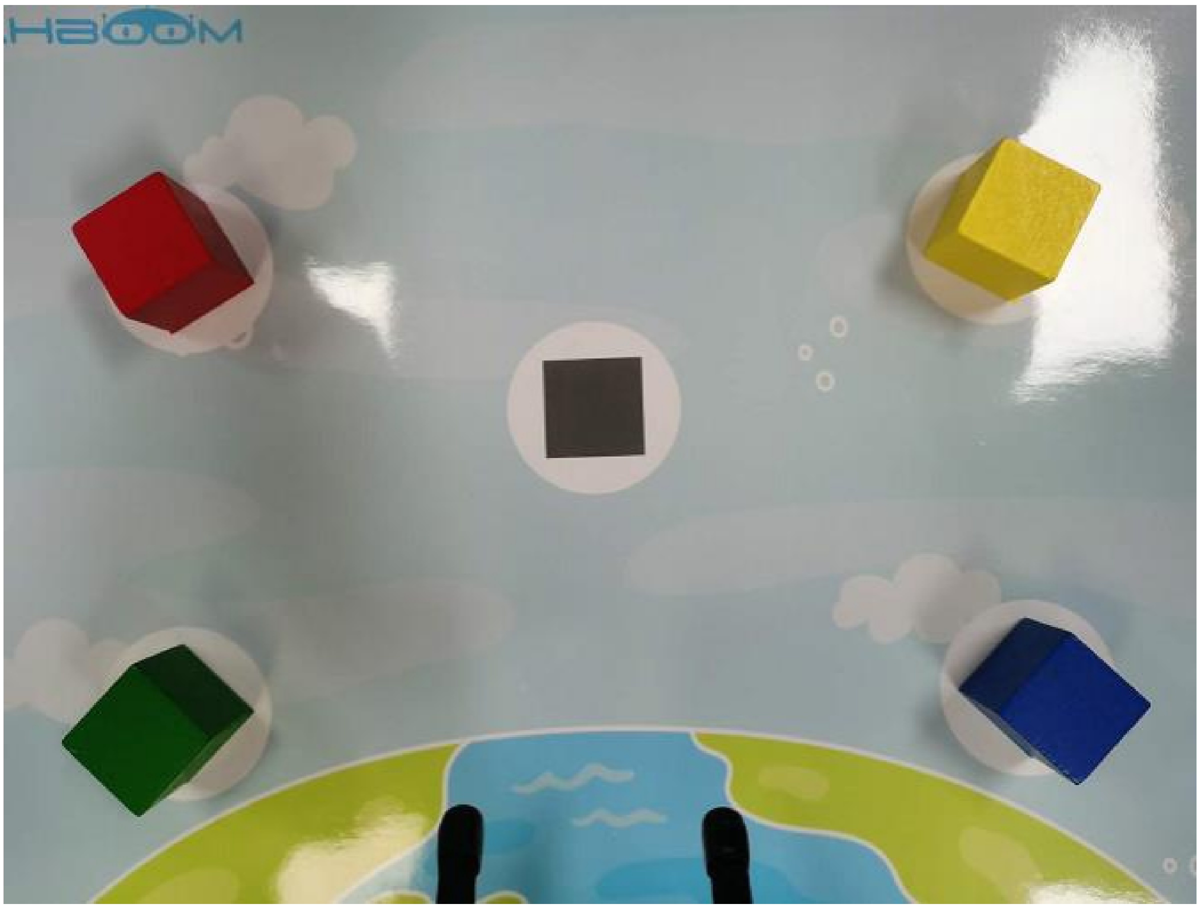
10	Diss	Thumb_down	
11	我爱你	ILY	
12	掌心向上	Palm_up	
13	双手比心 1	Heart_1	
14	双手比心 2	Heart_2	
15	双手比心 3	Heart_3	

16	数字 2	two	
17	数字 3	three	
18	数字 4	four	
19	数字 6	six	
20	数字 7	seven	
21	数字 8	eight	

22	数字 9	nine	
23	Rock	Rock	
24	竖中指	Insult	

2. Experimental placement

The experimental placement position is to place yellow building blocks in the yellow area, red building blocks in the red area, green building blocks in the green area, and blue building blocks in the blue area.



3. Code content

Code path:/root/Dofbot/6.AI_Visual/2.gesture_stack.ipynb

```
#bgr8 转 jpeg 格式
import enum
import cv2
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])
```

```
#导入相关的模块

import   threading
import   time
from   Arm_Lib import Arm_Device


#   创建机械臂对象
Arm   = Arm_Device()
time.sleep(.1)
```

```
#   定义手势识别函数部分
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

#   具体手势请看官方提供 https://ai.baidu.com/ai-doc/BODY/4k3cpywrv

hand={'One':'数字1','Two':'数字2','Three':'数字3','Four':'数字4',

      'Five':'数字5', 'Six':'数字6','Seven':'数字7',

      'Eight':'数字8','Nine':'数字9','Fist':'拳头','Ok':'OK',

      'Prayer':'祈祷','Congratulation':'作揖','Honour':'作别',

      'Heart_single':'比心心','Thumb_up':'点赞','Thumb_down':'Diss',

      'ILY':'我爱你','Palm_up':'掌心向上','Heart_1':'双手比心1',

      'Heart_2':'双手比心2','Heart_3':'双手比心3','Rock':'Rock',

      'Insult':'竖中指','Face':'脸'}
```

```
#    下面的key要换成自己的

"""    人体分析    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)  #设置帧率
g_camera.set(cv2.CAP_PROP_FOURCC,   cv2.VideoWriter.fourcc('M', 'J', 'P', 'G'))
g_camera.set(cv2.CAP_PROP_BRIGHTNESS,   40) #设置亮度 -64 - 64  0.0
g_camera.set(cv2.CAP_PROP_CONTRAST,   50) #设置对比度 -64 - 64  2.0
g_camera.set(cv2.CAP_PROP_EXPOSURE,   156) #设置曝光值 1.0 - 5000  156.0

ret,   frame = g_camera.read()
```

```
#    定义摄像头显示组件

image_widget    = widgets.Image(format='jpeg', width=600, height=500)  #设置摄像头显
示组件

display(image_widget)

image_widget.value    = bgr8_to_jpeg(frame)
```

```
#    定义转换显示中文函数

def   cv2ImgAddText(img, text, left, top, textColor=(0, 255, 0), textSize=20):

    if (isinstance(img, numpy.ndarray)):  # 判断是否OpenCV图片类型

        img =   Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

    # 创建一个可以在给定图像上绘图的对象

    draw = ImageDraw.Draw(img)

    # 字体的格式

    fontStyle = ImageFont.truetype(

        "simhei.ttf", textSize,   encoding="utf-8")

    # 绘制文本

    draw.text((left, top), text, textColor,   font=fontStyle)
    # 转换回OpenCV格式

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

```
#   定义不同位置的变量参数
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]

#定义抓取方块的状态
yellow_grabbed   = 0
red_grabbed   = 0
green_grabbed   = 0
blue_grabbed   = 0

#定义手势识别次数

Count_One   = 0
Count_Two   = 0
Count_Three   = 0
Count_Four   = 0
Count_Fist   = 0
```

```
#   定义移动机械臂函数,同时控制1-6号舵机运动, 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)



#   定义移动机械臂函数,同时控制1-5号舵机运动, 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)



#   定义夹积木块函数，enable=1：夹住，=0：松开

def   arm_clamp_block(enable):
    if enable == 0:
        Arm.Arm_serial_servo_write(6, 60,   400)

    else:
        Arm.Arm_serial_servo_write(6, 130,   400)

    time.sleep(.5)
```

```
#数字功能定义

def   number_action(index):

    if index == 1:

        # 抓取黄色的积木块

        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:

        # 抓取红色的积木块

        arm_move(p_top, 1000)
        arm_move(p_Red, 1000)

        arm_clamp_block(1)

        arm_move(p_top, 1000)

    elif index == 3:

        # 抓取绿色的积木块

        arm_move(p_top, 1000)

        arm_move(p_Green, 1000)

        arm_clamp_block(1)

        arm_move(p_top, 1000)

    elif index == 4:

        # 抓取蓝色的积木块

        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)

#   推倒积木块

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
```

```
#让机械臂运动到摄像头向前看的位置

arm_move_6(look_at,   1000)

time.sleep(1)
```

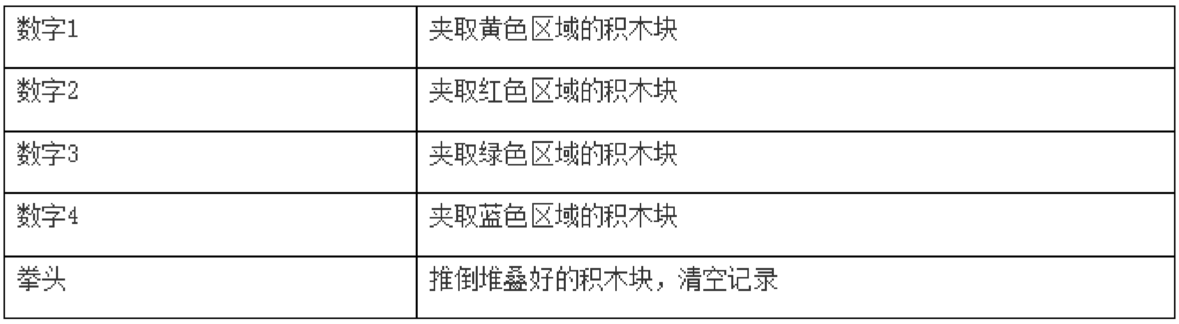
```
global   g_state_arm

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):
    # 开启机械臂控制线程
    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
```

```
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.拍照   """

        ret, frame = g_camera.read()



        """ 2.调用手势识别   """

        raw =   str(client.gesture(image_widget.value))

        text = demjson.decode(raw)

        try:
            res =   text['result'][0]['classname']
        except:
    #           print('识别结果：什么也没识别到哦~' )
    #           img = cv2ImgAddText(frame, "未识别", 250, 30, (0, 0 , 255), 30)
            img = frame
        else:
    #           print('识别结果：' + 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('识别结果：' + hand[res])
                    img = cv2ImgAddText(frame,   hand[res], 250, 30, (0, 255 , 0), 30)
                    if yellow_grabbed == 0:
                        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('识别结果：' + 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('识别结果：' + 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('识别结果：' + 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': #拳头

                Count_Fist = Count_Fist + 1

                Count_One = 0

                Count_Two = 0

                Count_Three = 0

                Count_Four = 0

                if Count_Fist >= 3:

                    print('识别结果: ' + 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
```

If the set gesture action is recognized, the robot arm will perform the corresponding action. When the robot arm recognizes a number for the first time, it picks up the building blocks and places them on the first layer. When it recognizes the gesture number for the second time, it picks up the building blocks and places them on the second layer. When it recognizes the gesture number for the third time, it picks up the building blocks and places them on the second layer. Put the building blocks on the third layer. When the gesture number is recognized for the fourth time, pick up the building blocks and put them on the fourth layer. The color of the building blocks picked up by the robot arm each time is determined by the number recognized. The number 1 means yellow, and the number 1 means 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. Unless a fist is recognized, all building blocks will be knocked down, the record will be cleared, and the game will start again.

The corresponding gestures and actions in this routine are as follows:



数字1	夹取黄色区域的积木块
数字2	夹取红色区域的积木块
数字3	夹取绿色区域的积木块
数字4	夹取蓝色区域的积木块
拳头	推倒堆叠好的积木块，清空记录