WeChat Plugin For PC(Windows)

– 微信小助手：自动回复、防撤回、远程控制

姓名：李潇睿 学号：41624401

北京科技大学计通学院 通信1604班

|  |
| --- |
| 摘要：在本报告中介绍了一款在 Windows下的微信客户端辅助工具，支持自动回复、消息防撤回、远程控制等功能。该软件利用了Opencv、SqLite、win32API、BaiduOCR等多种成熟的开发工具，简洁的实现了软件基本功能，达到了辅助微信聊天的目的，未来计划将其作为Windows服务置于服务器上，以代替itchat，避免被微信制约，实现AI微信小管家的目的。笔者尚未找到目前Windows平台有类似的外挂程序。 |
| 关键字： WeChat外挂;OpenCV;SQLite;OCR |

# 项目背景和意义

## 项目背景

如今微信早已走进千家万户，成为每个互联网居民传递信息、交流感情、完成工作和学习必不可少的重要一环。而随着使用者和好友数量的日益增长，微信的生态环境却日渐被破坏，广告、群聊灌水、在吗等诸多聊天时的问题困扰着广大使用者，与此同时，其也被赋予了越来越重要的使命，课程答疑、竞赛组队、班级管理、工作汇报、会议通知等越来越多生活中、工作中的重要环节必须通过微信来完成。

而在如此重要的微信上，自动化工具却是少之又少。在腾讯的企业导向下，官方并未推出任何一款管理软件，相比于QQ和Skype，其也缺乏 自动回复、丰富的群聊、发送大文件等诸多功能，另外作为数据的重要载体，微信上也存储了巨量的有价值的信息，而目前互联网上比较著名的项目仅有itchat一个，其用到的网页版微信端口也在日益收紧，不断有传言微信准备关闭网页版。

## 项目意义

在此基础上，PC版的微信外挂程序就显得尤为重要。目前仅有两名作者开发了Mac版的微信助手程序，其中一个还在去年被爆出作者利用DNS保留攻击窃取用户信息，因此删除了WeChatPlugin-MacOS的github仓库内容，然而截止我完成我的项目时，该仓库已经有12949个Star，足见用户对此功能的需求。

因此，我计划完成微信聊天防撤回、聊天信息记录、通过微信的文件传输助手远程控制PC端软件等功能。

# 需求分析

## 具体功能如下：

* 消息自动回复
* 消息防撤回
* 远程控制

所谓远程控制，是指通过微信来控制 PC上运行的其他应用，比如：

开启系统屏幕保护

清空回收站

注销、锁屏、休眠、关机、重启

退出 QQ、WeChat、Chrome、所有程序

网易云音乐(播放、暂停、下一首、上一首、喜欢、取消喜欢)

* 自动登录开关
* 可设置是否开机启动本脚本程序，实现开机自动登录微信的目的。

# 概要和详细设计

## 3.1代码流程

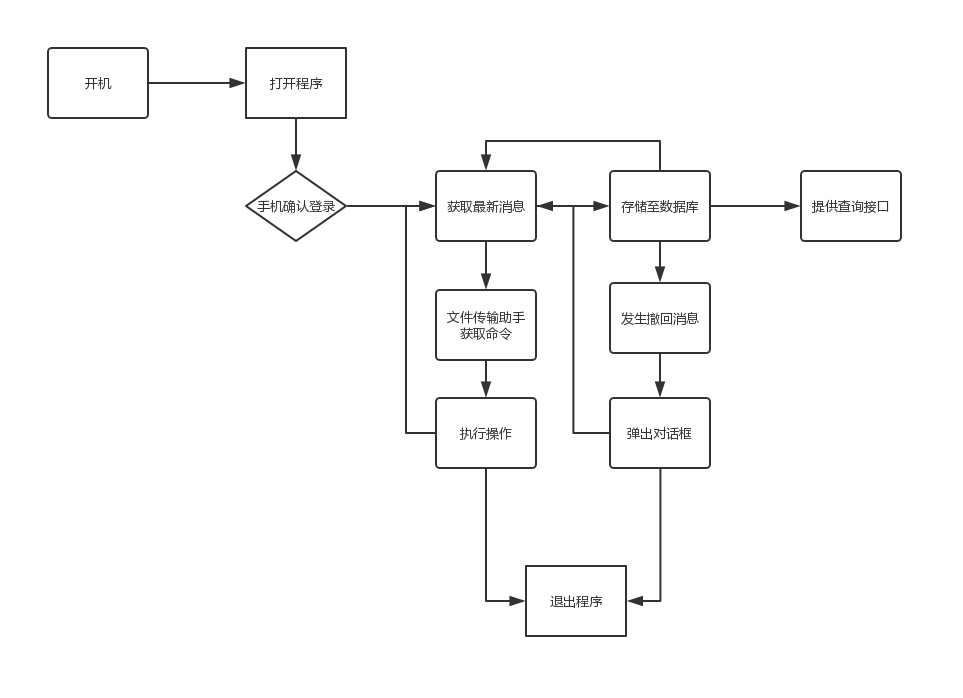


图 1 流程图

## 3.2 代码构成

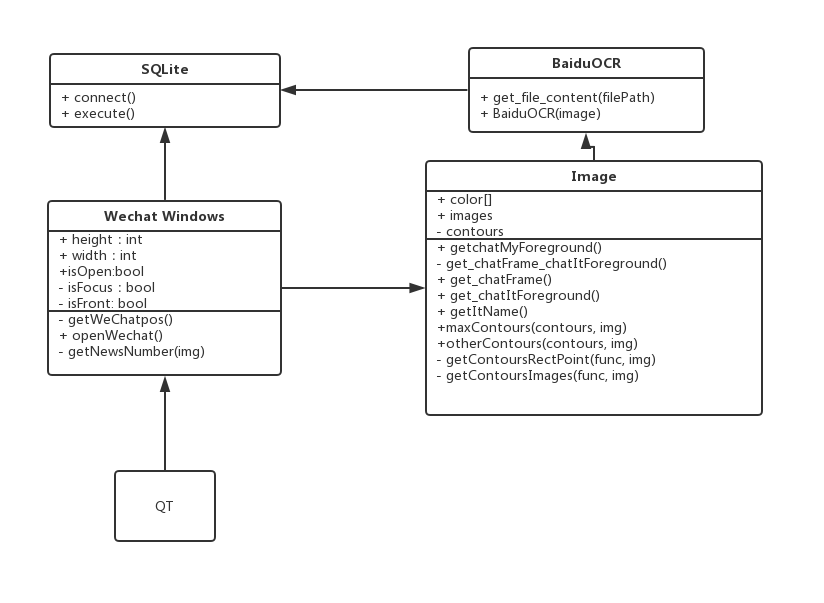


图 2 UML类图

## 3.3 数据库ER图

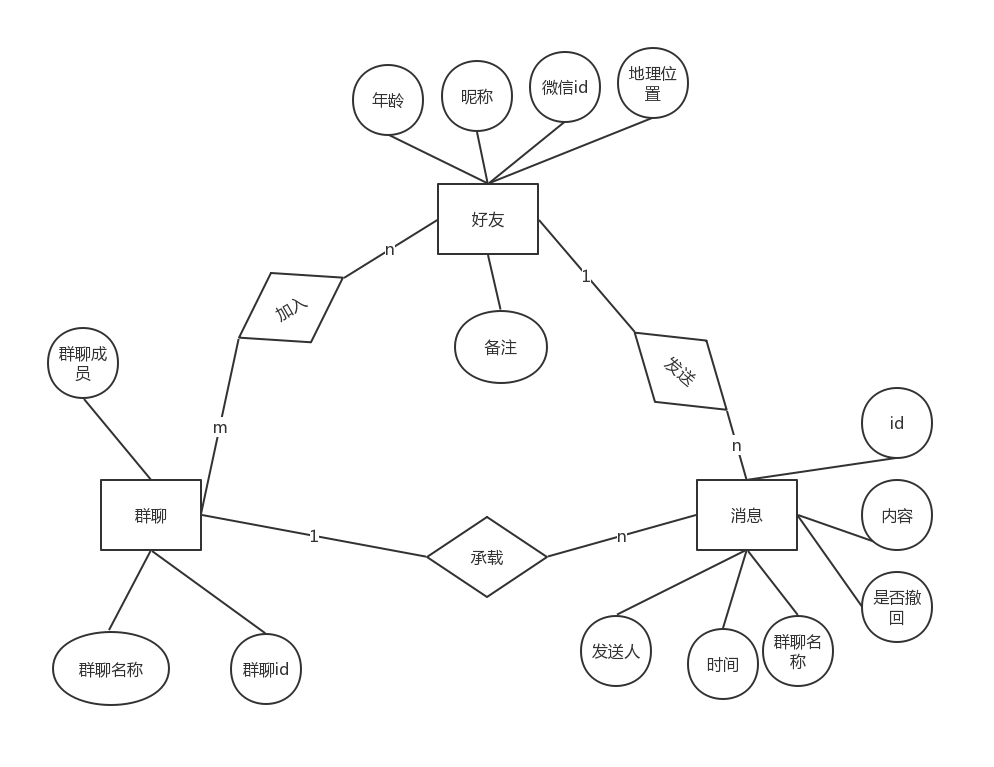


图 3数据库E-R图

## 3.4技术手段选型

因PC版WeChat并未开放任何接口，为方便Windows开发，使用Windows32自带API进行软件的打开与关闭。

使用Opencv-python进行界面的切割与状态判断，本项目中使用的是最新版本的OpenCV 4.1。

为保持轻量化的设计原则，也与微信本身的数据库设计尽量类似，故采用SQLite进行存储，直接使用SQLite的python库进行数据库表的建立和增删改查等操作。

在简体中文汉字识别方面，python离线版本的OCR库均不尽如人意，现在使用较多的是Tesseract[1]，而在使用神经网络训练的AI领域背景较为单一的简体中文是已经基本解决了的一个领域，是别的成功率很高，两者的对比可参考[2]，而百度云对个人开发者每天有5万次的OCR识别次数，足以满足本次作业的需要，故本次作业使用百度云OCR进行消息的读取。

在自动化领域，使用较多的python库为PyAutoGUI和PyWinAuto两种，其中前者主要着重于鼠标、键盘、截图屏幕的功能，后者虽也有类似功能但并不完善，主要着重于使用窗口、句柄、ListView、Button等进行窗口操作，实际实验中微信有一些崩溃的概率，而且它也不支持正则表达式，对中文的支持也不好。所以经过比较后，本项目还是采用了PyAutoGUI进行开发。

## 3.5 Requirements.txt

numpy==1.16.2

requests==2.18.4

Pillow==6.0.0

pyautogui==0.9.42

win32gui==221.6

opencv-python==4.1.0.25

baidu-aip==2.2.13.0

pywin32==224

# 代码实现

## 4.1软件平台：Win10+ Python3.7

## 4.2主要函数：

1、利用欧几里德距离判断颜色相似性：

|  |
| --- |
| def euclidean(p, q):  # 计算欧几里德距离,并将其标准化  e = sum([float(p[i] - q[i]) \*\* 2 for i in range(p.\_\_len\_\_())])  return 1 / (1 + e \*\* .5) |

几个数据集之间的相似度一般是基于每对对象间的距离计算。最常用的当然是欧几里德距离，其公式为:

https://images0.cnblogs.com/blog2015/751509/201506/281446588617494.png

2、创建数据库表

|  |
| --- |
| c.execute('''create table friends(\_friendId integer primary key autoincrement,  name text''')  c.execute('''create table message(\_\_message\_Id integer primary key autoincrement,  messageContent text,  \_friendId integer,  isWithdraw integer,  messageTime text,''') |

3、Opencv双阈值二值化图像

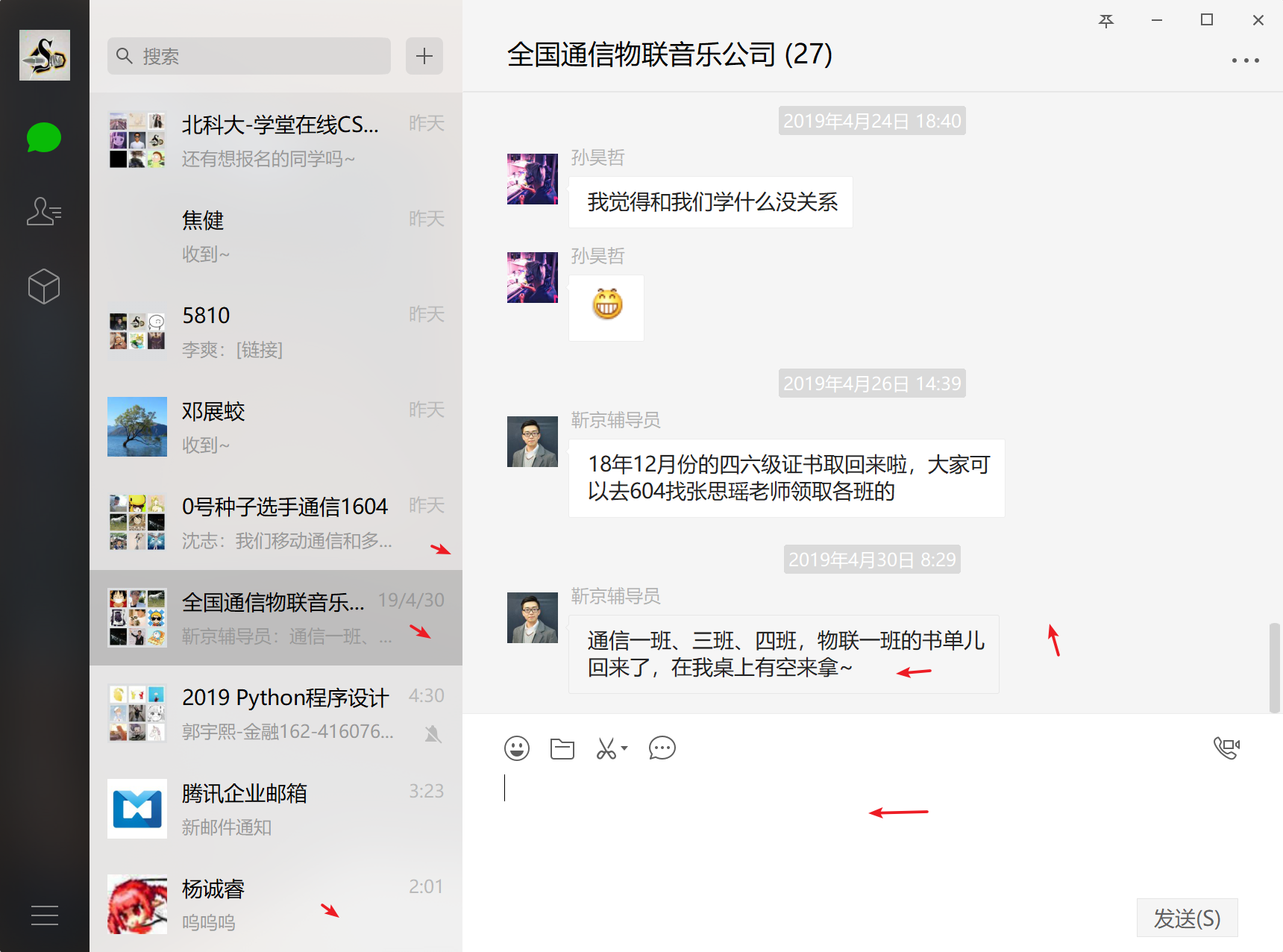
由于在微信窗口中各部分均有相当明显的颜色差异（见下图）

图 4 微信聊天窗口

故可以直接采用将图像二值化的方式将所需的部分提取出来。

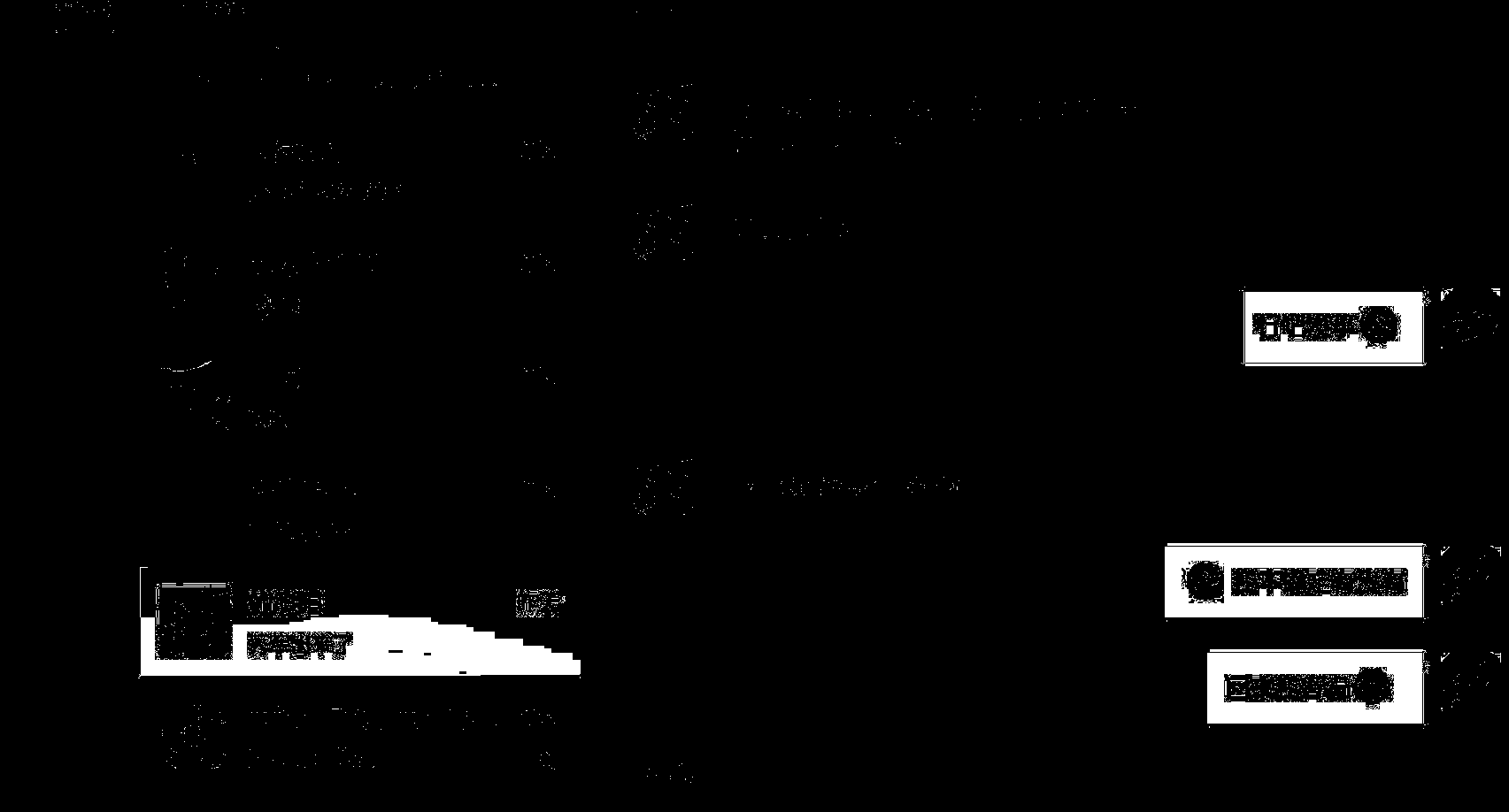


图 6 二值化后黑白图

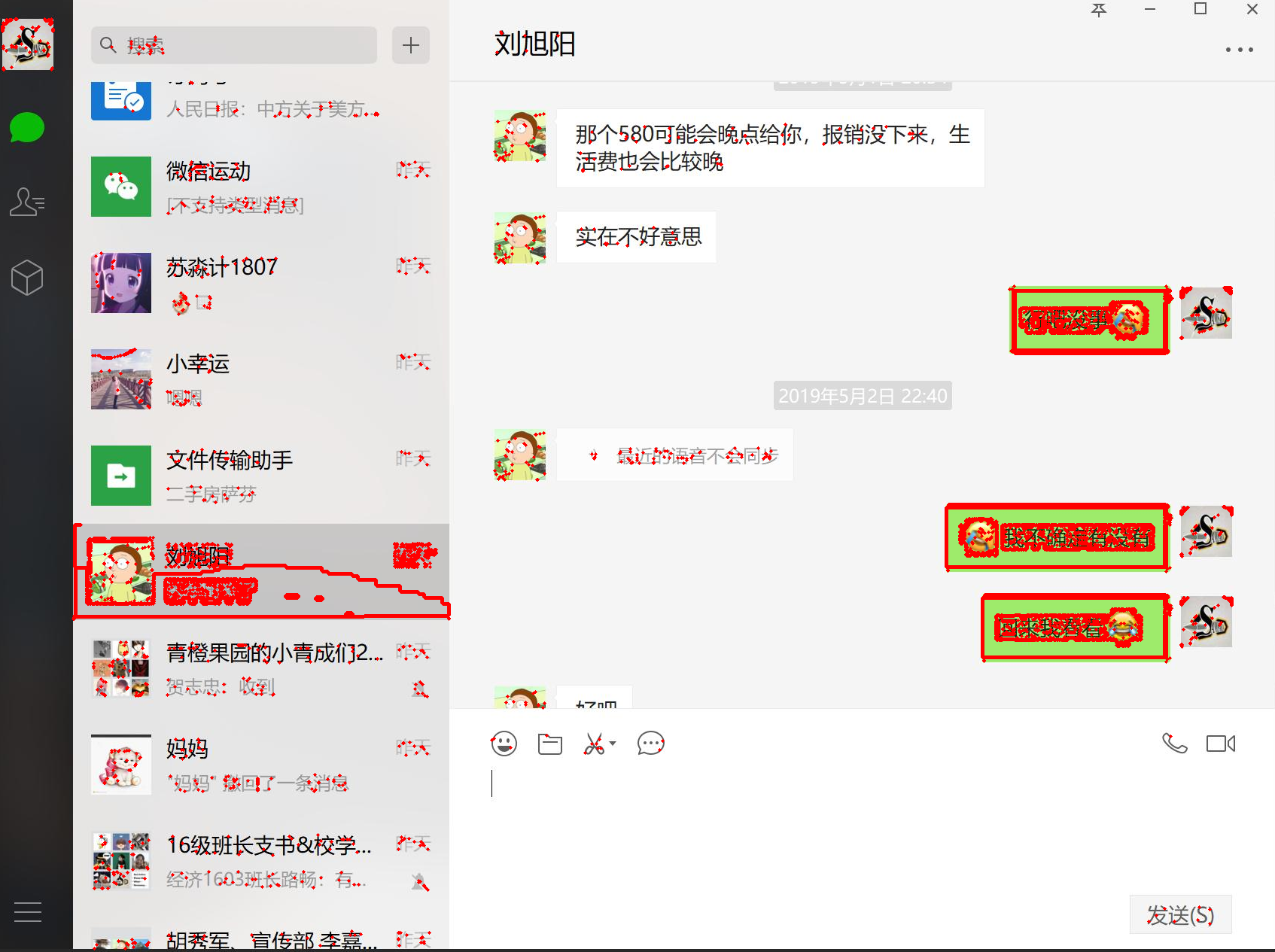


图 7 使用OpenCV的findcontours查找边缘后的切割结果

而OpenCV自带的函数中只有一个阈值，例如：

|  |
| --- |
| \_, binary1 = cv2.threshold(gray, 195, 255, cv2.THRESH\_BINARY) |

表示把超过195的灰度值映射到255上，小于195的值映射到0

|  |
| --- |
| \_, binary2 = cv2.threshold(gray, 197, 255, cv2.THRESH\_BINARY\_INV) |

表示把超过197的映射到0，小于197的映射到255。

这样我就做到了两图中只有195-197的部分为255，然后对两个结果进行与运算，得到的结果中就只有195-197是纯白色了。

代码中还实现了如区域抠图、判断文件是否存在、最大最小化程序窗口、遍历截图查找是否有红色气泡等功能，在此就不再赘述。

# 代码测试

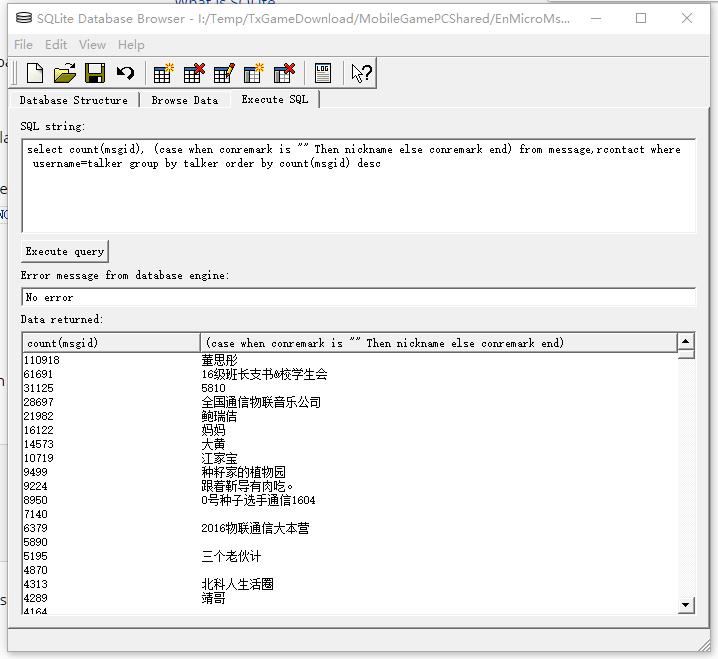


图 6 查询数据库中的朋友聊天数据总条数

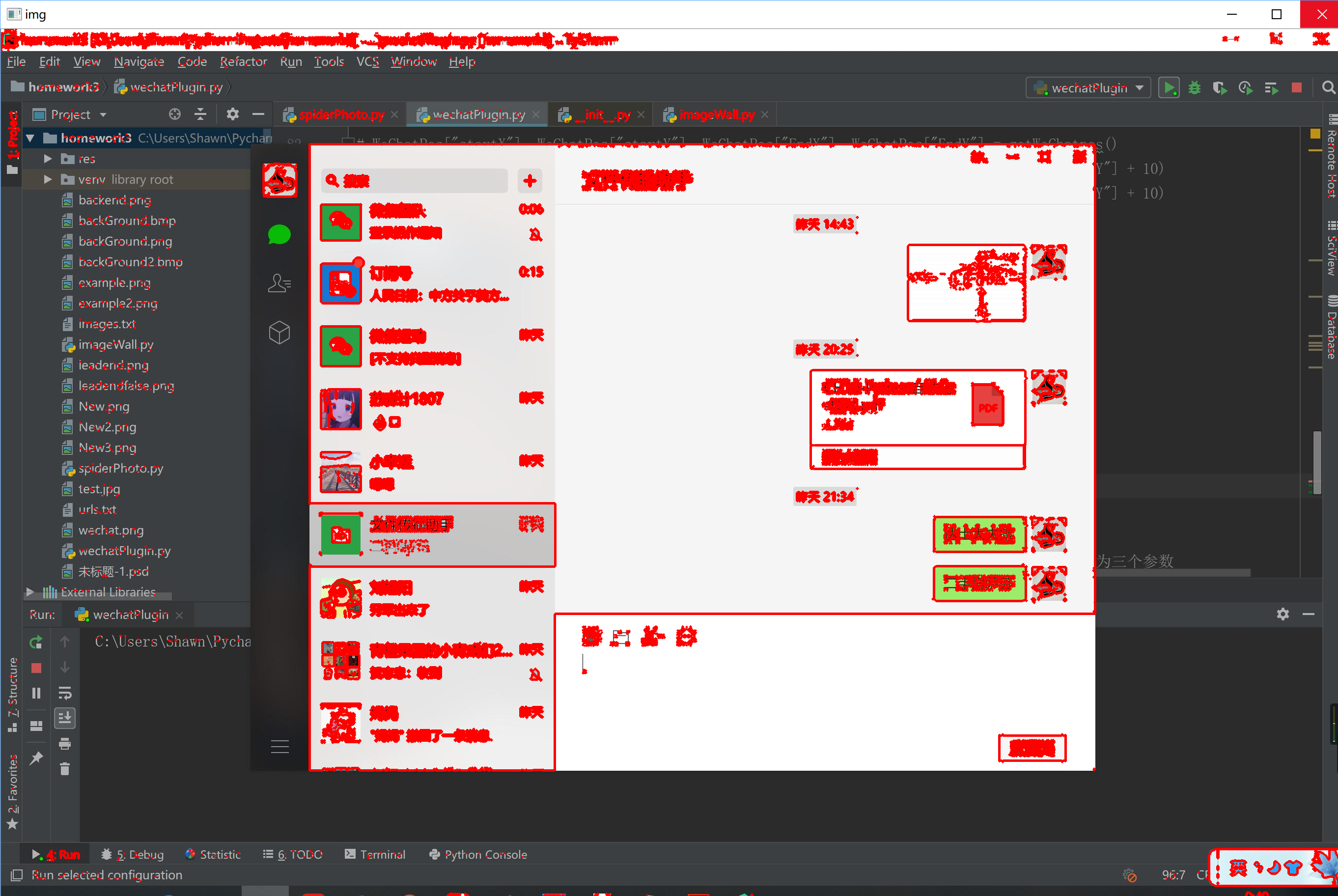


图9 OpencV取某一值时对界面的区域分割

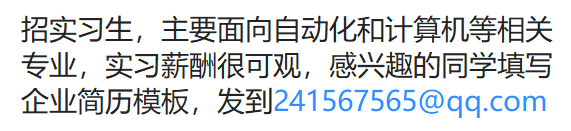


图 10 测试OpenCV截取截图

测试输出结果：{'log\_id': 6324550012620209129, 'words\_result\_num': 3, 'words\_result': [{'words': '招实习生,主要面向自动化和计算机等相关'}, {'words': '专业,实习薪酬很可观,感兴趣的同学填写'}, {'words': '企业简历模板,发到241567565@gg.com'}]}

# 六、结论和未来方向

本项目实现了一款在 Windows下的微信客户端辅助工具，支持自动回复、消息防撤回、远程控制等功能。该软件利用了Opencv、SqLite、win32API、BaiduOCR等多种成熟的开发工具，简洁的实现了软件基本功能，达到了辅助微信聊天的目的，虽然程序还有很多不足，但已基本可以使用，在实现本项目的过程中，我遇到了很多以往难以察觉的问题，比如Windows的任务栏居然不是纯黑色（有噪点还会隔一段时间一变），比如微信简约的界面下居然还有一堆灰度渐变等，但是幸好我也通过别的途径解决了他们。在这个过程中经历了很多的锻炼和成长，我感觉收获颇丰。

本项目其实距离完成还有很长的路要走，由于时间关系，很多模块都是从简设计的，没有考虑软件的鲁棒性。比如不同设备的显示器分辨率问题，微信上的公众号、服务号判断，实现的数据库也和原本设计的E-R有一定差距，还有诸如微信软件位置查找问题，数据安全等很多也需要考虑，主要是这个项目具有很好的实际应用价值和市场前景，我也会利用空闲时间努力把它完善。

# 七、致谢

感谢GitHub支持私有化仓库，因参考数量众多，CSDN的众多博客在参考文献处不能一一列出，对他们表示衷心的感谢。

# 八、参考文献

1. 匿名 Python图像处理之图片文字识别（OCR）（2018.06.09）[2018.06.09]<https://www.jianshu.com/p/137b7922485e>
2. BetaCat.未上线的猫 识别图片中的文字 - Tesseract 和 百度云OCR的对比（2018.01.16）[2018.01.16]

https://betacat.online/posts/2018-01-16/chinese-text-ocr-via-python/

# 九、代码

|  |
| --- |
| import sqlite3  import time  import cv2  import numpy as np  from aip import AipOcr  import os  import pyautogui  import win32api  import win32gui  selfFrame = '27292c'  selfSearch = 'dbd9d8'  chatBackground = 'f5f5f5'  chatItForeground = 'ffffff'  WeChatPos = {}  def changeColor(value):  value = value.upper()  digit = list(map(str, range(10))) + list("ABCDEF")  if isinstance(value, tuple):  string = '#'  for i in value:  a1 = i // 16  a2 = i % 16  string += digit[a1] + digit[a2]  return string  elif isinstance(value, str):  a1 = digit.index(value[1]) \* 16 + digit.index(value[2])  a2 = digit.index(value[3]) \* 16 + digit.index(value[4])  a3 = digit.index(value[5]) \* 16 + digit.index(value[6])  return a1, a2, a3  # 计算欧几里德距离：  def euclidean(p, q):  # # 如果两数据集数目不同，计算两者之间都对应有的数  # same = 0  # for i in p:  # if i in q:  # same += 1  # 计算欧几里德距离,并将其标准化  e = sum([float(p[i] - q[i]) \*\* 2 for i in range(p.\_\_len\_\_())])  return 1 / (1 + e \*\* .5)  def getWeChatpos():  """返回值：窗口左上角和右下角坐标"""  classname = "WeChatMainWndForPC"  titlename = "微信"  # 获取句柄  hwnd = win32gui.FindWindow(classname, titlename)  # 获取窗口左上角和右下角坐标  return win32gui.GetWindowRect(hwnd)  def openWechat():  win32api.ShellExecute(0, 'open', r'C:\Program Files (x86)\Tencent\WeChat\Wechat.exe', '', '', 1)  # screenWidth, screenHeight = pyautogui.size()  # currentMouseX, currentMouseY = pyautogui.position()  # initx, inity = pyautogui.position()  # print(pyautogui.locateOnScreen("wechat.png"))  # pyautogui.moveTo()  # pyautogui.click()  # pyautogui.moveTo(initx, inity)  def BaiduOCR(imagefile):  # 百度OCR识别  APP\_ID = '16205886'  API\_KEY = '11cgmBlpYydUMCTIe0MzcecW'  SECRET\_KEY = 'teW5LS8oR7dwSkGYoBmHC1ua51c6DhC2'  client = AipOcr(APP\_ID, API\_KEY, SECRET\_KEY)  """ 读取图片 """  def get\_file\_content(filePath):  with open(filePath, 'rb') as fp:  return fp.read()  image = get\_file\_content(imagefile)  """ 调用通用文字识别, 图片参数为本地图片 """  print(client.basicGeneral(image))  """ 带参数调用通用文字识别, 图片参数为本地图片 """  # """ 如果有可选参数 """  # options = {}  # options["language\_type"] = "CHN\_ENG"  # options["probability"] = "true"  # print(client.basicGeneral(image, options))  # \_, binary1 = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY)  # \_, binary2 = cv2.threshold(gray, 20, 255, cv2.THRESH\_BINARY\_INV)  # binary = cv2.bitwise\_and(binary1, binary2)  # cv2.imshow("img", binary)  # cv2.waitKey(0)  def getchatMyForeground():  \_, binary1 = cv2.threshold(gray, 195, 255, cv2.THRESH\_BINARY)  \_, binary2 = cv2.threshold(gray, 197, 255, cv2.THRESH\_BINARY\_INV)  binary = cv2.bitwise\_and(binary1, binary2)  cv2.imshow("img", binary)  cv2.waitKey(0)  return cv2.findContours(binary, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  def get\_chatFrame\_chatItForeground():  \_, binary1 = cv2.threshold(gray, 250, 255, cv2.THRESH\_BINARY)  \_, binary2 = cv2.threshold(gray, 255, 255, cv2.THRESH\_BINARY\_INV)  binary = cv2.bitwise\_and(binary1, binary2)  cv2.imshow("img", binary)  cv2.waitKey(0)  return cv2.findContours(binary, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  def get\_chatFrame():  contours, hierarchy = get\_chatFrame\_chatItForeground()  tempimg = img.copy()  return maxContours(contours, tempimg)  def get\_chatItForeground():  contours, hierarchy = get\_chatFrame\_chatItForeground()  return otherContours(contours)  def getItName():  \_, binary1 = cv2.threshold(gray, 188, 255, cv2.THRESH\_BINARY)  \_, binary2 = cv2.threshold(gray, 195, 255, cv2.THRESH\_BINARY\_INV)  binary = cv2.bitwise\_and(binary1, binary2)  cv2.imshow("img", binary)  cv2.waitKey(0)  return cv2.findContours(binary, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)  def maxContours(contours, img):  # c\_max = []  max\_area = 0  max\_cnt = 0  for i in range(len(contours)):  cnt = contours[i]  area = cv2.contourArea(contours[i])  # find max countour  if area > max\_area:  if max\_area != 0:  c\_min = [max\_cnt]  cv2.drawContours(img, c\_min, -1, (0, 0, 0), cv2.FILLED)  max\_area = area  max\_cnt = cnt  else:  c\_min = [cnt]  cv2.drawContours(img, c\_min, -1, (0, 0, 0), cv2.FILLED)  # c\_max.append(max\_cnt)  return max\_cnt  # cv2.drawContours(img, c\_max, -1, (0, 0, 255), thickness=3)  # cv2.imwrite("mask.png", img)  # cv2.imshow('mask', img)  # cv2.waitKey()  # def getNowChatName(func,img):  # point=getContoursRectPoint(func,img)  # for  def otherContours(contours, img):  c\_other = []  max\_area = 0  max\_cnt = 0  for i in range(len(contours)):  cnt = contours[i]  area = cv2.contourArea(contours[i])  # find max countour  if area > max\_area:  if max\_area != 0:  c\_min = [max\_cnt]  cv2.drawContours(img, c\_min, -1, (0, 0, 0), cv2.FILLED)  max\_area = area  max\_cnt = cnt  else:  c\_min = [cnt]  cv2.drawContours(img, c\_min, -1, (0, 0, 0), cv2.FILLED)  for i in range(len(contours)):  cnt = contours[i]  area = cv2.contourArea(contours[i])  if area != max\_area:  c\_other.append(cnt)  return c\_other  def getContoursRectPoint(func, img):  # step7：裁剪。box里保存的是绿色矩形区域四个顶点的坐标。我将按下图红色矩形所示裁剪昆虫图像。  # 找出四个顶点的x，y坐标的最大最小值。新图像的高=maxY-minY，宽=maxX-minX。  rect = cv2.minAreaRect(func()) # 得到最小外接矩形的（中心(x,y), (宽,高), 旋转角度）  return cv2.boxPoints(rect) # 通过box绘出矩形框  def getContoursImages(func, img):  # step7：裁剪。box里保存的是绿色矩形区域四个顶点的坐标。我将按下图红色矩形所示裁剪昆虫图像。  # 找出四个顶点的x，y坐标的最大最小值。新图像的高=maxY-minY，宽=maxX-minX。  rect = cv2.minAreaRect(func()) # 得到最小外接矩形的（中心(x,y), (宽,高), 旋转角度）  box = np.int0(getContoursRectPoint(func, img)) # 通过box绘出矩形框  Xs = [i[0] for i in box]  Ys = [i[1] for i in box]  x1 = min(Xs)  x2 = max(Xs)  y1 = min(Ys)  y2 = max(Ys)  hight = y2 - y1  width = x2 - x1  cropImg = img[y1:y1 + hight, x1:x1 + width]  cv2.imshow("cropImg", cropImg)  cv2.waitKey()  def getNewsNumber(img):  redNews = "ff3b30"  color1 = [0, 0, 0]  color2 = [0, 0, 0]  color2[0], color2[1], color2[2] = changeColor(redNews)  height = WeChatPos["EndY"] - WeChatPos["StartY"]  width = WeChatPos["EndX"] - WeChatPos["StartX"]  for x in range(WeChatPos["StartX"], WeChatPos["EndX"]):  for y in range(WeChatPos["StartY"], WeChatPos["EndY"]):  color1[0], color1[1], color1[2] = pyautogui.pixel(x, y)  if euclidean(color1, color2) > 0.9:  for i in range(0, 11):  if height / 11 \* i < y < height / 11 \* (i + 1):  return i  if \_\_name\_\_ == '\_\_main\_\_':  openWechat()  time.sleep(1)  WeChatPos["startX"], WeChatPos["startY"], WeChatPos["EndX"], WeChatPos["EndY"] = getWeChatpos()  # color1[0], color1[1], color1[2] = pyautogui.pixel(WeChatPos["startX"] + 10, WeChatPos["startY"] + 10)  # color2[0], color2[1], color2[2] = pyautogui.pixel(WeChatPos["startX"] + 12, WeChatPos["startY"] + 10)  # print(euclidean(color1, color2))  pyautogui.screenshot("test.jpg")  img = cv2.imread("test.jpg")  gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)  contours, ha = getchatMyForeground()  cv2.drawContours(img, contours, -1, (0, 0, 255), 3)  cv2.imshow("img", img)  cv2.waitKey(0)  # if os.path.exists('test.db') is False:  # conn = sqlite3.connect("test.db")  # c = conn.cursor()  # c.execute('''create table friends(\_friendId integer primary key autoincrement,  # name text''')  # c.execute('''create table message(\_\_message\_Id integer primary key autoincrement,  # messageContent text,  # \_friendId integer,  # isWithdraw integer,  # messageTime text,''')  # c.close()  # conn.close()  print(BaiduOCR("test.png"))  # cv2.imshow("img", img)  # cv2.waitKey(0)  # time.sleep(0.5)  # pyautogui.press('up')  # time.sleep(0.5)  # pyautogui.press('up')  # pyautogui.doubleClick()  # pyautogui.keyDown('shift')  # pyautogui.keyUp('shift')  # pyautogui.hotkey('ctrl', 'c') |