A Smart IoT Security System for Smart-Home Using Motion Detection and Facial Recognition

Conference Paper · May 2020

DOI: 10.1109/COMPSAC48688.2020.0-132

2 authors, including:

AKM Jahangir Alam Majumder
University of South Carolina Upstate
25 PUBLICATIONS **476** CITATIONS

A Smart IoT Security System for Smart-Home Using Motion Detection and Facial Recognition

AKM Jahangir Alam Majumder and Joshua Aaron Izaguirre
Division of Mathematics and Computer Science, University of South Carolina Upstate, SC, USA
majumder@mailbox.sc.edu, izaguirj@email.uscupstate.edu

Abstract - **The Internet of Things (IoT) combines the idea of technology providing value to our daily lives. One major category in which technology benefits us is the concept of security and privacy. Smartphones can act as a security alert system as the smartphone is the most widely-used smart device. Recently, the use of Artificial Intelligence (AI)-integrated smart IoT devices has significantly increased as well. In this paper, we developed a smart IoT security system for the smart-home. In which, a Raspberry Pi acts as a security system with a No Infrared (NoIR) Pi Camera Module to record videos and capture images. Also, a Passive Infrared (PIR) Motion Sensor is used to detect motion. We propose to use motion sensor data and the images gathered from the NoIR Pi Camera Module to predict a security threat using a face recognition classification technique along with our developed algorithm. The system has the capability to notify the user in case of any emergency. The proposed system can detect any security threat with an accuracy of 95.5% and a precision of 91%.**

Keywords – Smart IoT, Security System, Smart-Home, Motion, Face Recognition

I. INTRODUCTION

A. Background and Motivation

A smart-home consists of a computer, Smartphone, and other smart sensors or actuators that are equipped with an IoT connection. Internet of Things (IoT) involves interrelated computing devices transferring data over a network without requiring interaction between the computer and the user. This creates a platform for a security system by allowing the user to feel safe at home and feel safe leaving the home knowing that the user will be alerted when someone comes near the home. One key category in which technology benefits us is the concept of security and safety. With the fact that most of our population today carries technology around on their person as compared to the earlier years of smartphones makes it more appealing that smartphones can act as a security alert system [1-3]. Time and evidence are two factors necessary for proactive crime prevention.

The recent increase in availability and use of smart IoT devices and the ubiquity of smartphones allows the ability of individuals to monitor house/office/store security conditions on a continuous basis. This project's major impact for society is to establish a viable and easily available approach to the community by gathering data and identifying risky behaviors in the smart-home environment. In this research, we developed a system (smart IoT system) that allows the user to detect unauthenticated access in the smart-home.

B. Limitations of Previous Works

While smart IoT security systems are becoming more and more popular as technology continues to advance, many fail to provide the user with a device that uses machine learning. Many residence and retailers use the Electronic Article Surveillance (EAS) security system standard, which is problematic due to the high rate of false alarms associated with EAS [4-5]. We can solve many of the problems that come with the EAS security system and increase the efficiency of security systems by using real-time camera modules that keep a video log of each camera's video feed and alert the user from a smartphone device.

By allowing the IoT security device to capture images based upon motion, it will learn to differentiate between someone allowed in the smart-home vs an unauthorized person entering the smart-home. "In terms of IoT security, it's exceptionally wide scope involves multiple dimensions: trusted sensing, computation, communication, privacy, and digital forgetting" [6]. Our proposed system sends captured images and video files to the user as they occur, so this covers trusted sensing and communication. The open-source API we utilized for our notification system allows the user to set up specific devices to be notified through the API, which covers privacy. The data on the micro SDHC card and operating system, Raspbian, cover computation along with the IoT security device's ability of machine learning authentication through the Local Binary Pattern Histograms (LBPH) algorithm. Images and video files are sent to the user via notification; however, backup files are stored onto the micro SDHC card, which the user can access to obtain or delete, which covers digital forgetting [7-9].

C. Our Proposed Approach

In this research, our primary focus is on threat detection which will help to predict a security threat due to any unauthorized access in the smart-home environment. To address the issue of threat detection, the aim of this research is to determine a security threat using smartphones. Data from a No Infrared (NoIR) Raspberry Pi Camera Module and PIR Motion Sensor were used to validate the proposed approach and to identify a security vulnerability in the smart-home.

Our proposed approach combines computation with hardware designed to utilize motion detection based upon heat radiated from a person to trigger a positive differential change compared to the ambient heat radiated in the area the device is in. It also provides a dependable and fault-tolerant system by using machine learning to authenticate the motion detected by the PIR Motion Sensor to avoid false triggers on authenticated people in the area.

D. Major Contributions

In this paper, we propose to use a smart IoT security system designed for smart-homes which will allow the user to monitor the system via smartphone. Our major contributions are as follows: we

- *developed an IoT security system that will detect motion via PIR Motion Sensor that measures*

infrared light radiating from objects in its field of view

- *proposed authentication access to the smart-home by combining machine learning with the images captured via No Infrared (NoIR) Pi Camera Module.*
- *combined an open-source API with our IoT security system that will notify the user of unauthenticated motion via smartphone and provide image and video files to the user.*

The rest of the paper is organized as follows: in Section II, we discussed the system architecture's hardware and software components. In Section III, we explain data collection techniques for the PIR Motion Sensor and NoIR Pi Camera Module and continue with Section IV, which discusses our result analysis of the data collected. In Section V, we evaluate related works compared to our proposed smart IoT security system. In Section VI, we conclude the paper with some future research directions.

II. System Architecture

The strength of our proposed IoT security system is dependent on built-in wireless communication technologies to provide a low-cost solution with maximum freedom of wireless capability. In addition, we have used a smartphone and a developed security system with the NoIR Pi Camera Module and PIR Motion Sensor that are user-friendly and easy to install. The architecture of the system is shown in Figure 1.

In this research, each aspect of the hardware is adjusted and modified using the software to ensure an accurate and reliable IoT security system. The system depends on the PIR Motion Sensor to detect motion, send a high signal through the general-purpose input/output (GPIO), and run code based upon that high signal. The high signal from the PIR Motion Sensor will run code from the Raspberry Pi Zero W board's operating system, Raspbian, which is stored on the micro SDHC card. The code will allow the NoIR Pi Camera Module to record video and capture images will allow the open-source PushBullet API to send notifications with files to the user [10]. The captured images and videos are also stored onto the Micro SDHC card. The flowchart in Figure 1 represents the sequential order of how the hardware and software together notify the user of detected motion.

A. Hardware

All hardware components used in our developed security system are shown in Figure 2. Raspberry Pi Zero W (as shown in Figure 2a) [11], security device contains a NoIR Pi Camera (Contains no infrared filter for use at night shown in Figure 2e), a Passive Infrared (PIR) Motion Sensor (shown in Figure 2d) that detects motion based upon infrared light radiating from objects in its field of view, a fisheye lens that provides wide-angle views, a Samsung Pro Endurance 32GB Micro SDHC card (shown in Figure 2c) to store images and videos, and a Pi PIR Camera box case with a wall mount to protect the electrical components inside and mount the camera to point it at the area you would like to monitor [12-15]. An external battery pack shown in Figure 2b is used to power up the system. The complete IoT Security System is shown in Figure 2f.

The Raspberry Pi NoIR Camera Module Version 2.1 supports multiple different resolutions and frame rates. The focus lens remains in a fixed position. This camera model does not contain an infrared filter on the lens, which means it will produce better visuals during low-light settings. The fish-eye lens provides an ultra-wide 180-degree angle, which is important for a camera module with a fixed angle.

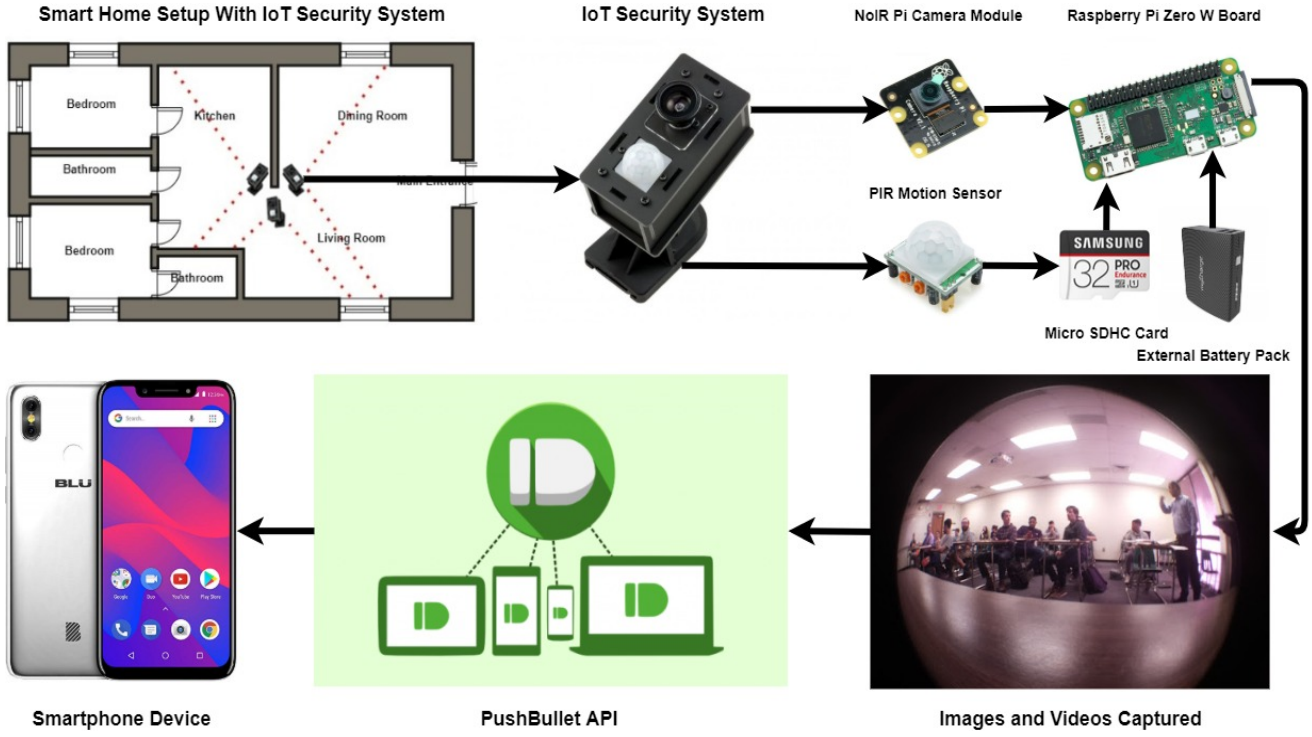


Figure 1: The Proposed IoT Security System Architecture

The PIR Motion Sensor can be set to detect motion from 5 meters to 7 meters. These are set to 7 meters. The sensors detect infrared light in a 100-degree cone extending from the sensor to the set range. The delay time, or how long the motion sensor will output high after motion is detected can be set from 0.3 seconds to 300 seconds. When the repeatable trigger is off, the output will automatically change from a high to a low level when the sensor output is high, and the delay time is over. When the repeatable trigger is on, it will keep the output high all the time when the sensor is retriggered repeatedly and will output low when idle and not triggered.

(a) Raspberry Pi Zero W Board (b) External Battery Pack (c) Micro SDHC Card

(d) HC-SR501 PIR Motion Sensor Module (e) NoIR Pi Camera Module (f) Complete IoT Security System

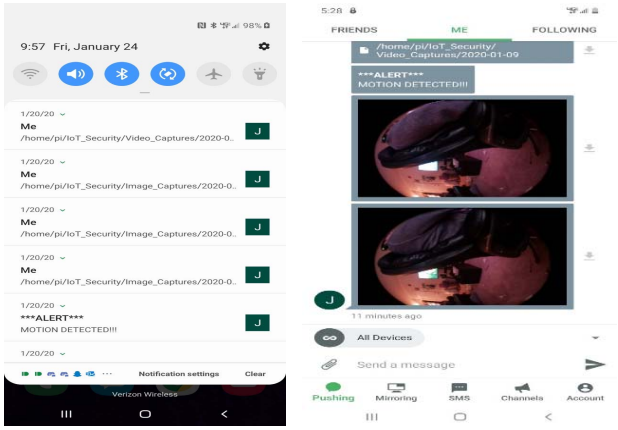
Figure 2. The Hardware components of Proposed IoT Security System

The Samsung Pro Endurance 32GB Micro SDHC Card is specifically designed for video monitoring cameras. It can read up to 100MBs per second and write up to 30MBs per second. It can record up to 43,800 hours of 4K and Full HD (1080p) recording and playback. The external case for the IoT security device provides compact and versatile protection for the electrical components embedded inside. The design of the case is tiny and discrete with mostly black material. It also includes a wall-mounting bracket that is crucial to set up the device at the right angle for surveillance [16-17].

B. Software

We have developed the software part of the project using the python language. A secure shell (SSH) program called PuTTY is used to communicate to the Raspberry Pi Zero W devices without having to connect them to a monitor or keyboard and mouse. To utilize notifications through multiple devices and allow files from the Raspberry Pi Zero W to be transferred through notifications, the open-source PushBullet API was used. After creating an account and downloading the app on the devices you want to be notified through, the program will allow you to be notified through each device set up when motion occurs.

According to our code in the main script of the software, the notifications will include a standard motion alert, a set amount of pictures that can be modified by changing the value of a single variable, and a video that records from beginning of the motion to five seconds after the last picture is taken by the NoIR Pi Camera Module. The five second amount can also be altered by changing a single line of code. Each picture and video are stored onto the Samsung Pro Endurance 32GB Micro SDHC card into separate picture and video files. These same files can be accessed through the open-source PushBullet API through notification, through the Raspberry Pi Zero W itself, and through programs such as WinSCP which can download the files from the Raspberry Pi Zero W onto your local drive. The smartphone screenshots for notification alert from smartphone (Figure 3a) and personal computer (Figure 3b) are shown in Figure 3.



(a) Notification alerts from smartphone (b) Notification alerts from Personal Computer

Figure 3. Text, image, and video alerts from the open-source PushBullet API application

III. DATA COLLECTION

Data collection consists of utilizing both PIR Motion Sensor and NoIR Pi Camera Module in different testing scenarios. Test cases for the PIR Motion Sensor involve checking whether a motion alert occurs when a person walks past the sensor within its range. Also, test cases involving extreme layering of clothing are used to see how well the PIR Motion Sensor detects heat through multiple layers of clothing. We can't test output signals with the NoIR Pi Camera Module like we can with the PIR Motion Sensor, but we can use the camera module to test our authentication access feature. Using the images from the camera module, we can use machine learning to train our IoT security device using the Local Binary Patterns Histograms (LBPH) algorithm with authorized people. If our device detects motion from unauthorized people, the user will be notified with image and video files via smartphone.

A. PIR Motion Sensor

Motion is detected via the PIR Motion Sensor's signal. When the GPIO input of the signal is low, no motion is detected at that current moment. On the other end, when the GPIO input of the signal is high, motion is currently being detected. We set the sensitivity of each PIR Motion Sensor to the medium range, which is about 5 meters. This was done to deter false positives. We set the time delay of each PIR Motion Sensor to the lowest setting, 0.3 seconds, so the sensor will check if motion is being detected or not as much as possible. We also set the repeatable trigger for each PIR Motion Sensor, which means that the GPIO input will remain high until the motion isn't detected anymore. For Figure 4, "VH" is the PIR Motion Sensor's high input and "VL" is the PIR Motion Sensor's low input. Also, "Tx" is the time duration during which the output pin remains

high after triggering and "Ti" is the time period where triggering is inhibited.

B. NoIR Pi Camera Module

When motion is detected, it is important that clear and accurate recordings and captures are made available to the user. While the NoIR Pi Camera Module doesn't detect

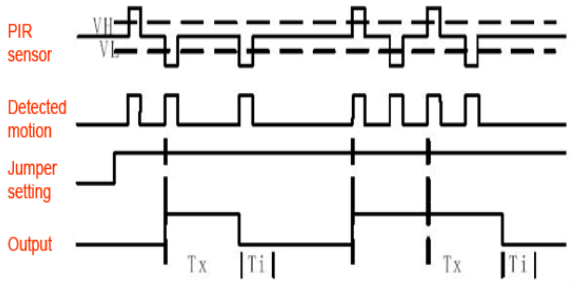
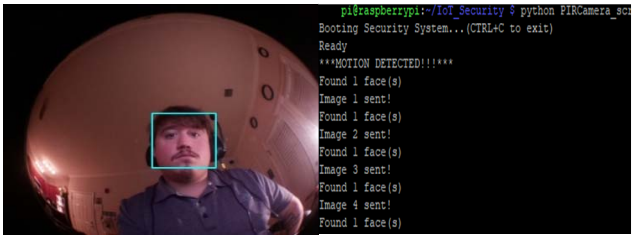


Figure 4. Repeatable trigger waveform for PIR Motion Sensor [18].

motion itself, it will store timestamps, images, and videos for the user based upon the PIR Motion Sensor detecting motion. The images taken from the NoIR Pi Camera Module will also be used for machine learning with the LBPH algorithm. The test cases for the camera module involved testing the facial recognition feature. This was done by running the script and allowing the program to use the NoIR Pi Camera Module and the OpenCV library to check whether the NoIR Pi Camera Module detected a face or not. Examples of the above tests are shown in Figure 5.



(a) Facial Recognition Image (b) Facial Recognition Terminal

Figure 5. The Facial Recognition Feature Using the NoIR Pi Camera module

C. Facial Recognition Feature

In order to combine an authentication system for our *Detecting-Motion-Sending-Notifications* procedure, we utilize one of the most popular algorithms for facial recognition, the Local Binary Patterns Histogram (LBPH) algorithm. This algorithm is based off the local binary pattern (LBP) operator, which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number [19]. Utilizing the LBP operator, we can use LBPH for a facial recognition feature by following steps: LBPH uses 4 parameters: Radius: Used to build the circular LBP and represents the radius around the central pixel. Neighbors: The quantitative value of sample points to build the circular LBP [20-21]. Grid X: The quantitative value of horizontal cells. Grid Y: The quantitative value of vertical cells. To train the algorithm, a dataset must be created with facial images of authenticated people in combination with an ID value. By applying the LBP operator, we create a new image, based up, on the original image, with highlighted facial characteristics.

This is done using two of our parameters: radius and neighbors. To create this new image, the following steps of the LBP procedure will be done: Split a part of the image into a window of 3x3 pixels, which is equivalent to a 3x3 matrix [22-24]. The central value of the matrix is taken to define the new values from the 8 neighbors. A new binary value is set for each neighbor of the central value [25-26]. We set the binary value to 1 if the neighbor value is equal to or higher than the central value. The binary value is set to 0 if the neighbor value is lower than the central value. The matrix will contain only binary values. We will add each binary value from each neighboring position from the matrix into a new binary value. Once we have that new binary value, we convert it to a decimal value and set it to the central value of the matrix, which equates to a pixel from the original image. Now that we have our new image, we extract the histograms using the grid x and grid y parameters by dividing the image into multiple grids.

Each histogram will contain 256 positions representing pixel intensity. Each histogram will be added together to create a new, bigger histogram. This new histogram will represent the characteristics of the original image. The LBPH algorithm is trained at this point. Each final histogram created will represent each image. Using the input image captured from the NoIR Pi Camera Module, we create a final histogram for it and compare it to the final histograms in our database of authenticated people. The formula we use to compare the two final histograms is the Euclidean distance:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2} \quad (1)$$

The algorithm output will be the ID from the image with the closest final histogram and the calculated distance. If the calculated distance is lower than the central value, the algorithm has successfully recognized the input image [27-31].

IV. Result Analysis

In this section, we introduce a developed approach to use our PIR Motion Sensor module to detect motion, our NoIR Pi Camera Module to create image and video files, and the open-source PushBullet API to send notifications of motion with the files to the user's device(s). We describe an approach to implement an authentication system using the Local Binary Patterns Histogram algorithm with our image and video files.

A. Detecting Motion and Sending Notifications

Considered a smart-home IoT environment, we first execute Detecting-Motion-Sending-Notifications procedure that will return a text notification, image files, and a video file when the PIR Motion Sensor module detects motion. Our *Detecting-Motion-Sending-Notifications* algorithm is executed with the following steps: Import required dependencies and initialize each input: pb (PushBullet API), GPIO (PIR Motion Sensor), camera (NoIR Pi Camera Module). Create a while loop equal to True so that the IoT security device will continue to run this algorithm after the reset occurs. When the GPIO input is equal to 1, send a text notification to the user by using the push_note() method for our pb input. Set up the video_file with a timestamp and file type, and use the start_recording() method for our camera input to capture

video. Create a for loop for how many images the user would like to receive per motion detected. After capturing and opening the image file with a timetistamp, use the uploadfile() method on our pb input to prepare the image file to send. Use the push_file() method on our pb input in combination with each image file created to send to the user via smartphone notification. Use the stop_recording() method on our camera input and use the uploadfile() method on our pb input to prepare the video file to send. Use the push_file() method on our pb input in combination with the video file created to send to the user via smartphone notification.

Algorithm 1 ***Detecting-Motion-Sending-Notifications***
Inputs: $PushBullet\ (pb), GPIO, Pi\ Camera$
Output: $Notification\ Alert = f\ (motion, image\_file, video\_file)$

1. Initialize $pb \leftarrow api\_key$, $GPIO \leftarrow$ PIR_PIN, $Pi\ Camera \leftarrow camera$
2. set tripped_status = True;
3. **while** (tripped_status == True) **do**
4. **if** (GPIO.input(PIR_PIN) == 1) **do**
5. set push ← pb.push_note() method to send text to user via **Notification Alert**;
6. set video_file with the timestamp and file type;
7. use camera.start_recording() method to start capturing for video_file;
8. set img_count for desired number of image files to create;
9. **for** (i in range(img_count)) **do**
10. open image_file with image_str to include the timestamp and file type;
11. use camera.capture() method to take an image on the NoIR Pi Camera Module;
12. **with(**(open(str(image_str), 'rb')) as pic
13. use pb.uploadfile() to prepare image_file;
14. **end with**
15. set push ← pb.push_file() method to send image_file to user via **Notification Alert**;
16. **end for**
17. use camera.stop_recording() to end the capture for video_file;
18. **with** (open(video_file, 'rb')) as vid use pb.uploadfile() to prepare video_file;
19. **with end**
20. set push = pb.push_file() method to send video_file to user via **Notification Alert**;
21. **while end**

The pseudocode of *Detecting-Motion-Sending-Notifications* is described in Algorithm 1 in further detail.

When testing the IoT security device's motion detection, we would run our *Detecting-Motion-Sending-Notifications* procedure multiple times in different scenarios. When the security device was active, the sample person would walk within the 7-meter range 50 times to test whether the device would detect the motion. The security device would detect the motion 49 times when the sample person would get within range of the PIR Motion Sensor and failed to detect motion once. When testing whether the device would trigger when no actual motion occurred, the device did not detect any motion tested 50 times each at one-minute intervals.

n = 100	Predicted Motion Detected: No	Predicted Motion Detected: Yes		
Actual Motion Detected: No	50	0	100%	0%
Actual Motion Detected: Yes	1	49	98%	2%

Figure 6: Confusion Matrix for Motion Detection Based Classification Accuracy

The confusion matrix for motion detected is shown in Figure 6. Our false positive (FP) equates to 1, false negative (FN) equates to 0, true positive (TP) equates to 49, and true negative (TN) equates to 50. To determine accuracy, we use the formula,

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)}. \quad (2)$$

Using this formula and our test case numbers, we figure out that the accuracy is equal to $\frac{(49+50)}{(49+50+1+0)} = 99\%$. To determine recall, or how often predicted motion is yes, we use the formula,

$$Recall = \frac{TP}{(TP+FN)}. \quad (3)$$

Using this formula and our test case numbers, we infer that the recall is equal to $\frac{49}{(49+0)} = 1$. To determine the precision, we use the formula,

$$Precision = \frac{TP}{(TP+FP)}. \quad (4)$$

Using this number and our test case numbers, we conclude that the precision equates to $\frac{49}{(49+1)} = 98\%$. We find a measurement that represents both precision and recall, the F-measure. To find the F-measure, we can use the formula,

$$F - measure = \frac{(2 \times Recall \times Precision)}{(Recall+Precision)} \quad (5)$$

Using the F-measure formula and our test case numbers, we can conclude that the

$$F - measure = \frac{(2 \times 1 \times 0.98)}{(1+0.98)} = 0.98.$$

In another test case for our *Detecting-Motion-Sending-Notifications* procedure, we would use a stopwatch to detect how fast our IoT security device would send a motion alert text message, an image alert text message, and a video alert text message. The line graph for this test case is shown above in Figure 7. The initial test case showed

extreme outliers with 12 seconds for a text alert and 20.5 for an image alert; however, the IoT security device was

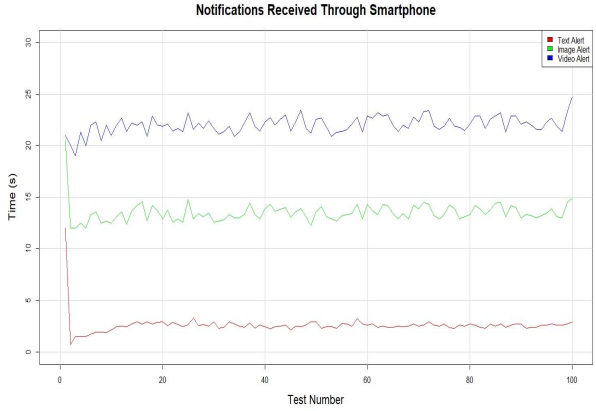


Figure 7. Notifications Response through Smartphone Line Graph

just booted up and could have needed more time for more accurate results. In test numbers 2-100, the numbers for each alert were more consistent with each other.

The average time in seconds for a text alert ended up being 2.60 seconds. The average time in seconds for an image alert ended up being 13.49 seconds. The average time in seconds for a video alert ended up being 22.04 seconds.

B. Facial Recognition

n = 100	Predicted Facial Recognition: No	Predicted Facial Recognition: Yes		
Actual Facial Recognition: No	50	0	100%	0%
Actual Facial Recognition: Yes	8	42	84%	16%

Figure 8. Confusion Matrix for Facial Recognition Based Classification Accuracy

The confusion matrix for facial recognition is displayed in Figure 8. Our false positive (FP) equates to 8, false negative (FN) equates to 0, true positive (TP) equates to 42, and true negative (TN) equates to 50. To determine accuracy, we use the same accuracy formula shown in equation 2. Using this formula and our test case numbers, we figure out that the accuracy is equal to $\frac{(42+52)}{(42+50+8+0)} =$ 92%. To determine recall, or how often predicted motion is yes, we use the formula in equation 3 again. Using this formula and our test case numbers, we infer that the recall for facial recognition is $\frac{42}{(42+0)} = 1$. Similarly, for precision using equation 4, we got $\frac{42}{(42+8)} = 84\%$ accuracy. Finally, the F-measure for facial recognition using equation 5 is $\frac{(2 \times 1 \times 0.84)}{(1+0.84)} = 0.91$.

C. Combined Analysis

Using the data results from our test cases from our *Detecting-Motion-Sending-Notifications* algorithm with an accuracy value of 99% and the facial recognition feature with an accuracy of 92%. The precision value for our *Detecting-Motion-Sending-Notifications* algorithm is 98% and the precision value for the facial recognition feature is 84%, which makes our combined analysis of precision for our IoT security system at 91%.

V. Related Works

A smart IoT security system for a smart-home is not a new idea; however, our concept of authentication is an additional feature that provides further security for a user.

In [32], the authors use an IoT based smart security and home automation system to notify the user of any trespassing and raises an alarm optionally. Unlike our design, the authors in [32] alert the user of every instance of movement that their PIR Motion Sensor detects.

In [33], the authors use an authentication security system to control and monitor the front door of the smart-home. The problem with this is that unauthenticated users may break in the home from another entrance. By avoiding the front door completely, this type of authentication security system will be useless, especially when capturing images and videos of the unauthorized people in the home itself.

Although it is not related to a smart-home, the authors in [34] developed an IoT based smart security and monitoring system for agriculture to identify rodents and threats to crops. These monitoring devices notify the user by delivering real-time data, but the user must visually analyze the data monitored. The attempted test cases provided an 84.8% success rate.

The authors [35] of a seemingly similar device fail to utilize facial recognition and express this fact, fail to express their algorithm for motion detection in a thorough and analytical way and fail to show any quantitative results for their device.

Another similar device for motion detection, but not facial recognition, is created by the author of [36]. Essentially it will detect the intruder through the PIR Motion Sensor and sound a buzzer. This author's faults lie with his human detection system, which is inaccurate when any part of the human body is covered and when the human has less contrast color with the background. Also, the accuracy of the author's motion detection and human detection system is less accurate at 89%, while our IoT security device's accuracy is 92% and contains more features.

Yet another device that we can compare ours to would be the authors in [37]. It is a Raspberry Pi device that includes a connection with a PIR Motion Sensor, a breadboard, and utilizes a NoIR Pi Camera Module and a Wi-Fi dongle. The results of this author's research essentially are just photos taken of the device and low-quality photos taken of a live stream of the device through an open-source application and through the author's pc. No facial recognition is used, no images, videos, or alerts are sent to the user, and the device does not have an external case for all the parts, which leads to negative exposure of the elements and the intruder. Also, with five references used, we cannot guarantee the reliability and research of this author.

VI. Conclusion

In this research, we developed a motion detecting IoT security system that can alert the user in case of any emergency and send notifications to his/her smart devices. Also, being able to utilize the proposed system with four

developed IoT security modules covering different zones of the smart-home's perimeter is crucial to a reliable and accurate security system. The results from tests are also presented to show that this approach provides a high degree of classification accuracy in detecting unauthenticated access in the smart-home.

In the future, we plan to use a more powerful Raspberry Pi with better processing power and collect more data simulating scenarios with different age group test subjects to improve the notification accuracy.

ACKNOWLEDGMENT

We would like to thank the Office of the Division of Mathematics and Computer Science at USC Upstate for the partial funding of this project. We would also like to thank anonymous reviewers for reviewing the early drafts of this paper.

REFERENCES

[1] A. H. Sanoob, J. Roselin and P. Latha, "Smartphone Enabled Intelligent Surveillance System," in IEEE Sensors Journal, vol. 16, no. 5, pp. 1361-1367, March 1, 2016.

[2] D. Pavithra and R. Balakrishnan, "IoT based monitoring and control system for home automation," 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, 2015, pp. 169-173.

[3] G. V. Poonam, and R., Y., Kalshetty. "Bluetooth based smart automation system using Android." International Journal of New Innovations in Engineering and Technology 7.3 (2017): 24-29.

[4] M. A. Mahler, Q., Li and A., Li, "SecureHouse: A home security system based on smartphone sensors," 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom), Kona, HI, 2017, pp. 11-20.

[5] R. M. Green, N., J., Basil. "Mobile device controller application for any security system". US Patent 8,489,065, 2013.

[6] T. Xu B., James. Wendt, and Miodrag Potkonjak. 2014. "Security of IoT systems: design challenges and opportunities." In Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD '14). IEEE Press, 417–423

[7]. W. Mark. "Building a Motion Activated Security Camera with the Raspberry Pi Zero." January 5, 2017.

[8] Q. Gou, L. Yan, Y. Liu, and Y. Li. 2013. "Construction and Strategies in IoT Security System." In Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing (GREENCOM-ITHINGS-CPSCOM '13). IEEE Computer Society, USA, 1129–1132.

[9] L. Ada, T. DiCola. "PIR Motion Sensor." Adafruit, Jan. 2014, https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor?view=all. [Accessed 7 May 2020]

[10] C. Hesse, A. V. Houck, R. Oldenburg. "Pushbullet API" 2015. https://docs.pushbullet.com/. [Accessed 7 May 2020]

[11] R. H. A. Trivedi, H. Mehta, A. B. Upadhyay. "Implementation of Web-Surveillance using Raspberry Pi" International Journal of Engineering Research & Technology (IJERT) Vol. 3 Issue 10, October-2014, IJERT.

[12] A. Shaik, and D. Kishore. "IoT based smart home security system with alert and door access control using smart phone." International Journal of Engineering Research & Technology (IJERT) 5.12 (2016): 504-509.

[13] M. H. A. Sohag & M. A. Ahamed (2015). Smart Home Security System Based on Microcontroller Using Internet and Android Smartphone. In International Conference on Materials, Electronics & Information Engineering, ICMEIE-2015 (pp. 1-5).

[14] A. D. Bagas, and W. A. C. Budi. "IoT-based Integrated Home Security and Monitoring System." Journal of Physics: Conference Series. Vol. 1140. No. 1. IOP Publishing, 2018.

[15] K. Mahesh, et al. "IoT based smart surveillance security system using Raspberry Pi." (2019).

[16] R. Rani, S. Lavanya, and B. Poojitha. "IoT Based Home Security System Using Raspberry Pi with Email and Voice Alert." International Journals of Advanced Research in Computer Science and Software Engineering ISSN (2018): 119-123.

[17] S. K. Chandra, and U. C. Pati. "IoT based intrusion detection system using PIR sensor." 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, 2017.

[18] L. Ada. "BISS0001: Micro Power PIR Motion Detector IC." http://www.ladyada.net/media/sensors/BISS0001.pdf. [Accessed 7 May 2020.]

[19] M. Pietikainen. "Local Binary Patterns" 2010. http://www.scholarpedia.org/article/Local_Binary_Patterns. [Accessed 7 May 2020]

[20] K. S. d., Prado. "Face Recognition: Understanding LBPH Algorithm." Towards Data Science, Nov. 10, 2017. https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b. [Accessed 7 May 2020.]

[21] M. Sahani, C. Nanda, Abhijeet Kumar Sahu and Biswajeet Pattnaik, "Web-Based Online Embedded Door Access Control and Home Security System Based on Face Recognition" 2015 International Conference on Circuit, Power and Computing Technologies [ICCPCT].

[22] D. A. R. Wati and D. Abadianto, "Design of face detection and recognition system for smart home security application," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, 2017, pp. 342-347.

[23] A. Aftab, et al. "LBPH based improved face recognition at low resolution." *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE, 2018.

[24] D. Farah, et al. "LBPH-based Enhanced Real-Time Face Recognition." *Int J Adv Comput Sci Appl* 10.5 (2019).

[25] Z. XueMei, and ChengBing Wei. "A real-time face recognition system based on the improved LBPH algorithm." 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP). IEEE, 2017.

[26] S. Nikolaos, and Dirk van den Heuvel. "Face recognition using local binary patterns histograms (lbph) on an fpga-based system on chip (soc)." 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2016.

[27] S. L. Suma, and S. Raga. "Real Time Face Recognition of Human Faces by using LBPH and Viola Jones Algorithm." International Journal of Scientific Research in Computer Science and Engineering 6.5 (2018): 6-10.

[28] S. Soe, and S. A. Nyein Oo. "Development of a Secured Door Lock System Based on Face Recognition using Raspberry Pi and GSM Module." *Development* 3.5 (2019).

[29] A. M. Jagtap, et al. "A Study of LBPH, Eigenface, Fisherface and Haar-like features for Face recognition using OpenCV." *2019* International Conference on Intelligent Sustainable Systems (ICISS). IEEE, 2019.

[30] A. Dorothy. S. Beatrice. Britto Ramesh Kumar, and J. Jerlin Sharmila. "IoT Based Home Security through Digital Image Processing Algorithms." 2017 World Congress on Computing and Communication Technologies (WCCCT). IEEE, 2017.

[31] Yashwant. "LBPH algorithm for Face Recognition." Open Genus IQ: Learn Computer Science. https://iq.opengenus.org/lbph-algorithm-for-face-recognition/. [Accessed 7 May 2020]

[32] R. K. Kodali, V. Jain, S. Bose and L. Boppana, "IoT based smart security and home automation system," 2016 International Conference on Computing, Communication and Automation (ICCCA), Noida, 2016, pp. 1286-1289.

[33] M. Sahani, C. Nanda, A. K. Sahu and B. Pattnaik, "Web-based online embedded door access control and home security system based on face recognition," 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], Nagercoil, 2015, pp. 1-6. doi: 10.1109/ICCPCT.2015.7159473

[34] T. Baranwal, Nitika and P. K. Pateriya, "Development of IoT based smart security and monitoring devices for agriculture," 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), Noida, 2016, pp. 597-602.

[35] P. Neha, S. Ambatkar, and S. Kakde. "IoT based smart surveillance security system using raspberry Pi." 2017 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2017.

[36] S. Nico & W. Ridwan. (2018). "Design of Smart Home Security System using Object Recognition and PIR Sensor." Procedia Computer Science. 135. 465-472. 10.1016/j.procs.2018.08.198.

[37] D. Arun, L. Abirami, M. Gomathi, G. Manju, S. Siva Sakthi. (2017). "IoT Based Smart Home Security System." International Conference on Latest Innovations in Applied Science, Engineering and Technology (ICLIASET 2017), March 2017.