Chatper13: Raspberry Pi sends email face recognition

The specific requirements of this project : Raspberry Pi start mail and OpenCV service. OpenCV real-time video surveillance of the camera, when the face is detected, the photo is taken, and the captured photo is sent to another mailbox by email in the form of a webpage picture.

The 2 method2 of python sends emails.
The first method is to enable the sendmail service on the Raspberry Pi. This method is more complicated, you need to set your own domain name, address, etc.,
The second method is to use SMTP service of other email service providers (qq, Netease 163, Google gmail) . This time we use smtp service of QQ email and a QQ mailbox as the sender of the mail.

First, we need to enter the QQ mailbox by the browser, find the "Settings" under the user name at the top of the page - "Account" - "SMTP Service" - "POP3 / SMTP Service" - "select open"

Figure 1-1 Enter setting of email

Figure 1-2 Switch language to English

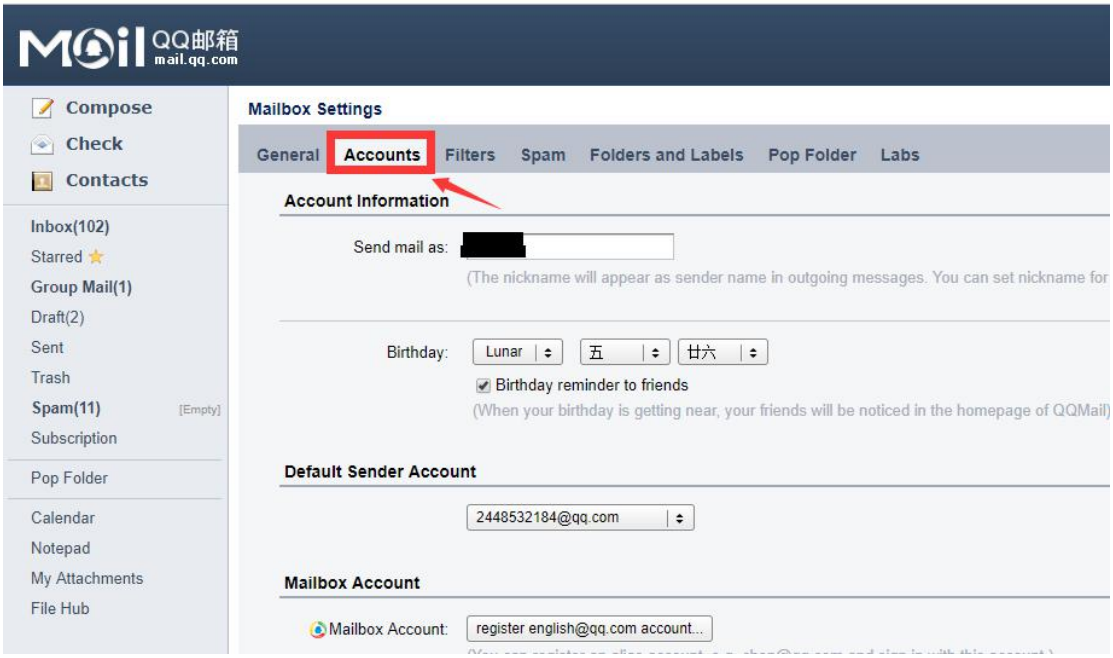

Figure 1-3 Select Accounts

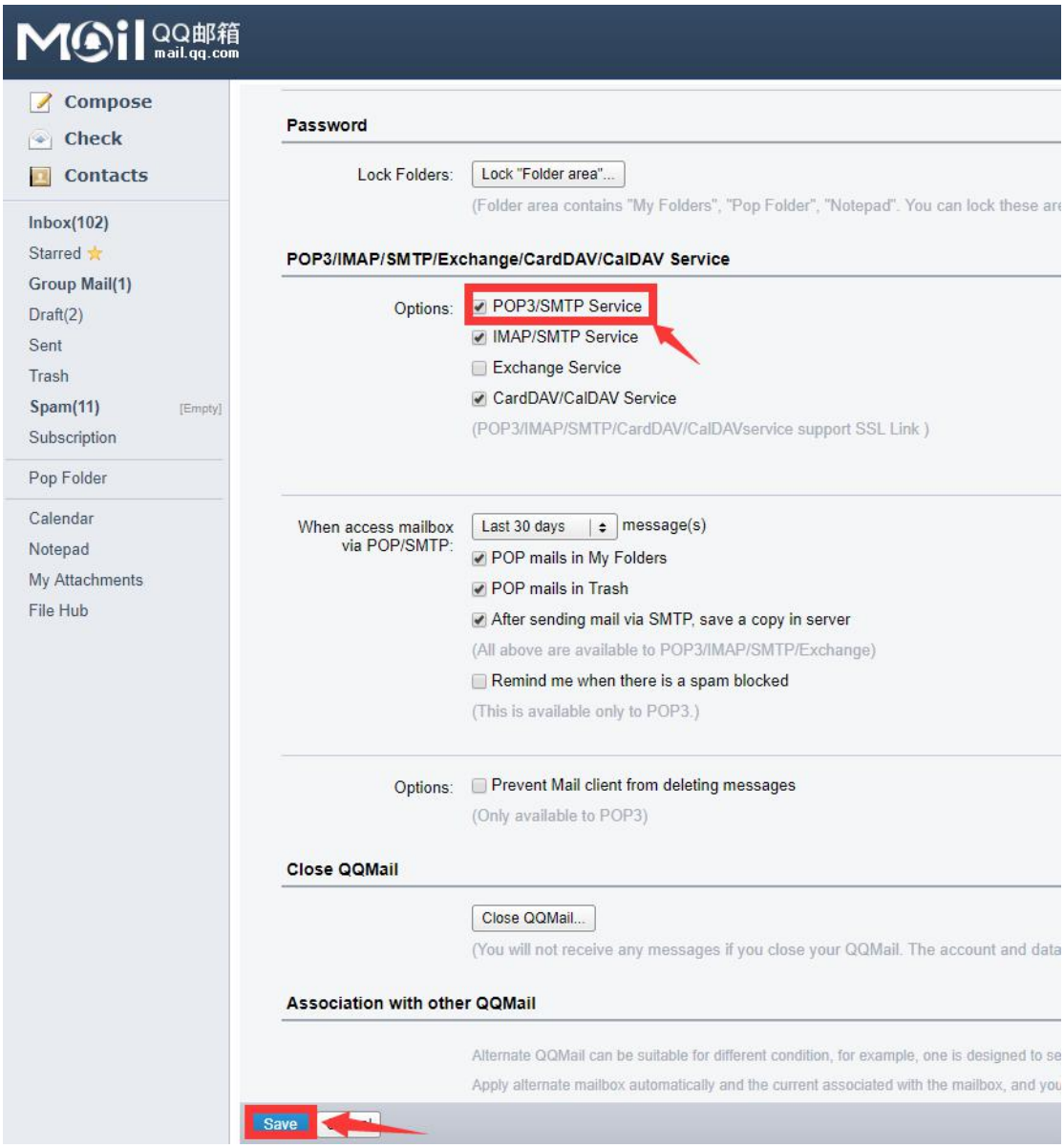

Figure 1-4 Start SMTP service

Note: The QQ mailbox SMTP service may be require you to set a separate password for your email, this is normal case, but this password should be kept in mind, so as not to affect the normal use of the mailbox.

After the completion, the system will automatically generate a set of keys. After the key is saved, our configuration of the STMP service is completed.

Figure 1-5 Get the key

First, we need to configure SMTP service.

```
sendDate=0
sender = “xxxxxxxxx@qq.com“          #your QQ email
password = “xxxxxxxxx”               #Key of STMP service in the QQ mailbox
receiver = "xxxxxxxxxx@xx.com"
#--------------------Mail service and port information---------------------#
smtp_server = "smtp.qq.com"
smtp_port = 465                      #STMP port of QQ
msg = MIMEMultipart('related')       #Use related to define the body of the
embedded resource
```

After the face is detected, the frame with the "face" is saved as a picture and saved locally. The file name is out.png. The path to the file needs to be reset by yourself. The existing out.png will be rewritten each time a new image is saved.

Then, setting the sender, receiver, and subject of the message.

Our requirement is to send a web page type message containing images, using the MIMEText() function. This method combines the text and images to be sent by html tags. Finally, add the generated web page to msg for sending.

```
img_file =
open('/home/pi/Adafruit_Python_PCA9685/face_warning_email/out.png',"rb")
                img_data = img_file.read()
                img_file.close()
                img = MIMEImage(img_data)
                img.add_header('Content-ID', '0')
                msg.attach(img)
                msg["From"] = Header("yaboom", "utf-8")
                msg["To"] = Header(receiver, "utf-8")
```

```
        msg["Subject"] = Header("face detected", "utf-8")
        #----------------Add the image as body content------------------
        message = MIMEText("<p>careful!!!!!</p><p>human approach your device</p><img src='cid:0'/>","html","utf-8")
    #plain Represent plain text
        msg.attach(message)
```

The source code of the program is located at:

/home/pi/yahboom/email_face/email_face.py

Before running the program, you need to modify the "sender", "password" and "receiver" in the file according to your actual situation.

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
    Created on Tue Nov  6 01:18:45 2018
    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file         email_face
    * @version      V1.0
    * @details
    * @par History

    @author: longfuSun
"""

from __future__ import division
import cv2

import time
import signal
import sys
reload(sys)
sys.setdefaultencoding('utf8')
#---------------------The above is the control coding method
import smtplib                                  #Import SMTP protocol package
from email.mime.text import MIMEText
from email.header import Header
from email.mime.multipart import MIMEMultipart  #Create a message body with multiple parts
from email.mime.base import MIMEBase
from email.mime.image import MIMEImage
import os.path                                  #Analysis path
from email import Encoders

sendDate=0
sender = "pi-motion@yahboom.com"                    #Send mailbox
password = "Yahboom0729"
receiver = "sdsunlongfu2017@163.com"                 #receive mailbox
#--------------------Mail service and port information----------------------
smtp_server = "smtp.yahboom.com"
smtp_port = 465                                 #STMP port of QQ
msg = MIMEMultipart('related')
```

```
smtp_port = 465                                    #STMP port of QQ
msg = MIMEMultipart('related')



cap = cv2.VideoCapture(0)
cap.set(3, 480)
cap.set(4, 320)
#Comparator xml location
face_cascade = cv2.CascadeClassifier( '123.xml' )

while True:
    ret,frame = cap.read()
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale( gray )
    max_face = 0
    value_x = 0
    font=cv2.FONT_HERSHEY_SIMPLEX
    #Record the time of shooting
    cv2.putText(frame,time.strftime("%Y-%m-%d %H:%M:%S",time.localtime()),(20,20),font,0.8,(255,255,255),1)
    if len(faces)>0:
        print('face found!')
        currentDate=time.time()

        for (x,y,w,h) in faces:
            cv2.rectangle(frame,(x,y),(x+h,y+w),(0,255,0),2)#0,255,0
            #max_face=w*h
            result = (x,y,w,h)
            x=result[0]
            y = result[1]

        #Avoid repeating shots in a short time, set the timestamp
        if currentDate-sendDate>600:
            cv2.imwrite("out.png",frame)


            img_file = open('out.png',"rb")
            img_data = img_file.read()
            img_file.close()
            img = MIMEImage(img_data)
```

```
            img.add_header('Content-ID', '0')    #The header of a normal attachment is different
            msg.attach(img)
            msg["From"] = Header("yaboom", "utf-8")
            msg["To"] = Header(receiver, "utf-8")
            msg["Subject"] = Header("face detected", "utf-8")

            message = MIMEText("<p>careful!!!!!</p><p>human approach your device</p><img src='cid:0'/>","html","utf-8")
            msg.attach(message)
            contype = 'application/octet-stream'
            maintype, subtype = contype.split('/', 1)
            try:
                #QQ must use .SMTP_SSL
                #Other servertry:.SMTP
                smtpObject = smtplib.SMTP_SSL(smtp_server , smtp_port)
                smtpObject.login(sender , password)
                #message.as_string() change the MIMEText object into a string
                smtpObject.sendmail(sender , [receiver] , msg.as_string())
                print ("发送成功")
            except smtplib.SMTPException :
                print ("发送失败！")
            smtpObject.quit()
            sendDate=time.time()
    cv2.imshow("capture", frame)
    if cv2.waitKey(1)==119:
        break

cap.release()
cv2.destroyAllWindows()
```

The effect of the operation as shown figure1-6 below.

Figure1-6