**智能家居应用-人脸识别门禁系统**

**<使用树莓派及阿里云视觉智能平台AI能力架设>**

**实验手册**

**物联网产品应用搭建训练营**

**目录**

[一、实验目的 3](#_Toc68794280)

[二、实验内容 4](#_Toc68794281)

[三、实验仪器、设备 4](#_Toc68794282)

[四、详细实验过程 5](#_Toc68794283)

[4.1人脸识别门禁软硬件环境构建 5](#_Toc68794284)

[4.1.1官网下载树莓派系统 5](#_Toc68794285)

[4.1.2系统写入SD卡 5](#_Toc68794286)

[4.1.3开启显示信号输出 5](#_Toc68794287)

[4.1.4接通电源开机 7](#_Toc68794288)

[4.1.5打开ssh连接及摄像头 7](#_Toc68794289)

[4.1.6修改静态ip地址 8](#_Toc68794290)

[4.2阿里云视觉智能开放平台应用 9](#_Toc68794291)

[4.2.1使用阿里云账号登录，成为开发者 9](#_Toc68794292)

[4.2.2创建AccessKey 9](#_Toc68794293)

[4.2.3能力开通 9](#_Toc68794294)

[4.2.4启动开发 10](#_Toc68794295)

[4.2.5测试 16](#_Toc68794296)

[4.3通过IOT Studio进行业务开发 17](#_Toc68794297)

# 一、实验目的

“物联网产品应用搭建训练营”是物联网专业的一门专业核心课程，要求已经学习了物联网专业的基础课程，包括物联网架构体系、物联网操作系统介绍、物联网无线网络特性、物联网云边协同、物联网与人工智能应用等内容。

本次课程是智能家居应用中人脸识别门禁的构建，希望通过本次课程的学习，帮助学员掌握：

1. 了解阿里云智能视觉开放平台；
2. 掌握阿里云视觉智能平台AI能力的封装和应用集成；
3. 可以通过阿里云视觉智能平台提供的基础AI能力，设计、开发和构建AI应用；

# 二、实验内容

以实验箱中的树莓派、及树莓派配套摄像头为基础硬件，通过简单开发（语言为python）调用阿里云视觉智能开放平台中的“人脸识别”相关能力的API，来实现人脸识别门禁的功能，同时将识别结果与IOT物联网平台联动，通过“钉钉”下发识别结果。

主要包括：

1. 树莓派系统的安装；
2. 搭建完成硬件环境；
3. 使用阿里云账号登录 [阿里云视觉智能开放平台]，成为开发者；
4. 人脸识别能力开通；
5. 调用人脸识别”相关能力的API完成代码工作；
6. 添加样本数据；
7. 测试验证；
8. 通过IOT Studio进行业务服务的开发，实现当’Score’大于0.5时，触发钉钉消息推送；
9. 完成实验；

# 三、实验仪器、设备

1、硬件：PC 微型计算机、1G以上内存，40G以上硬盘

2、软件：Windows 10，Android Studio, JDK , Android SDK

3、使用平台：

阿里云物联网平台；

阿里云视觉智能开放平台（目前免费）；

4、使用设备：

智慧物联开发套件（包含STM32L4762RG；树莓派3B+；）

# 四、详细实验过程

## 4.1人脸识别门禁软硬件环境构建

### 4.1.1官网下载树莓派系统

系统下载地址:<https://www.raspberrypi.org/software/operating-systems/>

系统SD卡安装工具Pi Imager:<https://www.raspberrypi.org/software/>

### 4.1.2系统写入SD卡

将SD接入电脑,使用Pi Imager将系统写入SD卡

Choose OS ,Use custom,选择下载的系统文件

choose SD CARD,选择sd卡

write,等待程序完成



### 4.1.3开启显示信号输出

树莓派默认关闭了显示信号的输出,安装完成需修改默认配置文件

安装完成后重新插拔SD卡,进入SD系统盘符,打开/boot/config.txt

修改注释项

hdmi\_safe=1

hdmi\_force\_hotplug=1

hdmi\_group=2

hdmi\_mode=82



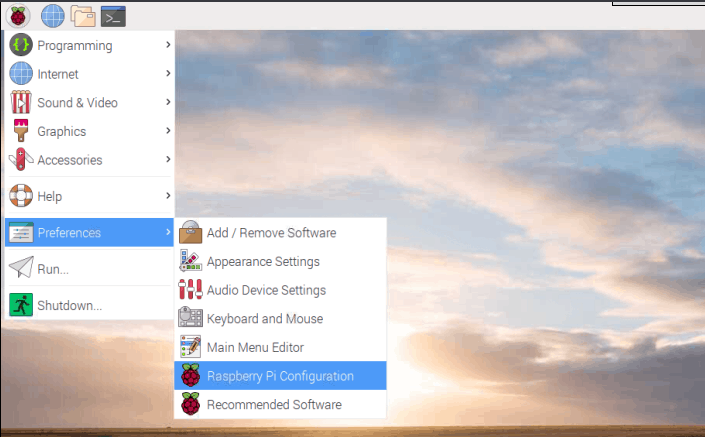
### 4.1.4接通电源开机

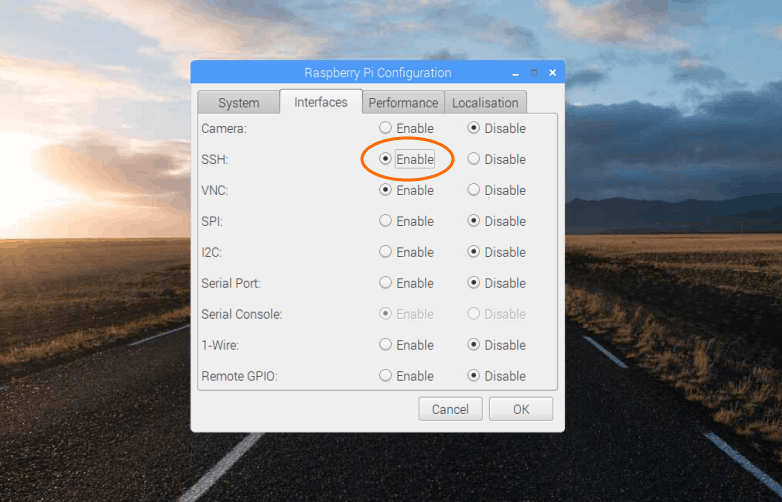
将SD插入树莓派,(记录开机显示分配的IP),并完成一些基本设置

### 4.1.5打开ssh连接及摄像头

SSH 在树莓派上是默认关闭的，因此安装好全新的 Raspbian 后打开树莓派时，你需要启用它。

首先通过菜单进入树莓派的配置界面,进入接口界面



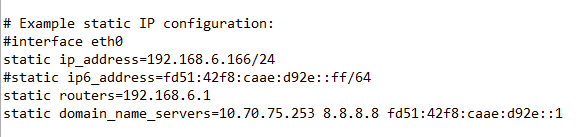


开启SSH连接及Camera并重启你的树莓派。

之后可通过PC端ssh进入系统,默认用户名pi,密码为开机时设置的密码

### 4.1.6修改静态ip地址

如果想自己分配静态ip地址,可以使用ssh修改,文件地址/etc/dhcpcd.conf



ip\_address是要配置的静态IP，后面要加上/24

routers是路由地址

domain\_name\_servers是DNS

修改之后重启设备生效: sudo reboot now

默认使用系统自带的python2.7

## 4.2阿里云视觉智能开放平台应用

### 4.2.1使用阿里云账号登录，成为开发者

使用阿里云账号登录阿里云视觉智能开放平台。如还未持有阿里云账户，请先注册阿里云账号。

地址：https://vision.aliyun.com

### 4.2.2创建AccessKey

在使用阿里云SDK和调用视觉智能开放平台的API之前，确保您已经生成访问密钥（AccessKey）。您可以在个人中心AcessKey管理中创建并管理属于您自己的AccessKey。



### 4.2.3能力开通

1）在能力页面上方单击立即开通，进入能力购买页面。

2）选择地域和要购买的能力，勾选服务协议后，单击底部立即开通。



### 4.2.4启动开发

**1）首先需要创建人脸数据库**

**添加SDK依赖**

pip install oss2

pip install aliyun-python-sdk-viapiutils

pip install aliyun-python-sdk-core

pip install viapi-utilspip install aliyun-python-sdk-facebody==1.2.18

(python示例代码)

#!/usr/bin/env python

#coding=utf-8

fromaliyunsdkcore.clientimportAcsClient

fromaliyunsdkcore.acs\_exception.exceptionsimportClientException

fromaliyunsdkcore.acs\_exception.exceptionsimportServerException

fromaliyunsdkfacebody.request.v20191230.CreateFaceDbRequest importCreateFaceDbRequest

client = AcsClient('<accessKeyId>','<accessSecret>','cn-shanghai')

request = CreateFaceDbRequest()

request.set\_accept\_format('json')

request.set\_Name("myFaceDB")

response = client.do\_action\_with\_exception(request)

# python2: print(response)

print(str(response, encoding='utf-8'))

**2）然后创建人脸数据样本**

(python示例代码)

#!/usr/bin/env python

#coding=utf-8

fromaliyunsdkcore.clientimportAcsClient

fromaliyunsdkcore.acs\_exception.exceptionsimportClientException

fromaliyunsdkcore.acs\_exception.exceptionsimportServerException

fromaliyunsdkfacebody.request.v20191230.AddFaceEntityRequest importAddFaceEntityRequest

client = AcsClient('<accessKeyId>','<accessSecret>','cn-shanghai')

request = AddFaceEntityRequest()

request.set\_accept\_format('json')

request.set\_DbName("myFaceDB")

request.set\_EntityId("test1")

response = client.do\_action\_with\_exception(request)

# python2: print(response)

print(str(response, encoding='utf-8'))

**3）给样本添加初始照片信息,用于摄像头拍摄照片之后对比识别**

(当前使用图片本地上传模式)

#!/usr/bin/env python

#coding=utf-8

from viapi.fileutils import FileUtils

fromaliyunsdkcore.clientimportAcsClient

fromaliyunsdkcore.acs\_exception.exceptionsimportClientException

fromaliyunsdkcore.acs\_exception.exceptionsimportServerException

fromaliyunsdkfacebody.request.v20191230.AddFaceRequest importAddFaceRequest

client = AcsClient('<accessKeyId>','<accessSecret>','cn-shanghai')

request = AddFaceRequest()

request.set\_accept\_format('json')

file\_utils = FileUtils("your own accessKey","your own accessSecret")

oss\_url = file\_utils.get\_oss\_url("/home/xxx.jpg","jpg",True)

print(oss\_url)

request.set\_DbName("myFaceDB")

request.set\_ImageUrl(oss\_url)

request.set\_EntityId("test1")

request.set\_ExtraData("姓名")

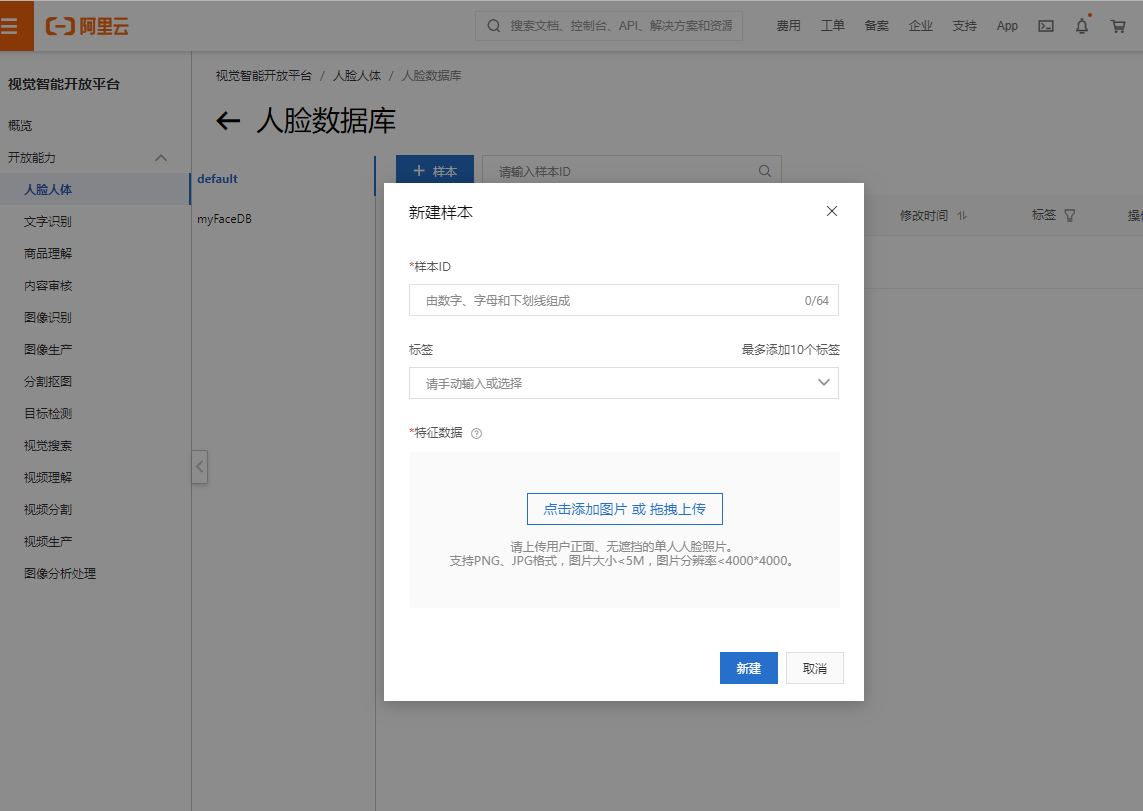
response = client.do\_action\_with\_exception(request)

# python2: print(response)

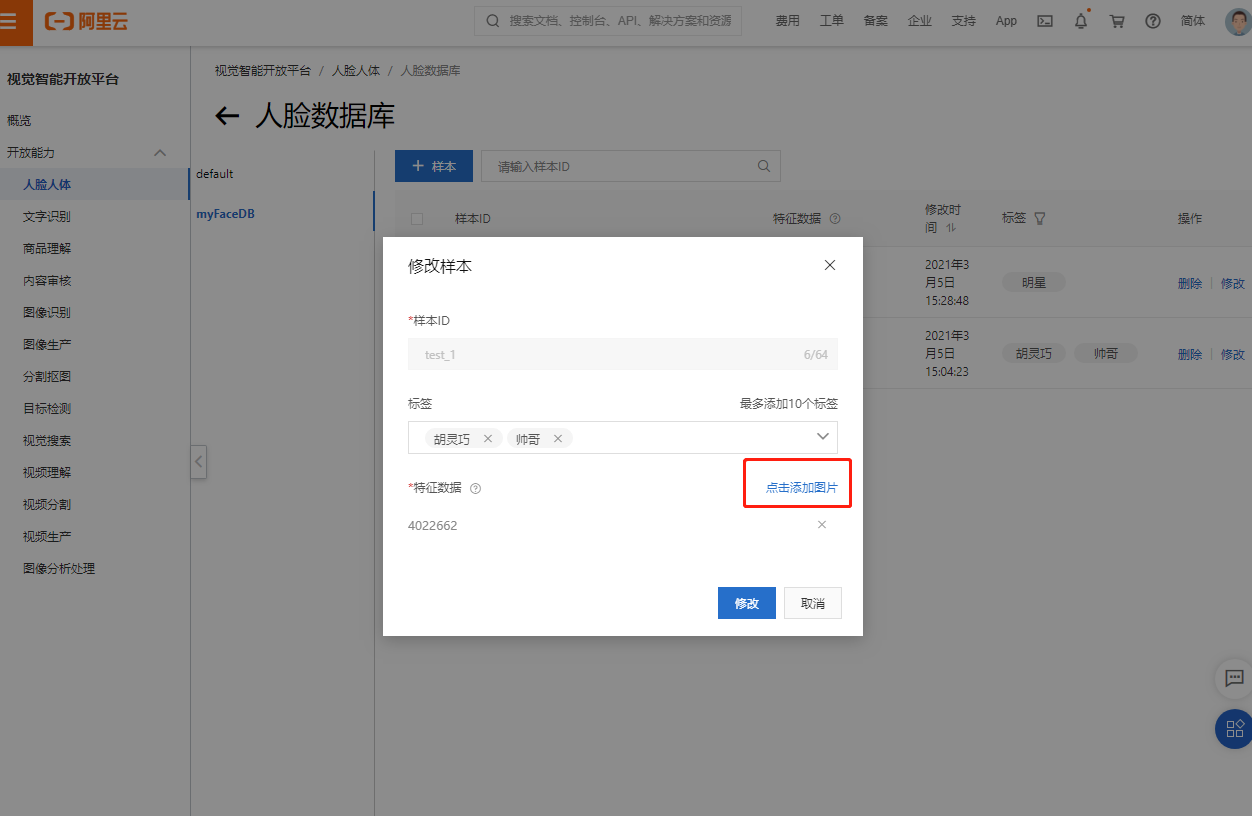
print(str(response, encoding='utf-8'))

也可以在视觉智能开放平台后台直接添加样本及样本图片

**添加样本**



**添加样本图片**



关机之后插入树莓派摄像头,启动系统

**4）参考示例代码完成编码**

sudo pip install opencv-python

// 安装opencv

sudo apt-get install -y libopencv-dev python-opencv libopencv-contrib-dev --fix-missing

// 安装一些其它工具

sudo apt-get install -y python-picamera python-pil python-tk

运行摄像头 Python示例代码(替换accesskey信息)

#!/usr/bin/env python

# coding=utf-8

import multiprocessing as mp

import os

import cv2

import time

from aliyunsdkcore.client import AcsClient

from aliyunsdkfacebody.request.v20191230.SearchFaceRequest import SearchFaceRequest

from viapi.fileutils import FileUtils

resX = 480

resY = 320

# The face cascade file to be used

face\_cascade = cv2.CascadeClassifier('/usr/share/opencv/haarcascades/haarcascade\_frontalface\_default.xml')

eye\_cascade = cv2.CascadeClassifier('/usr/share/opencv/haarcascades/haarcascade\_eye.xml')

search\_flag = True

face\_color = (192, 220, 240)

eye\_color = (200, 160, 164)

strokeWeight = 2

t\_start = time.time()

fps = 0

def get\_faces(img):

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, 1.3, 5)

return faces, img, gray

def search(filename):

client = AcsClient('<accessKeyId>','<accessSecret>', 'cn-shanghai')

file\_utils = FileUtils('<accessKeyId>','<accessSecret>')

PIC\_ROOT = os.path.dirname(os.path.abspath(\_\_file\_\_))

pix = filename.split('.')[1]

oss\_url = file\_utils.get\_oss\_url(PIC\_ROOT + "/" + filename, "jpg", True)

print(oss\_url)

face\_request = SearchFaceRequest()

face\_request.set\_accept\_format('json')

face\_request.set\_DbName("myFaceDB")

face\_request.set\_ImageUrl(oss\_url)

face\_request.set\_Limit(1)

response = client.do\_action\_with\_exception(face\_request)

print str(response)

# return str(response)

def draw\_frame(faces, img, gray):

global xdeg

global ydeg

global fps

global time\_t

global search\_flag

for x, y, w, h in faces:

cv2.rectangle(img, (x, y), (x + w, y + h), face\_color, strokeWeight)

cv2.putText(img, "Face No." + str(len(faces)), (x, y), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 255), 2)

face\_gray = gray[y:y + h, x:x + w]

eyes = eye\_cascade.detectMultiScale(face\_gray, 1.03, 5, 0, (40, 40))

for ex, ey, ew, eh in eyes:

cv2.rectangle(img, (x + ex, y + ey), (x + ex + ew, y + ey + eh), eye\_color, strokeWeight)

# Calculate and show the FPS

fps = fps + 1

sfps = fps / (time.time() - t\_start)

cv2.putText(img, "FPS : " + str(int(sfps)), (10, 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 0, 255), 2)

cv2.imshow("Frame", img)

# Main ######################################################################

if \_\_name\_\_ == '\_\_main\_\_':

camera = cv2.VideoCapture(0)

camera.set(cv2.CAP\_PROP\_FRAME\_WIDTH, resX)

camera.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, resY)

pool = mp.Pool(processes=4)

read, img = camera.read()

pr1 = pool.apply\_async(get\_faces, [img])

read, img = camera.read()

pr2 = pool.apply\_async(get\_faces, [img])

read, img = camera.read()

pr3 = pool.apply\_async(get\_faces, [img])

read, img = camera.read()

pr4 = pool.apply\_async(get\_faces, [img])

fcount = 1

while True:

read, img = camera.read()

if fcount == 1:

pr1 = pool.apply\_async(get\_faces, [img])

faces, img, gray = pr2.get()

draw\_frame(faces, img, gray)

eliffcount == 2:

pr2 = pool.apply\_async(get\_faces, [img])

faces, img, gray = pr3.get()

draw\_frame(faces, img, gray)

eliffcount == 3:

pr3 = pool.apply\_async(get\_faces, [img])

faces, img, gray = pr4.get()

draw\_frame(faces, img, gray)

eliffcount == 4:

pr4 = pool.apply\_async(get\_faces, [img])

faces, img, gray = pr1.get()

draw\_frame(faces, img, gray)

fcount = 0

fcount += 1

if cv2.waitKey(1000 // 12) & 0xff == ord("q"):

name = time.time()

name\_time = time.strftime("%Y%m%d%H%M%S", time.localtime(name))

cv2.imwrite('/home/pi/' + name\_time + '.jpg', img)

search(name\_time + '.jpg')

break

cv2.destroyAllWindows()

### 4.2.5测试

进入树莓派命令行,运行py脚本文件,可直接拉起摄像头,识别人脸.按下键盘 [ q ],进行拍照,并上传阿里云进行人脸识别,返回识别结果.

本地照片存储位置/home/pi/

## 4.3通过IOT Studio进行业务开发

**1）注册并登录阿里物联网开发平台**，地址为

https://iot.console.aliyun.com/lk/related-services

**2）然后按以下地址开始业务开发**

物联网平台--->相关服务---->物联网应用开发，

进入IOT Studio ,然后在“项目管理”中，新建项目，这里我们新建一个项目，名称是“人脸识别演示”；

新建一个业务服务，这里我们起一个名字叫“实现联动”；

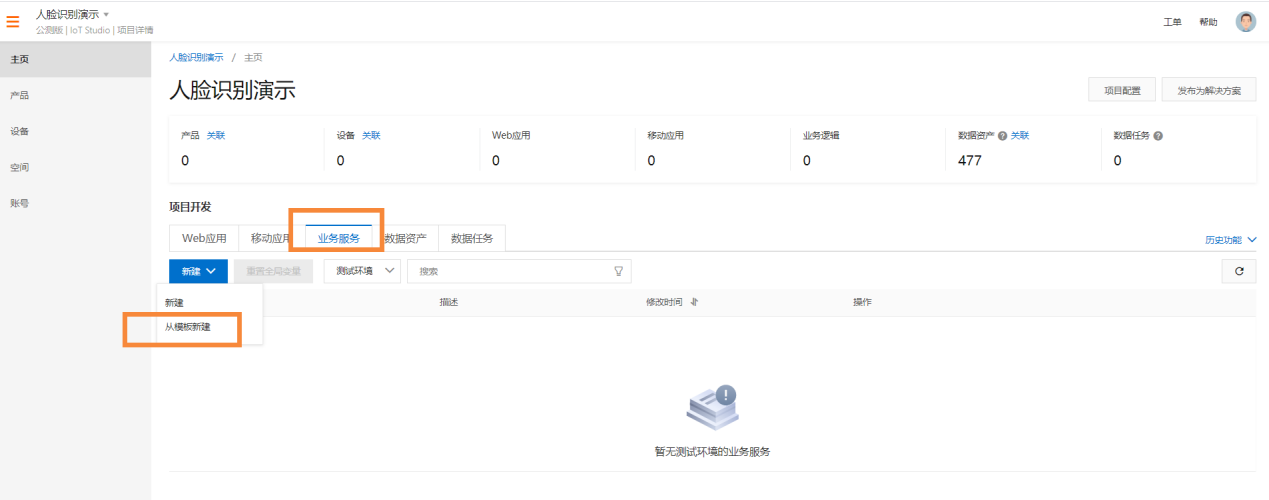


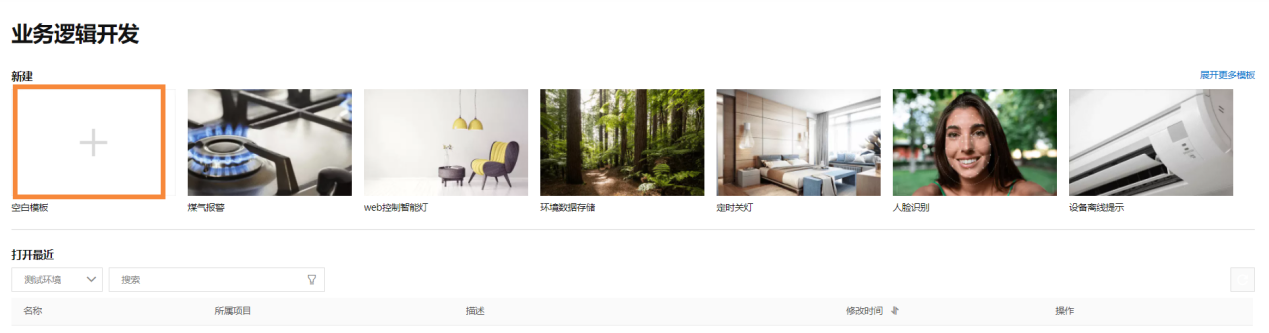




现在我们进入了项目首页，点击业务服务---->新建--->从模板开始--->新建

这样我们就进入了业务服务开发页面

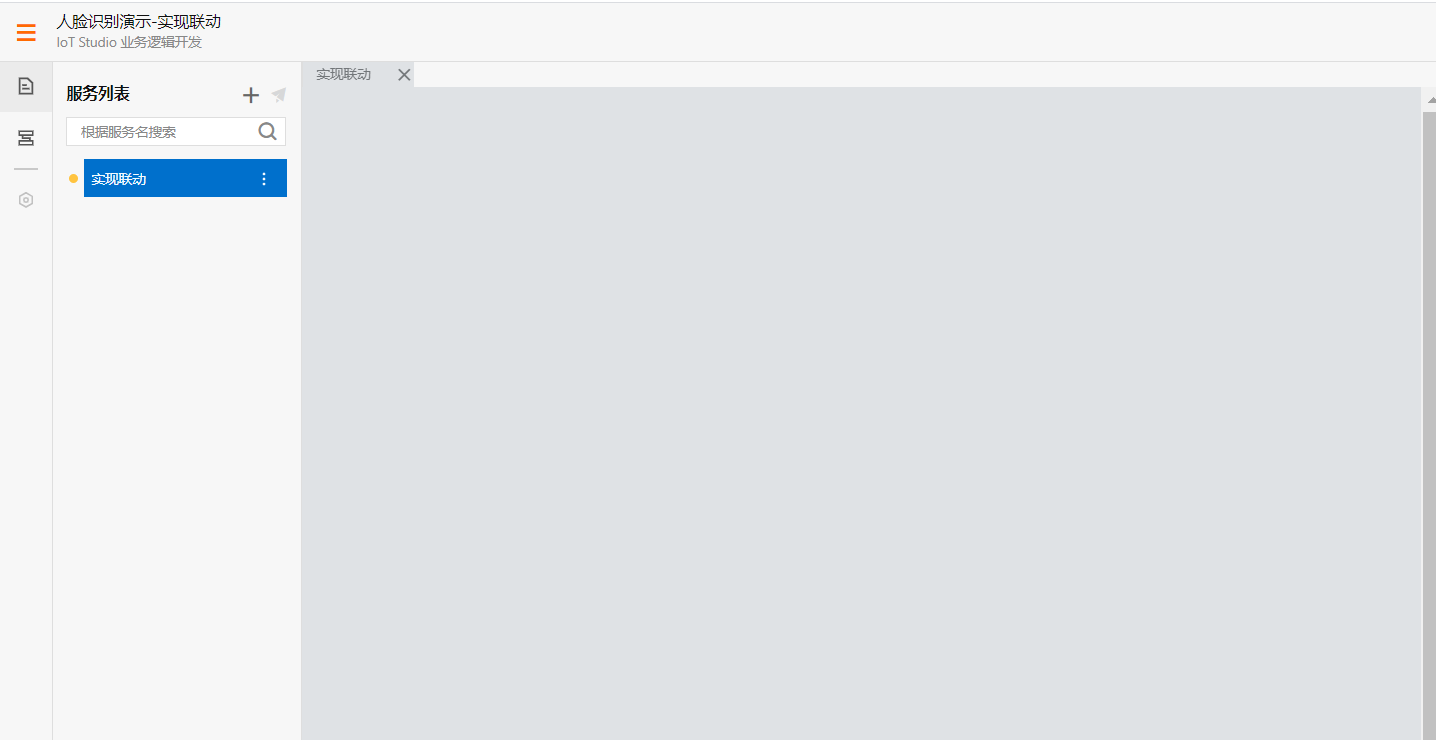






业务服务的开发与发布

这里是我们新建的业务服务页面



进行业务服务的开发，实现参数的解析以及根据条件触发相应动作，比如当’Score’大于0.5时，触发钉钉消息推送

输入数据示例

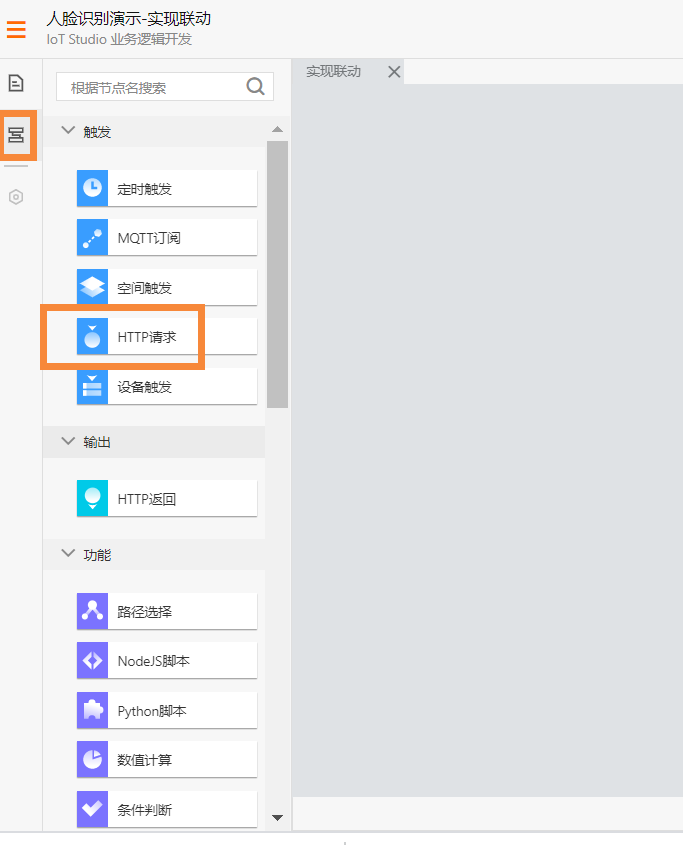
{’RequestId’:’02546217-3077-4B5B-8A1A-4A58C3F9E7B5’,’Data’:{’MatchList’:[{’FaceItems’:[{’FaceId’:’4023393’,’EntityId’:’huangbo’,’Score’:0.8057092428207397,’DbName’:’myFaceDB’,’ExtraData’:’黄渤’}],’Location’:{’X’:251,’Y’:90,’Height’:241,’Width’:174}}]}}

1：选择节点，然后拖拽触发条件 HTTP 请求

2：选择Python脚本tudo

3：选择HTTP返回

4：将模块进行链接并配置参数



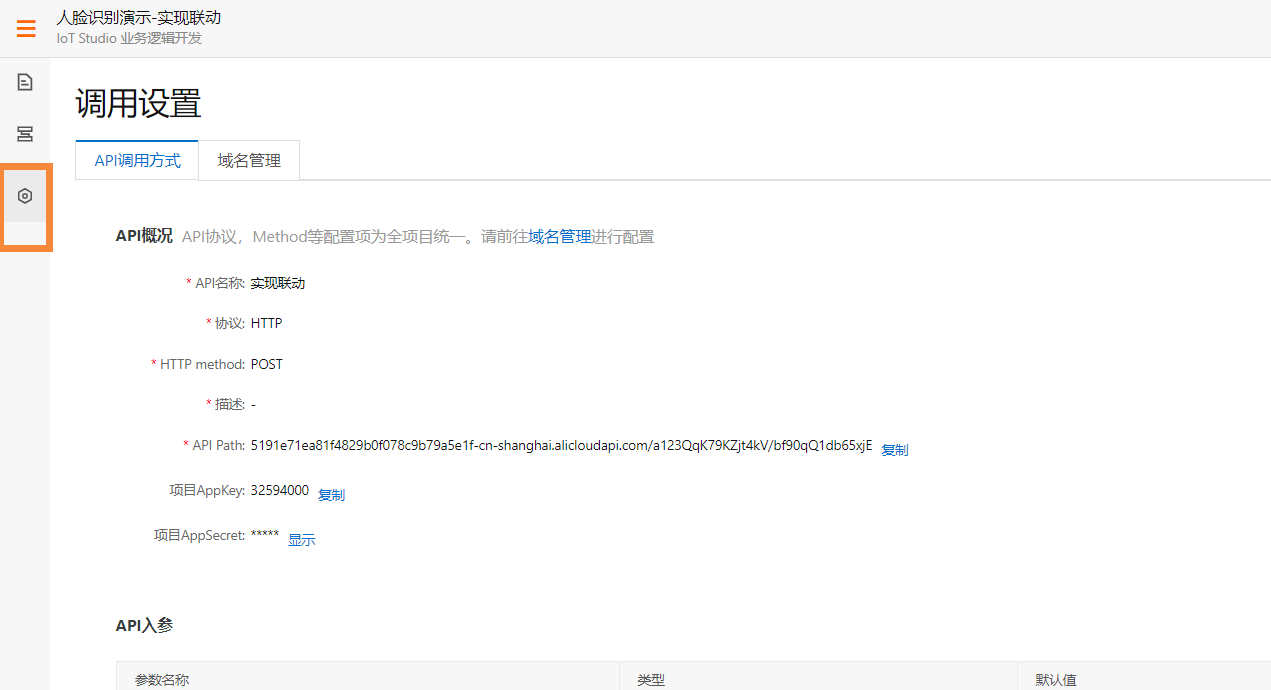
业务服务发布

现在我们业务服务已经完成了，我们开始部署并发布我们额业务服务，如图依次点击“保存”“部署”“调试”“发布”



业务服务的调用和参数信息

进入“服务调用设置”可以看到调用方法和示例代码



**上传数据到iot接口平台**

python依赖相关

<https://github.com/aliyun/alibabacloud-iot-api-gateway-python2-client>

示例:

import uuid  
  
from com.aliyun.api.gateway.sdk import client  
from com.aliyun.api.gateway.sdk.http import request  
from com.aliyun.api.gateway.sdk.common import constant  
  
url = "/a123ml8t9gu2zqTT/bfjc5oUj7hE9BV"  
  
host = "IMG_256http://4efd73d6c344454c9f1eab552d71964b-cn-shanghai.alicloudapi.com"  
  
cli = client.DefaultClient(app\_key="\*\*\*", app\_secret="\*\*\*")  
  
req\_post = request.Request(host=host, protocol=constant.HTTPS, url=url, method="POST", time\_out=30000, sign\_with\_body=False)  
  
bodyMap = {  
    'id': str(uuid.uuid4()),  
    'version': "1.0",  
    'request': {  
        'apiVer': "1.0.0"  
    },  
    'params': {  
        'action': 'TEST\_ACTION',  
        'test': '123'  
    }  
}  
req\_post.set\_body(bodyMap)  
req\_post.set\_content\_type(constant.CONTENT\_TYPE\_JSON)  
print(cli.execute(req\_post))

end