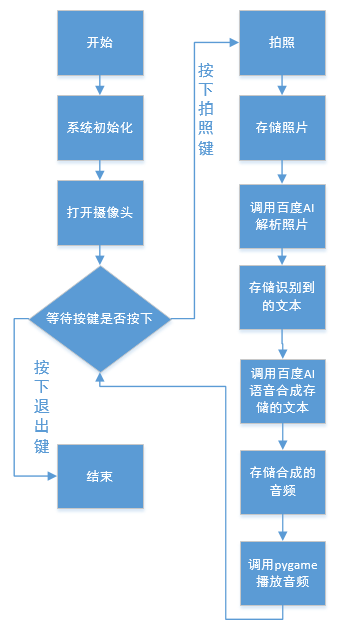
1. **程序基本思想：**

借助百度AI开放平台的通用文字识别和语音合成功能完成文字识别和公交报站功能。

接口1：通用文字识别-识别图片中的文字信息。

接口2：语音合成-将文字转为声音，可根据场景需求对发音人的语速、音调、音量进行灵活设置。

1. **程序流程图**

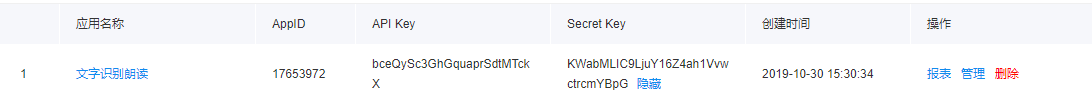
该程序的流程图如下：

1. **百度AI开发平台信息注册：**

app\_id=**'17653972'**app\_key =**'bceQySc3GhGquaprSdtMTckX'**secret\_key =**'KWabMLIC9LjuY16Z4ah1VvwctrcmYBpG'**

讲解：

登录百度AI平台，创建个人开发者账号。然后创建应用（就是你需要用到的一些接口，如文字识别等）.

常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的**应用列表**中查看。

1. **文字识别**

**def** critic\_word(img\_path):  
 client=AipOcr(app\_id,app\_key,secret\_key)  
 **with** open(img\_path, **'rb'**) **as** fp:  
 image=fp.read()  
 options = {}  
 options[**"language\_type"**] = **"CHN\_ENG"** options[**"probability"**] = **"true"** content=client.basicGeneral(image, options)  
 fname = time.strftime(**"%Y%m%d%H%M"**, time.localtime())  
 **with** open(fname + **'.txt'**, **'w'**, errors=**'ignore'**) **as** file:  
 **for** word\_result **in** content[**'words\_result'**]:  
 file.write(word\_result[**'words'**])  
 print(word\_result)  
 file.close()  
 print(**"fname: "**, fname + **'.txt'**)  
 **return** fname

讲解：

client=AipOcr(app\_id,app\_key,secret\_key)

//AipOcr是OCR的Python SDK客户端,就是建立跟百度平台对接。好比有个人叫百度，你正在跟它打电话，尝试通话，即建立连接。

**with** open(img\_path, **'rb'**) **as** fp:  
 image=fp.read()

//img\_path表示的是本地图片的存放地址，好比家庭地址，有地址才能找到它，找到这张图片。然后open-打开这张图片，‘rb’ 表示是以二进制读的方式打开（‘w’表示写）。

options = {}

//options是一个字典，里面放置了向百度请求文字识别时的一些参数。  
 options[**"language\_type"**] = **"CHN\_ENG"**

//识别语言类型，默认为“**CHN\_ENG“，即中英文混杂** options[**"probability"**] = **"true"**

// 是否返回识别结果中每一行的置信度，置信度就是翻译的准确度  
 fname = time.strftime(**"%Y%m%d%H%M"**, time.localtime())

//图片转化的文字数据需存放在一个文件中，fname即为文件名

// time strftime(format,时间) 函数接收以时间元组,并返回以可读字符串表示的当地时间,格式由参数format决定

content=client.basicGeneral(image, options)

//带参数调用通用文字识别, 图片参数为本地图片  
 **with** open(fname + **'.txt'**, **'w'**, errors=**'ignore'**) **as** file:  
 **for** word\_result **in** content[**'words\_result'**]:  
 file.write(word\_result[**'words'**])  
 print(word\_result)  
 file.close()

//content中存储了从百度ai平台上处理后返回的文字数据，它是一个字典数据类型，里面包含了很多返回参数。words\_result是其中之一，定位和识别结果数组。words\_result数组中的元素为{"words": " 文字翻译内容"}。

调用file.write（）将数据写入fname.txt中。

1. **语音朗读**

**def** speak(word\_path,index):  
 print(**"speak!"**)  
 client=AipSpeech(app\_id,app\_key,secret\_key)  
 size=getsizeof(word\_path)  
 music\_file=word\_path+**'-'**+str(index)+**'.mp3'  
 with** open(word\_path+**'.txt'**, **'r'**, encoding=**'GBK'**, errors=**'ignore'**) **as** file:  
 **while** (**True**):  
 content = file.read(1024)  
 **if** getsizeof(content) == 49:  
 **break** print(**"content: "**, content)  
 result = client.synthesis(content, **'zh'**, 1, {  
 **'vol'**: 5,  
 })  
 **if not** isinstance(result, dict):  
 **with** open(music\_file, **'wb'**) **as** f:  
 f.write(result)  
 f.close()

pygame.mixer.init()  
 pygame.mixer.music.load(music\_file)  
 pygame.mixer.music.play()  
 time.sleep(int(size/10))  
 file.close()

讲解：

client=AipSpeech(app\_id,app\_key,secret\_key) //同上

size=getsizeof(word\_path)//获得需要语音朗读的文本大小

music\_file=word\_path+**'-'**+str(index)+**'.mp3'** //命名我们需要写入的音频文件

1. **代码说明**

*# -\*- coding:UTF-8 -\*-***import** time  
**import** cv2  
**import** pygame  
**import** re  
**from** sys **import** getsizeof  
**from** aip **import** AipSpeech  
**from** aip **import** AipOcr

*#百度ai平台的个人信息*app\_id=**'17653972'**app\_key =**'bceQySc3GhGquaprSdtMTckX'**secret\_key =**'KWabMLIC9LjuY16Z4ah1VvwctrcmYBpG'**music\_index=0

在百度ai平台调用的个人注册信息，用于在与百度ai平台建立连接时表明自己的身份

*#调用-api文字识别***def** critic\_word(file\_path):  
 client=AipOcr(app\_id,app\_key,secret\_key) 与百度ai文字识别接口建立连接  
 img\_path=file\_path+**'.jpg'** 图片文件名

程序首先会创建图片文件，然后接收到识别的文字数据后，会创建文本文件，最后调用语音合成接口后创建音频文件。3个文件的文件名相同，仅后缀名不同。图片-jpg,文本-txt,音频-MP3。  
 **with** open(img\_path, **'rb'**) **as** fp: 打开一个图片文件  
 image=fp.read() 图片文件中的图片信息复制给image对象  
 options = {} 设定上传至百度ai平台时的操作信息  
 options[**"language\_type"**] = **"CHN\_ENG"** 设定文字识别的类型为中文英文混合options[**"probability"**] = **"true"** 设定文字置信度，就是识别的准确度，程序中无具体应用  
content=client.basicGeneral(image, options) 向百度ai发出图片和图片转文字的设定的请求，百度ai发送回的文本数据存储在‘content’中  
 *# 将图片转为文字，写入文本* flag = 0 信号量，初始值设为0  
 **with** open(file\_path + **'.txt'**, **'w'**, errors=**'ignore'**) **as** file: 创建并打开一个文本文件  
 **if** (content[**'words\_result'**]): # 看是否识别到文字**for** word\_result **in** content[**'words\_result'**]: 遍历文件数据，每次提取出一个词组，赋值给word\_result  
 print(**"word\_result: "**, word\_result)  
 cost = re.findall(**r'[1-9]+\.?[0-9]\*'**, word\_result[**'words'**])

在词组中找数字，查询结果赋值给cost  
 print(**"cost: "**, cost)  
 **if** len(cost): # 看文字中是否有数字,len()函数是求cost对象的长度，长度为0就是无数字，不进入该语句  
file.write(cost[0]) 向文本文件中写入公交号  
 file.write(**"路公交车即将到站。"**) 向文本文件中写入该话  
 flag = 1 flag设定为1,表明查找成功  
 **break  
 if** (flag == 0):

此时已完成遍历退出循环了，为有文字，但文字中没有数字的情况  
file.write((**"未识别到公交车号"**))  
 **else**: 此句是未查到数字即公交车号的情况  
 file.write((**"未识别到公交车号"**))  
 file.close()  
 print(**"fname: "**, fname + **'.txt'**)  
 **return** fname

*#调用api-语音朗读***def** speak(word\_path,index):  
 print(**"speak!"**)  
 client=AipSpeech(app\_id,app\_key,secret\_key) 与百度ai语音合成接口建立连接  
 size=getsizeof(word\_path)  
 music\_file = word\_path + **'-'** + str(index) + **'.mp3'** 音频文件名pygame.mixer.init() 音频朗读程序初始化  
 **with** open(word\_path+**'.txt'**, **'r'**, encoding=**'GBK'**, errors=**'ignore'**) **as** txt\_file: 打开文本文件  
 **while** (**True**):  
 content = txt\_file.read(1024) 一次读取文本文件中1024个字节  
 **if** getsizeof(content) == 49: 如果读取的content的大小为49，表明读取到末尾了，退出。49是测试得出来得大小。  
 **break** print(**"content: "**, content)  
 result = client.synthesis(content, **'zh'**, 1, {  
 **'vol'**: 5,   
 }) 将要朗读得文本发送给百度AI，content是文本内容，’zh’是合成音频为中文，vol设定音量  
 **if not** isinstance(result, dict):   
 **with** open(music\_file, **'wb'**) **as** f:  
 f.write(result) 写入音频  
 f.close()  
 pygame.mixer.music.load(music\_file) 加载音频  
 pygame.mixer.music.play() 播放音频  
 txt\_file.close()  
 time.sleep(int(size/10))  
*#主函数***def** main(index):  
 cap=cv2.VideoCapture(0) 打开摄像头  
 cap.set(3,900) 设定窗口大小  
 cap.set(4,900)  
 **while**(cap.isOpened()): 当摄像头是打开时  
 ret\_flag,Vshow=cap.read()   
 cv2.imshow(**'Capture'**,Vshow) 显示摄像头界面  
 k=cv2.waitKey(1) 等待键盘中的输入，1表示仅等待输入一个值  
 **if** k==ord(**'x'**): 键盘按下‘x’  
 index+=1  
 print(**'开始拍照'**)  
 fname = time.strftime(**"%Y%m%d%H%M"**, time.localtime()) 设定文件名，直接采用当前时间  
 cv2.imwrite(fname+**'.jpg'**,cv2.resize(Vshow, (1000, 600), interpolation=cv2.INTER\_AREA)) 将摄像头拍摄的界面存入文件中  
 word\_path = critic\_word(fname) 调用文字识别函数  
 speak(word\_path,index) 调用语音合成函数  
 **elif** k==ord(**'q'**): 按下‘q’键，退出循环，即退出等待拍照界面  
 **break** print(**'---end---'**)  
 time.sleep(1)  
 cap.release() 释放摄像头资源  
 cv2.destroyAllWindows() 关闭窗口

main(music\_index) 调用主函数，传入形参音频序号，初始值为0，因为音频无法被覆盖，所以每重新拍照一次，获得新的音频，音频文件名的序号加1.仅为了保证音频文件不重名。