IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Real Time Home Safety and Surveillance using Raspberry Pi and Artificial Intelligence

Parasa Tarun Sreekar [1][0000-0002-1482-2326], Marri Rohit[2][0000-0002-4259-5706], Mudadla Srikanth[3][0000-0002-8775-2066], Musturu Harsha Vardhan[4][0000-0002-7852-8820], Shama[5][0000-0001-9705-6681] and Ch. Ravi Sankar[6]

[1,2,3,4,5] B. Tech (ECE) Students, School of Electrical and Electronics Engineering,
Lovely Professional University, Punjab, India.
[6] Professor and Mentor, School of Electrical and Electronics Engineering,
Lovely Professional University, Punjab, India.

Abstract. Home Safety and Surveillance systems using IoT, and motion detection has made Human lives more secure, peaceful, and less worrisome, whenever our homes are not under our surveillance. However, these systems could not properly differentiate between user and intruder under certain conditions and there were many other factors which causes these systems to send false alert to the users. This is due to their lack of Intelligence (inability to understand, reason, interact and compute) about the household environment. Here in this paper, to overcome these problems we developed a Home Safety and Surveillance system by inducing Artificial Intelligence (A.I) and IoT with the existing Home Surveillance system which uses Raspberry-Pi as its core along with other devices and sensor modules. The proposed system uses sensors data along with the Artificial Intelligence applications such as Object-Detection, Face-Detection, Face-Recognition, in addition to our developed algorithm. The developed system is intelligent in its own way. It can find the cause of motion, verify the lightning conditions in the house. when faced with an indecisive situation, it can interact with the user. It is capable enough to reduce the false alerts to a minimal amount. When faced with emergency situations, it will alert the users via Telegram Application along with live feed of situation inside the house. Artificial Intelligence and IoT are two most major technologies that brings comfort to Human lives. When combined, these technologies made the system much smarter, more intelligent and a lot more dependable/trustworthy.

Keywords: Smart Home Surveillance, Artificial Intelligence, Raspberry-Pi, Smart Home Safety, Fire detection, Gas leak detection, IoT, Telegram Bot, Real-Time Home surveillance, Live Streaming Using Raspberry-Pi.

1 Introduction

1.1 Background and Motivation

On one hand, when integrated with electronics and technology, IoT has supplied many benefits for humans in several aspects and one of them is in the field of home security, surveillance, and monitoring. The user can leave home and stay less worried about safety and security of their home as these systems would alert them in case of safety or security issues. Also, these systems can send image or video feed along with the alert. On the other hand, Artificial Intelligence is the most advanced field with vast boundaries. It has many functionalities and applications in several different fields. When fused together, IoT and A.I makes these Smart Home Safety, Security and Surveillance systems more dependable, trustworthy, and a lot more useful for Humans in their Daily Life. Developing a Home Safety and Surveillance system that is integrated with IoT and is fused with A.I with high reliability and trustworthiness which adds value and benefits to Human Life is the motivation of this paper.

1.2 Limitations of earlier works

"With installed motion detectors at home, you expect an instant response to intruders. If a thief steps over the threshold or breaks in through the window, an alarm will be triggered. And you can rest assured that any high-quality alarm device can manage this task. But responding to motion is only half the battle: an important quality of a detector is the ability to figure out what exactly is moving. It must quickly and accurately detect a person, while also ignoring the natural interference and pets, etc.," as mentioned in [8].

- In [1], the proposed system is capable of surveillance and transmission of image or video to smartphone, but it cannot detect the cause of motion.
- In [2], the proposed system detects the movement of objects using pir sensor, captures the live image and directly applies face recognition. If the face is not matched, the system sends alert messages to user. The draw backs are, this type of system sends false messages in case of animals triggering motion or in case of person face covered with mask/cap/scarf. In both the cases the face recognition identifies the object as an unauthorized person.
- In [3], the proposed system uses pir sensor for motion detection and CCTV live feed for object detection when a person is identified by the object detection, it alerts the user. The drawback of this system is that it could only identify objects and decide if it is a person or not, but it neither can detect the visibility of the face of the person nor it can recognize the face of the person. These types of systems alert the user even when the user moves in the area.
- In [4], the proposed system alerts the user as the pir sensor detects the motion and it will also send captured photos and videos, but it cannot detect the cause of the motion which increases the false alerts. Also, it cannot send the live stream of the situation on user's smart phone.
- In [5], the proposed cannot detect the cause of the motion, cannot identify, and recognize the face of persons and because of this, the system could send false alerts to user in a lot of situations. This greatly reduces the reliability and credibility of the system. Also, it cannot show the live situation to the user in case of intrusion.

To summarize, in the earlier phase, the home security and surveillance systems mostly consisted of microcontrollers or microprocessors, sensors, actuators that are connected to internet and can communicate among themselves without user interaction. When needed, they would alert the user on their smartphones/computers/laptops. However, these systems did not have any intelligence and because of this they falsely alert the user under many circumstances. For example, a system that alerts the user whenever a motion is detected in the house, it alerts him even if the motion is due to the pets in the house. When the system is not turned off, and the user is present in the house, it would alert him about his motion. There were a lot more different scenarios where these systems are sending false alerts to their users. This raised a question about their reliability and trustworthiness. In the later phase to fill this gap came the computer vision which helped these systems in recognizing the face of the person as soon as motion is detected, and this type of approaches reduced the false alerts to some extent. But these systems also could not identify the cause of motion and there were many scenarios where the users face is not properly visible for recognition. For example, the user is wearing a mask and entered the house, or when pets/animals cause the motion etc., In these cases, or similar ones, these systems alert the users falsely as they could not properly recognize the face and identify the motion caused object as unauthorized person. To overcome such scenarios/situations with the help of Artificial Intelligence is the objective of this paper.

1.3 Proposed Approach

The proposed approach to reduce the false alerts/notifications from the system is: As soon as a motion is detected, the system first checks the lighting conditions in the room to ensure clear view if the room is dark, it turns the light and will continue further else it will directly continue to the next step. In the second step, the real-time video feed from web-camera is passed as frames of images to detect the cause of motion using the object detection algorithm it will continue to the third step only if the motion is due to a person, if the motion is not because of the person, it will not continue further and goes back to check motion detection. In the third step the system tries to verify the visibility of face of the person who triggered the motion sensor. If the face is properly visible for recognition, it will proceed to the next step else the system communicates with the person and asks the person to remove the object (like half/full mask, cap, scarf etc.,) on his/her face and after few seconds, it will again try to verify the visibility of the face. If the person removes the object on the face and passes the visibility check, it will go ahead to the fourth step else it will alert the users on the telegram application. The alert message will be about intruder and will consist of a live streaming link. By clicking on it, the user can see the household situation. In the fourth step the system passes the image frames of live video feed to face recognition algorithm and recognizes the face of the person that triggered the motion sensor. If the person/persons face/faces is in the list of known people, it will again start object detection if the object is person, it will wait for given amount of time (1 or 3 or 5 minutes) and again identify the object in the room this process goes on looping with given time (1 or 3 or 5 minutes) interval until the object detection gives the output of not a person. After this, it again goes back and wait for motion detection. If there were both recognized and unrecognized faces, the system sends a poll message to the users on telegram. The message will be 'There were unknown persons along with you, would you like to record the situation?' If the reply is 'yes', it will Start recording and it will again start object detection if the object is person, it will wait for given amount of time (1 or 3 or 5 minutes) and again identify the object in the room this process goes on looping with given time (1 or 3 or 5 minutes) interval until the object detection gives output of not a person. Once the output is not a person or if it receives 'stop recording' message from user, it will stop recording. If there were only unrecognized/unknown faces, the system will immediately send intruder alert message to the users telegram along with live streaming link. Apart from this, the system will also alert the users on their telegram along with live streaming link whenever there was a gas leak, or a fire is detected. A several other functions (i.e., responding to the voice commands and responding to bot function commands (for recording or household electrical appliances control) from user can also be added to the system to make the system more useful.

2 Methodology and Procedure

2.1 Hardware Components

To build the proposed system we used Raspberry-Pi 4B with 2GB RAM. This is used as the core of the system in which all the computations take place, and all the sensors are controlled along with collection of their information.

Raspberry-Pi: "The Raspberry Pi is a very cheap computer that runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, allowing you to control electronic components for physical computing and explore the Internet of Things (IoT)." [11]
The various components of Raspberry- Pi used in building the proposed model are:

- *"SD Card Slot*: This is used to install OS/booting/long term storage. The total memory of the SD card is about 8GB." [3]
- *USB Ports*: These ports can be used to connect different devices to the Pi. 700mA at 5A.
- *Audio Jack*: Digital audio is obtained if a set of speakers are connected.

- *"Ethernet Port*: This is used to connect the Pi to the Internet. It also plays a key role in updating and getting new software easier." [3]
- *"GPIO Pins*: Allows us to control and interact with devices connected to them." [3]
- *CSI Port*: This is used to connect Raspberry-Pi camera module.
- *Type-c port*: This is the port to which the power supply adapter is connected to the Pi.

Raspberry-Pi Camera Module (Rev 1.3): This is a camera module that can be connected to the CSI port on raspberry pi. We are using this camera module for continuous live streaming to a HTML web page that is accessible to the user.

USB Web Cam (Logitech c170): This is the web cam that can be connected to any of the 4 USB ports of Raspberry-Pi. We are using this to perform the object detection, face detection and face recognition.

PIR Motion Detection Sensor Module (HC-SR501): Passive Infrared sensor is used to sense movement in the room. It is interfaced with Raspberry Pi GPIO pins, as shown in the circuit diagram. It gives the output in digital form.

LDR Light Sensor Module: This is used to detect the presence of light in the room. It is interfaced with Raspberry-Pi GPIO pins, as shown in the circuit diagram. It gives the output in digital form.

Flame Sensor Module: This is the flame sensor infrared receiver Ignition Source detection module. It is used to detect accidental fire in the house. It is interfaced with Raspberry-Pi GPIO pins, as shown in the circuit diagram. It gives the output in digital form.

Gas Sensor Module (MQ-6): This is gas leak detection sensor that has high sensitivity to LPG, Propane, Isobutane and has good sensitivity to Combustible gas in wide range. It is used to detect the accidental gas leak in the house. It is interfaced with Raspberry-Pi GPIO pins, as shown in the circuit diagram. It gives the output in digital form.

2-Channel 5V Relay Module: This is a 5V,10A 2-Channel relay module that is used to control various electrical appliances with large current. It can be directly controlled with the logic signals from the Raspberry-Pi. It is interfaced with Raspberry-Pi GPIO pins, as shown in the circuit diagram. It takes the input in digital form.

5W Light Bulb: This is connected to relay module and is controlled using Raspberry-Pi. It is used to convert unfavorable lighting conditions in the house to favorable ones. It is interfaced with relay module, as shown in the circuit diagram.

Speakers: Speakers are used for getting voice output. With the help of python modules, we can make the raspberry-pi speak. The input will be the text we want to hear from Pi and the output will be from the spoken form of text from the speakers connected to Raspberry-Pi.

2.2 Software Requirements

Python Programming Language: To build the proposed model, Python is used as the main programming language along with necessary libraries and modules. Python is used for many purposes/functions in the working of this model. It is used to enable communication (i.e., controlling, commanding, passing instructions) between raspberry-pi and sensor modules, cameras, webpages, websites, telegram bot etc., It is used for training and executing the object detection, face detection and face recognition algorithms on the pi.

Jupyter-Notebook: "Jupyter Notebook provides an easy-to-use, interactive environment across many programming languages that doesn't only work as an IDE, but also as a presentation or education tool." [12]

MobaXterm: “MobaXterm provides all the important remote network tools (SSH, RDP, X11, SFTP, FTP, etc.,) to Windows desktop, in a single portable exe file which works out of the box.” [13]

RPi-Cam-Web-Interface: “RPi Cam Web Interface is a web interface for the Raspberry Pi Camera module. It can be used for a wide variety of applications including surveillance, dvr recording and time lapse photography.” [14]

2.3 Block Diagram

‘If the diagram is not clearly visible, please click on the image to download/view the Block Diagram from Google Drive.’

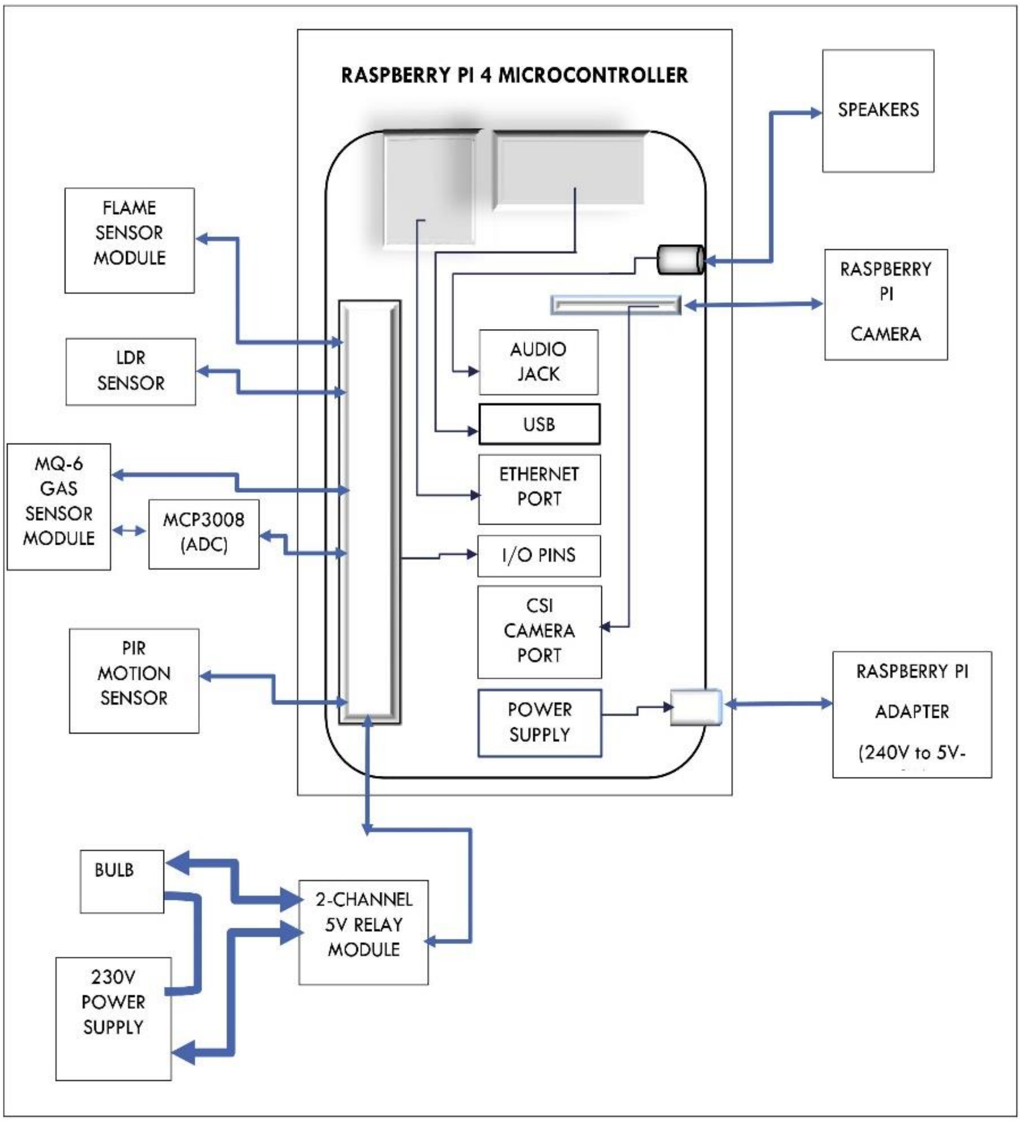

Figure 1: Block diagram illustration of the proposed model

2.4 Circuit Diagram

‘This diagram was made using Microsoft PowerPoint. The image of Raspberry-pi is taken from [9] and the Bread-Board image is taken from [10].

‘If the diagram is not clearly visible, please click on the image to download/view the Circuit Diagram from Google Drive.’

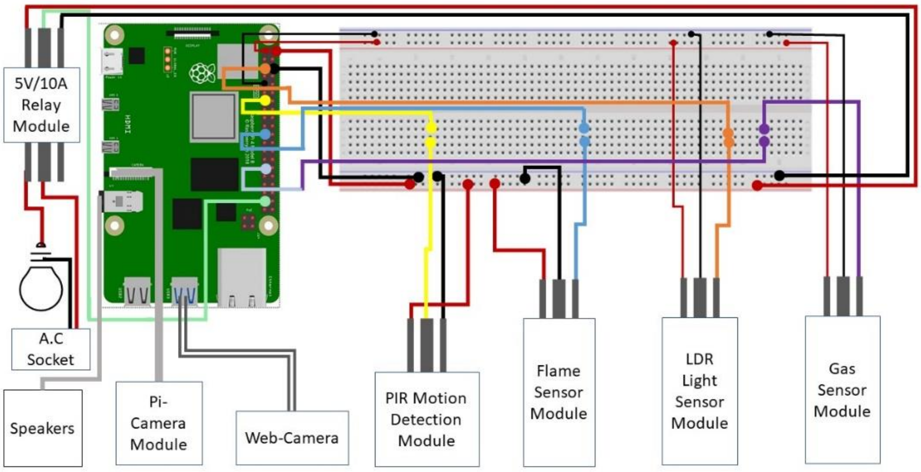

Figure 2: This figure illustrates the circuit connections of the proposed model.

2.5 Procedure

To build the proposed we’ve followed the procedure below.

- The first step is to install an Operating System in Raspberry-Pi.
  - This can be done this by following procedure mentioned in ‘STEPS TO INSTALL RASPBIAN OS’ section of [3]
- The second step is to connect Raspberry-pi to Display using HDMI cable or we can use VNC in case of unavailability of display.
  - To use VNC, we have used MobaXterm application. After installing it on the computer, we can connect to pi by selecting VNC as session type and simply entering IP address of pi. If faced with a problem, one can visit [15] to know the process.
  - VNC viewer can also be used in case of facing difficulties with MobaXterm application. The procedure to install the VNC viewer is as mentioned in [16]
- The third step is connecting all the components which includes Sensor modules, Webcam, Pi-Camera module, Ethernet Cable, Power Supply.
  - All these connections were made as per the circuit diagram shown in Figure 2.
- The fourth step is to install programming language and IDE. We have installed latest version of Python and installed Jupyter Notebook. This was done by executing the following commands in python terminal window.
  - sudo apt-get update && sudo apt-get upgrade
  - sudo apt-get install python3
  - pip3 install --upgrade pip
  - sudo pip3 install jupyter
- The fifth step is to install RPi-Web-Cam-Interface.
  - We have installed this by following the instructions in [14].
- The sixth step is to port forward the necessary ports such as FTP (21) and Webserver (8081). This was necessary so that we can connect to our Raspberry-Pi even when we are not on the same network as Pi.
  - This can be done by using port forwarding option on router admin page.
  - If the ISP (Internet Service Provider) has blocked these ports, then we can use Websites like the website mentioned in [17].

  - The procedure to connect the pi and the procedure for port forwarding is mentioned on the website [17].
- The seventh step is to download all the necessary python modules/libraries needed for the execution of the code.
  - This can be done by running the following commands in the terminal window of Pi.
    - sudo apt-get update && sudo apt-get upgrade
    - sudo apt-get install rpi.gpio
    - sudo apt-get install python3-bs4
    - sudo apt-get install python3-requests
    - sudo apt-get install python3-pandas
    - sudo apt-get install python3-numpy
    - pip3 install pytz
    - pip3 install pyttsx3
    - The final library is OpenCV to install this properly, we followed the instructions in "Setting Up Open-CV and Facial Recognition from Scratch" section in [19].
- The eight step is to create a telegram bot using Bot-Father and to get the API token. This was done by following the steps mentioned in [18].
- The ninth step is to importing necessary modules and packages.
- The tenth step is to train and build an objection detection model. We had done this by following the steps mentioned in [19].
- The eleventh step is to train and build a face detection model. We had done this by improvising the processes in [20,21].
- The twelfth step is to train and build a face recognition model. We had done this by improving the process mentioned in [22].
- We have used the functional programming approach using python.
  - We have used the 'def' method to define the functions that are useful in building the proposed model. Below is the list of functions, their functionality and output given by them.
    - **Motion_detected ():** This function uses the incoming digital data from PIR sensor module connected to the GPIO pin of Raspberry-Pi as input and returns the output as 'motion detected' or 'No motion'. If the pin state is high, it returns 'motion detected' as output. Else it returns 'no motion' as output.
    - **Fire_detected ():** This function uses the incoming digital data from flame sensor module connected to the GPIO pin of Raspberry-Pi as input and returns the output as 'fire detected' or 'No fire'. If the pin state is high, it returns 'fire detected' as output. Else it returns 'no fire' as output.
    - **Gasleak_detected ():** This function uses the incoming digital data from MQ-6 gas sensor module connected to the GPIO pin of Raspberry-Pi as input and returns the output as 'gas leak detected' or 'No gas leak'. If the pin state is high, it returns 'gas leak detected' as output. Else it returns 'no gas leak' as output.
    - **Room_is_bright ():** This function uses the incoming digital data from LDR sensor module connected to the GPIO pin of Raspberry-Pi as input and returns the output as 'room is bright' or 'room is dark'. If the pin state is high, it returns 'room is bright' as output. Else it returns 'room is dark' as output.
    - **Turn_on_the_light ():** This function sends an outgoing signal from Pi to Relay Module connected to GPIO pin of Raspberry-pi. When called, it sends high signal to relay module which switches on the light bulb connected to the relay module.
    - **Turn_off_the_light ():** This function sends an outgoing signal from Pi to Relay Module connected to GPIO pin of Raspberry-pi. When called, it sends low signal to relay module which switches off the light bulb connected to the relay module.
    - **Login_for_portforwarding ():** This is a function inside a function called 'Login_and_get_link ()', automatically logs in to the Pi-Tunnel website with the given credentials.
    - **Login_and_get_link ():** This function uses the 'Login_for_portforwarding' function and logs into Pi-Tunnel website and the navigates to 'Custom Tunnels' option on the webpage. After this it will extract the port forwarded address of port 8081 of raspberry-pi. The live stream which we started in fifth step will be running on this link. It returns the live streaming link as output and then logouts of the session. This link will be used to access the live stream from Pi-camera running on html webpage of RPi-Web-Cam-Interface This is a global link which forwards us to the local link

of pi on the mentioned port number so that it works fine even when we are not on the same network as Raspberry-Pi.

- **Send_message ():** This function takes the message as argument and uses the 'Login_and_get_link', gets the current live streaming link and adds the output from the function to the message argument. After that then sends the combined message to the telegram group in which the bot, we created in step eight is a member along with the users.
- **Start_object_detection ():** This function uses the model in tenth step and returns the output as 'person' or 'not a person' .
- **Start_face_detection ():** This function uses the model in eleventh step and returns the output as 'face is clear' or 'face is not clear'
- **Start_face_recognition ():** This function uses the model in twelfth step and returns the list of names of all the faces it recognizes if it cannot recognize the face then it gives name unknown to that face.
- **Ask ():** This function takes the text as argument and converts it to speech and sends it to speakers in the form of audio.
- **Start_recording ():** This function uses the output from the Login_and_get_link function, enters the webpage and clicks the start recording button on the page.
- **Stop_recording ():** This function uses the output from the Login_and_get_link function, enters the webpage and clicks the stop recording button on the page.
- **Send_poll_message ():** This function sends a poll message to the user with a preset message which is as follows: 'There were unknown people in the house along with household members. Would you like to start recording? If the replay from user is yes or if there was no reply from user within 5 min, it will call start_recording () function and will call Start_objet_detection () function. While the output from Start_object_detection () function is 'person', it will sleep for 1 minute. Once the output is 'not person', it will call Stop_recording () function and sends a message to user about details of the filename.

— Apart from these self-defined custom functions, we have used "quit ()" function to stop the running functions whenever their usage is finished.

- The final/decisive step is to embed/combine all the above functions according to the logic mentioned in the flowchart in figure 3.

2.6 Flowchart

The flowchart is made using the tool mentioned in [23]
'As the flowchart is of larger size, the image is not clear in figure 3 we have uploaded the image to Google Drive and embedded the link in the below figure.

Please click on the image to download/view the flowchart from Google Drive Clearly.'

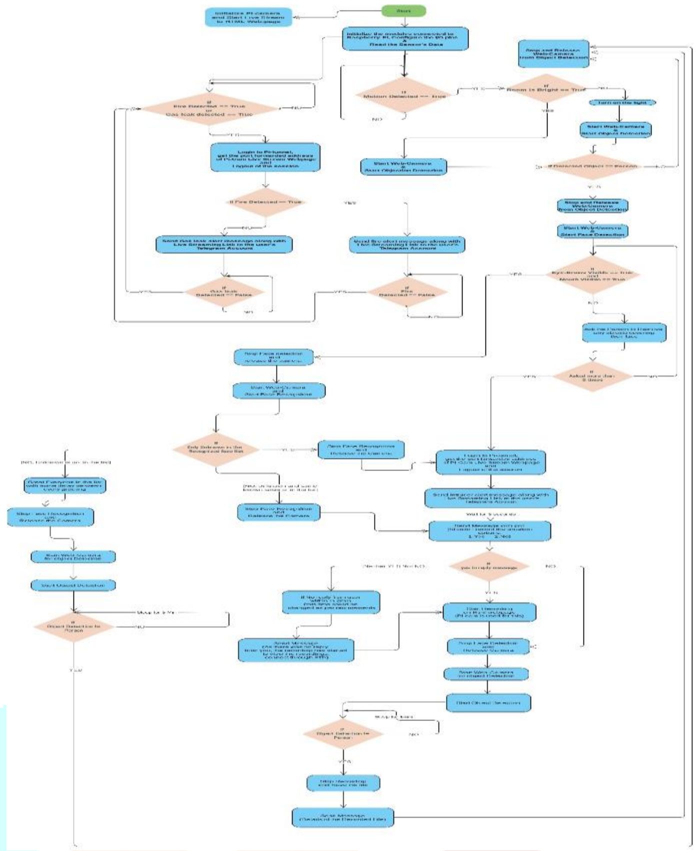

Figure 3 Flowchart Image

3 Results and Discussions

After following the above-mentioned procedure, the build model works exactly as per the proposed concept. Whenever a fire or gas leak is detected, the system sends the alert message along with the live streaming link to the telegram group. Also, when an unknown person enters in the absence of users, it will send an alert message along with live streaming link to the telegram group. When an unknown person is detected in the presence of users, it will ask whether to record the situation or not and goes ahead as per the defined logic. As a lot of computations must be done, there is a bit lag in the functioning, and this can be sorted out by implementing the multi-threading and multi-processing concepts of Python.

4 Conclusion

This paper is contributing a lot in solving the problems associated with the Home Safety and Surveillance systems. As these problems caused the downfall of the credibility and reliability of these systems, the proposed method to integrate Artificial Intelligence (A.I) and IoT with Home Safety and Surveillance systems has made the system more powerful. The proposed method and the build model have a lot of scope for improvements and experimentations. For example, only when the user identity is confirmed, the model could be further perfected to start taking the voice commands from the users and responding to them. The concepts of Deep Learning can be integrated with the proposed system where it captures the images of the user whenever they are in the house and uses them to train itself Which increases the accuracy to unimaginable heights.

References

1. Nayak, M., & Dash, P. Smart surveillance monitoring system using Raspberry Pi and PIR sensor. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6), pp. 7107-7109, ISSN: 0975-9646, (2014).
2. Madhuravani, B., Reddy, P. B., & Rashmika, M. (2017). Motion sensor and face recognition-based surveillance system Using Raspberry Pi. International Journal of Advanced Research in Computer Science, Vol. 8(5), E-ISSN:0976-5697, (2017).
3. Surantha, N., & Wicaksono, W. R. (2018). Design of smart home security system using object recognition and PIR sensor. Procedia computer science, Vol. 135, pp. 465-472, (2018).
4. Majumder, A. J., & Izaguirre, J. A. A smart IoT security system for smart-home using motion detection and facial recognition. 44th Annual (COMPSAC) Computers, Software, and Applications Conference, (pp. 1065-1071). IEEE, (2020, July).
5. Desnanjaya, I. G. M. N., & Arsana, I. N. A. (2021). Home security monitoring system with IoT-based Raspberry Pi. IJEECS, Vol. 22(3), pp. 1295-1302, ISSN: 2502-4752, (June 2021).
6. Kaundanya, C., Pathak, O., Nalawade, A., & Parode, S. Smart surveillance system using Raspberry pi and face recognition. International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6(4), pp. 622-623, ISSN 2278-1021, (2017).
7. Swati, S., Prof. Vijay Sharma. Internet of Things Approach for Face Detection & Face Recognition using Raspberry Pi. IJSRD, Vol. 5(10), ISSN: 2321-0613, (2017).
8. Why motion detectors react to animals and how to avoid it, Ajax Systems, https://ajax.systems/blog/what-is-pet-immunity-in-motion-detectors-and-how-to-use-it-correctly/.
9. Talking to your Raspberry Pi 3 — Android Things, Charles Muchene, https://medium.com/@charlesmuchene/talking-to-your-raspberry-pi-3-android-things-e250af6fdba9.
10. How I made an automated Jack-o'-lantern with a Raspberry Pi, Jessica Cherry, https://opensource.com/article/21/10/halloween-raspberry-pi.
11. What is Raspberry-Pi, https://opensource.com/resources/raspberry-pi
12. Python IDE Tutorial: Th Top 5 Python IDEs for Data Science, Paulo Henrique Vasconcellos, https://www.datacamp.com/community/tutorials/data-science-python-ide.
13. MobaXterm help page, https://mobaxterm.mobatek.net/documentation.html#1_1.
14. Rpi-Cam-Web-Interface, https://elinux.org/RPi-Cam-Web-Interface.
15. MobaXterm demo page, https://mobaxterm.mobatek.net/demo.html.
16. Setting Up a VNC Server on Your Raspberry Pi, knn002, https://www.instructables.com/Setting-up-a-VNC-Server-on-your-Raspberry-Pi/.
17. Pi-Tunnel, https://www.pitunnel.com/.
18. How to Create and Connect a Telegram Chatbot, https://sendpulse.com/knowledge-base/chatbot/create-telegram-chatbot.
19. Object and Animal Recognition with Raspberry Pi and OpenCV, Tim, https://core-electronics.com.au/guides/object-identify-raspberry-pi/.
20. Real-time Face detection | Face Mask Detection using OpenCV, Great Learning Team, https://www.mygreatlearning.com/blog/real-time-face-detection/.
21. Facial Landmark Detection (Eyes, Nose, Jaw, Mouth, etc.) using Python OpenCV, Ashish Choudhary, https://circuitdigest.com/microcontroller-projects/facial-landmark-detection-using-raspberry-pi-opencv.
22. How to Train your Raspberry Pi for Facial Recognition, Caroline Dunn, https://www.tomshardware.com/how-to/raspberry-pi-facial-recognition.
23. Visual Paradigm Online, https://online.visual-paradigm.com/.