Smart Security System: A Surveillance System Based on OpenCV and Android Platform

Nishant Ghanate[1], Kartik Bhagat[2], Sandeep Gamot[3]

[1-3]Dept. of Computer Engineering, Atharva College of Engineering, Maharashtra, India.

Publication Info

Article history:
Received : 18 February 2020
Accepted : 22 May 2020

Keywords:
OpenCV (Open Computer Vision), Smart surveillance, IoT (Internet of Things), Android, Firebase.

Corresponding author:
Nishant Ghanate
e-mail: *nishant7.ng@gmail.com*

Abstract

Image processing and computer vision techniques can now be implemented using computer processing. And with readily available libraries like OpenCV, it has become easier than ever to use such technologies to improve upon the traditional security systems for our benefit. By the creation of this project, we aim to make it a very easy process to secure the homes and workplaces at a relatively cheaper cost. Traditional CCTVs, in addition to being expensive, do not offer much in terms of control over devices. Our proposed system consists of developing a smart tool by which a computer or Raspberry Pi attached with a camera can be converted into a smart surveillance system. Further modules can be connected like temperature module, PIR sensor, etc. The aim of this project is to develop a smart and cost-efficient smart system and, as a result, provide a cheap and manageable option to small business owners and households.

1. INTRODUCTION

Computer vision is a highly branched scientific field that deals with things like how computers can be used to gain a high level of understanding from digital images or videos. From the perspective of engineering, its goal is to automate tasks that the human visual system is capable of.

Computer vision includes methods for image acquiring, image processing, image analyzing and understanding digital images, and extracting high-dimensional data from the real world to produce numerical or symbolic information, e.g., in the forms of decisions. OpenCV (Open source computer vision) is a library of programming functions mainly concerned with and focusing on real-time computer vision. Originally, it was developed by Intel. It was later supported by Willow Garage then Itseez, which was then later acquired by Intel.

The library is cross-platform and free for all to use under the open-source BSD license. Traditional surveillance security systems provide little in the way of taking action then and there as soon as a crime is being committed. Furthermore, it is very expensive and complicated to set up such a system.

The goal of this project is to create an open-source smart tool that can help individuals or institutes set up a low-cost, secure system by themselves. This will give them full control over their devices to secure their parameters and customize according to their needs.

1.1. Need and Motivation

It is important to secure our homes, offices, and other places of business to prevent certain crimes. The traditional surveillance systems do not have any way of alerting the owners if there are any criminal activities taking place. They just record/broadcast the feed. As a result, the owners are not able to take on the spot action to counter or prevent any intrusion or theft when it occurs.

1.2. Existing System

In today's day and age, it has become important to secure one's assets, be it workplace or home. If people overlook this, they may become vulnerable to various kinds of physical and cyber attacks on themselves, their work, and their family.

One of the aspects of such security is surveillance. Traditional CCTV setups provide little in the way of taking an on the spot action. CCTV setups depend on the users to go through the recorded feed or just observe the broadcasts continuously. Furthermore, they are quite expensive for a small household or business to own.

Being non-modular in design makes it difficult for users to understand them and debug them. Also, once set up, they are pretty much non-upgradable with new features. All these things make traditional CCTV setups very inefficient. [9]

The methodology of the current system makes use of Traditional CCTV cameras, which passively keep records of events happening in its visual scope. The current system just records what is happening and delegates the responsibility for taking actions on the concerned person rather than actively running a motion detection algorithm on the visual feed and having a fault-tolerant back-end to capture the event and alert the concerned person and doing

2010-2019 S-JPSET : Vol. 12, Issue Supplementary-1, ISSN : 2229-7111 (Print) and ISSN : 2454-5767 (Online)

so with the least cost possible.

Our methodology does not require any costly gadgets. Anything that can capture visual feed can be effectively evolved into a smart surveillance system.

1.3. Proposed System

Our system consists of two parts; the client and the server. The client is on an android-based cell phone, and the server is Google Firebase providing us with real-time authentication and database.

The proposed system finds its way into a variety of domains. The most basic setup will include a webcam connected to a computer or a raspberry pi device. After the code is executed on this device, the camera will start observing the environment, and as soon as it detects motion, it will notify the user on his/her phone. Firebase will be used as a database and a communication link between the device present on location and the phone that will be with the users. Additionally, this system can be modified to include counter-measures upon confirming a break-in or an attack, such as calling the police, etc.

Further sources of input, like a temperature sensor, LDR sensor, etc., can also be connected to the device with very small modifications to the code. This project will enable its users to set up a security system using the devices they already own and hence will prove to be very cheap and effective. [3][7][12][13]

The three essential parts of the system are the watcher interface, alert system, and user interface.

1.3.1. Watcher Interface

The Watcher Interface implements an Observer pattern that observes the camera feed and runs an Algorithm to detect motion using the OpenCV library that uses Image thresholding to detect motion.

1.3.2. Alert System

Whenever the algorithm detects motion, it sends an alert object to the user that consists of the snapshot, timestamp of the feed that triggered the motion detection algorithm, via Firebase back-end. [1][6][10]

1.3.3. User Interface

At the Firebase back-end, a messaging queue is responsible for routing the alert object to the corresponding user. This messaging queue is integrated in the Firebase itself. This alert object is sent as a JSON string over the internet. At the user's end, the JSON data is populated according to a template defined in the android application. [10]

2. WORKFLOW OF SYSTEM

Our system working has been depicted by the sequence diagram. The user in this system is the one using the application, the application frontend us the android application user interface used by the user and the back-end is the Firebase providing us with various features which were talked about earlier. [1][5][10]

The android application would receive the data from Firebase in real-time. In case there is an event captured by the camera, the stationary device would send a notification via the application along with the snapshot taken when such an event was triggered. [10]

2.1. Methodology

The software development process for this project is a mix of TDD or "Test-Driven Development" and the Spiral model, where critical modules were developed using TDD to ensure their logical correctness and their ability to perform their assigned task.

This includes the modules of :

- Authentication

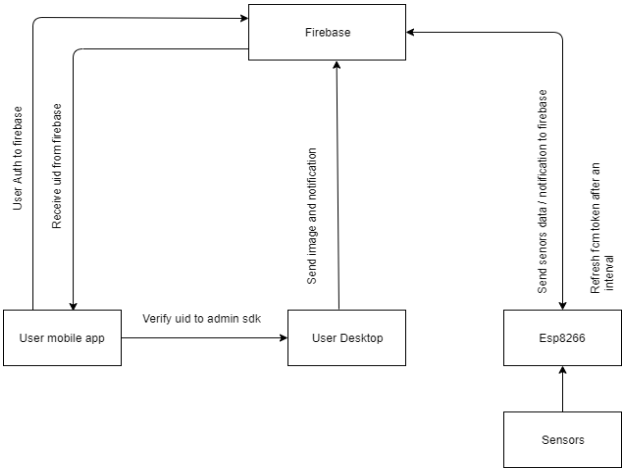

Figure 1: Proposed system

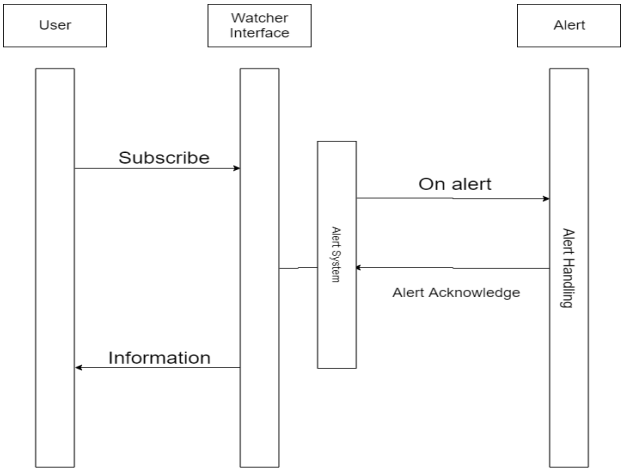

Figure 2: System workflow

 2010-2019 S-JPSET : Vol. 12, Issue Supplementary-1, ISSN : 2229-7111 (Print) and ISSN : 2454-5767 (Online)

- Motion detection [1]
- Data serialization
- Server error handling and re-routing
- Alert system

The trivial and non-critical part of the system is developed through the Spiral model. These include modules of :

- Data logging
- Desktop app error handling
- Networking (critical functionalities handled by external library)
- Image capturing

3. IMPLEMENTATION

3.1. User Authentication and Alert Events Dashboard

The user will be shown the authentication screen wherein the user will need a Google account to authenticate into the service. On successful authentication, the user will be forwarded to the alert events screen.

This screen is shown to the user after successful authentication from the database and authentication protocols. This screen shows the history of the events triggered according to the data from the sensors and camera feed. In case there is a motion-captured on the camera feed, the camera snaps a picture and sends it to the user along with an alert notification. The user can choose what actions to be performed next. (e.g., like calling the concerned authorities)

3.2. Sample Alert Notification and Contact List

The above screen shows a sample alert notification for a motion detection event triggered. The notification consists of a snapshot of the environment when the motion detection routine was triggered. the user may then decide to act on the alert by taking the appropriate decision to deal with the risk, or in case of a false positive, the user may choose to ignore the notification. [1]

The user may, if need be, choose from a selected list of contacts that may be better situated to deal with the emergency and act upon on behalf of the user. The contact list screen shows the list of contacts, and with a single click, the user can choose the contact from the emergency contact list.

The emergency contact list may consist of neighbors and concerned authorities or any other person that the user might trust.

3.3. Sign Out Page

The above screen can be accessed by the user to Sign out of the app and end the user session. The user activity and data are persisted on the database so that it can be accessed whenever desired.

4. CONCLUSION

As we know, the current security systems are expensive and difficult to implement. In this project, we have implemented the OpenCV library to monitor an environment and alert the user if there is any motion detected in it.

This design can be further customized using different sensors and tailored according to the users' needs with just a few changes in the program. This system can be extended and integrated with IoT, specifically smart home controllers so that the process of alerting can be automated and take the human out of the loop for real-time alerting and mitigation.

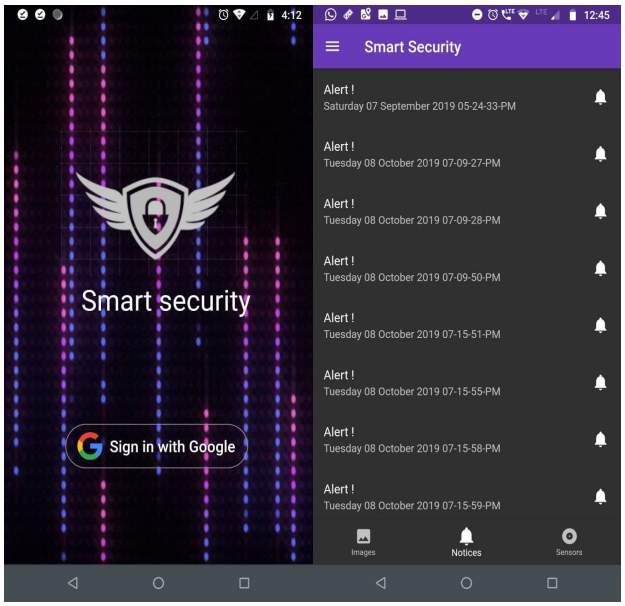

Figure 3: Authentication screen and alert screen

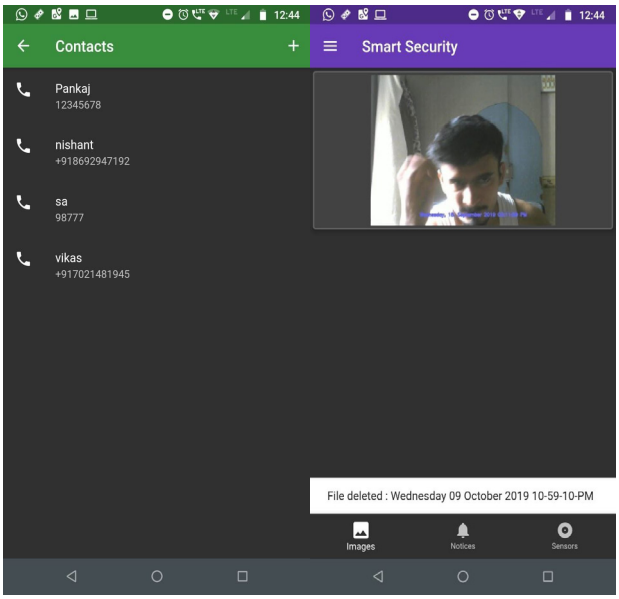

Figure 4: Alert and contact list

2010-2019 S-JPSET : Vol. 12, Issue Supplementary-1, ISSN : 2229-7111 (Print) and ISSN : 2454-5767 (Online)

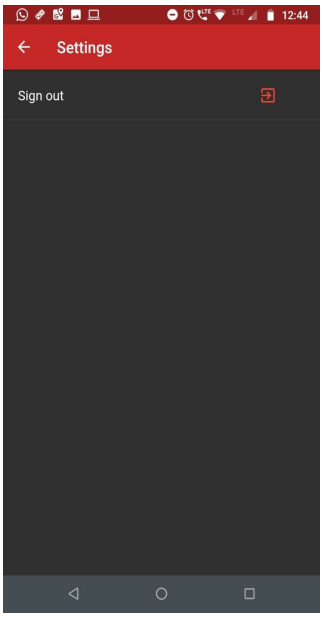

Figure 5: Sign out

One potential drawback of this project is that it may alert false positives. So to tackle this problem, we can further train a machine learning model to detect and filter out real threats from false positives.

Using this system, one with a little technical knowledge can easily convert an old laptop into a surveillance device and modify it according to their needs.

5. REFERENCE

[1] R.J. Radke, S. Andra, O. Al-Kofahi, B. Roysam, "Image change detection algorithms: a systematic survey," IEEE Transactions on Image Processing (Volume: 14, Issue: 3, March 2005), pp 294 – 307.

[2] Frazer K. Noble, "Comparison of OpenCV's feature detectors and feature matchers", 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP).

[3] Sourabh Sarkar, Srijita Gayen, Saurabh Bilgaiyan, "Android based home security system using Internet of Things (IoT) and Firebase," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA).

[4] Alex Jaimes, "Computer vision startups tackle AI," IEEE MultiMedia (Volume: 23, Issue: 4, Oct.-Dec. 2016), pp 94 96.

[5] Xin Li, Yiliang Shi, "Computer Vision Imaging Based on Artificial Intelligence," 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS).

[6] Raghav Bansal, Gaurav Raj, Tanupriya Choudhary, "Blur image detection using Laplacian operator and OpenCV," 2016 International Conference System Modeling & Advancement in Research Trends (SMART).

[7] D. Abhilash, Chandrashekar Chandrashekar, S. Shalini, "Economical, energy efficient and portable home security system based on Raspberry Pi 3 using the concepts of OpenCV and MIME", 2017 International Conference on Circuits, Controls, and Communications (CCUBE).

[8] Bhaumick Vaidya, Ankit Patel, Anand Panchal, Rangat Mehta, Krish Mehta, Parth Vaghasiya, "Smart home automation with a unique door monitoring system for old age people using Python, OpenCV, Android and Raspberry pi," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS).

[9] Blog: Advantages and Disadvantages of using Security Cameras, http://tiny.cc/y2vykz.

[10] Wu-Jeng Li, Chiaming Yen, You-Sheng Lin, Shu-Chu Tung, ShihMiao Huang, "JustIoT Internet of Things based on the Firebase real-time database", 2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE).

[11] P. Vigneswari, V. Indhu, R. Narmatha, A. Sathinisha, and J. Subashini, "Automated security system using surveillance," *International journal of current engineering and technology*, vol. 5, no. 2, 882–884, (2015).

[12] S. Suresh, J. Bhavya, S. Sakshi, K. Varun, and G. Debarshi, "Home monitoring and security system," in *2016 International Conference on ICT in Business Industry & Government (ICTBIG, 2016)*, 1–5.

[13] H. U. Zaman, T. E. Tabassum, T. Islam, and N. Mohammad, "Low cost multi-level home security system for developing countries," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS, 2017)*, 549–554.

[14] "Learning Python, 4th Edition", Author: Mark Lutz, O'Reilly Media, 2010.

[15] "A Practical Introduction to Computer Vision with OpenCV," Author: Kenneth Dawson-Howe, Wiley Publication, 2014.