

IOT Based Smart Surveillance using Voice Recognition

V.Saravanan, G.Maheswari, G.Aswini, S.Mariya Epsiba

Department of Electronics and Communication Engineering, Jeppiaar SRR Engineering College,
Padur, Chennai, Tamil Nadu, India

vsaranmamba@gmail.com, maheswariganesh12@gmail.com, aswinigajendran26@gmail.com, epsiba.sarah@gmail.com

Abstract– Surveillance has used in lot of applications like home nursing , taking care of elderly people etc. Smart surveillance using voice recognition system is capable of enhancing situational awareness across multiple scales of space and time. It describes IOT based remote control and surveillance architecture using Raspberry Pi. This project make use of sounds around the environment and detect the intrusion using voice recognition technique and using sound sensor detector module we can identify the intrusion using decibel level. Once the recognition is done and the intrusion is found, it sends the alert notification from the server to remote administrator over android application. Then admin can take an appropriate action and alert local security. In this project we use Raspberry Pi as a main source which is the best and efficient platform for IOT. Sound sensor detector module is connected in the raspberry pi board. The data are continuously stored in the cloud. We can able to retrieve the data whenever and wherever we need from the cloud service using internet connection. We use mics to recognize and compare the sounds around the environment.

Keywords– Raspberry pi, Sound sensor detector module, IOT, Voice recognition technique, Cloud service, Android application.

I. INTRODUCTION

Now a day's surveillance is very essential to fulfill our safety aspects as robbery, illegal entry has been a problem in small houses to huge industries. It has several applications. Through the surveillance we can investigate the incidents and also can prevent the potentially catastrophic incidents. The Internet of Things (IoT) is communication network which interrelates computing devices and machines provided with unique identifiers (UIDs) and has the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. Raspberry Pi is a remarkable device fully functional in the computer in a tiny and it is a low-cost package. Sound sensor detector module allows to measure the sound levels in decibels (dB) easily. In our project the voice in the environment is converted to text using Google Speech-to-Text conversion. when the already given input and the current converted word matches then it alerts the admin by using Firebase Cloud Messaging (FCM). The message is sent to the admin in the form of push notification over an android application.

II. PROPOSED SYSTEM

Starting from one camera in each streets and roads video surveillance were introduced to assist the police and guards in preventing crime. The number of cameras is increasing rapidly. Difficulties can occur if one operator has to monitor multiple camera views at the same time. Many Anti-Social behaviors are preceded by some kind of Noise. Example gunshots, breaking glass, screaming.

Cameras with audio detection facilities adds a great value to the surveillance.

Taking into account of all these facts we proposed a new surveillance system using IOT in which most of the work load is done by the computer. The solution is to detect the audio using a mic. It is connected to RASPBERRY Pi and converted into text format using GOOGLE SPEECH-to-TEXT. This further alert the security or the user through Firebase cloud service. The mic continuously listens for the predefined unusual noises and initiates an alert. These are already commercially available surveillance systems but expensive. They are introduced for specific task in a specific environment. The main advantage of our project is that we can receive alerts through mobile phones in case of any emergency through firebase. This makes the system to be effective and can be used by common people who want to take care of family or other official resources. This solution makes both language independent as the google API can understand 100's of languages and privacy protected. This ensures immediate attention by the security operators or the local admin.

III. SYSTEM ARCHITECTURE

This architecture diagram consists of USB Microphone and sound sensor which are connected with Raspberry Pi board and This hardware setup is connected to IOT cloud service through the program running in the Raspberry pi board in Python language. Then the Cloud service send alert message to the Android mobile phones over the Android application.

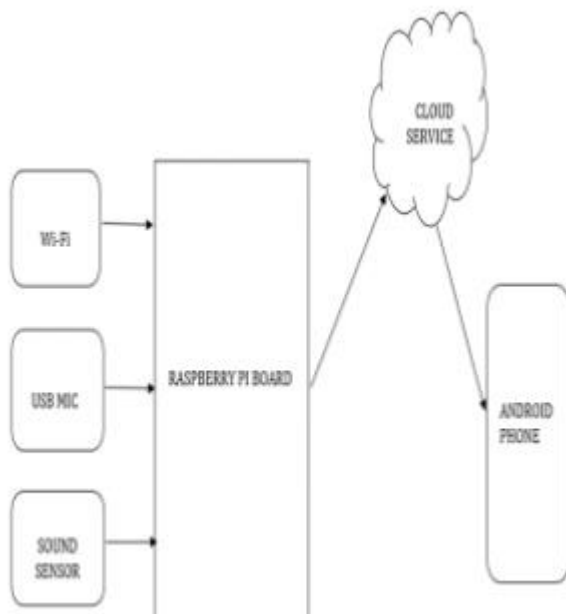


Fig.1. Block Diagram.

1. Raspberry Pi

Raspberry Pi- A credit card sized computer, in which any HDMI input gadget is able to connect and for an operation a keyboard is required. It is a Linux based microcomputer based on ARM architecture. It was mainly built for developing open source game. The device is estimated to cost about \$35 depending on the model. The latest model of Raspberry Pi 3 is Model B+.



Fig. 2. Raspberry Pi.

There are five Raspberry Pi models exist. They are: Model B+, Model A+, Model B, Model A and the Compute Module All these models use the same SoC (System on Chip - combined CPU & GPU), the BCM2835, but other hardware features differ.

• Basic Specifications:

We can use SD Card Slot to install OS/booting/long term storage. It's total memory SD card is about 8GB. Micro USB Power Port provides 5V/2.5A.

If HDMI display is not available RCA Video Out is connected to display. It is used to carry audio and video signals. They are also called as A/V jacks.

Ethernet Port plays significant role in updating and getting new software as it is used to connect to the Internet.

We can use HDMI OUT (High Definition Multimedia Interface) port with HDTVs and monitors with HDMI input.

We can control and interact with real world using GPIO 40 pin interface.

• Raspberry Pi Operating Systems:

The Raspberry Pi runs under Linux form operating systems as well as there is a master version of Linux based kernel well known as Raspbian. Raspbian is a Debian based free operating system that is optimized for Raspberry Pi hardware. Raspbian provides more than a pure OS. It comes with over 35,000 packages and precompiled software bundled in a good format for easy installation on Raspberry pi . An Important note is that the Raspberry Pi does not operate in a Windows environment. To get access to Pi from windows we require Putty Software. Putty is an SSH and TelNet client. The Raspberry pi is an extra powerful device that arises power with some responsibilities. Because of this reason in this project we have made a use of 'python' script for converting the speech to text and comparing it with the given input.

The Raspberry Pi can be used for security system at a relatively low cost. It controls on the concept of the IoT. A security system to be installed can be very costly from security companies, however with some skills and time the Raspberry Pi can be programmed to be a safety by means of as much as we need, for a fraction of the cost.

• Programming The Raspberry pi:

The Raspberry Pi has to be programmed with a suitable programming language to enable communication with the outside world. These languages are Java, FORTRAN, Pascal, Python, C, C++ etc. Each language has its own syntax and meaning. Raspberry Pi can be programmed using any of these languages but in this project, we have used python language.

Python: Python is a high-level language. It is also an interpreted, interactive, as well as object-oriented scripting language. It helps programmers to write clear, logical code for both small- and large-scale projects. Python is a general-purpose language, was designed to be highly understandable. It uses English keywords frequently while other languages use punctuations and has rarer syntactical constructions than other languages. It is administered at runtime by the interpreter so we need not to compile our program before executing it. This is analogous to PERL and PHP. we can actually stand at a Python prompt as well as interact with the interpreter directly to write your programs. Python provisions

Object-Oriented style or the technique of programming which compresses code within objects.

Beginner's Language: Python is an impressive language and easy to learn for the beginners and It supports the development of a wide range of applications from simple text processing to WWW browsers to games. Python is resulting from several other languages, like ABC, Modula-3, C, C++, Algol-68 and other scripting languages.

In our project after installing Raspbian operating system, we have to update it . Then download the python 3.6 and install it. Make python 3.6 as default version. So that the program runs successfully.

2. Usb Mic

The use of Mic is to sense the sound. USB Mic is needed to connect with the Raspberry pi board. The Raspberry Pi is capable of recording audio through its USB 2.0 ports using the advanced Linux sound architecture (ALSA). The Raspberry Pi can sample at 48kHz at a bit depth of 16-bits, which allows the user to record and playback fairly good quality audio. The Python audio analysis is a great tool for engineers interested in acoustic or audio processing and even signal processing techniques.

• Selecting and Recognizing the Usb Microphone:

Selection of microphone is depending on the application of the user. Here we use USB Mini Mic. It filters out unwanted background noise. It is used for VoIP, Voice recognition etc. It is small and portable. Its sensitivity is -47dB to +4dB. Its frequency response is 100Hz to 16kHz. After acquiring the USB mic and plugging it into one of the Raspberry Pi USB ports, we need to check the Raspberry Pi if it is communicating with the audio device or not using this command.

```
pi@raspberrypi: ~ $ lsusb -t
```

If the output is similar to "Class=Audio, Driver=snd-usb-audio" this response, then the USB mic is ready for audio recognition.

Installing the Correct Audio Tools for the Raspberry Pi:

Using the Python library 'pyaudio' to record and play audio data from the USB mic. Before get started with 'pyaudio,' it is needed to ensure that the Raspberry Pi has all the necessary prerequisites by installing the following packages:

```
pi@raspberrypi: ~$ sudo apt-get install libportaudio0 libportaudio2 libportaudiocpp0 portaudio19-dev
```

If this is successful, then we can download the 'pyaudio' library. Here we have installed Python 3.6 with 'pip3'.

```
pi@raspberrypi: ~$ sudo pip3 install pyaudio
```

After the two installs above have been successful, open Python 3.6 and import pyaudio.

TESTING THE USB MIC AND Pyaudio:

```
>>> import pyaudio
>>> p = pyaudio.PyAudio()
>>> for ii in range(p.get_device_count()):
>>> print(p.get_device_info_by_index(ii).get('name'))
```

This should give the output of the index of each audio-capable device on your Pi. We should take note of the index of the USB device, because we need to adjust the pyaudio device index according to the above sequence. For example, our USB device index is "2" (index 0 is ALSA blank, index 1 is IEC958/HDMI, etc). The output .wav file should be 3 seconds long and is sampled at 44.1kHz with a maximum resolution of 16-bits. Depending on the microphone used, the sample rate can be increased to 48kHz. The bit-depth can also be changed. The Pi, with a high-quality microphone, is capable of mid-tier audio recording (16-bit, 48kHz).

3. Sound Sensor Detection Module

The sound sensor detection module is used to detect sound easily and is basically used for detecting sound intensity. This detection module can be used for security, switch, and monitoring applications. It makes use of a microphone which supplies the input to an amplifier, peak detector and buffer. When the sound is detected by the sensor, it processes an output signal voltage and sent it to a microcontroller then performs necessary processing. The sound sensor detection module works by the use of thin piece of material called diaphragm that vibrates when hit by sound waves. It converts by the vibration of diaphragm into an electrical signal that is sent to the LEGO brick, which knows that a sound has been heard.

It has VCC (3.3V DC to 5V DC) pin, GND pin, Output pin. This sensor can able to determine the noise levels within decibels at 3kHz to 6kHz frequencies. It is very simple to use and gives digital output.

• Specifications:

1. Operating voltage 3.3V-5V and operating current is 4~5mA
2. The voltage gain is 26dB
3. Output model: digital switch outputs (0 or 1, which indicates low or high level)
4. With a mounting screw hole
5. PCB size: 3.4cm * 1.6cm
6. It is used in the important areas like Security system for Home/Office, Spy Circuit, Home Automation, Robotics, Smart Phones etc.

4. Google Speech to Text

Google speech to text is used to convert audio to text through neural network models as it has a great Speech Recognition API .It enables easy integration into developer applications . This API converts spoken text into written text, briefly Speech to Text. Main theme of this API application is that send audio to receive a text transcription. It can able to process both real time streaming and prerecorded audio. The API can recognize 120 languages to support the users has excellent results for English language and developers. In our project we use mic to detect the audio which is recognised by google speech to text and it is converted to text format.

Before begin:

1. Sign in to your Google Account.
2. Set up a Cloud Console project.
 - Create or select a project.
 - Enable the Cloud Speech-to-Text API for that project.
 - Create a service account and enable billing.
 - Download a private key as JSON.
3. We should set the environment variable `GOOGLE_APPLICATION_CREDENTIALS` to the path of the JSON file that contains the service account key. It provides authentication to the work, This variable applies only to the current shell session, when a new session is opened, then set the variable again.

Example: Linux or macOS, Windows.

4. Install and initialize the Cloud SDK.

Further to start it has three different methods to work on google speech to text.

Using client libraries or `gcloud`, or command. Here client libraries in python language is used.

This library can be installed in a virtualenv using `pip`. `virtualenv` is a fair tool to create isolated Python environments. The basic problem is one of dependencies and versions, and indirectly permissions. With the help of `virtualenv`, it's possible to install this library without the need of system install permissions, and without clashing with the already installed system dependencies.

To install python package :

- `Pip install --upgrade google-cloud-speech`.
- check the pip list. This shows `google-cloud-speech`.
- Open command prompt and set json path.

`Pip` is a python package installer. The `pip` is installed here to access the required packages. There will be a need to install a few packages: `PyAudio`, `PortAudio` and `Speech Recognition`. `PyAudio 0.2.9` is required and need to compile that manually.

We have created a service account in google speech-to-text.

COST: Cost in a Google speech-to-text platform is free for first one hour. Then the cost is based on seconds when it starts detecting the audio.

• **Speech Requests:**

Cloud Speech-to-Text has three important methods for performing speech recognition. These are listed below:

- **Synchronous Recognition (REST and gRPC):** It sends audio data to the Speech-to-Text API, and performs recognition on the sent data, and it returns results after all audio has been processed. This request method is limited to audio data of 1 minute or less in duration.
- **Asynchronous Recognition (REST and gRPC):** It sends the audio data to Speech-to-Text API and long running operation is initiated. Using this operation, we can periodically poll for recognition results. This type of request is used for audio data of any duration up to 480 minutes.

Streaming Recognition (gRPC only): It performs recognition on audio data which is provided within a `gRPC` bi-directional stream. This type of requests is designed for real-time recognition purposes, like capturing live audio from a microphone. This type of speech request provides in term results while capturing the audio, allows result to appear, for example, while a user is still speaking.

5. **Firestore Cloud Messaging:**

It is formerly called as Google cloud Messaging(`GCM`). It is the cross platform messaging solution . It is used to send messages and notifications for android , iOS and web applications, without having to worry about the server code. It can be used at free of cost and it has no quotas . `FCM` is performed via cellular network or `Wi-Fi`. In the term of responsiveness, `FCM` has a better result (quicker) than `SMS`. It's intended to be reliable, with 98% of messages delivered to connected devices in 500ms or less, as well as massively scalable, with an infrastructure that delivers over a trillion messages `Firestore Cloud Messaging(FCM)` is a part of `Firestore`, it is also used for other `Firestore` services. It supports two types of messaging technique. One is Notification messages and other is Data messages. As our app is in the background ,the `Firestore SDK` will automatically process the message and display it in the device's system .Since the `Android` system has default push notification feature, this is one of the easiest method to send push notification. In data messages we can send customized data elements to the client application. The limits of notification message is 2kB and data message is 4kB.We can distribute the message to a single device or a group of device or the device subscribed to some topics. The purpose of using `Firestore` service is ,it is useful for developing both new application and rewrite an existing one .The dynamic contents can be easily stored and retrieved using `Firestore`. We can develop the application without any form of custom coding the backend (server). It provides automatic data synchronization, messaging, authentication services, analytics, file storage etc.

6. **Android Application:**

To receive the push notification from the cloud service we have to create an android application. Add the project with proper name to `Firestore` console and add `Firestore` to our android app and register the app in `Firestore`. Then download the json file. Using this json file in android studio we have created the basic `FCM` supported app and enabled the push notification and then installed it .Now our android is ready to receive message .Then it will send the .apk file .Then using that apk file we installed the app to whoever needed this project and app .Every device that have application supported with `FCM` will have a `Registration Id` that represent a unique device. `Registration Id` sometime called as token. An application developer can use `HTTP` API protocol to send notification

(in JSON format) to target device with their token. For one request, FCM allow up to 1000 token.

```
{
  "to": "Registrstion Id",
  "notification": {
    "body": "Content of Notification",
    "title": "Title of Notification"
  }
}
```

This is FCM Request Notification Format. FCM message delivery is unpredictable. FCM delivers the push messages to a big portion of the subscribers in a reasonable timeframe (in 10 seconds) . That's why FCM is still feasible to serve as a medium for delivering messages or notifications.

IV. EXPERIMENTAL SETUP AND WORKING



Fig.3. Setup.

Power supply is given to Raspberry Pi board and connect wi-fi to it or else connect the router as an interface between Raspberry Pi and the Monitor. We can connect the Raspberry Pi board directly to HDMI port and for other devices we need router. Then check the IP Address of Raspberry Pi. Using that IP address in command prompt with the help of Secure shell command (SSH) we can access the Raspberry Pi and the documents in it. Add credentials and run the program. It will wait for the response once the speech to text is detected and it will compare with the input. If it is matched it will alert the user. When the sound is above the decibel which is set as the threshold the sensor detect the sound and alert the situation.

VI. CONCLUSION

The ability to detect hazardous events on both the indoor (home) and outdoor(road) can be the matter of life and death, or it can mean the difference between normal life and the life with major handicap. In this work, the proposed monitoring system using Raspberry pi, Microphone and Sound detection sensor has been completed and tested which is linked to a cloud service. It will effectively improve a detection and accuracy rate. Through this project we can provide a highly secured environment. It will be very useful in-home nursing and elderly care. This project will be very efficient in indoor applications.

REFERENCES

- [1]. Neha Patil, Shrikant Ambatkar and Sandeep Kakde "IoT Based Smart Surveillance Security System using Raspberry Pi" conference paper 2017 International conference on Communication and Signal Processing (ICCSP). DOI: 10.1109/ICCSP.2017.8286374
- [2]. Anna Stahlbrost, Annika , Hollosi "Audio Monitoring in Smart Cities" M. Conference Paper 2014.
- [3]. Chinmaya Kaundanya , Omkar Pathak , Akash Nalawade, Sanket Parode" Smart Surveillance System using Raspberry pi and Face Recognition" International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 4, April 2017.
- [4]. Sayed Khushal Shah, Zeenat Tariq, Yugyung Lee" Audio IoT Analytics for Home Automation Safety" 2018 IEEE International Conference on Big Data,5181-5186,2018.
- [5]. D.Pavithra, Ranjith Balakrishnan, "IoT based monitoring and control system for home automation", Proceedings of 2015 Global conference on communication technologies (GCCT 2015).
- [6]. Sharmin Akter, Rehana Afroz Sima, Md. Sohid Ullah, Syed Akhter Hossain "Smart security surveillance using IoT" 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (trends and future directions) (ICRITO).
- [7]. M.Mahadi Adul Jamil, M. Shukriahmad "A pilot study: Development of home automation system via Raspberry Pi" 2015 2nd International Conference on Biomedical Engineering (ICoBE), 30-31 March 2015, PENANG.
- [8]. Dhanshrai Mali R.T. Patil, Nagaraj Dharwadhar Chaitanya R. Devale, OmprakashTembhurne "Real-Time Surveillance System Using Raspberry Pi "Proceedings of International Conference on substantiable Computing in

Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur -India, February 26-28,2019

- [9]. Suchitra Khoje “Implementation of IOT-Based Smart Video Surveillance System.” International Conference on Advances in Intelligent Systems and Computing, P.771-780,2017.

AUTHOR PROFILE



G.Maheswari
Department of Electronics
and Communication
Engineering, Jeppiaar SRK
Engineering College,
Padur, Chennai, Tamil
Nadu, India
maheswariganesh12@gmai
l.com