International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072

# Intelligent Home Security System using Artificial Intelligence

**Kunal Salvi[1], Rayyan Shaikh[1], Prof. Arshi Khan[2]**

*[1]Student, Department of Electronics Engineering, M.H. Saboo Siddik College of Engineering, Byculla, Mumbai, Maharashtra, India*
*[2]Assistant Professor, Department of Electronics Engineering, M.H. Saboo Siddik College of Engineering, Byculla Mumbai, Maharashtra, India*

---

**Abstract -** *Intelligent Home Security System using Artificial Intelligence is the project undertaken to replace traditional means of security with modern IOT and A.I. based systems. Some commercial products based on image recognition are readily available, but such single security level systems can be easily breached. To overcome these drawbacks, we have implemented an A.I. based 2 level security system that can be easily scaled and can be packed with more features without loss in performance.*

***Key Words***: **Artificial Intelligence, Machine Learning, Tensorflow, Image Recognition, Voice Recognition, Spectrogram, IOT, MQTT**.

## 1. INTRODUCTION

There are currently 7.6 billion people on Earth [1]: 3.7 billion are connected to the internet; almost 50% of that connected population lives in Asia, 24% of whom reside in India [2] for which the number comes up to 440 million. Investment in digitisation and urbanisation and friendly regulatory policies hold key to ensuring that India continues to advance on its path of socioeconomic progress. The market potential of all things IOT in India alone is predicted to be $9 billion by 2020[3]. India is one of the key countries poised for large-scale implementation of IOT projects - not only to be able to set new standards but also as a key geography to anticipate the emergence of a new humanism embracing people and devices.[4] A.I. has become a thing of magic now a days and almost every company wants to integrate a part of it in their project. A.I. gives an edge to the devices that traditional hard-coded logic can't compete with.

With each passing day, overlap between IOT devices and A.I. is increasing. The combination of both, one serving as a tool for data acquisition and deployment while the later acting as a tool for computation. Applications of A.I. in IOT are endless and one of them is it use in home security. As majority of Indian homes still use traditional mechanical locks and tower bolts, home breaks are inevitable. Apprehending the culprit is sometimes impossible and important meetings are missed if no one's at home. To counter this, we developed our project by implementing a two-tier security system, which is server based, provides good enough accuracy and provide a lot of features for commercial use.

## 2. Methodology

As home security is the issue of this paper, we went with a two-tier security approach with image identification being one and voice identification being second. The system that we designed is a server-client model, where a Raspberry-Pi and a NodeMCU are clients and an Acer laptop being a server. The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote teaching of basic computer science in schools and in developing countries [5]. The Raspberry Pi 3 used is this project is a Raspberry Pi 3 Model B+ which hosts a Broadcom BCM2837B0 SOC which has 4x Cortex-A53 cores and runs on 1.4GHz.[6] It also has a 10/100 Mbit/s Ethernet, 802.11 b/g/n/ac dual band 2.4/5 GHz wireless, Bluetooth 4.2 BLE.[7] Raspberry Pi is responsible for acquiring data from the subject and transmit the data through TCP sockets to the server. NodeMCU is an open source IoT platform which includes firmware which runs on ESP8266 Wi-Fi SoC from Espressif Systems and the hardware is based on the ESP-12 module.[8][9][10] Due to cost restraint and low speed internet the decision of running a "server" on the laptop was made. The server is a laptop which is connected to the same home Wi-Fi router to which the Raspberry Pi and the NodeMCU is connected. To make the system completely wireless, the data communication between the server and the Raspberry Pi is done through TCP Sockets and the communication between the server and the NodeMCU is done through an IoT protocol called MQTT. MQTT stands for Message Queuing Telemetry Transport and is an ISO standard publish-subscribe-based messaging protocol.[11] It works on top of the TCP/IP protocol. It was designed for connections with remote location where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. In our system, the publisher is the server, the client is a NodeMCU and the server is an open MQTT server provided my iot.eclipse.org. Whenever the subject is in proximity of the Raspberry-Pi, which is connected with a USB camera, the RPi take a bunch of photos and a voice sample of the subject saying the command "Open the door". Theses files are then sent to the server through TCP socket.

The IP address of the server connected to the home router is known to the RPi. After the files are transmitted, the socket communication is closed and the RPi is out of the loop. After the files are received, they are stored folder. The image file is read, resized and moved in the test folder. The wav file is read, and a spectrogram is created. A spectrogram is a graph of spectrum frequencies of a signal as it varies with time. The jpeg file of spectrogram is saved in the test folder. After saving both the files, the image recognition model tests those images and gives a confidence level which is compared against a threshold.

Once the threshold is crossed, that means the both the images are of the authorized personality or administrator. After authentication, the server sends a MQTT message to the NodeMCU on a specific topic. After receiving the message, the NodeMCU operates the server to be moved downwards thus opening the tower lock. If the threshold is not crossed even by one confidence level, the condition holds false and the server sends an email alert with an image file attached and an SMS to the administrator. Both the RPi and the server are in infinite loop and the RPi is runs the python script at boot.

## 3. System Architecture

The block diagram of the system is shown in Fig-1.

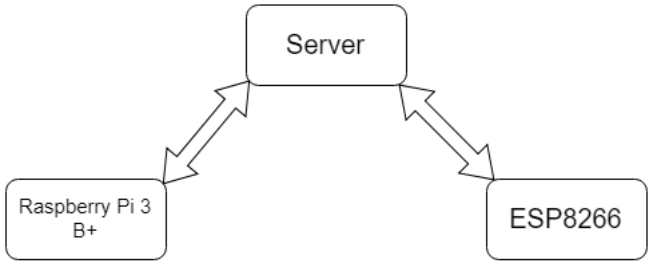


**Fig-1**: System Block Diagram

The RPi has Raspbian OS running on it with all the useful python modules installed. The images are captured using a USB camera instead of a PiCam so that both the voice and wav files can be captured while keeping the cost of materials low. The images are processed with the help of on python library called OpenCV. OpenCV (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision.[12] The voice sample is of duration of 4 seconds and it is recorded using Pyaudio module. The image file is in .jpg format and the voice sample is in .wav format. Both these files are sent to the server through TCP sockets from where all the recognition and control operations take place. After recognition the server sends the appropriate commands to the NodeMCU

## 4. Face and Spectrogram Recognition

Image Recognition, in the context of machine vision, is the ability of software to identify objects, places, people, writing and actions in images. In this project we have used image recognition to identify the face of the authorized personnel and the spectrogram of the voice of the same. To achieve this we used Google's Inception v3, which is a widely-used image recognition model that has been shown to attain greater than 78% accuracy on ImageNet dataset. It is based on the original paper: "Rethinking the Inception Architecture for Computer Vision" by Szegedy, et. al. [12] The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.

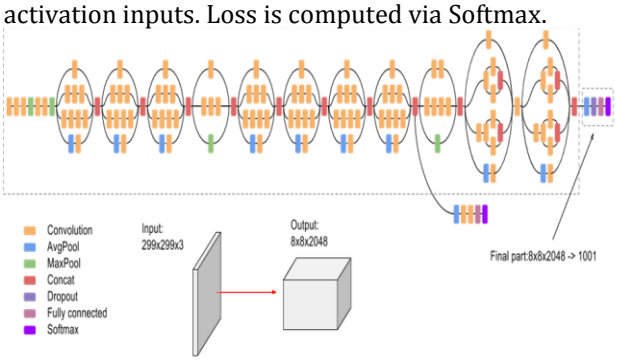


**Fig-2**: Architecture of Inception model

The model is trained on 20,000 images of the administrator and around 10,000 images of the second co-author of this paper. Just like images of faces, images of spectrogram of both, the administrator and the co-author were taken and labelled appropriately. OpenCV was used for acquisition and augmentation of the images. To spectrogram, a certain amount of noise was added while keeping the fidelity of audio intact. With this approach, the need for second model is eliminated and the code size is significantly reduced. But this results in longer augmentation and training sessions. Inception model is based on TensorFlow library from Google. The editors used for this project is eclipse with Pydev plugin, Idle and Arduino. The training of model took almost 6 hours to complete and augmentation took almost 1 hour.

## 5. Results

1] Data Acquisition:

**Fig-3**: Acquired Admin Faces

2] Training of images:

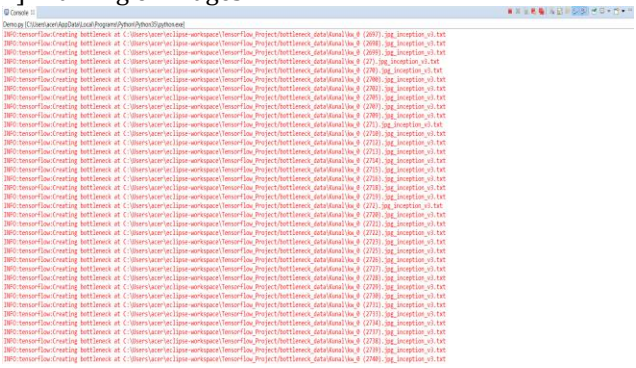

**Fig-4**: Training

3]Completion of Training:

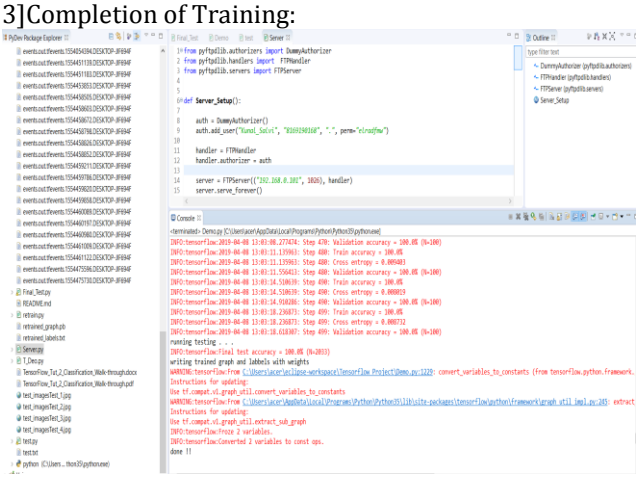

**Fig-5:** After training is done

4] Image Recognition Output:

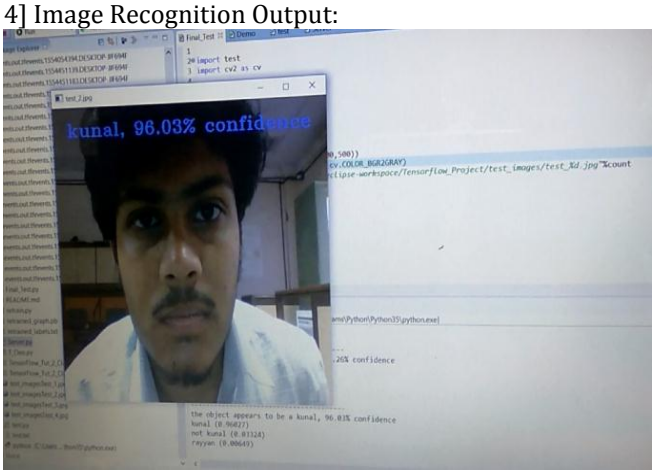


**Fig-6:** Image Recognition Output

5] Voice Spectrogram Output:

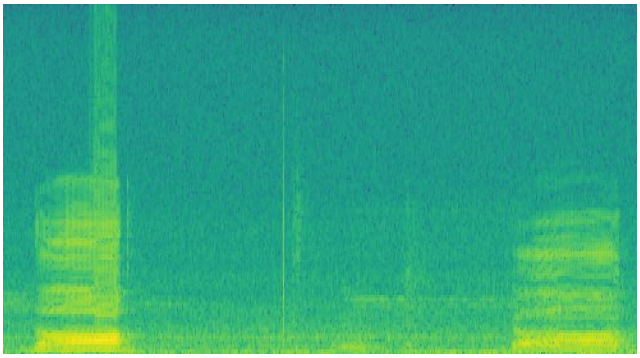

**Fig-7**: Voice Spectrogram of voice signal


**Fig- 8:** Confidence level of voice spectrogram

6] Email Received when the threshold is not crossed

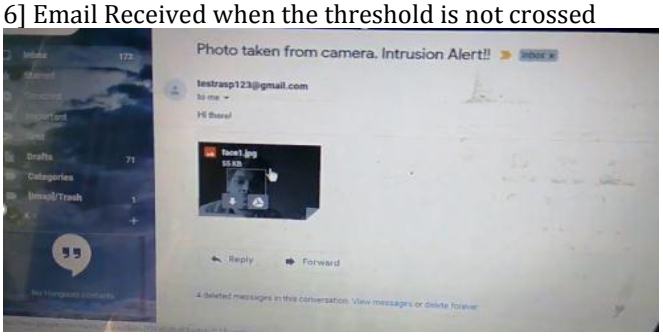


**Fig-9:** Email alert from server

7] SMS Received:


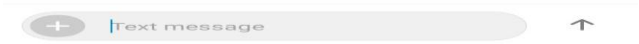


**Fig-10:** SMS from server

8] When door is closed:

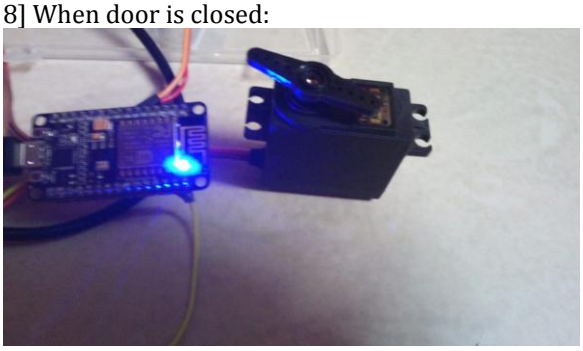

**Fig-11**: Servo motor at 0 degree

**Fig-12:** Servo at 180 degrees

## 6. Conclusion

The scope of the project was to implement principles of A.I. and IoT in the security sector. We were able to implement our face and voice recognition system efficiently and with low bill of materials. The code can be more refined, and more features can be added in order to make it into a full-fledged product for commercial use. Instead of an ESP8266 SoC, a Bluetooth enabled ESP32 can be used in order to add Bluetooth support so that the admin can still enter the house even when there is a power outage.

## References

1. WorldOMeters, Current World Population, January 2018, http://www.worldometers.info/world-population/
2. Internet World Stats, Internet usage statistics, December2017, http://www.internetworldstats.com/stats.html
3. Deloitte, Internet of Things (IoT) to be the next big thing for operators—TMT India Predictions 2017, https://www2.deloitte.com/in/en/pages/technology-media-and-telecommunications/articles/tmt-india-predictions2017-press-release.html
4. India IoT REPORT, Tata Communications, https://www.tatacommunications.com/wp-content/uploads/2018/02/IoT-Report.pdf
5. Cellan-Jones, Rory (5 May 2011). "A£15 computer to inspire young programmers". BBC News.
6. Adams, James. "Compute Module 3+ on sale now from $25". raspberrypi.org. Retrieved 29 January 2019.
7. "Raspberry Pi 3B+ Specs and Benchmarks - The MagPi Magazine". The MagPi Magazine. 14 March 2018. Retrieved 17 August 2018.
8. Zeroday. "A lua based firmware for wifi-soc esp8266". Github. Retrieved 2 April 2015
9. Systems, Espressif. "Espressif Systems". Espressif-WikiDevi. Retrieved 3 June 2017.
10. Brian Benchoff. "A DEV BOARD FOR THE ESP LUA INTERPRETER". Hackaday. Retrieved 2 April 2015.
11. "ISO/IEC 20922:2016 Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1". iso.org. International Organization for Standardization. June 15, 2016
12. Rethinking the Inception Architecture for Computer Vision" by Szegedy, et. al.