A siren identification system using deep learning to aid hearing-impaired people

A siren identification system using deep learning to aid hearing-impaired people

Arturo Esquivel Ramirez, Eugenio Donati, and Christos, Chousidis[*].

School of Computing and Engineering, University of West London, St Mary's Road, W55RF, London

Abstract

The research presented in this paper is aiming to address the safety issue that hearing-impaired people are facing when it comes to identifying a siren sound. For that purpose, a siren identification system, using deep learning, was designed, built, and tested. The system consists of a convolutional neural network that used image recognition techniques to identify the presence of a siren by converting the incoming sound into spectrograms. The problem with the lack of datasets for the training of the network was addressed by generating the appropriate data using a variety of siren sounds mixed with relevant environmental noise. A hardware interface was also developed to communicate the detection of a siren with the user, using visual methods. After training the model, the system was extensively tested using realistic scenarios to assess its performance. For the siren sounds that were used for training, the system achieved an accuracy of 98 per cent. For real-world siren sounds, recorded in the central streets of London, the system achieved an accuracy of 91 per cent. When it comes to the operation of the system in noisy environments, the tests showed that the system can identify the presence of siren when this is at a sound level of up to -6db below the background noise. These results prove that the proposed system can be used as a base for the design of a siren-identification application for hearing-impaired people.

Keywords: Siren; Siren detection; Real-time siren detection; Spectrograms; Deep learning for audio; Convolutional Neural Network; Embedded systems; Automobile safety; Road safety; hearing-impaired people; Visual indication

1. Introduction

Among the road safety problems that hearing-impaired are facing in modern society is their inability to identify the sound of sirens around them. This makes people with hearing problems to be further exposed to hazardous conditions, especially when they are driving or walking in a busy urban environment. Sirens are used as the main warning method from ambulances, police, and fire extinguishing vehicles but also to provide warnings in the case of fire, robberies, and other hazardous events. Enabling deaf people to identify the presence of a siren in their surrounding environment improves their safety and reduces their overall stress when they are in public spaces. This consecutively can help in improving their quality of life.

The scope of this research is the development of an efficient siren identification system that will identify the presence and the type of a siren in the surrounding environment and will provide this information to hearing-impaired people.

The development of an automated siren identification system, based on conventional signal processing techniques and algorithmic programming, has a significant level of difficulty as often the siren sound is mixed with high-level background noise [1]. Also, several

[*] Corresponding author
Email addresses: Christos.chousidis@uwl.ac.uk (Christos Chousidis), arturoesquivelrmz@outlook.com (Arturo Esquivel Ramirez), eugenio.donati@uwl.ac.uk; eugenio.donati@gmail.com (Eugenio Donati)

environmental sounds such as music, and car horns can have the same or similar frequency and spectrum to sirens. This increases the complexity of the design of such a system [2].

The work presented in this paper describes the development of an autonomous system that identifies the presence of a siren sound in the environment in real-time and communicate this information using visual signals. The proposed system constantly monitors and analyses the sound from the environment and when a siren sound appears it informs the user. The identification mechanism of the proposed system is based on ML techniques that seem to be the appropriate alternative method to conventional DSP for these types of problems [3]. The system presented in this work utilises a Convolutional Neural Network (CNN) in combination with a classification network. The incoming sound is converted into spectrograms which are inserted into the network that handles sound as a sequence of images. After the classification of the incoming sound is completed, the system communicates with an embedded hardware which is responsible for passing the outcome to the user through visual aids.

The main contribution of this work is the development of a novel, Artificial-Neural-Network based framework for the identification of a siren sound within urban environmental noise. This in turn can be the basis for the development of a standalone device (portable or embedded in vehicles) that can potentially improve road safety for people with partial or total hearing loss.

An additional contribution that results from the above is the creation of a complete dataset of ten thousand labelled data for the training of a neural network (NN), which is the only one of its kind available. In addition, a method of creating this dataset using a combination of MATLAB and a Digital Audio Workstation (DAW) is developed for this purpose. The authors are willing to share this dataset with the research community upon request.

The remaining of this paper is organized as follows. Section 2 presents and analyses the results of a survey, conducted especially to identify the needs of this research. In this survey, a sufficient number of hearing-impaired people were questioned to detect the need and the characteristics of a potential automated siren identification system. Section 3 present the most recent stage in the development of siren detection system. Section 4 describes and analyses the design of the deep neural network used for this application and explains the techniques that have been used to achieve the desired results. In section 5, the hardware interface responsible for the visual communication with the user is presented and analysed. In section 6, the method of generating the training dataset is presented by using the MATLAB computational platform and a Digital Audio Workstation. Section 7 describes the training and validation process of the neural network and the real-time operation of the system. Section 8 is dedicated to the testing and assessment of the system. A series of real-world tests are described, and the objective assessment results are presented and analysed. Finally, in section 9, the conclusions are presented, and the overall development is assessed. Moreover, several further developments are proposed to improve the efficiency, feasibility and the usability of the proposed system.

2. Hearing-impaired people survey

According to the United Nations Office for Disaster Risk Reduction (UNDRR), there is a significant issue when it comes to deaf people encountering hazardous situations and disasters, as these events are mainly communicated to the crowd via siren sounds [4]. To understand the magnitude of the problem and identify the need and the characteristics of a potential automated siren identification system, a survey was conducted within people with diagnosed hearing disability [5] [6]. Thirty (30) people with hearing problems participated in the survey which was conducted in January 2019 with the help of the sign-language interpreter Ms Diana Martinez.

Among the participants, 10% feel unsafe in their environment in terms of hazardous situations which involve their hearing limitations. The majority of the participants identify as their main concern in terms of safety, the inability to identify alarms. 90% of the participants identify the siren sound as the most vital in terms of safety while 96.7% to believe that a warning sound identification system will improve their safety and quality of life.

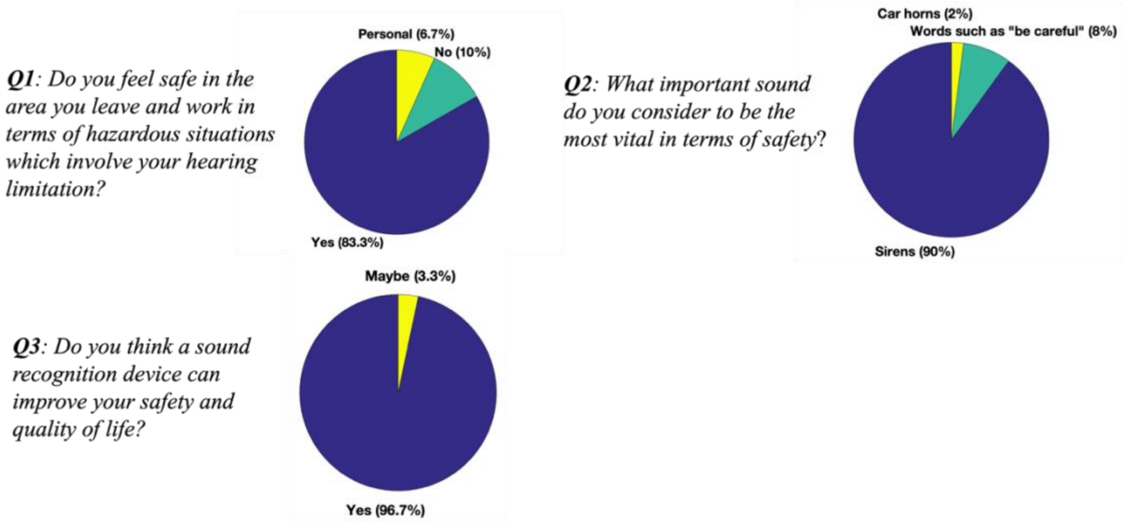

Fig. 1 - Hearing-impaired people survey

Participants in their comments highlight the importance of the development of an automated siren identification devise and believe that this should be able to communicate siren presence information using other senses such as visual signals, vibrations or smell. They also propose that this should be able to be implemented in vehicles as the safety at the road is among their main concern. Figure 1 shows the detailed results of the survey in three key questions.

3. State of the art

Most siren-detection systems that appear in the literature are based on digital signal processing (DSP) techniques. This creates an issue in the operation of these systems, especially with different types of sirens used in the US and Europe but also when the level of siren sound is very close to the environmental noise and especially below it.

F. Meucci et all. in [7] are aiming to resolve the problem of poor identification using spectral analysis, such as FFT or Short Time Fast Discrete Fourier Transform (ST-FDFT). These methods often fail to efficiently detect a siren sound since their acoustic energy is distributed across the spectrum. This is because the siren sound is usually produced by saturated push-pull type amplifiers. In their work, a pitch detection algorithm based on a time-domain technique called Module Difference Function (MDF) is implemented. The system performs well when clean audio is available, but accuracy drops significantly when the signal to noise ratio drops below 0dB. Authors at [8] provide a low computational solution for the problem using a combination of analogue electronics and simple signal processing. The system is based on an analogue envelope detector to determine the energy variation of the siren sound in time. The authors demonstrate the successful operation of their system without providing detailed information about its accuracy. In addition, the system is built and tested only in a simulation environment, and it is limited to the two most common siren signals (yelp and wail), excluding the Hi-Lo and Priority Sirens. J. Schröder et all. are investigating the development of a siren detection system using AI [9]. The authors are comparing five different classifiers, hand-labelled part-based models (PBM-H), positive training data part-based models (PBM+), discriminative part-based models (PBM-), hidden Markov models using mel-spectrogram, (HMM-Mel), and hidden Markov models with mel-frequency cepstral coefficients (HMM-MFCC). The tests were made with both clean and noise-added sound with Signal-to-Noise Ratio (SNR) ranging from 20dB to -20dB. The accuracy results presented in this paper are ranging from 0.86% to 0.56% however, this is because in these tests siren sounds with significantly low SNR are also considered. The main limitation of this research is that it is restricted to German police and, generally, European siren sounds. In addition, a user interface that communicates the results to the user is not included in the system. Finally, the authors in [10] use also a similar method based on an artificial neural network model and MFCCs in combination with a Fourier decomposition method. The accuracy of this method is significantly high reaching values up to 98%. However, the limitations here are first, that much of the training data are not real-world siren sounds but samples collected from the internet, second, the siren sounds are focused on the ambulances and the classification outcome is limited to the presence or non-presence of siren and third, that a real-life test hasn't been performed. Finally, a user interface is not considered here as well.

The model we propose in this paper is aiming to address some of the above-mentioned issues by providing a system that works in a satisfactory manner with all types of siren sounds from Europe and the US, provide a detailed classification in terms of the type of the siren that identifies and also features a user interface, based on embedded hardware. This allows an easy setup and reset of the system and provides a simple and direct communication of the output using visual aids making it suitable for elder and hearing-impaired people.

4. The Neural Network design

The main area in audio where machine learning is employed for sound identification is voice recognition. Within voice recognition, extensive implementation of sound identification can be found in speech recognition (SR). In SR, several machine learning techniques are employed for the recognition of words or sequences of words. Conventionally, most speech recognition applications are based on HMMs. As the short-time stationary characteristics of speech can be thought of as a Markov model, HMMs turn out easy to train and computationally sustainable [11]. However, the main drawback of this approach is that it results in statistically inefficient modelling of non-linear data [12]. Another approach to SR that seeks to tackle the non-linearity issue is the use of Artificial Neural Networks (ANN). The ability of ANNs to find meaning in non-precise and complex data allows a more efficient training for non-linear data when compared to HMM [13]. Examples across the literature have proposed deep learning based SR systems that deliver better performances than traditional HMM approaches. Such an example is proposed by Microsoft in 2012 where the latest version of Microsoft Audio Video Indexing Service yields, based on deep learning, 30% less error than conventional HMM models. Despite the ability of ANNs to classify non-linear data, the inability to represent temporal information makes such an approach highly effective on single words but much less effective on continuous speech [14]. A solution to this issue is proposed in [14] where ANNs are employed for the pre-processing of the speech data to facilitate the learning and deployment of HMM-based systems.

Another area of voice recognition that is extensively addressed in the literature is Singing Voice Detection (SVD) which focuses on identifying the singing voice within a music track. In SVD, the use of ANNs is widely more established when compared to SR. The use of CNNs with image recognition through audio spectrograms appears as one of the most employed methods in the literature. An approach to such a technique is proposed in [15] where Mel spectrograms are generated from the music tracks for the training and deployment of the model. Another example of CNN and image recognition is shown in [16]. where the network is fed with an image representing a map of coefficients generated through STFT. Such implementations highly differ from those employed by SR as their aim is the detection of a singing voice within a larger audio file and not the identification of a specific phoneme.

In this research, a system for the identification of siren sounds through deep learning models is proposed. Even though it does not involve voice, such a system relates to the technique employed in SVD. As opposed to SR, SVD mainly focuses on finding a singing voice within a large piece of audio as, for instance, a music track. On the other hand, SR is usually performed on clear voice signals where the voiced sound is the sole element of the recording. A real-time siren identification aims to detect a siren sound within traffic that, not only is a noisy environment, but also features several environmental sounds. This characteristic shows similarity to the detection and identification of a singing voice using spectrograms and CNNs more than to the identification of a clear phoneme. This led the authors to address the siren detection and identification problem using this approach.

4.1 Convolutional Neural Network

Implementing a deep learning (DL) application for audio requires the design a NN able to accept in its input layer data that contain spectral and time information. The aim of this research is to create the foundation for the development of a standalone application that can be used by elderly and hearing disabled people. Therefore, the limitations when it comes to computational power are important. The technique chosen for this application is based on image recognition. This means that the sound recorded from the environment is converted into a spectrogram and then the pixel values of this generated image are used as the input for the NN. The use of spectrograms is highly suggested for these applications as siren sound, contrary to regular speech and music, creates clear features in terms of spectrum and its variation over time, leading into a high level of accuracy. However, using spectrograms directly to a classic NN requires a very large number of nodes in the input layer consequently increasing the demand for computational power. The proposed application generates spectrograms in grayscale with resolution 134x134 pixels. Each pixel holds a value that represents its level within the grayscale from 0 to 1. This value is used then to feed a neuron at the input layer. Therefore, the size of the input layer is expected to be 17956 neurons. To improve accuracy, the Convolutional Neural Network (CNN) technique was used [17]. This technique extents the size of the network at the input side by adding a “Feature Extraction Network” (FEN) before the actual classification network. The task of FEN is to modify the input data and extract important characteristics such as patterns or shapes. To achieve this, a filtering method is applied using convolution [18].

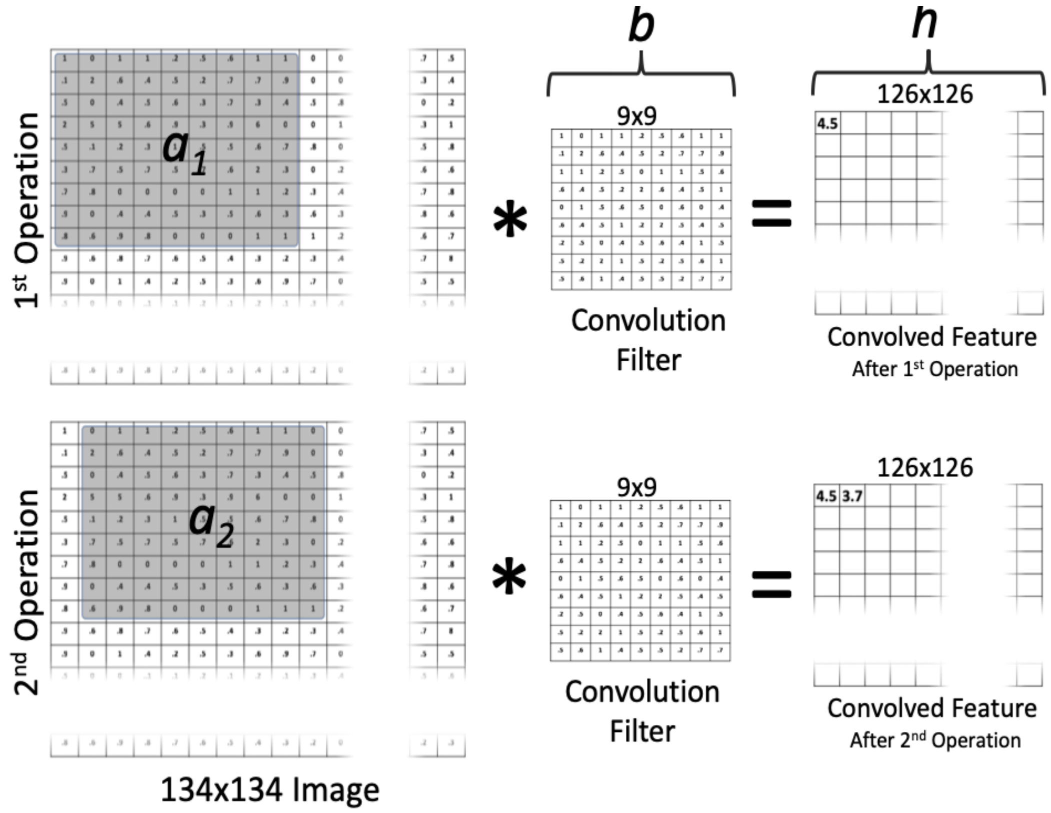

Fig. 2 - Convolution operation at the FEN

The filters are relatively small matrices that convolve successively with the matrix generated by the spectrogram. The convolution filters used in this application are matrices with a size of 9x9 elements and the operation is described in figure 2. Each element of the output matrix *h* is the result of the operation described by equation 1.

$$h[n] = \sum_{n=1}^{N} a_k[n] \cdot b[n] \tag{1}$$

The last layer of the FEN is called the Pooling Layer and it is used to reduce the size of the image. This reduces the required resources in terms of memory, computational power and processing. However, before this layer, the output of the convolution layer passes through the ReLU activation function to keep only relevant data. This function converts any negative value to zero (0) while the rest remains the same.

4.2 Overall structure of the network

The overall structure of the NN is shown in figure 3. The FEN *(in the grey background)* consists of 20 convolution filters. The second layer of the FEN *(pooling layer)* consists of 20 matrices which outputs a size of 20x63x63 elements. This leads to an overall number of 79380 values. The classification network consists of an input layer, one hidden layer, and an output layer. To receive the values of the pooling layer, the input layer of the classifier has 79380 nodes/neurons. The hidden layer consists of a set of 100 nodes. Finally, the output layer has 5 nodes.

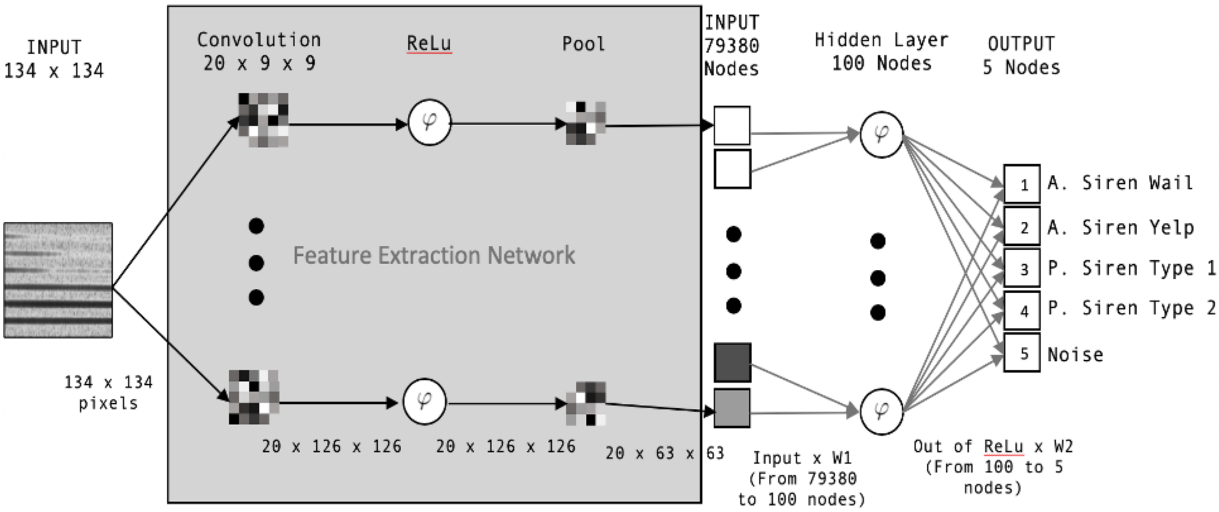

Fig. 3 - Architecture of the overall Neural Network

The system is designed to identify and classify four (4) types of sirens, these are:

- Wail siren
- Yelp siren
- Priority siren type 1 (Hi-Lo)
- Priority siren type 2

It is also able to provide a 5th type of classification when just noise is identified. Therefore, the output layer of the overall NN consists of 5 nodes. Table 1 describes of the overall network.

Layer		**Number of Elements**
Future Extraction Network	Input (Spectrogram)	134 x 134 (17956) Nodes
	Convolution Filters	20 Filtered Spectrograms of 126 x 126 pixels
	ReLU	20 x 126 x 126
	Pool	20 x 63 x 63
Classification Network	Input Layer	79380 Nodes
	Hidden Layer	100 Nodes
	Output Layer	5 Nodes

Table 1 - Elements of the overall neural network

5. The physical interface

The structure of the siren identification system proposed in this paper it is shown in figure 4.

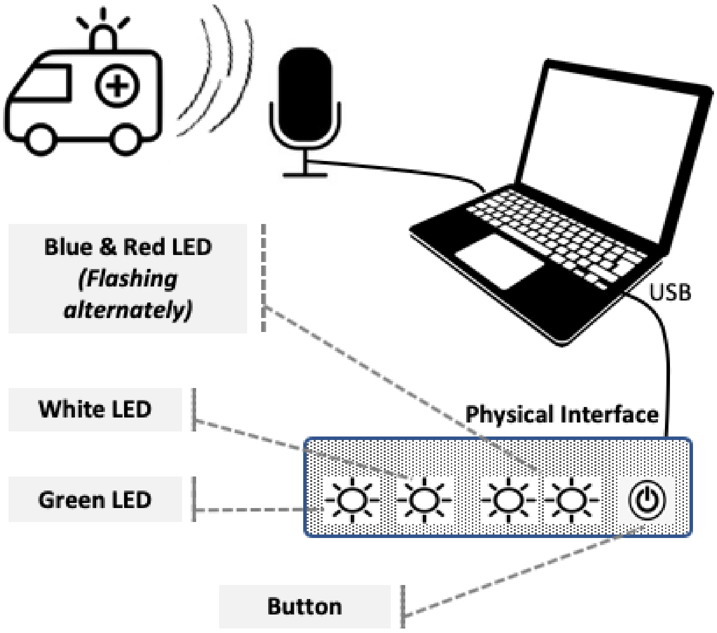

Fig. 4 - System's setup and the Physical Interface

The monitoring of the sound is achieved by using a microphone connected to a computer. The NN is implemented in MATLAB. The network processes incoming sound in real-time and activates one of the 5 output nodes, shown in figure 3. The outcome of the process is then communicated to the user through a physical interface that is connected to the computer via USB [19]. However, the vital information for the end-user is the presence of a siren sound in the vicinity area. Therefore, the physical interface is designed to provide a visual alert when a siren sound is detected without giving any additional information about the type. The interface has 4 coloured LEDs and a button. Table 2 gives a brief description of components' operation.

Component	Function
White LED	Starting/Sleeping device and loading trained data
Green LED	Noise identification *(no siren detected)*
Blue & Red LED	Siren identification *(LEDs Flashing alternately)*
Button	• 1 click: Siren acknowledged by the user • Pressed for 3 seconds: Sleep On/Off • Pressed for 6 seconds: Turn off the system

Table 2 - Physical interface, components and operation

The white LED indicates that the system is loading. This happens when MATLAB runs the code for the first time. Also, when the button is pressed for more than 3 seconds, the white LED indicates that the system is on sleep mode in which no audio is recorded, and only a standby code is running in a while loop (until the button is pressed again). The green LED represents that there is no siren detected, only noise (e.g. traffic noise, crowd noise or environmental sounds such as rain, wind etc.). When the button is pressed for less than 3 seconds, it means that the user has noticed that there is a siren sound present and it deactivates the system for the next 10 seconds. If this happens, the green LED turns on. The red and blue LEDs flash alternately only when the system identifies a siren. These LEDs won't work when the device is in sleep mode or after the acknowledged button is pressed. Finally, if the button is pressed for more than 6 seconds, the device switches off.

The physical interface was build using the Arduino open-source development board that uses the ATMEL ATmega328P microcontroller. This is a low computational power system that is able though to handle the operations of the visual communication and control between MATLAB and the user. The reason that this board was chosen instead a single, standalone microcontroller is that MATLAB provides an “Arduino Support” package that allows communication between the board and the software via USB, in real-time with minimum programming requirements.

It is worth mentioning here that the final system is expected to be an independent application without the requirement of an internet connection, cyberthreats and data protection, therefore, are not expected to be an issue.

6. Generation of the training dataset

One of the main problems that had to be addressed performing this research was the generation of the training data. Deep learning is an efficient method to create computational models when algorithmic methods fail, however, it requires a significant amount of data to train these models [20]. In our case, there is no related database with siren-sounds available for training. Therefore, training data had to be generated as part of this project. The generated training data consists of 10,000 spectrograms using a combination of 3 different samples from each type of siren and 5 different samples of ambient noise that

includes traffic noise, people talking and rain sounds. These samples were a combination of original recordings from the streets in London and downloaded sounds. Table III shows the distribution of the 5 different of data subsets along with their classification (label).

Label	Category	Sample	Spectrograms
1	Wail Siren	3	2,000
2	Yelp Siren	3	2,000
3	Hi-Lo Siren	3	2,000
4	Priority Siren	3	2,000
5	Noise	5	2,000
Total			**10,000**

Table 3 - Training data (distribution and classification)

The samples were placed in a loop of 12 minutes in Logic Pro X. The aim here was to get different spectrograms of the same siren type by including different ambiance sounds. Automation affecting parameters such as: Doppler effect, reverb and EQ, was also included to increase the diversity of the produced data. Figure 5 demonstrates the way that the siren and background noise recordings are arranged. It is shown in this figure that the background sound changes every 2 minutes to create a more realistic set of training data. For each category shown in table 3 a setup similar to one shown in figure 5 was created.

A script created in MATLAB was used to record the sound and to generate and store the spectrograms. To train the system properly, it is necessary to store the spectrograms in a random order. For this, a variable that gets 10,000 random integers was created, to name each spectrogram with a different random number.

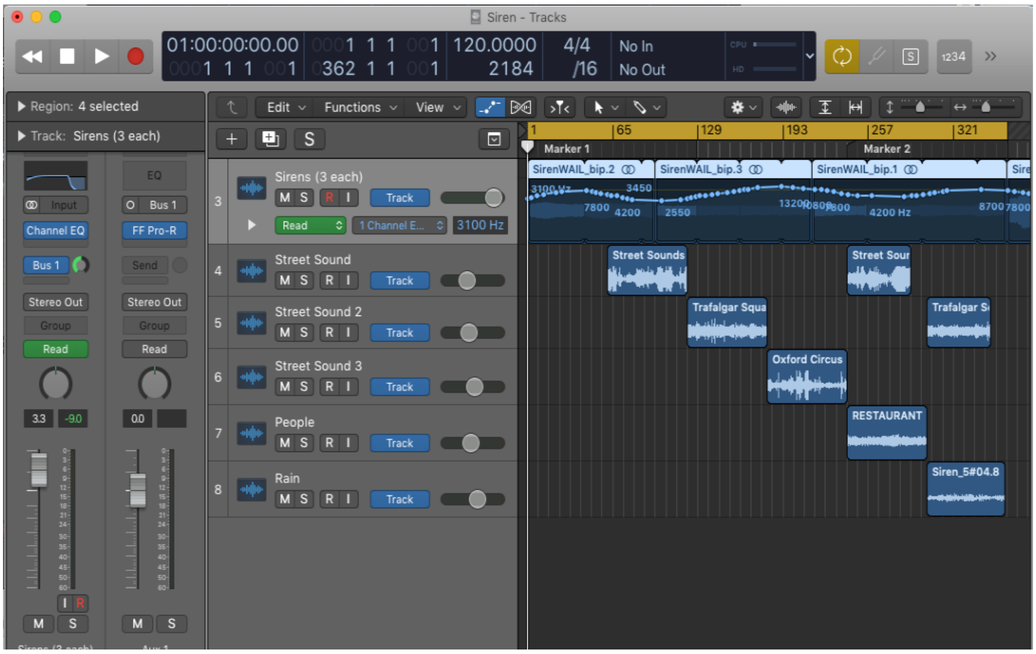

Fig. 5 - Siren and ambient sound arrangement

7. The training process

The first step of the training is to load the training data. A dedicated function was created in MATLAB for that purpose. The function stores all 10,000 images and labels in two new variables. Then, the filters and weights are created using random numbers. 20 filters are created with a matrix size of 9x9, and two sections of weights. Weights are responsible to transfer data from a layer to the next one. For instance, each node of the input layer multiplies a random number and the sum of these multiplications that connects to a node of the next layer is stored in that node. Once the filters and weights are created, the images and labels are split into two groups; the first 8,000 spectrograms are used to train the model, and the rest 2,000 are used for the validation process [21].

In the next step, the training of the network takes place. A training function was also created to perform this task using the basic MATLAB instructions, as long as for this application no deep learning toolboxes are used. For the training process an *epoch loop* is implemented. An epoch represents the process in which an entire dataset is passed once, both forward and backward through a NN. However, this is too big information to be handled by the computer at once, so, the dataset is divided into several batches. For the dataset of 8,000 spectrograms, 80 batches are used, which contains 100 mini-batch loops. All these conforms an epoch loop and there are 3 epoch loops in this system.

For every batch loop, the training function restores the delta-weights to zeroes, (Back-Propagation process), in order to make calculations for every 100 spectrograms. However, at the end of the process, the weights are averaged and summed. After the model has been trained, the updated weights and filters are stored in a MATLAB file. The following steps give a brief description of the training process.

1. Load spectrogram.
2. Apply filters through convolution.
3. Process with ReLU Activation Function.
4. Reduce image size in the Pooling Layer.
5. Get weighted summed from the Pooling Layer and first weights.
6. Process hidden layer with ReLU Activation Function.
7. Get weighted summed from the hidden Layer and second weights.
8. Get output using Softmax Function.
9. Back-Propagate using Delta Function and apply Momentum equation.
10. Repeat from Step 2 a hundred times and then update Weights and Filters

Lastly, a testing function is called for the validation process with the 2,000 remaining spectrograms. The testing function is very similar to the training function that described in this section; the only difference is that at the end it compares the output of the network to the correct output, it sums every result and then it averages it, giving the accuracy in percentage.

8. Test and objective evaluation of the system

The results from the testing function after the training process show that the model has been successfully trained with an accuracy of 99.65%.

8.1 Basic operation testing

The first series of tests were made using the same siren samples used for the training. The sound was played by a speaker and recorded through a microphone connected to the computer. In this test, the system was able to identify the siren with an accuracy of 98%. This figure comes from monitoring the reactions of the machine (whether it shows that a siren was recognized or not), over a 200 second trial. In this first test 196 over 200 seconds were categorized successfully. The performance graph in figure 6 shows the results of this test. The next test was performed using real-world scenarios. The siren sounds were recorded on the streets and then was edited into one recording with a length of 200 seconds. In this test, the accuracy dropped to 91% as it can be seen in figure 7.

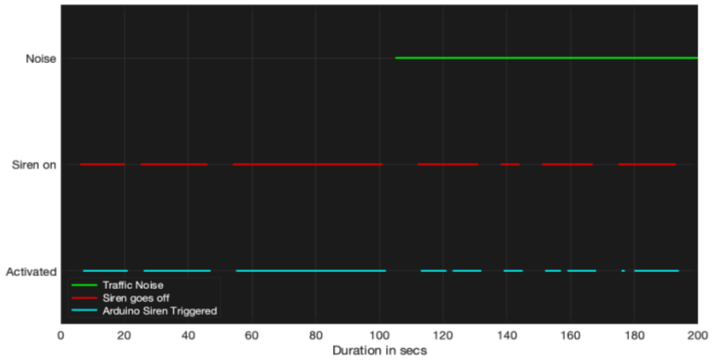

Fig. 6 - Accuracy based on training sounds

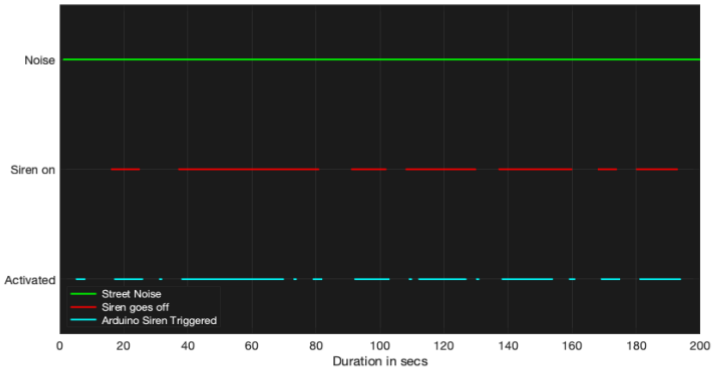

Fig. 7 – Accuracy based on real-world scenarios

8.2 Advanced operation testing

An advanced real-time test was also conducted to show objectively how this system performs in real conditions and in a variety of environments. For this, a 5.7-minute recording was created that includes 7 random cases of ambulance and police sirens (all recorded in London) along with traffic and pedestrian noise. The first three cases took place in Oxford Circus, the fourth case near Camden Town, and the last three cases in Trafalgar Square. Table 4 shows the details of the real-world scenarios used for the test.

Case	Siren Type	Location	Observations
1	Wail & Hi-Lo	Oxford Circus	-
2	N. A	Oxford Circus	Bus sound with high pitch, similar to siren
3	Priority & Wail	Oxford Circus	-
4	Wail	Camden Town	Low traffic and pedestrian noise
5	Yelp & Priority	Trafalgar Square	-
6	Yelp (Hi-Lo very distant)	Trafalgar Square	-
7	Wail & Yelp	Trafalgar Square	Siren distorts (clip gain) and rapid changes of siren type

Table 4 - The 7 real-world scenarios

This test is aiming to investigate the operation of the system by comparing the SPL values between the traffic noise and the siren. In other words, we want to find out how early and at what SPL levels the siren is identified by the system in comparison with the background noise.

The following protocol was created for this purpose. First, for each case, the audio was re-recorded in a noise-controlled environment using the NTi, XL2 audio analyser that can export the SPL data for the whole recording [22]. For the new recording, a spectrum analysis was performed to identify the frequency of the siren. Then, during the time that the siren was audible, a spectrogram was generated every 0.5 seconds in MATLAB. The spectrogram allows us to identify the relationship in dB between the background noise and the siren. Performance graph resulting from this process for all 7 cases are shown in figure 8 to 14. It is important to mention that this is a manual process that requires a significant amount of time to be implemented, however, it provides a clear and accurate illustration of the performance of the system. The green line represents the level of background noise, the red line the level of the siren sound. The blue line shows when the siren identification indicators (blue & red LEDs in figure 4) are active in the physical interface.

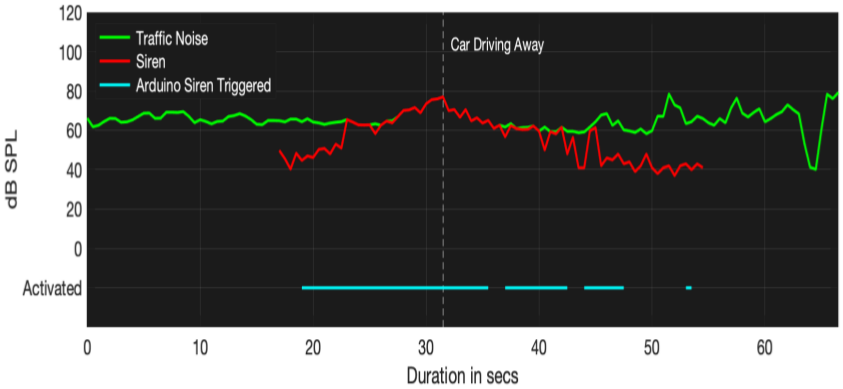

Fig. 8 - Siren identification (case 1)

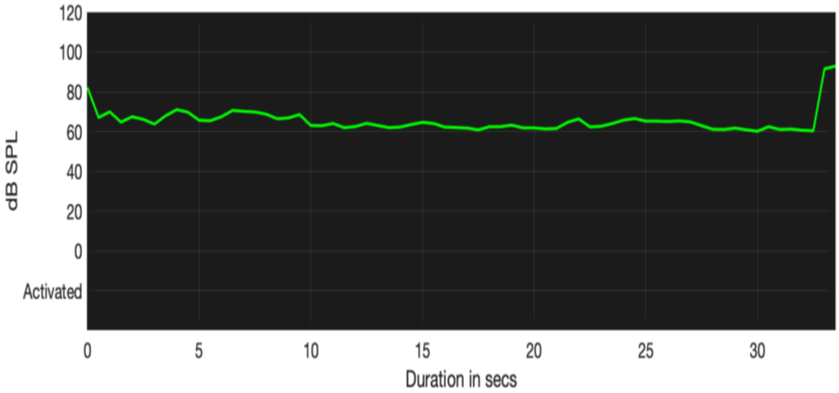

Fig. 9 - Siren identification (case 2)

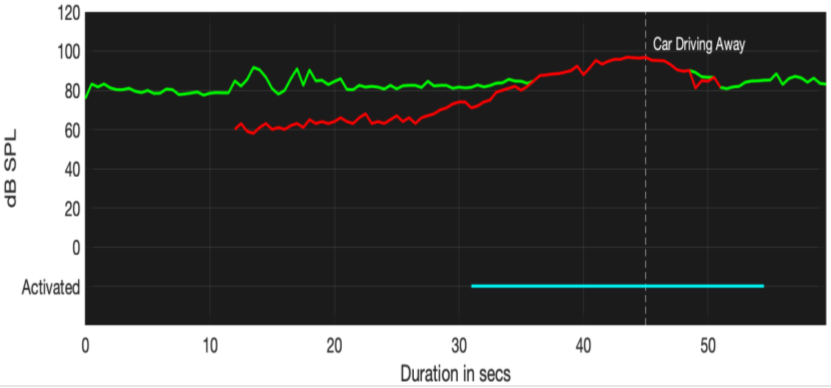

Fig. 10 - Siren identification (case 3)

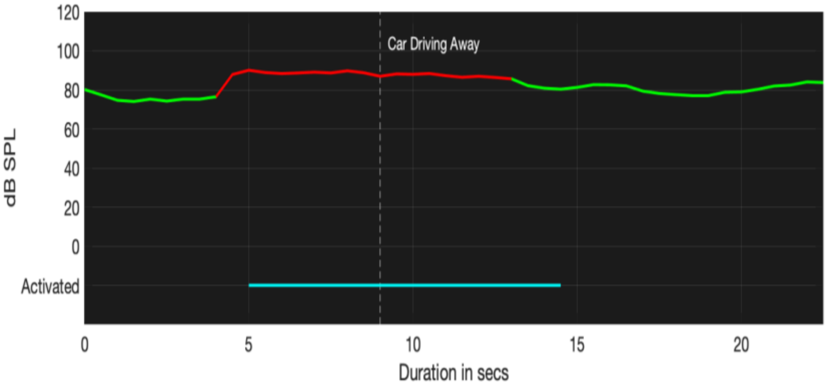

Fig. 11 - Siren identification (case 4)

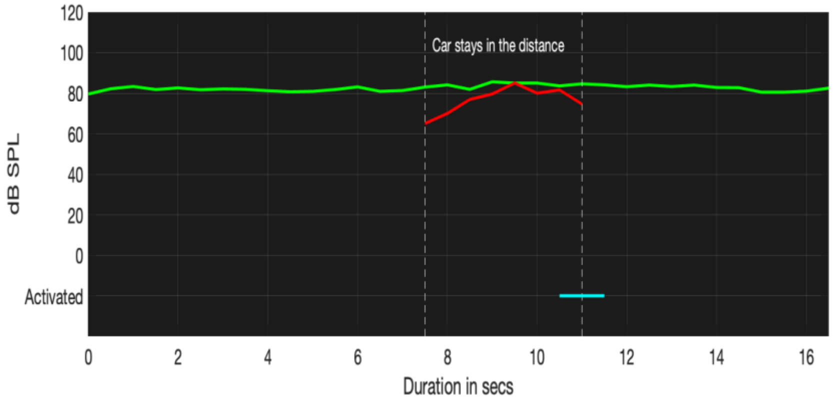

Fig. 12 - Siren identification (case 5)

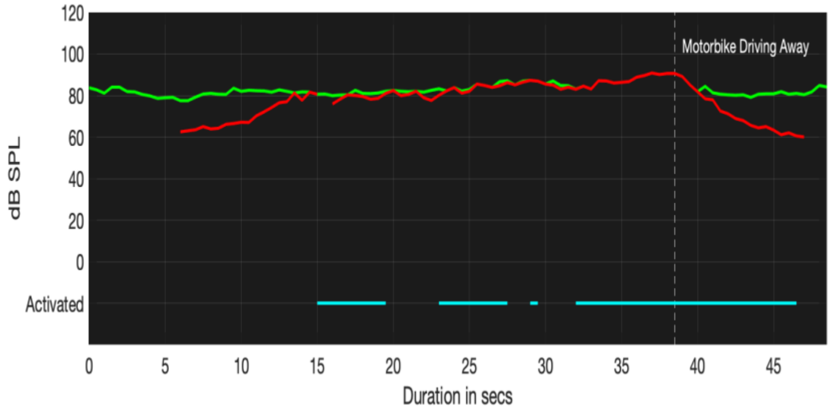

Fig. 13 - Siren identification (case 6)

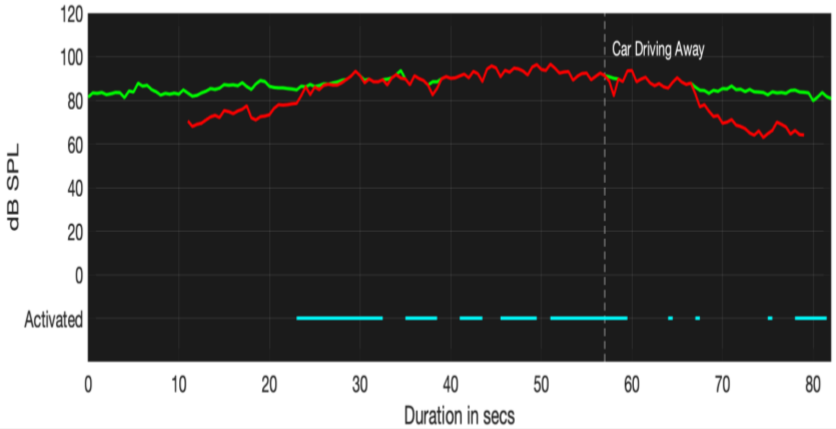

Fig. 14 - Siren identification (case 7)

A quantitative analysis of the test results presented in the seven real-world cases of this advanced operation testing is provided here. It is worth mentioning that this is the only test in the literature of an automated siren detection system that looks at cases where multiple siren sounds are combined, something that is not uncommon in the busy streets of a modern metropolis.

- In case 1, we have simultaneously a "Wail" and a "Hi-Lo" siren recorded in Oxford Circus in central London. It is shown in figure 8 that the system identifies the combined siren sound in real-world environment when this is 18 dB below the environmental noise, and it is able to identify the presence of the combined siren sound for the 70.7% of the time they appear.

- In case 2, we do not have a siren sound. The recording (fig 9) is from a bus the engine of which was producing a high pitch sound similar to a siren. It is shown in this graph that the system successfully distinguished this sound from a siren.
- In case 3, we have a gain a combined sound from two sirens (Priority and Wail), and the behaviour of the system is shown in figure 10. Here the detection of the siren is at achieved at approximately 7dB. below the background noise. For the period that the siren level is at 0dB the system accuracy is 100%. Overall accuracy for the whole period of the siren sound presence is 51.3%.
- Case 4, shown in figure 11 is a single “Wail” type siren, recorded in Trafalgar Square. Here the background noise and the siren are matching, and the system can identify the present of the mixed siren-sound with accuracy of 100% with a delay of 2 seconds in the detection.
- In case 5, we have again a combined siren sound (Yelp & Priority type). The operation of the system is shown in figure 12. The performance of the system in this extreme case is relatively poor (14.3%). This is mainly because the combined siren sound is constantly below the environmental noise and the duration of the sound is only 3.5 seconds.
- Case 6 is again a combined siren sound consists of a “Yelp” and “Hi-Lo” type with the second being at a far distance from the recording position. The behaviour of the system is described in figure 13. The accuracy at 0dB is 72% and the overall accuracy for the whole period of the siren sound presence is 58.5%.
- In case 7 finally, we have again a mixed siren sound consists of “Wail” and “Yelp” type. Here we have a case where the siren is very loud and close to the microphone and causes a distortion in the recorded signal. In addition, the type of siren changes very fast (police car with multiple siren sounds). This extreme case it is described in figure 14. It is shown here that the detection it is achieved at 6dB below the background noise, the accuracy at 0dB is 66.25% and the overall accuracy for the whole period of the siren sound presence is only 48.5%.

8.3 Analysis of the measurement

From the results, it can be seen that, in general, the device recognizes efficiently when a siren is present and when there is just noise in the environment. The model works better when there is low or no street noise, as shown in figures 6 and 7 as well as when the siren sound meets, or it is very close to the noise level. However, the performance does not get directly affected when the siren approaches or moves away from the observation point (case 6 & 7). In most cases, the sirens were not recognized efficiently when their level is 6dB, or more, below the noise level. The graph in figure 15 presents analytically the performance of the system in various conditions.

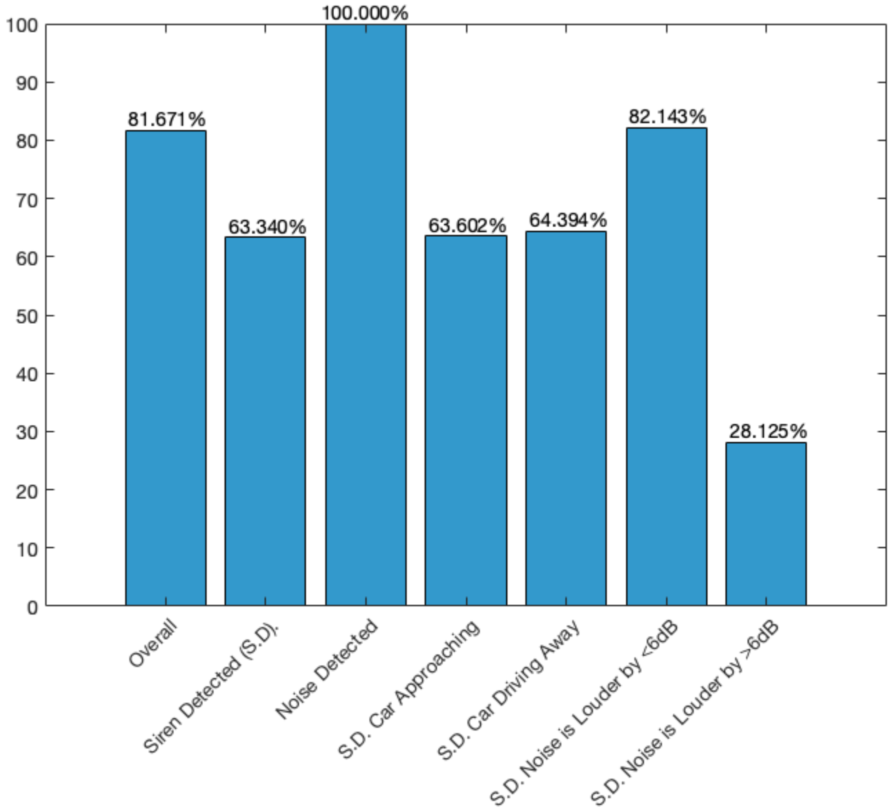

Fig. 15 - Analytic performance results

9. Conclusions

The design, the development and the testing of a siren identification system using deep learning are presented in this paper. A survey that was initially conducted among people with diagnosed hearing disability showed that a siren identification system will improve their safety and quality of life. The system was designed with a proof-of-concept philosophy in mind. A deep NN application was implemented in MATLAB that runs on a regular computer. A hardware interface was also built to visually communicate the sirens detection with the user. In a real-world application, this should be a standalone device or an application that runs in a mobile phone or a car's computing system. An issue that had to be addressed was the lack of available datasets for the training of the system. This was resolved by generating the training data as part of this project. Finally, a series of extensive tests were performed using real-world scenarios to assess the performance of the system. The tests showed that the system is capable of recognizing siren sound especially when the sound level of the siren is close to the level of background noise.

The authors believe that the system performance can be improved by using a more diverse training dataset. A significant development also would be the implementation and testing of the system in a standalone platform or as an application for mobile phones. Finally, the applicability of the system will benefit from the design of a sound localization subsystem employing a circular array that will be able to identify the direction of the incoming siren sound. Further future development would be the enrichments of the system with the capability of identifying additional hazardous events creating in this way an innovative assistive system for the support of hearing-impaired people.

References

[1] Jiang, Xiaodong, et al. "Siren: Context-aware computing for firefighting." International Conference on Pervasive Computing. Springer, Berlin, Heidelberg, 2004.

[2] Joy, Ivan L. "Siren detector." U.S. Patent No. 3,992,656. 16 Nov. 1976.

[3] Uemura, Satoshi, et al. "Outdoor acoustic event identification using sound source separation and deep learning with a quadrotor-embedded microphone array." The Abstracts of the international conference on advanced mechatronics: toward evolutionary fusion of IT and mechatronics: ICAM 2015.6. The Japan Society of Mechanical Engineers, 2015.

[4] McElroy, A. (2013) Sirens sound yet deaf people left standing. Available at: https://www.unisdr.org/archive/34528 (Accessed: December 2018)

[5] Hersh, Marion, James Ohene-Djan, and Saduf Naqvi. "Investigating road safety issues and deaf people in the United Kingdom: an empirical study and recommendations for good practice." Journal of prevention & intervention in the community 38.4 (2010): 290-305.

[6] Ohene-Djan, James, Marion Hersh, and Saduf Naqvi. "Road safety and deaf people: the role of the police." Journal of prevention & intervention in the community 38.4 (2010): 316-331.

[7] F. Meucci, L. Pierucci, E. Del Re, L. Lastrucci and P. Desii, "A real-time siren detector to improve safety of guide in traffic environment," 2008 16th European Signal Processing Conference, 2008, pp. 1-5.

[8] R. A. Dobre, V. A. Niţă, A. Ciobanu, C. Negrescu and D. Stanomir, "Low computational method for siren detection," 2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME), 2015, pp. 291-295, doi: 10.1109/SIITME.2015.7342342.

[9] J. Schröder, S. Goetze, V. Grützmacher and J. Anemüller, "Automatic acoustic siren detection in traffic noise by part-based models," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 493-497, doi: 10.1109/ICASSP.2013.6637696.

[10] B. Fatimah, A. Preethi, V. Hrushikesh, A. Singh B. and H. R. Kotion, "An automatic siren detection algorithm using Fourier Decomposition Method and MFCC," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225414.

[11] Nassif, A.B., Shahin, I., Attili, I., Azzeh, M. and Shaalan, K., 2019. Speech recognition using deep neural networks: A systematic review. IEEE access, 7, pp.19143-19165.

[12] Y.Zhang,‘‘Speechrecognitionusingdeeplearningalgorithms,’’Stanford Univ., Stanford, CA, USA, Tech. Rep., 2013, pp. 1–5. [Online]. Available: https://scholar.google.com/scholar?as_q=Speech+Recognition+Using+ Deep+Learning+Algorithms&as_occt=title&hl=en&as_sdt=0%2C31

[13] Trivedi, P.A., 2014. Introduction to various algorithms of speech recognition: hidden Markov model, dynamic time warping and artificial neural networks. International Journal of Engineering Development and Research, 2(4), pp.3590-3596.

[14] Padmanabhan, J. and Johnson Premkumar, M.J., 2015. Machine learning in automatic speech recognition: A survey. IETE Technical Review, 32(4), pp.240-251

[15] Schlüter, J. and Grill, T., (2015). Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks. In ISMIR (pp. 121-126).

[16] You, S.D.; Liu, C.H.; Chen, W.K. (2018). Comparative study of singing voice detection based on deep neural networks and ensemble learning. Hum.-Centric Comput. Inf. Sci. 2018, 8, 34

[17] Chandna, Pritish, et al. "Monoaural audio source separation using deep convolutional neural networks." International conference on latent variable analysis and signal separation. Springer, Cham, 2017.

[18] Salamon, Justin, and Juan Pablo Bello. "Deep convolutional neural networks and data augmentation for environmental sound classification." IEEE Signal Processing Letters 24.3 (2017): 279-283.

[19] Arduino (2018) Arduino Yun. Available at: https://store.arduino.cc/arduino-yun (Accessed: November 2018)

[20] Ko, Tom, et al. "Audio augmentation for speech recognition." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[21] Shah, T. (2017) About Train, Validation and Test Sets in Machine Learning. Available at: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7 (Accessed: December 2018)

[22] NTi Audio (2018) Sound Level Meter Specifications. Available at: https://www.nti-audio.com/Portals/0/data/en/XL2-Specifications.pdf (Accessed: April 2019)