

**SINHGAD INSTITUTE OF TECHNOLOGY**

***DEPARTMENT OF COMPUTER ENGINEERING***

**(ACADEMIC YEAR 2018-19)**

**PROJECT REPORT ON**

**SECURITY CAMERA SYSTEM**

**Submitted by**

**Patil Smitesh Nitin(TEB-23)**

**Somvanshi Pruthveeraj Pradip(TEB-42)**

**Yamle Aniket Ramdas(TEB-58)**

**Under Guidance of**

**Dr. MS Chaudhari**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. no** | **Contents** | **Page no.** |
| 1 | ABSTRACT | 3 |
| 2 | HARDWARE AND SOFTWARE REQUIREMENTS | 4 |
| 3 | HARDWARE DESCRIPTION | 5 |
| 4 | SOFTWARE DESCRIPTION | 9 |
| 5 | INTERFACING AND EXECUTION STEPS | 11 |
| 6 | CODING OF THE PROJECT | 14 |
| 7 | OUTCOMES | 17 |
| 8 | REFERENCES | 18 |

**1.ABSTRACT**

Security is considered to be of concerns in today’s era and hence secure systems that monitor an environment as per user needs have become lucrative as well. This mini project of ours is aimed to deliver a robust, low cost security measures that can be used in a private space that detect any unwanted intrusion.

This system is designed such that it is highly mobile, easy to set and easy to maintain. It has almost no maintenance cost. We make use of many technologies like google cloud vision api for face detection, twilio for rest services ,embedded sensors like picamera and pir motion sensor . And thus we have tried to implement a working model of security services at a small scale that can be easy scalable to bigger systems if needed.

**2.REQUIREMENTS**

**1.Hardware Requirements**

* Raspberry Pi 3 b+
* PiCamera Module
* PIR Motion Sensor
* Female to Female Jumper Cables
* Power Supply (Power Bank)
* Display Monitor
* HDMI Connector for Display

**2.Software Requirements**

* Google Cloud Vision Api
* Twilio Python Library
* Os, Smtplib, Io, Email Libraries
* Noobs Operating System on Raspberry Pi
* Motion Sensor and Pi Camera Libraries

**3.Hardware Description**

**1.Raspberry Pi**

****

The **Raspberry Pi** is a series of small [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and in [developing countries](https://en.wikipedia.org/wiki/Developing_countries). The original model became far more popular than anticipated,selling outside its [target market](https://en.wikipedia.org/wiki/Target_market) for uses such as [robotics](https://en.wikipedia.org/wiki/Robotics). It does not include peripherals (such as [keyboards](https://en.wikipedia.org/wiki/Keyboard_(computing)) and [mice](https://en.wikipedia.org/wiki/Mouse_(computing))) and [cases](https://en.wikipedia.org/wiki/Computer_case). However, some accessories have been included in several official and unofficial bundles.

The organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation). After the Pi Model B was released, the Foundation set up Raspberry Pi Trading, with [Eben Upton](https://en.wikipedia.org/wiki/Eben_Upton) as CEO, to develop the third model, the B+. Raspberry Pi Trading is responsible for developing the technology while the Foundation is an educational charity to promote the teaching of basic computer science in schools and in developing countries.

According to the Raspberry Pi Foundation, more than 5 million Raspberry Pis were sold by February 2015, making it the best-selling [British computer](https://en.wikipedia.org/wiki/British_computer). By November 2016 they had sold 11 million units, and 12.5m by March 2017, making it the third best-selling "general purpose computer".In July 2017, sales reached nearly 15 million. In March 2018, sales reached 19 million

* **Specifications**

**Processor:** Broadcom BCM2837 Quad core (ARM Cortex-A53) SoC  
**Memory:** 1GB LPDDR2  
**Ethernet:** 10/100 BaseT socket  
**USB:** 4 x USB 2.0 ports  
**Connection:** GPIO Header 40-pin  
**HDMI:** Rev 1.3 and 1.4   
**Storage:** microSD card slot for loading operating system and data storage  
**Power:** USB connector for 5.1V / 2.5A dc

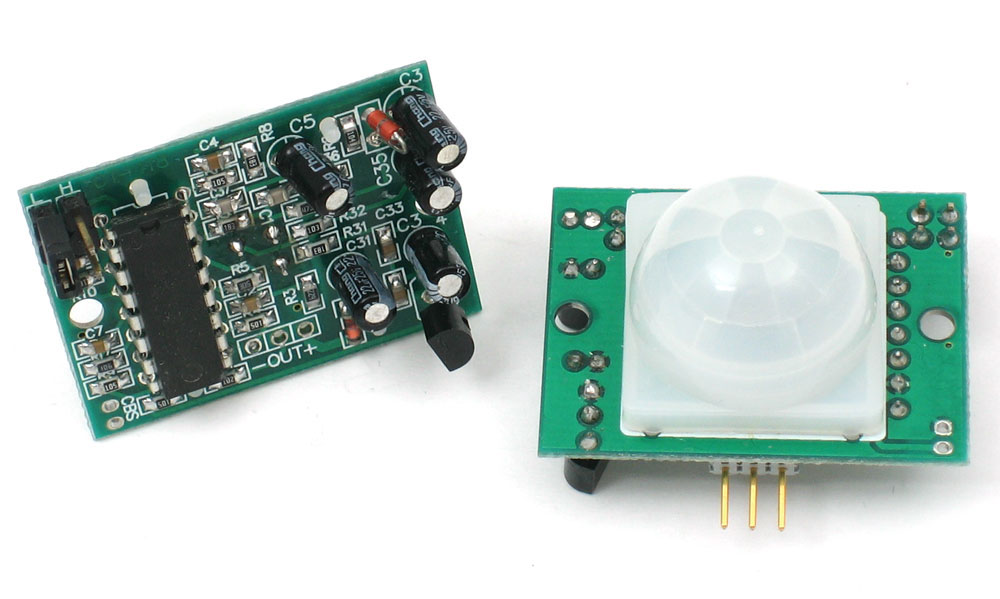
**2.PiCamera**



Many questions regarding picamera are based on misunderstandings of how the camera works. This chapter attempts to correct those misunderstandings and gives the reader a basic description of the operation of the camera. The chapter deliberately follows a [lie-to-children](https://en.wikipedia.org/wiki/Lie-to-children) model, presenting first a technically inaccurate but useful model of the camera’s operation, then refining it closer to the truth later on.

The Raspberry Pi Camera Module is an official product from the Raspberry Pi Foundation. The original 5-megapixel model was [released](https://www.raspberrypi.org/blog/camera-board-available-for-sale/) in 2013, and an 8-megapixel Camera Module v2 was [released](https://www.raspberrypi.org/blog/new-8-megapixel-camera-board-sale-25/) in 2016. For both iterations, there are visible light and infrared versions

**3.Motion Sensor**



PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

PIRs are basically made of a [pyroelectric sensor](http://en.wikipedia.org/wiki/Pyroelectric) (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.

Along with the pyroelectic sensor is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the [BISS0001 ("Micro Power PIR Motion Detector IC")](http://learn.adafruit.com/system/assets/assets/000/010/133/original/BISS0001.pdf), undoubtedly a very inexpensive chip. This chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor.

**4.Software Description**

**1.PiCamera Python Library**

This package provides a pure Python interface to the [Raspberry Pi](https://www.raspberrypi.org/) [camera](https://www.raspberrypi.org/learning/getting-started-with-picamera/) module for Python 2.7 (or above) or Python 3.2 (or above).

# Basic Functions

The following recipes should be reasonably accessible to Python programmers of all skill levels. Please feel free to suggest enhancements or additional recipes.

## 1.Capturing to a file

Capturing an image to a file is as simple as specifying the name of the file as the output of whatever [capture()](https://picamera.readthedocs.io/en/release-1.13/api_camera.html#picamera.PiCamera.capture) method you require:

from time import sleep

from picamera import PiCamera

camera = PiCamera()

camera.resolution = (1024, 768)

camera.start\_preview()

# Camera warm-up time

sleep(2)

camera.capture('foo.jpg')

## 2. Capturing to a stream

Capturing an image to a file-like object (a [socket()](https://docs.python.org/3.4/library/socket.html#socket.socket), a [io.BytesIO](https://docs.python.org/3.4/library/io.html#io.BytesIO) stream, an existing open file object, etc.) is as simple as specifying that object as the output of whatever [capture()](https://picamera.readthedocs.io/en/release-1.13/api_camera.html#picamera.PiCamera.capture) method you’re using:

from io import BytesIO

from time import sleep

from picamera import PiCamera

# Create an in-memory stream

my\_stream = BytesIO()

camera = PiCamera()

camera.start\_preview()

# Camera warm-up time

sleep(2)

camera.capture(my\_stream, 'jpeg')

**2.Twilio**

If you have used Lyft, Airbnb, or Netflix, you have probably used Twilio. In fact, Twilio powers communications for over 40,000 businesses around the world.

So what is Twilio? Simply put, Twilio is a developer platform for communications. Software teams use Twilio APIs to add capabilities like voice, video, and messaging to their applications. This enables businesses to provide the right communications experience for their customers. Behind Twilio APIs is a Super Network, a software layer that connects and optimizes communications networks around the world. This is what allows your users to reliably call and message anyone anywhere.

With Twilio, you can reach customers in the ways they prefer, and engage with them effectively using context related to that interaction. As customer experience can increasingly make or break your brand, [**programmable communications**](https://www.twilio.com/learn/twilio-101/why-businesses-need-programmable-communications) has become more crucial than ever to the success of businesses today.

## 1.Install the Library

The easiest way to install the library is from PyPi using [pip](http://www.pip-installer.org/en/latest/), a package manager for Python. Simply run this in the terminal:

pip install twilio

Python Code for Sending Message

from twilio.rest import Client  
  
# Your Account SID from twilio.com/console  
account\_sid = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"  
# Your Auth Token from twilio.com/console  
auth\_token = "your\_auth\_token"  
  
client = Client(account\_sid, auth\_token)  
  
message = client.messages.create(  
 to="+15558675309",   
 from\_="+15017250604",  
 body="Hello from Python!")  
  
print(message.sid)

**6.Pi to Laptop Connection**

**1.Connect Ethernet rj45 cat6 cable to pi and laptop**

**2.Install Advance IP Scanner,Putty,VNC Viewer**

**3.Open cmd in windows and give ipconfig**

**4.Copy Ethernet to Ethernet IPv4 address**

**5.In Advance IP Scanner input that ip as ex 192.168.0.0-255**

**6.Find Raspi Ip**

**7.Input it in Putty -> Hostname**

**8.Login as username -> pi with password ->raspberry**

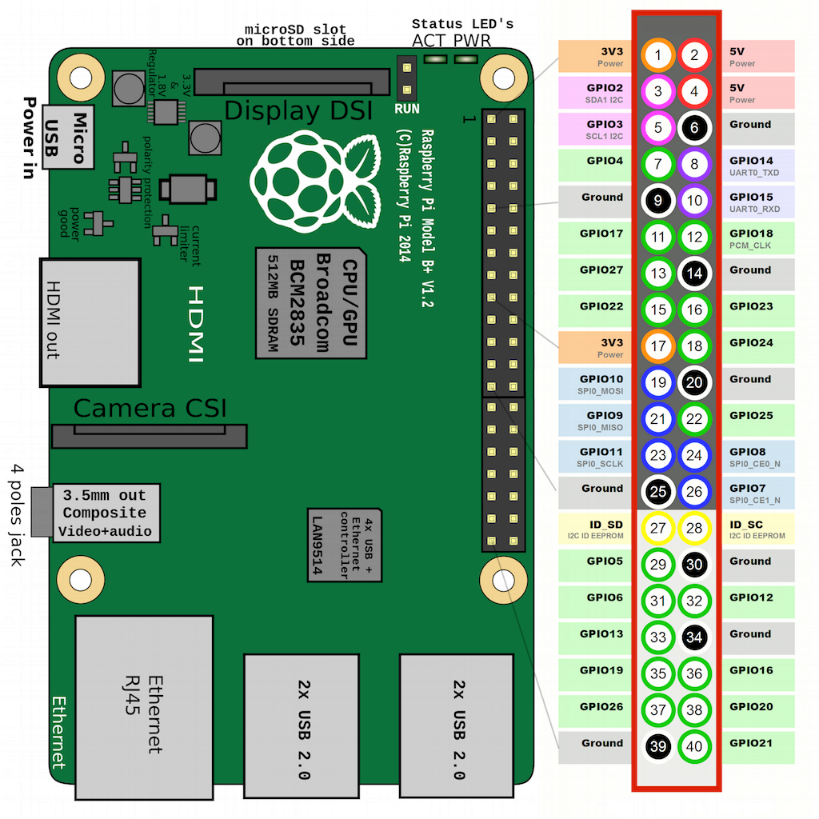
**9.Install tightvncserver->sudo apt-get install tightvncserver**

**10.Then input IP of raspberry in VNC Viewer as ex->192.168.0.1:1**

**5.Workflow and Execution Steps**

* **Step 1 :** Insert your picamera on picamera receiver just behind the Ethernet port . Make sure that sunny side of the picamera is facing in a opposite direction of the Ethernet port
* **Step 2 :** Connect motion sensor to your raspberry pi through gpio pins , by using female to female jumper cables. Connect the ground of motion sensor to GPIO pin number 6 , power to pin number 2 and output pin to 23





**Step 3 :** Boot up your raspberry pi and goto terminal and type : sudo apt update

sudo apt full-upgrade

sudo raspi-config

Select interfacing options and enable camera and reboot your device

* **Step 4 :** Now we need to set up google cloud vision api and the twilio rest services .For setting up twilio services register and create an account on twilio and then install twilio python library by following command

**sudo pip install twilio**

This will install twilio in your system. After registering for twilio you will get your mobile number and authorization tokens keep a note of them

* **Step 5 :** For cloud services you need to register for google cloud goto cloud.google.com and register after registering goto billing and enter details then in the api section go and enable google-cloud-vision api afterwards goto credentials and create a new service account and generate a json key that will be saved on your system keep a note of this key
* **Step 6 :** Now on your raspberry pi run command

sudo pip install --upgrade pip

sudo apt-get install libjpeg8-dev

sudo pip install google-cloud-vision

sudo pip install --upgrade google-api-python-client

sudo pip install --upgrade Pillow

sudo apt-get install python-picamera

* **Step 7 :** You need to set path to the json file that can be done in following steps :

export GOOGLE\_APPLICATION\_CREDENTIALS=filename.json

sudo nano ~/.bashrc

GOOGLE\_APPLICATION\_CREDENTIALS="/home/pi/vision123.json"

* **Step 8 :** You need to set your email details for sending an email do the following steps :

sudo apt-get install ssmtp

sudo apt-get install mailutils

sudo nano /etc/ssmtp/ssmtp.conf

* **Step 9 :** Edit the file as :

root=smitesh22@gmail.com

mailhub=smtp.gmail.com:587

hostname=raspberrypi

AuthUser=smitesh22@gmail.com

AuthPass=YourEmailPassword

FromLineOverride=YES

UseSTARTTLS=YES

UseTLS=YES

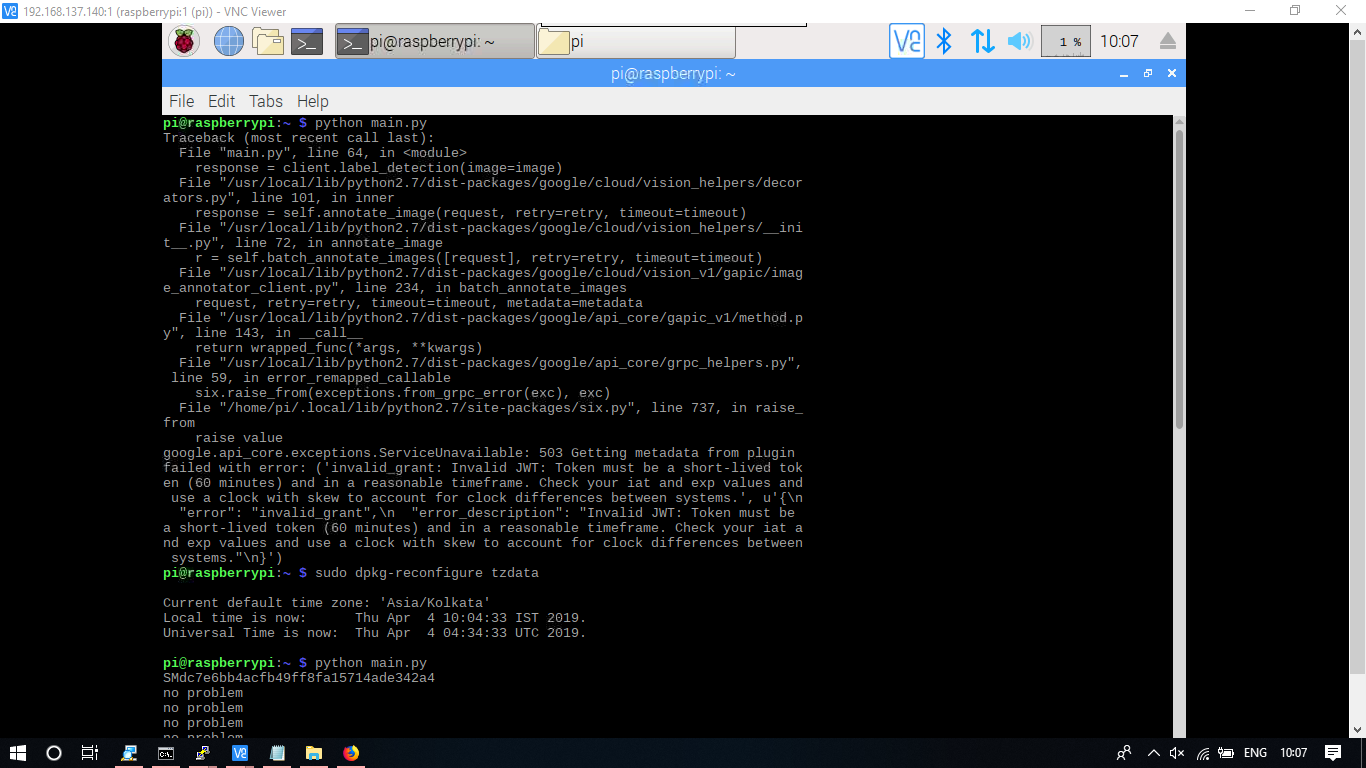
* **Step 10 :** Test if smtp is working or not by running :

**echo "Hello smitesh" | mail -s "Testing..."** [**smitesh22@gmail.com**](mailto:smitesh22@gmail.com)

* **Step 11 :** Now your have done with the interfacing part and you now need to execute the code that is given in next section
* **Issues**

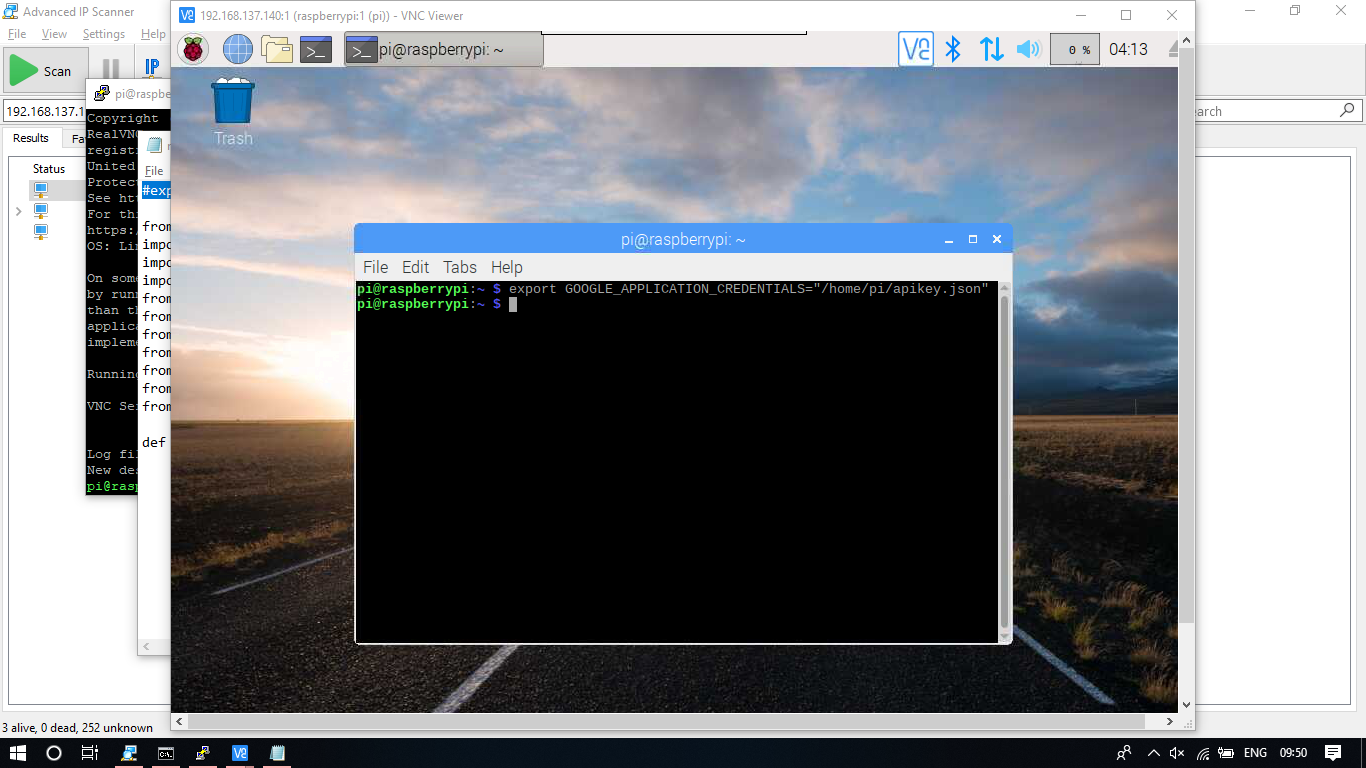
1. TimeZones : While running goole cloud vision api some problems may occur if timezones of your pi and google server are not in sync this can be done by running

sudo dpkg-reconfigure tzdata



1. Setting Enviroment Variable : Cloud vision requires us to set a path to the .json key to access the cloud api this can be done by running

export GOOGLE\_APPLICATION\_CREDENTIALS="/home/pi/apikey.json"



**6. Coding of the project**

**Main.py file**

**from gpiozero import MotionSensor**

**import os**

**import smtplib**

**import io**

**from email.mime.multipart import MIMEMultipart**

**from email.mime.text import MIMEText**

**from email.mime.base import MIMEBase**

**from email import encoders**

**from twilio.rest import Client**

**from google.cloud import vision**

**from google.cloud.vision import types**

**def SendMail():**

**fromaddr = "smitesh22@gmail.com"**

**toaddr = "smitesh4250@gmail.com"**

**msg = MIMEMultipart()**

**msg['From'] = fromaddr**

**msg['To'] = toaddr**

**msg['Subject'] = "Intrusion!!!"**

**body = "Alert Intrusion Detected!!"**

**msg.attach(MIMEText(body, 'plain'))**

**filename = "abc.jpg"**

**attachment = open("/home/pi/abc.jpg", "rb")**

**p = MIMEBase('application', 'octet-stream')**

**p.set\_payload((attachment).read())**

**encoders.encode\_base64(p)**

**p.add\_header('Content-Disposition', "attachment; filename= %s" % filename)**

**msg.attach(p)**

**s = smtplib.SMTP('smtp.gmail.com', 587)**

**s.starttls()**

**s.login(fromaddr, "Howudoin?@123")**

**text = msg.as\_string()**

**s.sendmail(fromaddr, toaddr, text)**

**s.quit()**

**pir = MotionSensor(23)**

**while True:**

**pir.wait\_for\_motion()**

**os.system("raspistill -o abc.jpg")**

**client = vision.ImageAnnotatorClient()**

**# The name of the image file to annotate**

**file\_name = os.path.join(**

**os.path.dirname(\_\_file\_\_),**

**'abc.jpg')**

**# Loads the image into memory**

**with io.open(file\_name, 'rb') as image\_file:**

**content = image\_file.read()**

**image = types.Image(content=content)**

**# Performs detection on the image file**

**response = client.label\_detection(image=image)**

**labels = response.label\_annotations**

**for label in labels:**

**if str(label.description)=="Face":**

**SendMail()**

**os.system(“python3 sms.py”)**

**2.SMS.py**

**from twilio.rest import Client**

**account\_sid = 'ACb484601b74bcc508fa5e63e71445f6d8'**

**auth\_token = 'e6a78b05f26df690e91a18d6e8d82191'**

**client = Client(account\_sid, auth\_token)**

**message = client.messages \**

**.create(**

**body="Alert Check Mail",**

**from\_='+12029198401',**

**to='+919011535518'**

**)**

**print(message.sid)**

**Execution :**

On Terminal :

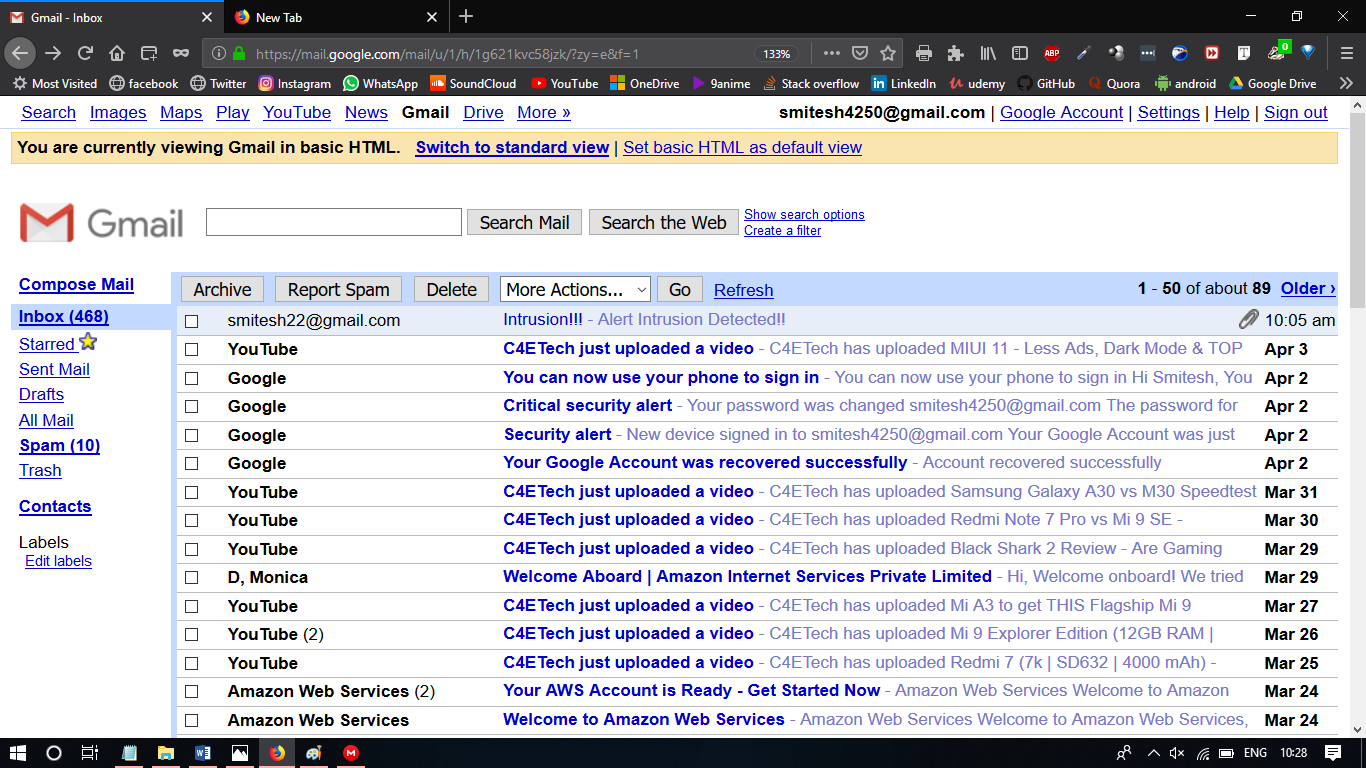
**export GOOGLE\_APPLICATION\_CREDENTIALS="/home/pi/apikey.json"**

**python main.py**

Note : Our google cloud vision api uses python 2 but twilio requires python 3 to run hence we cannot execute them in one file hence we created a separate SMS.py file that is called in out main python code

**7.Outcomes**

1. You will get a email on specificed email with the image of the intruder detected



2.You will also get a alert on your mobile device that has been configured by using twilio

\

**8.References**

* Raspberry Pi : <https://www.raspberrypi.org/documentation/>
* Raspberry Pi Installation Guides : <https://www.raspberrypi.org/documentation/installation/noobs.md>
* PiCamera Hardware : <https://picamera.readthedocs.io/en/release-1.13/fov.html>
* PIR Sensor : <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview>
* PiCamera Library : <https://picamera.readthedocs.io/en/release-1.13/>
* Twilio : <https://www.twilio.com/docs/libraries/python>
* Google Cloud Vision : <https://cloud.google.com/vision/docs/>
* Pip : <https://pip.pypa.io/en/stable/>
* Python : <https://docs.python.org/3/>