EL 203
Embedded Hardware Design

Group 46 Project Report

Project on
Safari Pokedex:
Animal Sound Identification System using Raspberry Pi

Instructor: Prof. Amit Bhatt

Group Members:
1. Karnav Kikani (201401067)
2. Aditya Joglekar (201401086)
3. Anusha Phadnis (201401098)
4. Parth Kanakiya (201401100)
5. Rajdeep Pinge (201401103)

Contents

Introduction:

In a forest safari, a variety of sounds of different animals and birds can be heard all around, but a very less proportion of those animals can actually be seen, especially from a big vehicle carrying a large group of people in it. Moreover, it is hard to identify which animal is around, only based on the sound heard inside the vehicle. So, it would be better if there was a device that could identify what animal is in proximity to it, and then inform all others sitting inside, and could add some information about it, along with clicking a picture or taking a video. This would **enhance the Safari experience for tourists** and at the same time allow them to know more about the animals they are there to watch.

Problem Statement (What was promised to be delivered):

When, an animal is nearby and makes a sound, the objective is to identify the animal based on its sound using a Raspberry Pi, and then broadcast its information in real time to the smart-phones connected to it wirelessly. Our project also finds the direction from which the sound is coming, and takes a video by facing the camera in the closest direction out of the four pre defined ones. **We have been sufficiently successful in achieving all the targets mentioned in the proposal. We have also identified the limitations of the project and the possible extensions which have been mentioned at the end of the report.**
We will be demonstrating the above, i.e. classifying animals based on the recorded sound and then taking a picture of it with a webcam using sound sensors.

Theory:

- ### Raspberry Pi:

Raspberry Pi is a small sized, single board Linux computer. It is an example of SoC (System on a Chip), which supports many GPIO ports and I/O ports for connections with a monitor, keyboard, mouse, webcam etc. We are using the Raspberry Pi 2 for **processing of the sound signal** and for creation of a WiFi hotspot to **send messages about the information of the animal** to the phones connected to it.
The Raspberry Pi board can be converted into a wireless access point using a WiFi module. Here, will just use this as a WiFi hotspot to connect the smartphones with the Raspberry Pi, where it will be acting as a server hosting the animal data, and the smartphones will be acting as clients receiving data regarding the detected animal.

- ### Machine Learning algorithm (Random Forest):

For training and classification of the sounds on the Raspberry Pi, we are using an algorithm for machine learning called the ***'Random Forest Classification'*** **algorithm**. This algorithm constructs multiple decision trees during training and then uses ***mode***

classification or ***mean prediction*** to finally output the class which a particular input belongs to. We are using mean prediction for classification.
As we had limited training data (around 40 samples per class), **we used the above mentioned technique rather than using the more popular logistic classifier or a neural network which require a large amount of training data**.

- **Wifi-module and server on Raspberry Pi (local access point and local server):**

One the animal is identified, the corresponding output is stored into a file. This output file is then processed by the server on Raspberry pi and the users connected to local wifi access point are shown an **interactive information** about the classified animal on their smartphones. This information along with pictures of the animal will help in making the safari more enjoyable.

- **Arduino and Sound Sensors:**

Arduino Uno is another embedded microcontroller that can be used to read analog or digital inputs and give analog or digital outputs. We are using Arduino because unlike Raspberry Pi, it can read analog inputs, which is exactly what we need to do in case of the sound sensors that we are using.
The **LM393** sound sensor detects sound and gives the amplitude or intensity of that sound as its output. This can be used to fairly detect the direction from which the sound is coming.

- **Servo Motor:**

Servo motor is a DC motor that can be controlled in terms of its position, which can be specified by giving some specific input to it. It is connected to a motor shield driver that makes the task of giving input to the motor easier. The shield has its input pins connected to the Raspberry Pi and the output pins connected to the motor.
This motor can be attached to any device to make it rotate as much as needed. In this case, a camera is being attached to the motor to make it rotate and take pictures or videos.

Our Solution:

Our solution to the problem statement above consists of the following parts:

1. Recording the sound:

To record the sound, a USB microphone has been connected to the Raspberry Pi through a sound card and it will be placed at a strategic location on the vehicle being used, such that maximum intensity of the sound can be obtained. A button has been connected such that whenever a person hears a sound and wants the whole process to start, he/she can press the button. Whenever the button is pressed, **a script starts running on the**

Raspberry Pi which starts recording audio from the microphone for a fixed interval of time and then sends that recorded audio to the next step.

Problems faced: Raspberry Pi doesn't have the facility to record the analog input. Therefore, we need an **external sound card which will convert the recorded analog sound into a digital sound** at a specific sampling rate. Common sampling rate of 44.1 kHz has been used to take the samples. Also the sample size can be changed based on the requirements. We have experimentally determined the sample size to be approximately minimum of 5 seconds for the correctness of further process.

The learning algorithm is **sensitive to noise**. If the input signal is too noisy, then the machine learning algorithm will have trouble classifying it. Thus the accuracy of the algorithm is dependent on the quality of sound recorded which depends on factors such as the quality of the microphone used, the environment in which the input is recorded.

A solution to make the algorithm robust, will be to **train it on a noisy data, ie take training samples from the environment in which the prediction is to be done, so that on an average the noise present in the environment will be accounted for by the training algorithm.**

2. Classification of the sound:
As mentioned, for classification of the sound, we are using a machine learning algorithm called Random Forest Classification from the standard machine learning open source library called *scikit-learn*[2]. Firstly, **feature extraction needs to be done** from the recorded audio for which we used the python library called *librosa*[3]. Features extracted from the audio consist of

- **Zcr (Zero-crossing rate)** - The rate at which the signal makes sign change transitions. It is used to represent the frequency of the signal
- **RMS (Root Mean Square) energy** - It roughly represents the **'loudness'** of the sound signal
- **Mel- frequency cepstral coefficients (mfcc)** - These coefficients are the current state of the art because they **capture very important information** about the input sound signal. Sounds generated by an animal are filtered by the shape of its vocal tract including tongue, teeth etc. This shape determines what sound comes out. If we can determine the shape accurately, this should give us an **accurate representation of the sound being produced**. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope.
- **Spectral centroid**- The **spectral centroid** is a measure used in sound processing to characterise a spectrum. It indicates where the "center of mass" of the spectrum is. Perceptually, it has a connection with the impression of **"brightness" or intensity of a sound**.

- **Spectral rolloff-** It represents whether the s**ound is “lingering” or no**t. Sounds which are sharp and quick will have faster rolloffs.

Thus each of these features captures a **different aspect of the input signal**. The learning algorithm **assumes that these features are discriminatory** in nature i.e. using these features, it can “learn” to predict the appropriate class of the input sound.
This part of the problem is in the “feature extraction using sound processing” domain. Once features are obtained, all that remains is to run the training algorithm and then use the model generated to predict “unseen, new” input data.
During training (done beforehand), the algorithm works on these features to create decision trees and then stores these trees. After we have extracted features from the audio, we send them for classification which uses mean prediction on the output obtained for various parts of the recording (the class which has occurred in the results the highest number of times, is the class which comes out as the result). This output helps us decide what information to display.

Problems faced:

1. Large amount of time was taken for **installing machine learning libraries** due to storage medium being the external memory card. **Low processing power of Raspberry Pi** was another reason for the time taken.
2. Earlier, the **plan was to train the model on the available data set on the PC** and then copy the trained model to the Raspberry Pi so that it uses the model for prediction when sound is recorded. This is because, machine learning algorithms are computationally intensive and once training is done prediction can be done using the model generated. But the system architecture of both the systems is different. The Raspberry Pi (1 and 2) are 32-bit debian operating system while our PC runs a 64-bit Ubuntu based on Linux. **Hence the PC trained model did not work on the Raspberry Pi.** Therefore, we had to find out other ways of training.
    a. One possible solution is to install a 32-bit python in PC and then train the model but the dependencies on scikit-learn machine learning library made this solution impossible to implement.
    b. Another solution was to take a Raspberry Pi 3, supporting 64-bit debian. This solution was not feasible with the time and cost constraints.
    c. Therefore we had to take the third option which is to **train the whole model on the Raspberry Pi in 32-bit system. The actual training took about 2 hours** every time but the result was successful and the Pi was able to classify the animal correctly.
3. Latency in the prediction - This is again a problem of the low processing power of the raspberry pi platform. **The prediction which could be done on the PC instantaneously(~4-5 sec) took around 30 seconds to finish on the Raspberry Pi 2. Thus, our system classifies an incoming sound signals 30 seconds after it is recorded.**

4. Lack of **wild life data** - Our project had to classify wild animal sounds but unfortunately we were not getting good quality data for these animals. There is a popular ML saying “The algorithm is as good as the data”, if data is not of a good standard the algorithm is helpless. Hence we were forced to use animals for whom good quality datasets were available, but none of these animals were wild.

3. Setting up a local Wifi access point on Raspberry Pi using Wifi module[4]:
Using a WiFi module, the Raspberry Pi board can be converted into a WiFi hotspot server to which any mobile device can connect. This requires basic knowledge of networking. Note that this **does not need internet connectivity,** hence it does not limit the use of this product.

Problems faced: This **seemingly simple task of setting up a Wifi access point** on the Raspberry Pi has many intricacies which must be handled. The main problem was that along with the **local access point and server we had to also start a DHCP server to assign unique IP addresses to the connected devices which is needed in accordance with the networking protocols**. It was very difficult to set up this part since the **uniqueness of IP must be maintained** for each device otherwise there will be a problem of duplicate IPs in the subnet. This problem was solved by us using the isc dhcp server which assigns unique IP address whenever a new host connects to it.

4. Uploading Information on Server and making it visible to the user:
We have created a **local PHP server on the Raspberry Pi**. This server was set up beforehand using a python script and it displays a HTML page which can be seen by any device that is connected to it. The **page is refreshed after a fixed interval** (~10 sec) so that new updates can be seen automatically on the user’s smartphone automatically.
The messages **(that is, the HTML page) consist of information about the animal** that has been detected. It displays a pre-written webpage which gives information to users in accordance with the output of the classification algorithm. The same page can be seen on the smartphones connected to the device through a specific IP address on their browser (Here, the **IP address is 192.168.42.1:9000 where the server port is 9000**).

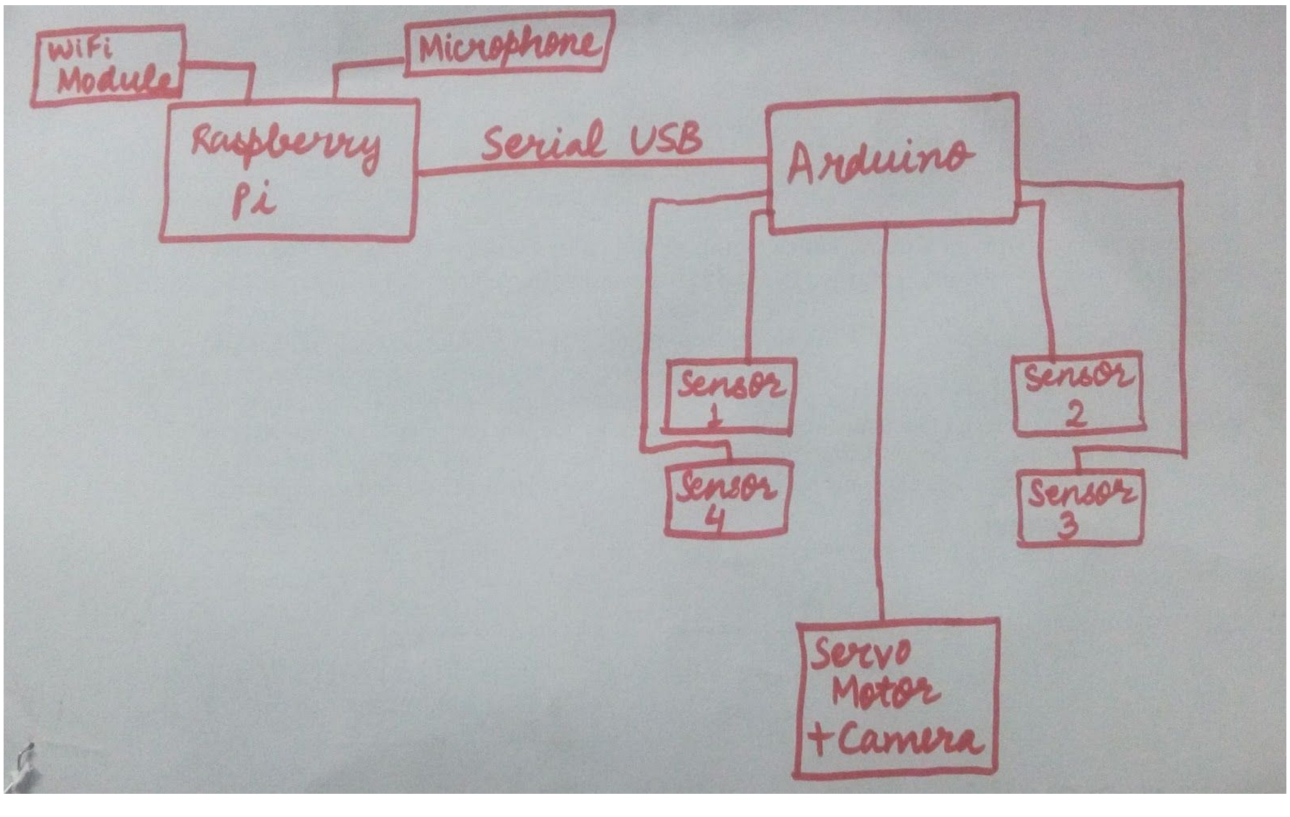
5. Taking pictures using Servo Motor and webcam:
The Raspberry Pi on executing this task, sends a signal to the Arduino to start working. The Arduino is connected to four sound sensors which point in four different directions. It processes which sensor is exhibiting the highest intensity of sound and based on that, sends to the Raspberry Pi via a serial USB cable, the number of steps that the motor should move. This is calculated in the following way:
If the sound intensity is maximum in a particular sensor, then the motor would stop right at the position of that sensor. A camera is attached to the motor which then automatically starts taking a picture or video, as coded in the script written.

The main aim of the algorithm is to enhance the safari experience of the tourist. It has been established that real time feedback is a good way of achieving this. The system thus works as an automated tour guide!

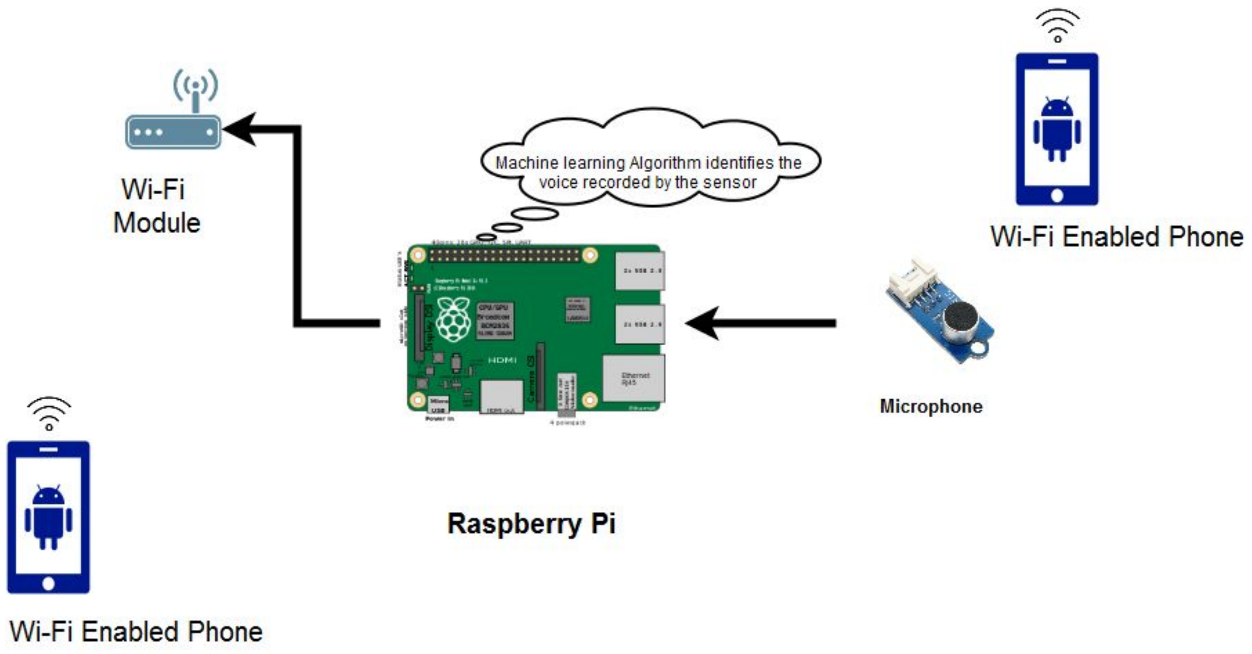
Components Used:

- Raspberry Pi 2 Board
- Microphone (currently we are using a simple headphone)
- WiFi Tenda^R module
- Smartphones
- Push button to trigger the programme using Raspberry Pi GPIO pins.
- Arduino Uno
- Servo motor
- Webcam
- LM393 Sound sensors

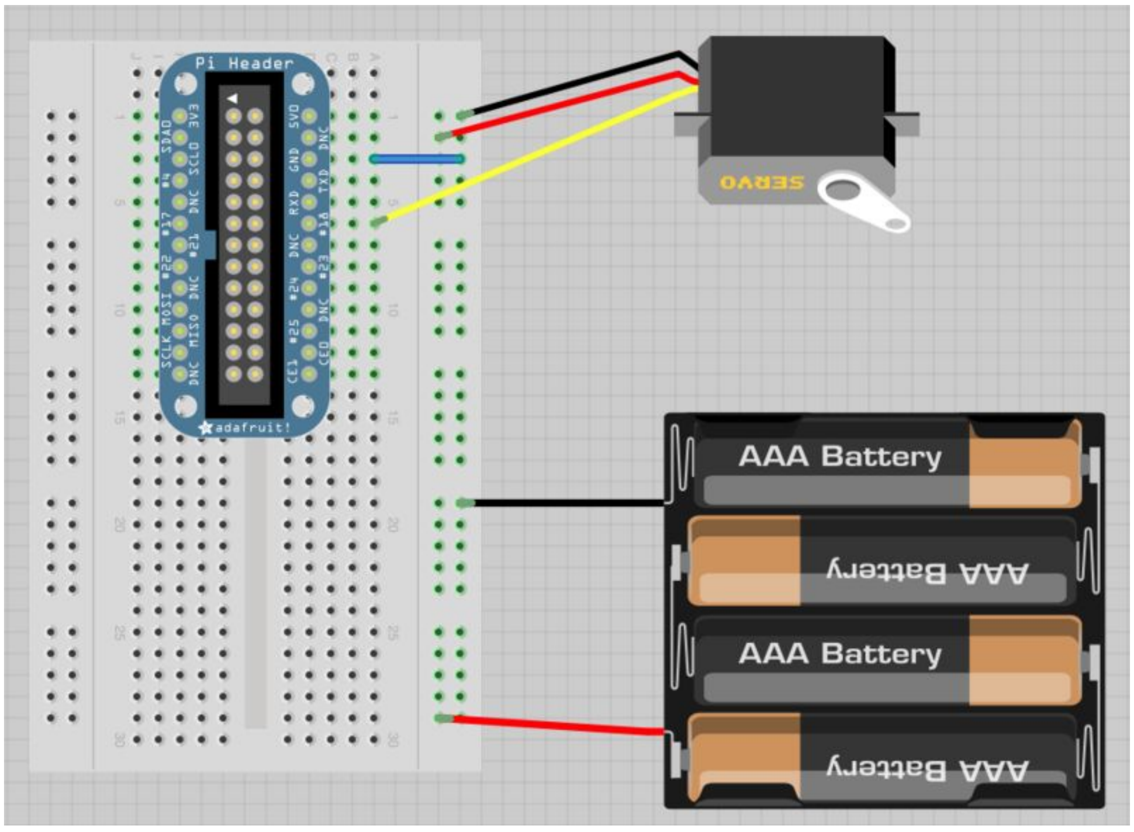
Block diagram of the entire project:



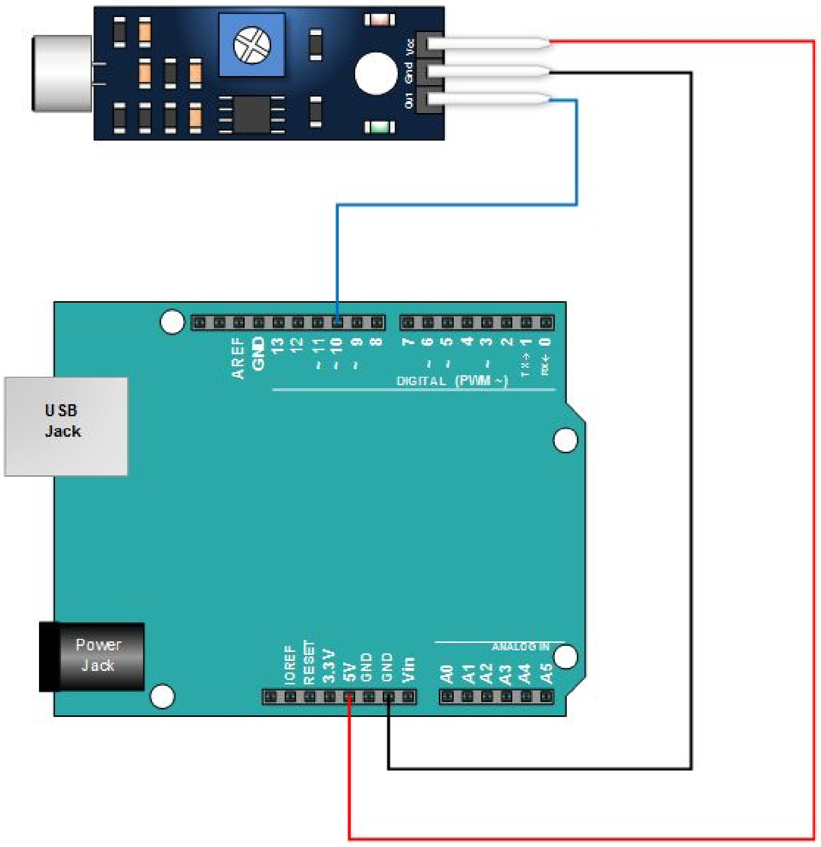
Block Diagram for identification:



Connections for Servo motor and Raspberry Pi:



Connections for Arduino and Sound sensors:



Sample input and output:

A sound sample of a German Shepherd dog (obtained from the Internet) was played near the microphone and the push button was pressed. The output obtained was:

```
Dog
Dog
Dog
Dog
Cat
```

This signified, according to the mean prediction used, that it is a dog's sound.
So, the information about dogs was broadcast in the form of a webpage, to the mobile phones which were connected to the Raspberry Pi's WiFi hotspot.

The model was also extensively tested on random data samples taken from the internet and gave sufficiently correct results.
The system was able to achieve around 85% accuracy. This is pretty good considering the limited data that we had at our disposal.

Conclusion:

This project gives us a fairly good accuracy of detecting animals by their sound. It can easily be attached to a Safari vehicle and can be used efficiently in place of a Safari guide who would have to recognize the sound based on his/her own judgement. However, it has a few limitations which are all due to less efficiency of some hardware components, which can be resolved.

Limitations and Future scope:

The limitations that the project has, can be resolved in the following ways:

1. Noise reduction:
In practical scenarios, especially in a forest, noise would create a problem in classifying the sounds. This could be resolved by using a bandpass filter, or by training the machine with already noisy sounds. This could improve the accuracy of the device. In our dataset, we have some noisy samples to make the system take into account the noise.

2. Microphone sensitivity:
The microphone used is very basic and less sensitive and hence the range of detection gets limited. If we can obtain a very sensitive microphone, then the distance range over which an animal can be detected, would increase. We can also use multiple microphones to increase the quality of recording.

3. Variety:
This algorithm can be used for training the machine with a lot more number of sounds and classes, which would be useful for detection of an increased variety of wildlife.

4. Automation:
If a sensitive sound sensor or microphone can be obtained, then the need for the push button being used, can be eliminated as the sound sensor would be able to detect a change in intensity of the sound around it and would be able to start recording on its own.

5. Extension - creating databases for regions:
This project could be used in region-wise mapping of wildlife in a forest, by keeping a log of the location where it detected a particular animal.

6. **Using actual wild-life data** - Currently, we have trained our system on domestic animals as the more data and better quality was available for it. To make it useful in the jungle, we would need data of wild animals. There are websites which host such data which can be explored.

Acknowledgements:

Firstly, we would like to thank Prof. Amit Bhatt, for making this course enjoyable and practical. During the course we also learnt a lot of important ideas and principles important for our professional lives. The project inspired us to combine our interests, hobbies and our technical expertise to create something unique from it. We learnt about the intricacies involved in the design of embedded systems and had to improvise to deal with these difficulties.

We learnt to make do with the available tools and technologies, sometimes things would not work out, then we had to change our plan after much brainstorming. This brings us to the random group policy, thanks for allocating random groups. It teaches us to deal with people, adjust and to work together to make a good product.

We would also to thank our TA's who answered our doubts, difficulties with great patience and provided honest, cogent insights and suggestions.

It was indeed fun working on this project!

References:

[1] Baby cry project: https://github.com/giulbia/baby_cry_detection/tree/master/ . This project helped us verify that our idea could actually be implemented.

[2] Scikit-learn machine learning library - http://scikit-learn.org/stable/documentation.html

[3] librosa sound analysis and feature extraction library - https://github.com/librosa/librosa

[4] Setting up Wifi Access Point on Raspberry Pi for isc dhcp (internet systems consortium): https://cdn-learn.adafruit.com/downloads/pdf/setting-up-a-raspberry-pi-as-a-wifi-access-point.pdf

[5] Data Set taken from Ph. D. student Mr. Dharmesh who is specifically working on classification of environmental sounds. We are obliged to him for giving us a good quality database and helping us in need.