A
Project Report
on
"Surveillance updates on Email using Raspberry Pi"

Prepared by

Amartya Singh [17EC088]
Rahul Thakkar [17EC094]

Under the guidance of

Dr. Sagar B. Patel
Prof. Dhara M. Patel

Submitted to

Charotar University of Science & Technology for Partial Fulfillment of the Requirements for the Degree of Bachelor of Technology in Electronics & Communication

EC 346 – Group Project-I

of 5th Semester of B.Tech

Submitted at

DEPARTMENT OF ELECTRONICS & COMMUNICATION

Faculty of Technology & Engineering, CHARUSAT

Chandubhai S Patel Institute of Technology

At: Changa, Dist: Anand – 388421

June-November 2019

CERTIFICATE

This is to certify that the report entitled "**Surveillance updates on Email using Raspberry Pi**" is a bonafide work carried out by **Amartya Singh** and **Rahul Thakkar** under the guidance and supervision of **Dr. Sagar B. Patel** and **Prof. Dhara M. Patel** for the subject **Group Project-I (EC346)** of **5th Semester of Bachelor of Technology in Electronics & Communication at Faculty of Technology & Engineering (C.S.P.I.T.) – CHARUSAT, Gujarat.**

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the Subject specified for 5th semester of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under the supervision of:

Dr. Sagar B. Patel
Assistant Professor
Department of Electronics & Communication
C.S.P.I.T., CHARUSAT-Changa.

Prof. Dhara M. Patel
Assistant Professor
Department of Electronics & Communication
C.S.P.I.T., CHARUSAT-Changa.

Dr. Trushit K. Upadhyaya
Head of Department,
Department of Electronics & Communication
C.S.P.I.T., CHARUSAT- Changa, Gujarat.

Chandubhai S Patel Institute of Technology (C.S.P.I.T.)

Faculty of Technology & Engineering, CHARUSAT

At: Changa, Ta. Petlad, Dist. Anand, Gujarat - 388421

ABSTRACT

This security camera on detecting an object will send an Email to the administrator with the pictures of the detected object. Administrator can also watch live streaming of camera from any device that is capable of running any browser.

Features:

✓ Object Detection
✓ Sends an Email with an image of detected object
✓ Live Streaming on any device with a browse

Application:

✓ Intruder / Pet / Baby monitoring system
✓ Wildlife photography

Advantages:

✓ No need of continuous monitoring
✓ On detecting Object pictures are provided on Email
✓ Live Streaming on any device

Disadvantages:

✓ Picture quality is not great
✓ Not good for low light conditions
✓ Latency & frame drops during live streaming

Skills Required:

Python, OpenCV, Raspberry PI, SMTP (Simple Mail Transfer Protocol) Configuration, IP configuration, Hardware Assembly & Casing.

Hardware Required:

Raspberry Pi 3B+, PI Camera, Pi Camera Cable, Micro-USB to USB cable, Casing, Power supply adapter, micro-SD card, SD card Adapter.

Software Required:

Open CV, Python, Raspbian OS, Any other OS like windows / ubuntu, Any browser, SMTP interface, IP configuration tool

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide **Dr. Sagar B. Patel** and **Prof. Dhara M. Patel** and coordinator of E&C department of CSPIT, **Dr. Trushit Upadhyaya**, for their exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by them time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Mentor **Prof. Jignesh J. Patoliya** for their cordial support, valuable information and guidance, which helped me in completing this task through various stages.

Amartya Singh (17EC088)
Rahul Thakkar (17EC094)
CSPIT, CHARUSAT

TABLE OF CONTENT

List of Figures

Abbreviations

US	Ultra Sonic
APP	Application [For Android Application]
SONAR	Sound Navigation and Ranging
RADAR	Radio Detection and Ranging
Email	Electronic Mail
URL	Uniform Resource Locator
MP	Mega Pixel
OS	Operating System
RAM	Random Access Memory
GPU	Graphics Processing Unit
CPU	Central Processing Unit
SMTP	Simple Mail Transfer Protocol

1: Introduction of Project

There are different types of surveillance system available like thermal camera, night vision camera, infrared camera, color camera, RADAR, SONAR & many other. This project falls under the category of color camera. A color camera surveillance system is the most preferred type as it makes user feel like he himself is there monitoring everything with his own eye & it is the most affordable type of surveillance available.

1.1: Improvements in available technology:

Currently in most cases a user has to continuously monitor the live video coming from camera to detect intruder. So, in this product we have removed compulsion of continuous monitoring by adding a feature which will Email the user with the photo of the intruder & then user can use the live stream feature to monitor what the intruder is doing.

1.2: Project Overview:

This project uses Raspberry Pi 3B+ & Pi Camera. On detecting any intruder, it will send an Email to the registered mail ID of the user. This Email will consist the picture of face of intruder. So, as the user got notified about the current situation. If the user thinks he need to keep an eye on the detected intruder then he can use the provided URL to start live stream video.

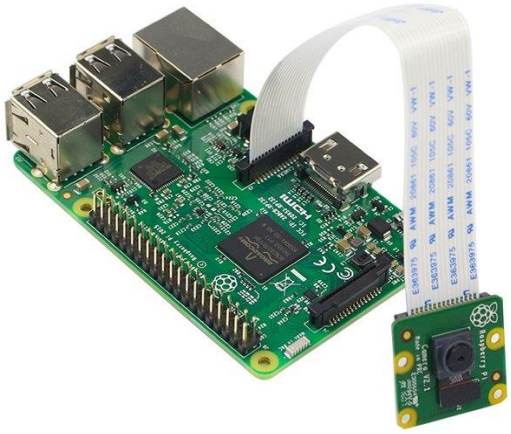

Figure 1: Raspberry Pi with camera module

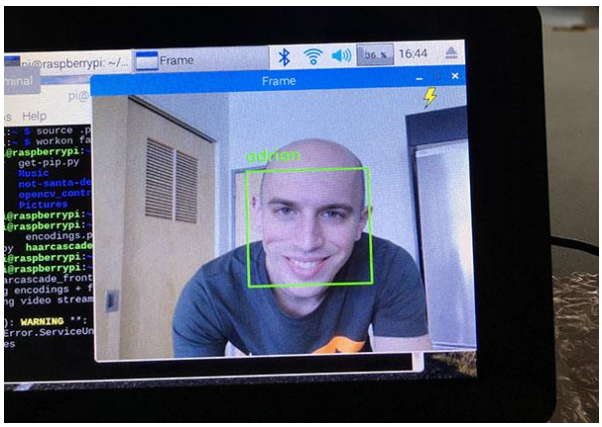

Figure 2: Live Stream

2: Project Description

This Project is using less hardware component but many software. All the function played by each hardware & software is described below:

2.1: Hardware Description:

Raspberry Pi 3B+:

It is the main processing unit of the project which is equipped with Raspbian Buster OS. All the processing of the video will be done on this onboard computer. All the programming is there on memory card which inserted in to this. So, basically this is the brain of our project. It is equipped with a dedicated slot to install the camera module.

Pi Camera Module:

It is installed in the port provided on raspberry pi specially dedicated to install camera. It is a color camera which will help raspberry pi to monitor a area & capture the required image. This camera is of 5MP. So, basically this will act as eye for our project.

2.2: Software Description:

Raspbian Buster OS:

This OS is designed specially for raspberry. This is a Linux based OS.

Open CV:

Open CV is open source library used for image processing. Many functions are prewritten in this library that are frequently used for image processing. It is also bundled with readymade model database for face & body detection.

Python:

This is the chosen programming language for our project because it is compatible with Open CV & preinstalled on Raspbian OS.

2.3: Block Diagram:

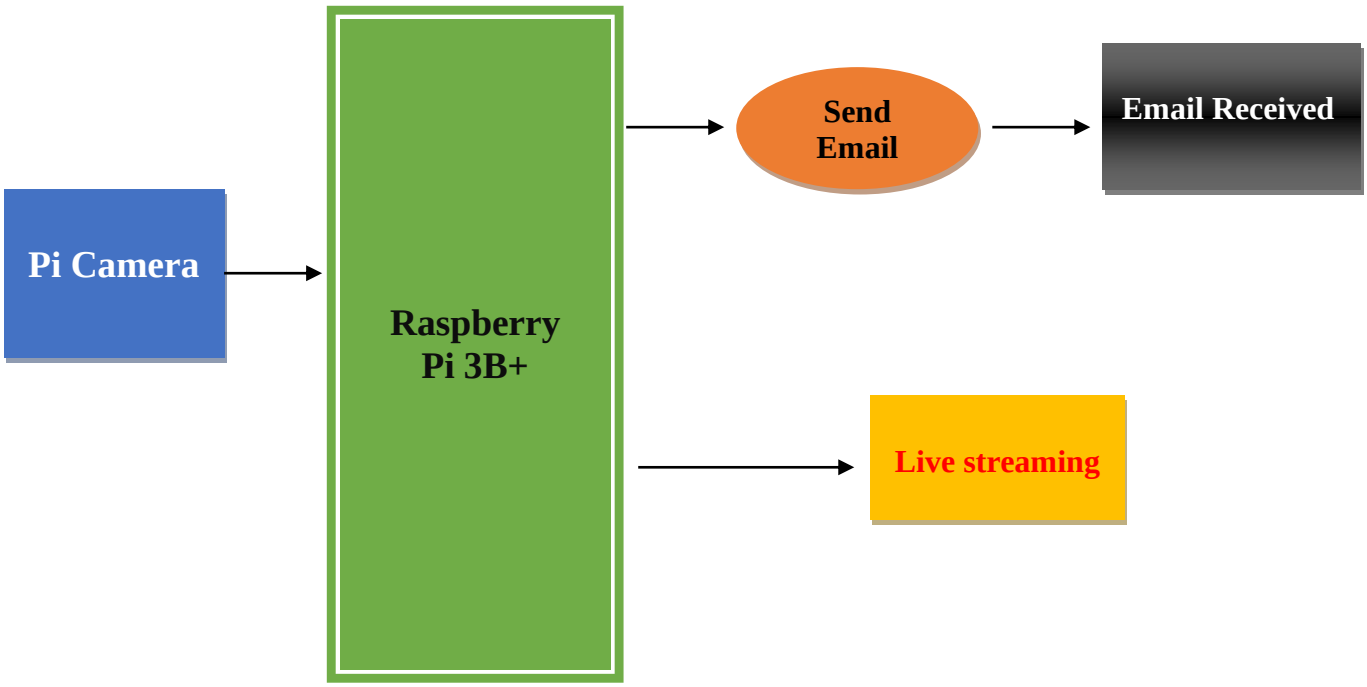

Figure 3: Block Diagram

2.4: Schematic Diagram:

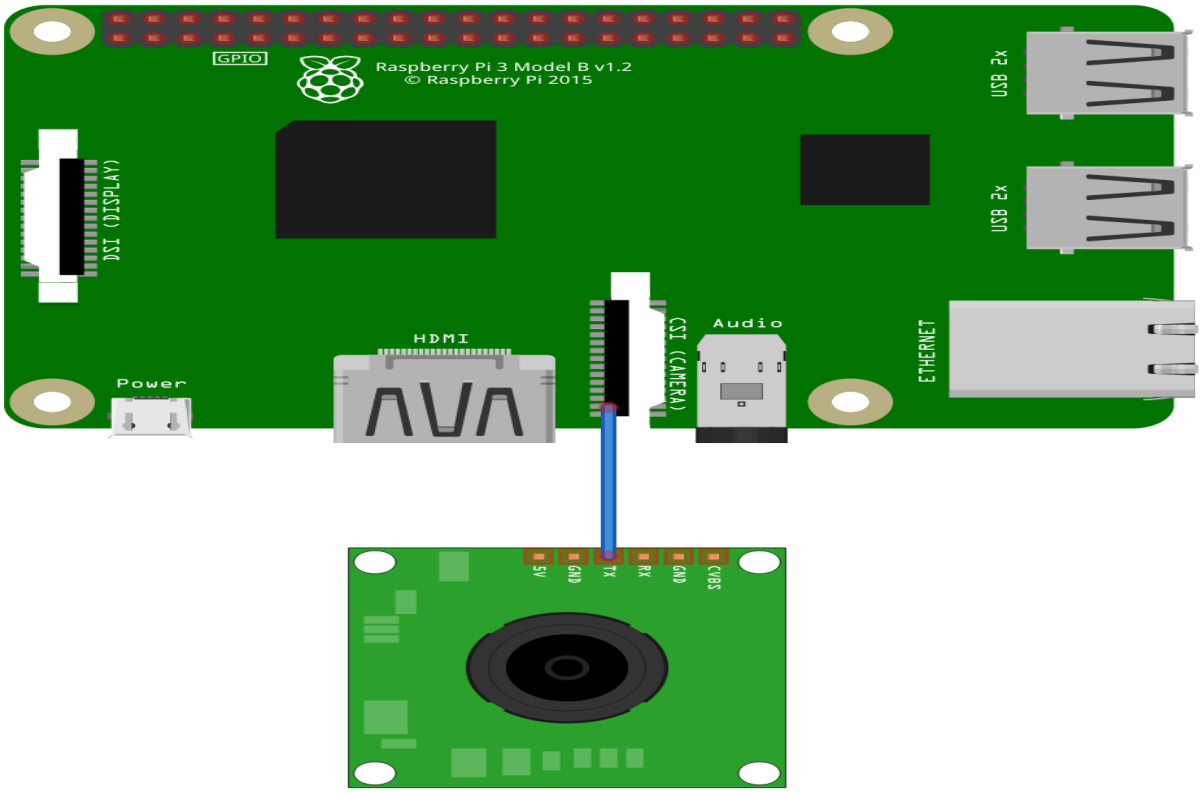

Figure 4: Schematic Diagram

3: Component Details

Here list of all components with their specification is provided. Data sheet of each & every component gives all the data of a component so please refer it.

3.1: Component List:

- Raspberry Pi
- 5V Power Supply
- Pi Camera
- Memory Card

3.2: Component Technical Specifications:

Raspberry Pi 3B+ Specifications

- SoC: Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
- GPU: Broadcom Videocore-IV
- RAM: 1GB LPDDR2 SDRAM
- Networking: Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
- Bluetooth: Bluetooth 4.2, Bluetooth Low Energy (BLE)
- Storage: Micro-SD
- GPIO: 40-pin GPIO header, populated
- Ports: HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
- Dimensions: 82mm x 56mm x 19.5mm, 50g

Pi Camera Specifications

- Resolution: 5 MP
- Lens Focus: Fixed Focus
- Image Size(Pixels): 2592Ã—1944
- Interface Type: CSI(Camera Serial Interface)
- Sensors: Omnivision 5647 fixed-focus
- Aperture: 2.9
- Focal Length: 3.29
- FOV: 72.4Â°
- Length (mm): 25
- Width (mm): 23
- Height (mm): 8
- Weight (gm): 3

4: Implementation of Project

The complete hardware setup should be done as per instruction given below because all the Python code has been written as per that configuration only. Copy the same code as provided because it will only work with the hardware setup we have done before. So, strictly follow the hardware & software setup instruction

4.1: Hardware Implementation:

- Connect 5V power supply to Raspberry Pi.
- Insert camera module in the camera port provided on Raspberry Pi

4.1: Software Implementation:

Step 1:

Create a folder named as smart-security-camera in the home directory of Raspbian OS. Then create a camera.py python file to operate on camera inside smart-security-camera folder.

camera.py

```
import cv2
import imutils
from imutils.video.pivideostream import PiVideoStream
import time
import numpy as np

class VideoCamera(object):
   def __init__(self, flip = False):
      self.vs = PiVideoStream().start()
      self.flip = flip
      time.sleep(2.0)

   def __del__(self):
      self.vs.stop()
   def flip_if_needed(self, frame):
      if self.flip:
```

```
            return np.flip(frame, 0)
        return frame

    def get_frame(self):
        frame = self.flip_if_needed(self.vs.read())
        ret, jpeg = cv2.imencode('.jpg', frame)
        return jpeg.tobytes()

    def get_object(self, classifier):
        found_objects = False
        frame = self.flip_if_needed(self.vs.read()).copy()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        objects = classifier.detectMultiScale(
            gray,
            scaleFactor=1.1,
            minNeighbors=5,
            minSize=(30, 30),
            flags=cv2.CASCADE_SCALE_IMAGE
        )

        if len(objects) > 0:
            found_objects = True

        # Draw a rectangle around the objects
        for (x, y, w, h) in objects:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

        ret, jpeg = cv2.imencode('.jpg', frame)
        return (jpeg.tobytes(), found_objects)
```

Step 2:

Now create a mail.py python file to send Email inside smart-security-camera folder.

mail.py

```
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText
from email.MIMEImage import MIMEImage

# Email you want to send the update from (only works with gmail)
fromEmail = 'camerap80@gmail.com'
```

```
fromEmailPassword = 'thegreatamartya'
# Email you want to send the update to
toEmail = 'realtechcorner@gmail.com'

def sendEmail(image):
        msgRoot = MIMEMultipart('related')
        msgRoot['Subject'] = 'Security Update'
        msgRoot['From'] = fromEmail
        msgRoot['To'] = toEmail
        msgRoot.preamble = 'Raspberry pi security camera update'

        msgAlternative = MIMEMultipart('alternative')
        msgRoot.attach(msgAlternative)
        msgText = MIMEText('Smart security cam found object')
        msgAlternative.attach(msgText)

        msgText = MIMEText('<img src="cid:image1">', 'html')
        msgAlternative.attach(msgText)

        msgImage = MIMEImage(image)
        msgImage.add_header('Content-ID', '<image1>')
        msgRoot.attach(msgImage)

        smtp = smtplib.SMTP('smtp.gmail.com', 587)
        smtp.starttls()
        smtp.login(fromEmail, fromEmailPassword)
        smtp.sendmail(fromEmail, toEmail, msgRoot.as_string())
        smtp.quit()
```

Step 3:

Now create a main.py python file inside smart-security-camera to do operations.

main.py

```
import cv2
import sys
from mail import sendEmail
from flask import Flask,render_template,Response
from camera import VideoCamera
from flask_basicauth import BasicAuth
import time
import threading
```

```
email_update_interval=600  # sends an email only once in this time interval
video_camera=VideoCamera(flip=True)  # creates a camera object, flip
vertically
object_classifier=cv2.CascadeClassifier("models/facial_recognition_model.xml"
)  # an opencv classifier

# App Globals (do not edit)
app=Flask(__name__)
app.config['BASIC_AUTH_USERNAME']='pi'
app.config['BASIC_AUTH_PASSWORD']='thegreatamartya'
app.config['BASIC_AUTH_FORCE']=True

basic_auth=BasicAuth(app)
last_epoch=0


def check_for_objects():
   global last_epoch
   while True:
      try:
         frame,found_obj=video_camera.get_object(object_classifier)
         if found_obj and (time.time() - last_epoch) > email_update_interval:
            last_epoch=time.time()
            print "Sending email..."
            sendEmail(frame)
            print "done!"
      except:
         print "Error sending email: ",sys.exc_info()[0]


@app.route('/')
@basic_auth.required
def index():
   return render_template('index.html')


def gen(camera):
   while True:
      frame=camera.get_frame()
      yield (b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')
```

```
@app.route('/video_feed')
def video_feed():
   return Response(gen(video_camera),
                   mimetype='multipart/x-mixed-replace; boundary=frame')


if __name__ == '__main__':
   t=threading.Thread(target=check_for_objects,args=())
   t.daemon=True
   t.start()
   app.run(host='0.0.0.0',debug=False)
```

Step 4:

To make these programs run automatically as we provide power supply to it. Go to terminal & follow these steps:

```
sudo nano /home/pi/.bashrc

//In this file add these 2 lines at the end:

cd smart-security-camera
python main.py

//press ctrl+x -> y -> Enter
// As this will run everytime you open a terminal so to close it press ctrl+c
```

5: Application of Project

- **Intruder Alert System**
  This technology can be used to secure an area where no person is allowed to enter. If anyone enters then the administrator will be notified and they can take appropriate actions.

- **Kids / Pets Monitoring System**
  This technology can be used to keep an eye on your kids if they are alone at home as it is equipped with live stream. So, all the actions in home can be enjoyed from anywhere in the world.

- **Wildlife Photography**
  This technology can be used for wildlife photography because a photographer has not to be present there continuously. As soon as an animal passes from the visible area it's picture will be clicked.

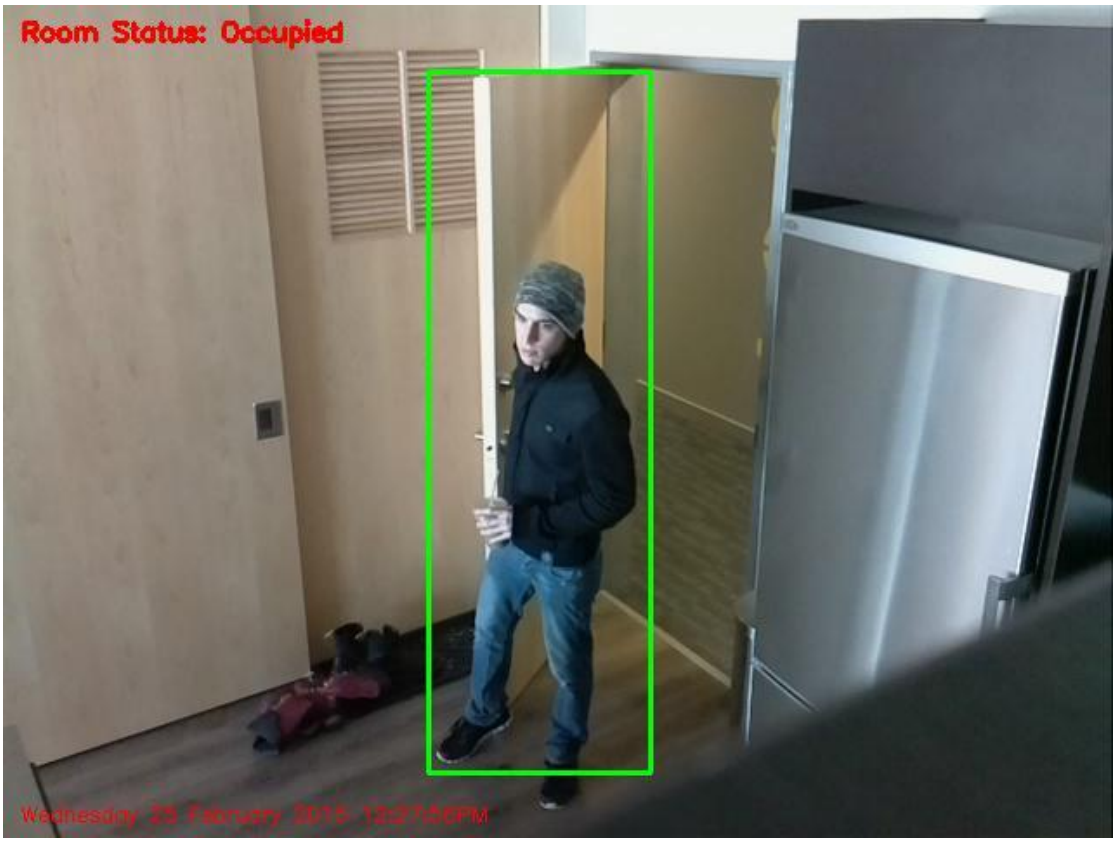

Figure 5: Intruder Detected

6: Conclusion

While working on this project, we learnt to interface Pi camera with Raspberry Pi. So, it makes easier to learn interfacing of other components with Raspberry Pi when needed in future. We also learnt to troubleshoot many issues related to hardware & software. Many facts came to be known related to all this component while working on this project.

While working on this project We have gained a lot of hands on experience with Python, SMTP, Open CV. We think that experience of working on Raspbian OS will help us in future to operate on another Linux based OS. We also learned to use Flask which converts our python code to a basic web application.

7: References

Special Thanks To:
https://www.hackster.io/hackershack/smart-security-camera-90d7bd

https://medium.com/@snipecharts/raspberry-pi-project-how-to-make-a-smart-security-camera-system-c26e63540828

GitHub Link:
https://github.com/HackerShackOfficial/Smart-Security-Camera

Data Sheets:
https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf

https://robu.in/product/raspberry-pi-camera-module/

Official Website:
https://www.raspberrypi.org/

https://opencv.org/

Open CV installation Guide:
https://medium.com/@patrick_ryan/building-opencv-4-10-on-raspian-buster-and-raspberry-pi4-64669bd2eb74