Smart Surveillance Vigilant Detection and Notification System

Gayathri Prabhu M , Soorya Krishna K, Apoorva C and Apoorva
ECE Department, Srinivas Institute of Technology, Mangalore

***Abstract*—Surveillance system plays a key role in maintaining the security in today's life. But, fail to provide the feature of avoiding the unfortunate happenings. This work comes up with an idea of smart surveillance system designed such a way that the user is notified when upcoming dangers are detected. In this work, motion of the intruders and the presence of a human faces are detected using image processing algorithms. As the motion detected in the locality of the surveillance area, a short video clip is recorded and sent to the user along with an alert message through the email server controlled by Raspberry Pi. When the cloud is not available, the data is stored locally in the Raspberry Pi and is sent to the user when the connection restarts. Live streaming video from the camera is accessed on any Internet enabled device. Surveillance camera rotates with the help of a DC motor. Also, temperature and gas sensors are integrated with the system.**

Keywords—Surveillance; IoT; Raspberry Pi; Motion Detection; Intruders; OpenCV

I. INTRODUCTION

In today's world, CCTVs (Closed Circuit Televisions) are installed everywhere for the surveillance and monitoring. Installation of the CCTVs helped to bring the transparency and improve the efficiency in the system. Traditional video surveillance is used in short distance monitoring because PC (Personal Computer) is used as a monitor host and connected with a coaxial cable. Initially, it was dominated by analog cameras connected using coaxial cables. To decrease the cost and increase the performance, digital switching systems are used and now IP (Internet Protocol) based delivery of data [1].

In conventional surveillance system, proof of crime committed is obtained but it cannot be stopped at that moment unless it has been brought to the notice to the concerned persons. If a building caught fire, timely knowledge helps in preventing a huge loss and can also prove life Savior. Just recording the events will not help in anything [1].

Detection and tracking of moving objects are important tasks for visual based surveillance systems. Video surveillance application pay attention to a wide area and so omni directional cameras are used. Video data is captured from a mobile camera, compressed into the Moving Picture Experts Group (MPEG) format, transferred through the Internet under control of the processor. Monitor client will receive the compressed data frame to restructure and recompose video images. IoT (Internet of Things) video monitor systems provide a practical solution for remote wireless monitoring with low cost.

CCTVs use video cameras to transmit signals to a specific place on a limited set of monitors. It differs from broadcast television in which the signal is transmitted though Point to Point (P2P) or Point to Multi point (P2MP) or mesh wired or wireless links [2]. Surveillance of the public using CCTV is common in many areas around the world. In industrial plants, CCTV equipment is used to observe parts of a process from a central control room for example when the environment is not suitable for humans.

Design and implementation of a low-cost system monitoring based on Raspberry Pi in which Motion Detection algorithm written in Python was described in [3]. In addition to this, the proposed system uses the motion detection algorithm to significantly decrease storage usage and save investment costs. The algorithm for motion detection is being implemented on Raspberry Pi, which enables live streaming camera along with detection of motion using Pi camera.

Design and implementation of a low cost smart security camera with night vision capability using Raspberry Pi and OpenCV was described in [4]. Design was used inside a warehouse facility. It has human detection and smoke detection capability that can provide precaution to potential crimes and potential fire. The Raspberry Pi with Open Source Computer Vision software handles the image processing, control algorithms for the alarms and sends captured pictures to users email via Wi-Fi. The system uses ordinary web cam but its IR filter was removed in order to have night vision capability.

Security alarm system using low processing power chips using Internet of things which helps to monitor and get alarms when motion is detected and sends photos and videos to a cloud server was described in [5]. IoT based application is used remotely to view the activity and get the notifications when the motion is detected. The photos and videos are sent directly to a cloud server. When the cloud is not available, then the data is stored locally on the Raspberry Pi and sent when the connection resumes.

In [6], Pyroelectric Infrared sensors (PIR) and pressure sensors are used as the alert group in windows and doors where an intruder must pass through. These low-power alert sensors wake up the MCU (Micro Controller Unit) which has power management for the ultrasonic sensors and PIR sensors. This state transition method saves a large number of sensors required for the alert power. It also uses the Majority Voting Mechanism (MVM) to manage the sensor groups to enhance the probability of multiple sensors sensing. After the MCU sends the sensor signals to the embedded system, the program starts the web camera. This sensing experiment shows that it reduces the systems power consumption.

The above systems do not recognize faces, whereas this work is designed to recognize the faces and send alerts. The surveillance system is integrated with a gas sensor and a temperature sensor for additional security. The temperature sensor will update its data to cloud server when it exceeds threshold and notifies the user through mail. This system will identify the motion and record a small video clip, which is sent to the user through mail along with a live streaming link. Thus, this system is advantageous than the conventional CCTV by reducing the storage memory requirement. It also alerts the user to take action during the intrusion or accident in the surveillance area.

The organization of this paper is listed below. Section II describes the proposed work. Section III describes the soft- wares and libraries used in this work. Section IV describes the implementation of the Surveillance Detection and Notification System. Section V discusses the simulation results. Conclusion is given in Section VI.

II. PROPOSED FRAMEWORK

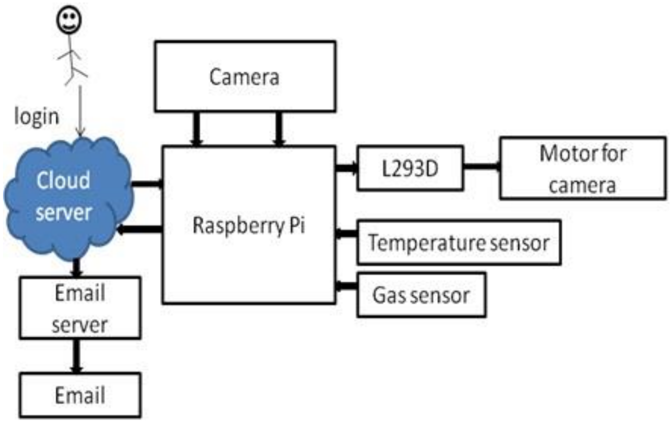

Fig. 1: Block diagram of the Proposed system

Figure 1 shows the proposed system to implement smart surveillance vigilant detection and notification System [7]. In this, Raspberry Pi controls the entire system. Camera is used to monitor the surveillance area and a motor is used to rotate the camera. L293D IC is a motor driver. Sensors such as temperature sensor and gas sensor are used to supervise the area. Any disturbance in the surveillance area results in user receiving an alert through email.

III. SOFTWARES AND LIBRARIES USED

A. Software's and Libraries are Used

In this work, following software's and libraries are used.

- Raspbian is used as the Operating system in Raspberry Pi. It is a Debian-based computer operating system for Raspberry Pi [8].
- Mail Server is used to send an alert or notification to the user if there is any activity in the surveillance area. Mail server receives e-mails from client computers and delivers to other mail servers. When you press the “Send” button in your e-mail program, program will connect to a server on the network/Internet and termed as Simple Mail Transfer Protocol (SMTP) server [9].
- ThingSpeak is an open source IoT application and API is used to store and retrieve data using the Hypertext Transfer Protocol (HTTP) over the Internet or via a Local Area Network. When “Send” button is pressed in the e-mail program, the program will connect to SMTP server. ThingSpeak enables the creation of sensor logging applications, location tracking applications and a social network of things with status updates.
- OpenCV-Python is used to solve computer vision problems. It makes use of Numpy, which is a highly optimized library for numerical operations with MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This makes easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.
- Deep Neural Network (DNN) library is used to classify the images. Deep learning is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms.
- Motion library is used to detect the motion and it is a built-in library available on Raspberry Pi. Motion is a highly configurable program that monitors video signals to watch birds, check in on your pet, create time-lapse videos and more.
- Adafruit library is used to read data from sensor. Many small embedded systems exist to collect data from sensors, analyze the data, and either take an appropriate action or send that sensor data to another system for processing. Creating new drivers is a relatively easy task, but integrating them into existing systems is both error prone and time consuming. By reducing all data to a single type and settling on specific, standardized SI units for each sensor family the same sensor

types return values that are comparable with any other similar sensor.

- OpenCV is used as a platform to write face detection and motion detection programs. OpenCV is an open source library that includes several hundreds of computer vision algorithms. OpenCV has a modular structure which includes several static libraries. Following modules are used in this work.
  - Core functionality - A compact module defining basic data structures including the dense multidimensional array and basic functions used by all other modules.
  - Image processing - This module includes linear and non-linear image filtering, geometrical image transformations like resize, affine, perspective warping generic table-based remapping, color space conversion, histograms, and so on.
  - Video - Includes motion estimation, background subtraction, and object tracking algorithms.
  - Calib3d -This module includes basic multiple-view geometry algorithms, single and stereo camera calibration, object estimation, stereo correspondence algorithms and elements of 3D reconstruction.
  - Features2d - Includes salient feature detector, de-scriptor, and descriptor matchers.
  - Objdetect - This module includes Detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
  - Other than these, Highgui, Video I/O, Graphics Processing Unit (GPU) libraries are used.

IV. IMPLEMENTATION

The block diagram shown in Figure 1 is implemented by considering each block at a time and then integrated to make the complete system. The system is first interfaced with sensors and is communicated to the user through the server. Later the camera is interfaced to this system in order to surveillance the area. By using the motion detection algorithm, surveillance is made and notified to the user if movement is detected. The motor is rotated in order to increase the area of surveillance which is powered by a 5V supply.

A. Flowchart to Detect Temperature, Gas and Face

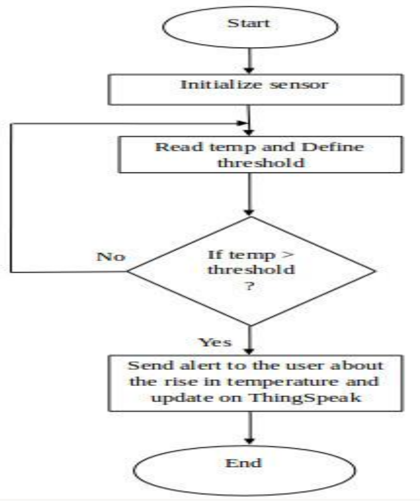

Fig. 2 : Flowchart to Detect Temperature

Temperature sensor is interfaced with Raspberry Pi and reads the temperature from the surveillance area. If this value is greater than that of the threshold value set, an alert message is sent to the user. Flow chart to detect the temperature is shown in the Figure 2.

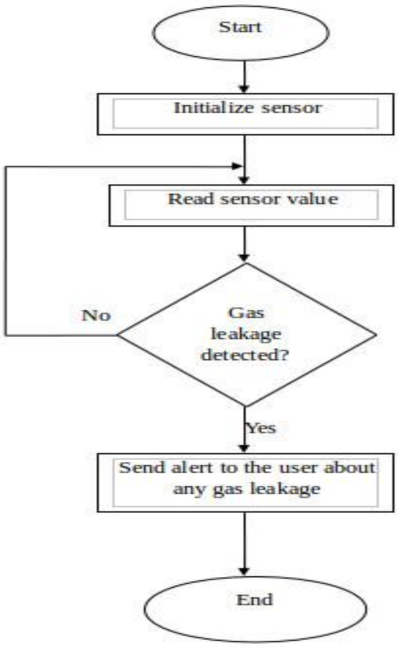

Fig. 3 : Flowchart to Detect Gas

Gas sensor is interfaced with the Raspberry Pi and is used to detect if any leakage of gases in the surveillance area. If so, an alert is sent to the user. Flowchart to detect gas is shown in Figure 3.

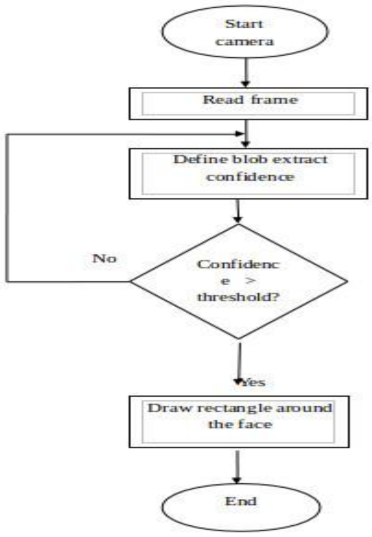

Fig. 4: Flowchart for Face Detection

Once the frame is captured from the video, it is further processed to check whether any face is detected and threshold is defined to identify the face. Flowchart for the face detection is shown in Figure 4.

B. *Motion Detection Algorithm*

Motion detection algorithm is used for finding the presence of human in the surveillance area. The algorithm is as follows.

- The video of the surveillance area is continuously streamed by the USB camera and is processed frame by frame by the Raspberry Pi.
- The frames are then converted into gray scale as the system does not need the colored content of the images to be processed for motion detection.
- The background of the video stream is largely static and unchanging over consecutive frames of a video. Therefore, by modeling the background, substantial changes can be monitored. If there is a substantial change, a motion is detected.
- The first frame is considered as a reference image with respect to which the next sequence of images is subtracted to get the blob greater than the threshold.
- The threshold is a minimum area predefined for detecting a comparable object rejecting the small animals like lizards, rats, etc.
- Every consecutive images which are needed to be compared will be having minute difference may be due to noise. Hence, to eliminate this, the images are blurred using Gaussian blurring.
- The blurred images are processed easily.
- On motion detection, a text is displayed on the video regarding the room occupancy status as Occupied.

C. Flowchart to Implement Surveillance Vigilant Detection and Notification System

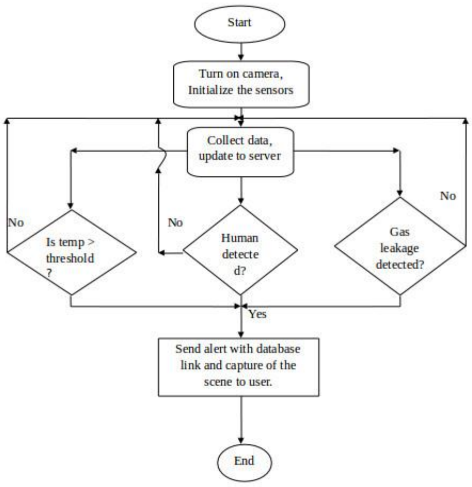

Fig.5: Flowchart to Implement the Complete System

Combining the individual cases discussed in section IV-A and IV-B forms the flow chart of complete Surveillance Vigilant Detection and Notification system. All the three cases are considered and if any one turns out to be true, then the alert is sent. Flow chart corresponding to this is shown in Figure 5.

V. SIMULATION RESULTS

A. Temperature Sensor Output

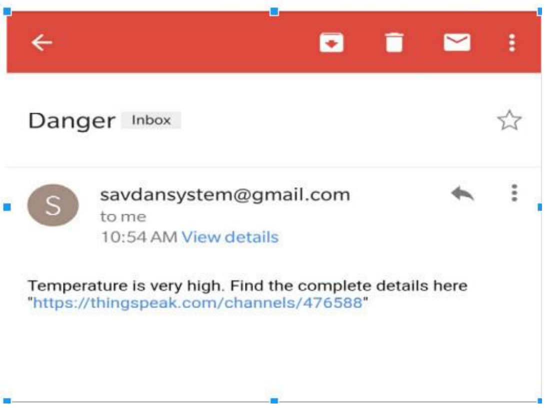

Fig. 6: Alert mail of high temperature

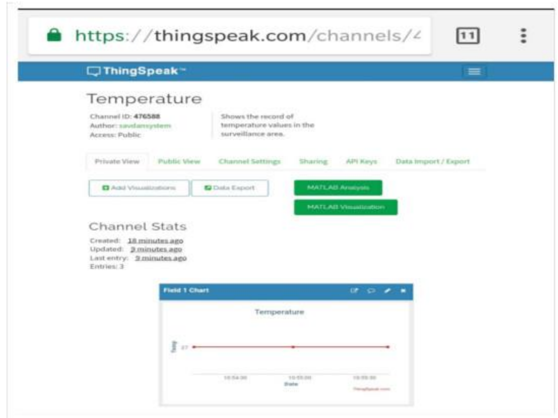

Fig. 7 : ThingSpeak platform result representation

Temperature sensor continuously senses the temperature in the surveillance area and as soon as it exceeds a certain threshold, an email is sent to the user. It also contains link where a graphical representation of the temperature versus date (or time) that can be viewed as shown in Figure 6 and 7.

B. Gas Sensor Output

Figure 8 shows the information to user, when any type of harmful gases such as H_2 , CO, CH_4 , Smoke etc. are detected in the area of surveillance. Intimation is sent through an email that LEAKAGE in gas is detected.

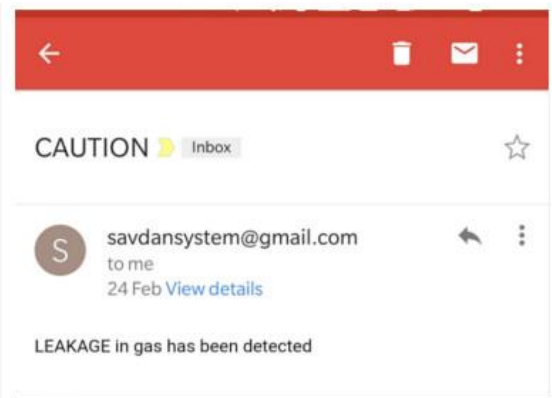

Fig. 8 : Alert mail of detecting the gas

C. Face Detection

Human face is detected using OpenCV deep learning module. The face is marked with a square box. This is possible not only for downloaded images or video but also in the live video streaming. As the system is designed to detect human face, it can differentiate between human and non-human faces. Two cases of results are shown in Figure 9 and 10 .

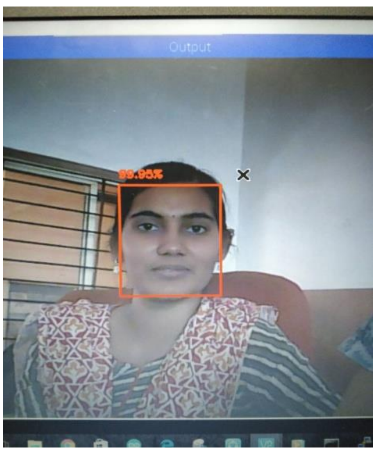

Fig. 9: Individual face detection

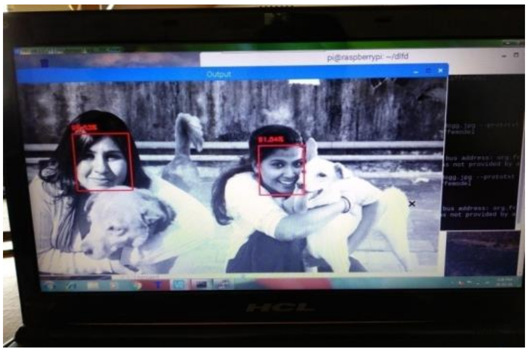

Fig. 10: Multiple human face detection

In this system, face detection was not accurate when the face was covered with mask. To overcome from this disadvantage, system uses motion detection algorithm to know the status of surveillance area and is shown in Figure 11. Efficiency of the designed system is checked at different

Fig. 11 : Room occupancy status

random time with respect to false/true positive or false/true negative. Here, True positive indicate the right result (alerting the user) for right trigger whereas false positive indicates that alerting the user when there is no motion detected. False negative denotes that not alerting the user when there is motion and True negative is that not alerting when there is trigger. Different trials are taken at different times and is shown in Table I. Efficiency of the system is found that more than 90%.

TABLE I : CHECKING THE EFFICIENCY OF THE DESIGNED SYSTEM

Trials	True Positive	True Negative	False Positive	False Negative
1	5	1	0	2
2	7	1	0	1
3	6	2	0	0
4	4	2	1	0
5	5	1	0	0
6	4	0	0	0
7	6	1	0	1
8	10	3	0	2

VI. SIMULATION RESULTS

The proposed system is used in home security and other control applications. Deploying Raspberry Pi and webcam helps to detect report and monitor intrusion events by the users. System alerts the user and helps to reduce the damages. Use of Cloud network in the system allows for storage of captured images and recorded videos. This system ensures that there is no need to store entire streaming data in the system memory and thus reduces the memory space. It also enables the camera to cover a wider region by rotation up to 360 degree. If the server connection is lost then system stores the video capture locally and sends it once the server connection is re-established.

Fig. 12: Complete setup

REFERENCES

[1] Sean Dieter Tebje Kelly, Nagender Kumar Suryadevara, and Subhas Chandra Mukhopadhyay, "Towards the Implementation of IoT for Environmental Condition Monitoring in Homes", IEEE Journal, Vol .13, No. 10, October 2013.

[2] S. Hong, D. Kim, M. Ha, S. Bae, S. Park, W. Jung and J. E. Kim, "SNAIL: An IP -based wireless sensor network approach to the internet of things", IEEE Wireless Commun, Vol. 17, no. 6, pp. 34-42, Dec. 2010.

[3] Bui Dinh Mao and Eui-Nam Huh, "Low cost real-time system monitoring using Raspberry Pi", IEEE Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference, pp 1-4, 10 August 2015.

[4] Alexander C. Abad and Elmer P. Dadios , "Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV", IEEE 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), pp 2-6, 22 January 2015.

[5] Neelam Sharma and Anurag T yagi, "An Internet of things approach for motion detection using Raspberry Pi", IEEE 2014 International Conference on Intelligent Computing and Internet of Things (ICIT), pp 3-4, 25 May 2015.

[6] Ying-Wen Bai, Zi-Li Xie and Zong-Han Li , "Design and Implementation of a Home Embedded Surveillance System with Ultra-Low Alert Power", IEEE publication, pp 71-73, 01 November 2015.

[7] S. Naga Jyothi andK. Vijaya Vardhan , "Design and Implementation of Real Time Security Surveillance System using IoT", International Conference on Communication and Electronics Systems (ICCES), 21-22 Oct. 2016.

[8] https://www.raspberrypi.org/products/raspberry-pi-2-model-b

[9] Sarthak Jain, Anant Vaibhav and Lovely Goyal, "Raspberry Pi based interactive home automation system through E-mail", IEEE 2014 International Conference on

Optimization, Reliabilty, and Information Technology (ICROIT), pp 1-3, 17 April 2014.

[10] Simon Monk, "Raspberry Pi Cookbook: Software and Hardware Problems and Solutions", O'Reilly Media, 2 edition, 12 June 2016.