VIBE - WITH PEOPLE WITH HEARING IMPAIRMENT

Velmurugan S[1] and Prabhakar A[1]
[1]Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai, India-600036

NOTE: Corresponding author: Velmurugan S, ee21s131@smail.iitm.ac.in

Abstract – *Hearing loss significantly impacts daily life, leading to communication difficulties, social isolation, and increased risk of accidents. This can cause frustration, anxiety, and depression. The use of sound indication devices improve the quality of life for individuals with hearing impairment by providing them with a means to detect sounds in their environment. We have developed Vibe, a device that alerts a user to specific sounds (a baby's cry, an alarm, a door knock, a vehicle horn, or a spoken name) in their environment. Our device uses EdgeML, a form of machine learning, to perform tasks locally on the device rather than relying on cloud-based services. This approach has the advantage of reduced latency and improved privacy. The EdgeML model was trained on a locally collected dataset, for better applicability to the Indian environment. Devices such as Vibe will help individuals with hearing impairment navigate their surroundings safely and confidently, leading to increased independence and improved mental health.*

Keywords – Fourier transform, hearing loss, machine learning, neural networks

1. INTRODUCTION

People with hearing impairments will rely on different methods to indicate sounds in their environment. A common method is the use of visual cues, such as hand gestures or facial expressions, to indicate important sounds. For example, if someone hears a knock on the door, they might wave their hand to get the attention of a person with hearing impairment and then point to the door to indicate that someone was there. Animals have also been trained to assist individuals with hearing impairments [1]. Service dogs assist people with disabilities by alerting them to important sounds in their environment. These dogs serve as both companions and ears for their owners, providing them with a greater sense of independence and safety. However, training service dogs requires significant resources, including time and money [2]. Hence, there is also interest in developing automatic sound detection systems that can replace or supplement the role of service dogs.

One of the key challenges in developing good audio localization and detection devices is finding algorithms that can accurately classify environmental sounds, despite variations in the recording conditions and the presence of background noise [3]. Recent advancements in edge machine learning (EdgeML) have made it possible to perform sound classification on low-power devices such as smartphones and wearable devices. EdgeML enables on-device processing, reducing the need for data transfer and increasing privacy. Additionally, it can help overcome challenges related to limited connectivity and resource-constrained environments [4].

Our work begins with collecting diverse environmental sound data, in an Indian environment, including sounds commonly experienced by the hearing-impaired. We process the data using a spectrogram and Mel-Frequency Cepstral Coefficients (MFCCs) [5]. We then label the pre-processed data with the corresponding environmental sound class. Finally, we choose a suitable lightweight machine learning model and train it using the labeled dataset, optimizing its accuracy and deploying it into edge device [6]. These steps have been proposed and validated in the remaining portions of this work.

2. LITERATURE STUDY

The initial step of environmental sound classification involves gathering data. There are a number of openly accessible audio datasets available to researchers. ESC-50, ESC-10, Ultrasound8k, and BDlib datasets have been used extensively in environmental sound classification research using machine learning. These datasets contain high-quality audio recordings of environmental sounds, and their annotations provide ground-truth labels that are essential for supervised learning models. Self-collected sound samples can be useful for specific research projects, but they may require more effort for processing and analysis. Additionally, the quality of the recordings and the accuracy of the annotations may vary, which can affect the performance of machine learning models. Therefore, using established datasets like ESC-50, ESC-10, Ultrasound8k, and BDlib can provide more reliable and consistent results.

Piczak [7] utilized the ESC-50 dataset and achieved recognition accuracy of approximately 81% among untrained human participants, while baseline methods achieved a mean accuracy of 44%. Meanwhile, Toffa et al. [8] proposed an alternative approach for sound classification, combining texture feature Local Binary Pattern (LBP)

with audio features. They evaluated the effectiveness of this approach using support vector machines, random forest, and k-nearest neighbor on the ESC-10 and ESC-50 datasets.

In Piczak's work [7], using a standard classifier yielded an accuracy of 73%, with a separate human benchmark accuracy of 96%. Thwe and War [9] used the ESC-10 dataset for the classification of sound events using time-frequency representations like spectrograms and multi-support vector machines. Salamon and Bello [5] used the ultrasound8K dataset and compared a holistic framework with a solo feature approach based on PCA-enhanced log-scaled mel-spectrogram patches. Their baseline classification accuracy was 68%, while the average classification accuracy was 73.6%.

Divya et al [10] proposed to support individuals with hearing impairments with an Internet of Things (IoT) approach. Their system used a microphone to capture audio signals, which was then processed using digital signal processing techniques such as filtering and amplification. The processed signals were transmitted wirelessly to a smartphone application, which provided visual feedback in the form of text, images, or vibration. The authors evaluated the system using a dataset of audio recordings of speech and environmental sounds. The system achieved high accuracy in recognizing speech and environmental sounds, demonstrating its potential for supporting people with hearing impairments. In a similar study, Garcia-Ortiz and Rodriguez [11] propose a mobile-based solution for sound recognition and classification to support individuals who are deaf or hard of hearing. The study used a dataset of environmental sounds, which was collected using a smartphone app. The dataset included 14 classes of environmental sounds, with a total of 300 recordings. The methodology of the study involved processing the audio recordings using a Convolutional Neural Network (CNN) architecture to classify the sounds into their respective categories. The CNN architecture was optimized using the Adam optimizer, and the model trained using a cross-entropy loss function. The model was then deployed on a mobile device for real-time sound recognition. The authors evaluated the model's performance using a test dataset and achieved an accuracy of 83.33% in sound recognition.

Table 1 – Comparison between different types of open-source datasets

Dataset	Labels	Classes	Duration of the sample	Publication year
Ultrasound8k	8732	10	$\leq 4s$	2014
BDlib	120	12	$\leq 10s$	2015
ESC-50	2000	50	$\leq 5s$	2015
ESC-10	400	10	$\leq 5s$	2015
Ultrasound	1302	10	$\leq 4s$	2014

ProtoSound, introduced by Jain et al [12], allows Deaf and Hard of Hearing (DHH) users to customize a sound recognition model by recording a few examples of the sounds they want to recognize. This enables DHH users to create personalized and fine-grained categories that are tailored to their individual needs. The model was evaluated on two real-world sound datasets and showed significant improvement over state-of-the-art sound recognition systems. In addition, it was deployed on a mobile device and evaluated by 19 hearing participants, who found that it accurately learned sounds across diverse acoustic contexts. In previous work on the Sound-Watch [13], Jain et al utilized a VGG Lite architecture to underscore the significance of carefully considering the number of layers in a deep CNN model. While additional layers can enhance accuracy by capturing more features from the training data, an excessive number may led to increased complexity and memory usage, potentially resulting in overfitting. Jain et al reported a latency of approximately 6 seconds on the edge device and the deployment on the market-available Ticwatch Pro Android watch (with specifications of 4×1.2GHz, 1GB RAM) costing around 15,000 rupees. Their work highlighted potential challenges related to misclassifications, latency, and privacy concerns. Additionally, their findings were based on a relatively small participant pool, which could limit the generalizability of the results.

VisAural [14] is a wearable sound localization device for people with impaired hearing. It uses an array of head-mounted microphones to detect the direction of a sound and places LEDs at the periphery of the user's visual field to guide them to the source of the sound. VisAural was evaluated with nine people with hearing impairments, and the results showed that it was effective in helping them locate sounds. One of the main limitations of VisAural is that it takes a few seconds for VisAural to detect the direction of a sound and update the LEDs. This can be a problem in situations where the user needs to quickly locate a sound, such as when an ambulance is approaching, or a car honking.

Tara Matthews et al [15] devised a prototype peripheral visualization system tailored for non-speech audio, employing a blend of color, motion, and spatial cues to represent distinct sound events. Through a user study involving ten deaf participants, the system demonstrated an impressive 82% accuracy in identifying specific sound occurrences. Users also reported a high level of ease in interpreting the visualizations, with no discernible interference in their ability to carry out concurrent tasks. However, we note that the evaluation was confined to a small-scale study with deaf participants, highlighting the need for further research to gauge the system's effectiveness in broader real-world settings. Additionally, the system was restricted to a limited set of sound events, and lacked the capability to ascertain the direction of sound sources, highlighting a need for further algorithmic development in this regard.

Finally, Liu et al [16] presented UbiEar, a smartphone-based acoustic event sensing and notification system for

hard-of-hearing people. UbiEar uses a lightweight Deep Convolution Neural Network (DCNN) to enable location-independent acoustic event recognition on commodity smartphones. It also includes a set of mechanisms for prompt and energy-efficient acoustic sensing. UbiEar was evaluated in a controlled experiment with 86 hard-of-hearing students. The results showed that UbiEar can achieve an average accuracy of 91.2% in acoustic event recognition, even in noisy environments. UbiEar was also found to be energy-efficient, consuming only 0.1% of the battery life per hour. However, the device was only evaluated in a controlled environment with a limited set of sounds.

Table 2 summarizes the different spectral approaches and datasets used by researchers. Their work has accuracy ranging from 70-90%. Therefore, the choice of a suitable dataset depends on the purpose of the sound classification task, and the accuracy of classification can be improved by selecting appropriate methodologies and algorithms. Instead of relying on pre-existing datasets, we gathered our own data for training the machine learning model. This was necessary because the characteristics of sound samples in our specific region of India are distinct from those found in widely available datasets. The necessity of using a self-collected dataset is explained in detail in Section 5. For example, the Indian vehicle horn differs significantly from a foreign vehicle horn, both in perception and in the presence of a lot of ambient noise.

Our current research focuses on a tailored solution for individuals with hearing impairments (DHH). We aim to develop a highly efficient and cost-effective model that minimizes resource requirements while maximizing portability and accessibility. Through the integration of EdgeML techniques, we aim to create a solution that not only conserves resources but also significantly lowers costs. Our priority lies in achieving the lowest possible latency and power consumption, ensuring that our technology seamlessly integrates into daily life.

Table 2 – Comparison of different datasets, methodologies and accuracy

Authors Used	Dataset	Method		Accuracy
		Feature extraction	Algorithm	
Mustaq and Su[[17]]	ESC-50	Mel spectrogram	DenseNet	84.66%
		Log-Mel spectrogram		78.55%
	Ultrasound 8K	Mel spectrogram	DenseNet	88.52%
		Log-Mel spectrogram		84.74%
	ESC-10	Mel spectrogram	DenseNet	81.25%
		Log-Mel Spectrogram		71.25%
Nicolae-Cat˘ alin Ristea.[[18]]	ESC-50	Spectrogram	Separable transformer architecture	91.13%
	SCV2	Spectrogram		98.51%
Salamon and Bello.[[3]]	Ultrasound 8k	MFCC	Deep CNN	79%
Koutini et al.[[19]]	ESC-50	Spectrogram	CNN with MLP classifier	96.8%
	Open Mic			-
	DCASE 20			76.3%
Martín-Morató et al [[20]]	ESC-30	Spectrogram	CNN with SoundNet Network Architechture	77%
	Urbansound 8k			73.96%
	DCASE2017 T4			-
Zhang, Z et al [[21]]	ESC-50	Log-gammatone spectrogram	Convolutional RNN	86.5%
	ESC-10			94.2%
	DCASE 2016			88.9%
Luz et al [[22]]	Urbansound 8K	Mel-spectrogram	CNN with SVM and Random Forest Classifier	96.8%
	ESC-10			86.2%
Das et al [[23]]	Urbansound 8K	Mel-spectrogram	CNN with SVM	84.2%
David Elliott et al [[22]]	ESC-50	Mel-spectrogram	CNN	67.71%
	Office sounds			95.31%

3. DATA COLLECTION AND PREPROCESSING

This section outlines the procedures and techniques used for data collection and includes details about the research design, participants or subjects, materials and equipment, and statistical procedures. Contributions from Clarke School for the Deaf and Balavidyalaya School, both in Chennai, have helped guide our device development for individuals with hearing impairments, including those who are deaf-blind. Clarke School's input gathered from a sample size of 20 individuals, emphasized the importance of capturing essential auditory cues such as doorbells, horns, a baby crying, fire alarms, ambulances, and alarm clocks. The user interaction also highlighted the need for waterproofing the device and incorporating both vibration and visual indicators. Similarly, the insights from Balavidyalaya, derived from a sample size of 15 individuals, including alumni, underscored the significance of identifying various sounds, including morning alarm clocks, pedestrians approaching from behind, doorbells, vehicle horns, and potential safety hazards like gas leaks or overflowing tanks. These specific sound classes were considered during our initial data collection, aligning closely with the input received from both schools.

We have used the house of quality, shown in Fig. 1, to finalize the specifications of the device. This methodology helped us prioritize user needs and translate them into technical specifications [24]. User requirements included factors such as lightweight design, comfort, customization options, affordability, user-friendliness, power indication, battery backup features, visual indicators, and haptic feedback (vibration). Technical requirements were delineated to address attributes like size, battery life, Bluetooth connectivity, display, mobile application integration, power consumption, production cost, power switch, haptic feedback, and sound recognition. Relationships between customer and technical requirements were then assessed, quantifying the strength of their connection on a scale of 0 to 10. Additionally, importance ratings were calculated to ascertain the overall significance of each technical requirement, factoring in the

priority of the associated customer requirement. The current state of each technical requirement was evaluated, providing a clear understanding of the existing status in relation to the desired objectives. The direction of improvement was indicated by arrows, guiding the development process towards enhancements, degradation, or maintenance of current attributes.

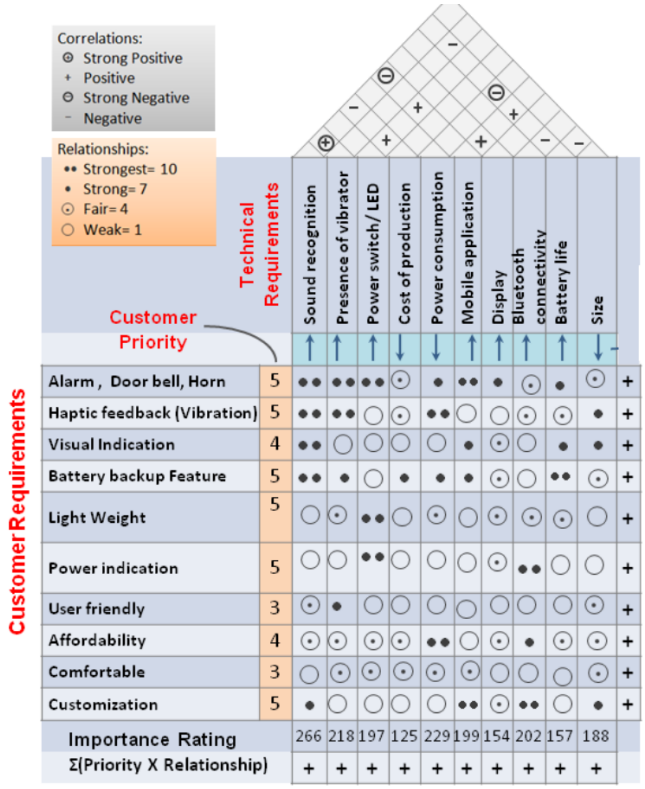

Fig. 1 – House of quality

We used the Android sound recording app called AroundSound [25] to collect sound samples under environmental conditions. Table 3 describes the details of our self-collected dataset. Preprocessing of sound samples includes three stages: subsampling, de-noising, and silence portion detection and removal, following the algorithm structure in Fig. 2.

Table 3 – Self-collected dataset for five classes

Class	Duration	Samples per class
Baby cry	≃5s	100
Vehicle horn	≃5s	100
Alarm	≃5s	100
Door Knock	≃5s	100
Calling Name	≃2s	40

3.1 Subsampling

Sound samples were recorded at a sampling rate of 44.1 kHz using AroundSound. To reduce the computational complexity of training a neural network and the memory requirements of the training and deployment process, we subsampled the data to 16kHz [26]. Authors

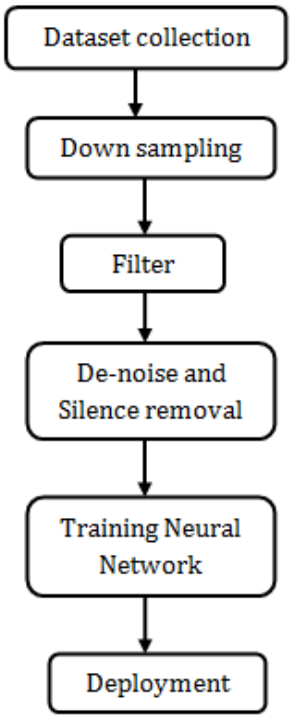

Fig. 2 – Structure of an algorithm

from various studies [27, 28] identified that a lower sampling rate can help prevent overfitting, a common issue in machine learning where the model becomes too specific to the training data and performs poorly on new data.

3.2 De-noising

Sound samples often contain noise; in our case, it is mostly ambient noise (including traffic noise, people talking, and wind), which would negatively impact the performance of a machine learning model. Turget's work [29] focused on removing any distracting sounds from collected baby crying sounds and emphasizing specific points during the preprocessing stage. This pre-emphasis can done using spectral subtraction, Wiener filtering, and adaptive filtering [30, 31]. The goal is to improve the Signal-to-Noise Ratio (SNR) of the sound file. One approach is to use bandpass filters, keeping only a range of frequencies, to remove noise from sound files before training the model [32, 33]. We used SciPy, a scientific Python library, to design bandpass filters that removes the noise in an identified frequency range for each sound class. The BPF filter is designed with the following upper and lower limit frequency ranges.

- Vehicle horns: 2.5 kHz - 4.1 kHz
- Baby cry : 500 Hz - 1500 Hz
- Alarm : 2 kHz - 4 kHz
- Door Knock : 100 Hz - 200 Hz
- Human voice: 200 Hz -500 Hz

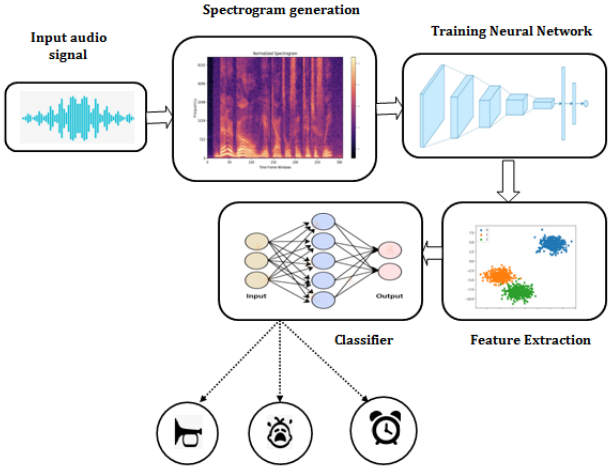

Fig. 3 – Sequential process of methodology

3.3 Silence detection and removal

Removing silent portions from the recorded sound files before training a neural network also helps improve the accuracy of the model prediction [34]. Silent portions in the sound files are often caused by background noise, interference, or gaps between sound events. They reduce the Signal-to-Noise Ratio of the sound files and make it more difficult for the neural network to distinguish between different sound events. Silence removal can also be done manually with the help of the truncate silence option in Audacity [35].

4. METHODOLOGY

The approach we follow for our proposed machine learning-based environmental sound classification system is shown schematically in Fig. 3. It involves several steps, including data collection and preprocessing, feature extraction, labeling, model training, evaluation, and deployment to edge devices.

4.1 Spectrogram feature extraction

We use a spectrogram to extract features from preprocessed data samples [36]. We compute the Short-Time Fourier Transform (STFT) of the audio signal, dividing the signal into overlapping frames and applying a Fourier transform to each frame [37]. The variation in frequency content of the audio signal over time is then presented as a 2-D figure. Spectrogram parameters are the settings that can be adjusted to control the appearance and quality of the spectrogram. The key parameters used are,

- Window size: 1000ms
- FFT size: 128
- Frequency range: 0-8kHz
- Noise floor: (-52dB) It is used to specify the minimum amplitude threshold for audio samples in the dataset. Any samples below this threshold will be considered as noise and discarded during training and testing.

Fig. 4 represents the spectrogram of the alarm sound sample; the frequency of the sample falls within the range of 2 to 4 kHz. The strength of each frequency component is represented by varying intensities of the colors. Here, red in color areas represents stronger frequency components. These extracted features from the sample can then be used as inputs to the next stage, i.e., training a convolutional neural network.

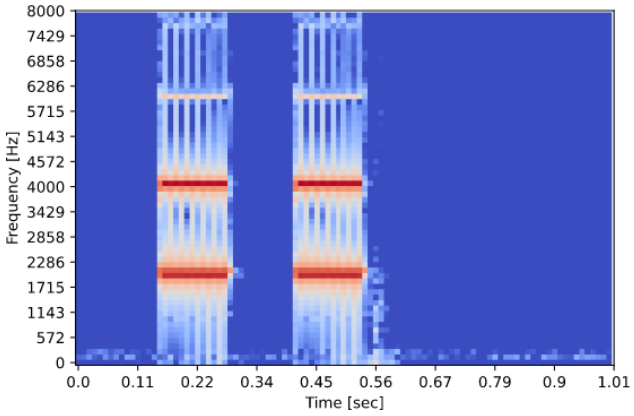

Fig. 4 – Spectrogram of a ringing alarm.

4.2 Neural network architecture

Our proposed neural network architecture is composed of multiple layers, each with its own set of functions and purposes implemented using Python. Our self-collected dataset was divided into training (80%) and test (20%) sets randomly. We used external sounds to validate the particular class. We designed a six-layer Convolutional Neural Network (CNN) with an input layer and a reshape layer, followed by two 1D convolution or pool layers. We also included a dropout layer with a rate of 0.5 to reduce overfitting [38]. The function of each NN layer is described as follows.

1. Input layer (6,435 features): The input layer is the first layer of the neural network; this layer will take the preprocessed audio samples as input
2. Reshape layer: Input audio signals are first transformed into a spectrogram. The resulting spectrogram is a 2D matrix of frequency bins and time frames. However, the input to the neural network must be in the form of a 3D tensor, where the third dimension represents the number of input channels (i.e., mono or stereo audio)[39][40].
3. 1D convolution/pool layer (8 neurons, three kernel size, one layer): 1D convolutional layer with eight neurons and a kernel size of 3 would take as input a 1D tensor (i.e., a vector) and apply eight filters, each with a size of 3, to produce eight feature maps. The output would be a 3D tensor with dimensions (batch_size, sequence_length - 2, 8), where

batch_size is the number of samples in a batch and sequence_length is the length of the input sequence.

4. Dropout layer: A dropout layer with a rate of 0.25 is added after the first dense layer. During each training iteration, 25% of the neurons in the first dense layer will be randomly dropped out [38].
5. 1D convolution/pool layer (16 neurons, three kernel size, one layer): A 1D convolutional layer with 16 neurons and a kernel size of 3 would take as input a 1D tensor (i.e., a vector) and apply 16 filters, each with a size of 3, to produce 16 feature maps. The output would be a 3D tensor with dimensions (batch_size, sequence_length - 2, 16), where batch_size is the number of samples in a batch and sequence_length is the length of the input sequence. Adding a 1D pooling layer after the convolutional layer would further reduce the dimensionality of the output feature maps. A typical choice for the pooling operation in a 1D CNN is to use the maximum value in each sliding window of pool [41, 42].
6. Output layer: This is responsible for producing the final output of the network based on the input data, and uses a softmax activation function [43]to produce a probability distribution over the different classes in the classification problem. The class with the highest probability is then selected as the predicted output of the network.

4.3 Edge device architecture

We have adopted an RP2040-based Arduino microcontroller with an inbuilt MEMS-type omnidirectional microphone for our work. The RP2040 is suitable for edge machine learning applications due to its low cost, low power consumption, and high performance. It has 264KB of RAM, and a dual Arm Cortex-M0+ core that can run at up to 133 MHz, with enough processing power to handle ML models [44]. The features of our edge device are:

- Wearable: designed as a wearable watch, as shown in Fig. 6.
- Haptic and visual feedback: specific vibrations to match a few pretrained sounds. An OLED display shows what sound has been captured and predicted.
- Sensitivity range : ~ 5 m
- Operating time: ~ 28 days per charge
- Deep sleep mode: sound activity/threshold-based wake-up
- Customizable: custom sounds can be added through Over The Air (OTA) configuration.

5. USING REGION-SPECIFIC DATASETS

To demonstrate the necessity of a self-collected dataset, we conducted an experiment with the primary objective of assessing the performance difference between our

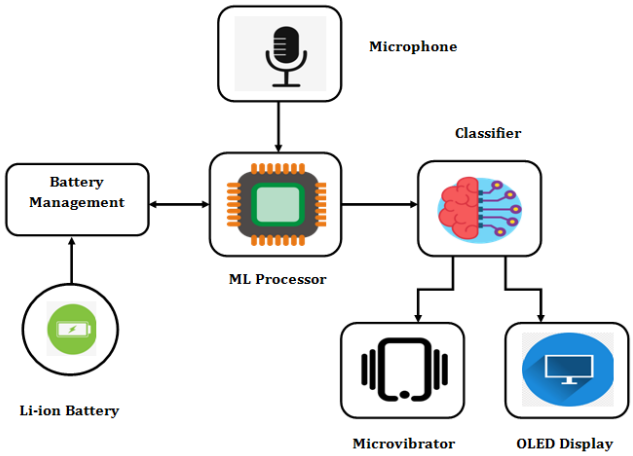

Fig. 5 – Wearable edge device architecture

Fig. 6 – Wearable device design

self-collected datasets from the Indian region and open datasets available in the urban sound library. Initially, we trained our model using an Urbansound8K dataset comprising car horn and siren sounds [45]. Subsequently, we evaluated the model's performance using Indian car horns and siren sounds. The confusion matrix provided in Table 4 illustrates the classification results. Notably, the overall accuracy of the model is found to be only 33%. Further analysis reveals variations in the F1 scores for different sound classes. Specifically, for the 'Car Horn' class, the F1 score is 0.54, indicating a relatively better performance compared to other classes. However, for the 'Noise' class, the F1 score is considerably low at 0.1, suggesting a significant challenge in accurately classifying this sound category. Additionally, the 'Siren' class exhibits an F1 score of 0.34, indicating room for improvement in its classification performance.

Table 4 – Confusion matrix - Classification results of Indian sounds trained on the Urbansound8k dataset.

Actual / Predicted	Car Horn	Noise	Siren
Car Horn	42.20%	53%	3.20%
Noise	0.00%	100%	0.00%
Siren	12.90%	0%	21.10%
F1 Score	0.54	0.1	0.34

The findings of this experiment underscores the importance of utilizing region-specific datasets for training and

testing sound classification models. The low accuracy and varying F1 scores highlight the need for a self-collected dataset tailored to a specific region, in our case India, which can potentially lead to improved model performance and accuracy in sound classification tasks. We emphasize the significance of dataset selection and its impact on the effectiveness of sound indication devices employing 1D CNN-based machine learning.

Additionally, our approach allows for customization in noise handling based on specific use case requirements. By adopting this method, we overcame the limitations of the UrbanSound8k library, where noise was artificially added in the background. This approach substantially boosts the performance and precision of our sound classification model, especially in noisy settings, ultimately enhancing the relevance and reliability of our research outcomes.

6. DEPLOYMENT: CHALLENGES AND OPTIMIZATION TECHNIQUES

We have used the CMSIS-NN, a neural network inference library optimized for execution on Arm Cortex-M processors [46]. This library offers a range of functions for tasks such as loading and executing CNN models, as well as conducting common operations like convolution, pooling, and activation. We encounter multiple challenges when deploying trained CNN algorithms on the RP2040 processor due to its constrained resources. Firstly, a CNN model often exceeds the RP2040's limited memory capacity, necessitating techniques like quantization, pruning, and knowledge distillation to reduce their size. Additionally, the RP2040's relatively slow processing speed results in prolonged execution times for a CNN model, prompting us to optimize model architecture and employ efficient neural network operations. Finally, despite being a low-power processor, running a CNN model on the RP2040 can still lead to significant power consumption, driving us to implement strategies such as dynamic power management and efficient operation implementations to mitigate this issue. We employed a variety of techniques to optimize the model architecture. This encompassed the use of smaller kernels, a reduction in the number of layers, and the implementation of more efficient activation functions. We discovered that the utilization of strides convolutions and depth-wise separable convolutions substantially decreased the computational cost of a model without compromising accuracy.

7. RESULTS AND DISCUSSIONS

We have evaluated the performance of our model using various metrics such as a confusion matrix, accuracy, and F1 score. To assess the model's performance, in Case-I, we trained a machine learning model with five classes of datasets that included noise and silence. In Case-II, we used two categories of datasets after performing denoising and removing the silent portion. Evaluation metrics for the Case-I and -II are discussed in terms of their accuracy and performance.

7.1 Evaluation metrics for Case-I

We first trained a machine learning model using a dataset of 500 audio samples collected from five different classes (baby cry, alarm, door knock, vehicle horn, and spoken name) of environmental sounds. Each category contains 100 samples, along with noise and silent segments. A confusion matrix is shown in Table 5, summarizing the performance of the classification algorithm. Rows represent the actual class, while the columns represent the predicted class. The percentages indicate the proportion of samples correctly classified as each class with off-diagonal elements gives an indication of error in classification. The model correctly classified 91.7% of alarm sounds but wrongly classified 1.7% as a baby's cry, 6.7% as a door knock, and 0% as a vehicle horn or spoken name.

Table 5 – Case-I: Confusion matrix for classification of five classes of sound samples with noise and silence segments

	Alarm	Baby Cry	Door knock	Vehicle horn	Spoken Name
Alarm	91.7 %	1.7%	6.7%	0%	0%
Baby Cry	0%	98.6%	1.4%	0%	0%
Door knock	1.8%	0%	98.2%	0%	0%
Vehicle Horn	0%	1.9%	26.4%	71.7%	0%
Calling Name	0%	12.8%	1.2%	4.6%	81.4%

The F1 score is calculated using the precision and recall for each class. Precision is the proportion of true positives (TP) among all predicted positives (TP + FP), while recall is the proportion of true positives (TP) among all actual positives (TP + FN). For example, we can calculate the precision, recall, and F1 score for each alarm class as follows, and similarly, it will be calculated for other classes as well.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{91.7}{91.7 + 1.7 + 6.7 + 0 + 0} = 0.89$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{91.7}{91.7 + 0 + 1.8 + 0 + 0} = 0.98$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 0.93$$

With Case-I methodology we achieved the maximum accuracy of 88.3% with the prediction.

7.2 Evaluation metrics for Case-II

Adding noise as a separate category to the training data can make an ML model more robust with sound classification [47, 48]. In Case-II, we trained the ML model with only two classes of sound samples after performing denoising and removal of the silent portion. In the Case-I study, we observed that with a separate noise class, the model may mis-classify noise as a meaningful sound, leading to better predictions.

Table 6 – Case-II: Confusion matrix for classification of three classes of sound samples without noise and silence segments

	Vehicle horn	Siren	Noise
Vehicle horn	98.3 %	1.7%	0%
Siren	4.2%	95.8%	0%
Noise	0	0%	100%
F1 score	0.97	0.97	1.0

From Table 6, we observed that the confusion matrix of Case-II shows that the classifier has performed well for vehicle horn and siren classes, with high accuracy rates of 98.3% and 95.8%, respectively. Also, reducing the number of classes can potentially increase the accuracy of the ML model with prediction. A few technical glitches observed from the results shown in Table 5 e.g. ambient noise was recognized as vehicle horn and a door knock not at all recognized by the device.

7.3 Experimental test setup

In this, the experimental test shown in Fig. 7 was conducted in a closed room under normal ambient noise level conditions, i.e., 45 dB SPL. We placed table-top wireless speakers at various locations inside the room and played different sound samples from multiple trained classes towards the device from mobile to evaluate its performance.

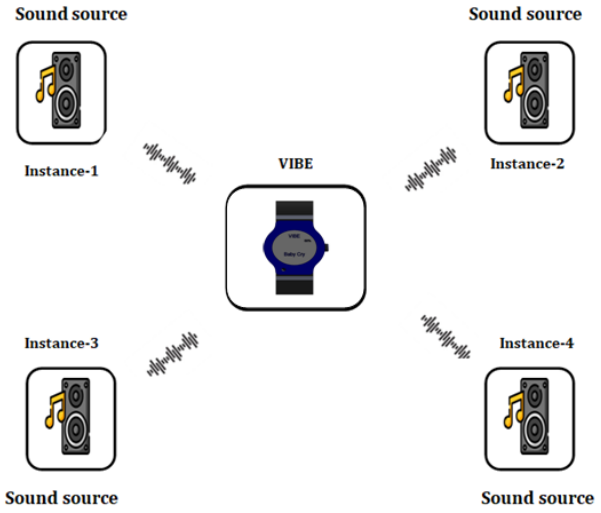

Fig. 7 – Experimental setup

The objective of this experiment was to test the device's ability to predict and detect sounds accurately, as well as to determine if any time delay was present. Keeping the sound source 5 feet away from the device at different locations allowed us to evaluate the device's ability to detect and recognize sounds from various directions since we used an omnidirectional microphone.

The variation in prediction times arises from the tailored design of the 1D Convolutional Neural Network (CNN) architecture for distinct sound classes. The "Alarm" class benefits from an optimized architecture that efficiently extracts its specific features, leading to faster recognition. In contrast, other classes with unique spectral signatures require different configurations, potentially involving deeper networks or alternative structures, resulting in longer prediction times. This tailored approach ensures optimal responsiveness to each class's intricacies.

To assess the reliability of the device's sound recognition capabilities, a measure of confidence level was incorporated into the experimental setup. The confidence level of "High", "Medium" or "Low" denotes the degree of certainty the device assigns a particular class label to a given audio stimulus. As a metric it provides insights into the robustness of the recognition process.

Table 7 – Experimental setup results: device performance for different classes of sounds with prediction

Stimuli played	Confidence level	Recognised class	Prediction time
Baby cry	High	Baby cry	5-10 sec
Alarm	Medium	Alarm	2-5 sec
Vehicle horn	Low	Vehicle horn	5-10 sec
Door Knock	Medium	Door knock & Vehicle horn	5-10 sec
Calling a name	High	Recognised	5-10 sec

8. CALIBRATION AND DEPLOYMENT

We conducted field tests of the device with three Deaf and Hard of Hearing (DHH) users, obtaining valuable feedback regarding its performance, battery life, and user comfort. Calibration of the device was done in the following ways:

- Sensitivity adjustment: The current sensitivity of the PDM microphone is approximately -42 dB FS (decibels full scale). This means it will produce an output voltage of 1 V peak-to-peak for an input Sound Pressure Level (SPL) of 94 dB. The sensitivity was varied by adjusting the gain of the microphone amplifier. The gain is typically specified in decibels (dB).
- Threshold setting: A threshold is defined to specify the minimum sound level (typically around 45 decibels) required to activate the indicator. This threshold was tailored to suit the user's specific needs and preferences.
- Filtering and signal processing: Implementing a filter that passes only the frequencies of interest. This can be done, for example, to filter out background noise and isolate the sound of a specific class.

Currently, we are in the process of initiating a pilot deployment in collaboration with Clarke School for the Deaf and Balavidyalaya school, both in Chennai. In this phase, we will evaluate the performance of our system in live educational environments and gather feedback directly from end users. Additionally, we are working on a comprehensive deployment plan that includes strategies for system calibration, user training, and ongoing support. This will ensure that our technology is seamlessly integrated

into educational settings and maximize its effectiveness for DHH students.

9. LIMITATIONS AND FUTURE WORK

Our results demonstrated that our approach can achieve competitive results for specific sound classification. One potential limitation of our work is the size and diversity of the dataset used for training and testing the model, which is limited to only 500 audio samples in Case-I and a limited number of classes in Case-II. The model's performance is evaluated using traditional metrics like accuracy and F1 score. It would be beneficial to use additional metrics, such as Area Under the Curve (AUC), to evaluate the model's performance more comprehensively. We observed a somewhat poorer performance in noisy environments, as loud background noise made it difficult for the system to identify and classify the target sound accurately. In very noisy environments, the sound classification system would sometimes mistakenly classify the background noise as the target sound.

In future work, we plan to explore different architectures and hyperparameters to improve the EdgeML, mainly predicting vehicle horns in heavy traffic situations. We will also collect more data to increase the diversity and size of our dataset. Finally, our model with edge devices will be customizable by users for their specific needs.

10. CONCLUSION

Our study has demonstrated the feasibility of using machine learning algorithms for environmental sound classification on a wearable platform for individuals with hearing impairments. Our approach achieved high accuracy in recognizing various environmental sounds, providing users with real-time feedback and enhancing their awareness of the surrounding environment. This technology has the potential to significantly improve the quality of life for individuals with hearing impairments, enabling them to interact more effectively with the environment and the people around them. Further research can focus on improving the accuracy of the classification algorithm, exploring additional features for sound analysis, and integrating the platform with other assistive technologies. Overall, this study provides a promising direction for future research and development of wearable platforms for individuals with hearing impairments.

ACKNOWLEDGEMENT

We are grateful to Sony Pictures Networks India for their financial support. We thank Ms. Indulakshmi from the Enability Foundation for Rehabilitation for her initial idea, as well as to the teachers and alumni from the Clarke School for the Deaf and Balavidyalaya School in Chennai.

Table 8 – Abbreviations and their definitions

Abbrev.	Definition
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Networks
ESC	Environmental Sound Classification
LBP	Local Binary Pattern
MEMS	Micro Electro Mechanical Systems
MFCC	Mel-Frequency Cepstral Coefficients
MLP	Multi-layer Perceptron classifier
OLED	Organic Light-Emitting Diode
PCA	Principal Component Analysis
SNR	Signal to Noise Ratio
SPL	Sound Pressure Level
STFT	Short Time Fourier Transform
VGG	Visual Geometry Group

REFERENCES

[1] John V Van Cleve. *Deaf history unveiled: Interpretations from the new scholarship*. Gallaudet University Press, 1999.

[2] In-Chul Yoo and Dongsuk Yook. "Automatic sound recognition for the hearing impaired". In: *IEEE Trans. on Consumer Electronics* 54.4 (2008), pp. 2029–2036.

[3] Justin Salamon and Juan Pablo Bello. "Deep convolutional neural networks and data augmentation for environmental sound classification". In: *IEEE Signal processing letters* 24.3 (2017), pp. 279–283.

[4] Md Mohaimenuzzaman, Christoph Bergmeir, Ian West, and Bernd Meyer. "Environmental Sound Classification on the Edge: A Pipeline for Deep Acoustic Networks on Extremely Resource-Constrained Devices". In: *Pattern Recognition* 133 (2023), p. 109025.

[5] Justin Salamon and Juan Pablo Bello. "Unsupervised feature learning for urban sound classification". In: *2015 IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015, pp. 171–175.

[6] Xiaohu Zhang, Yuexian Zou, and Wenwu Wang. "LD-CNN: A lightweight dilated convolutional neural network for environmental sound classification". In: *2018 24th Intl. Conf. on pattern recognition (icpr)*. IEEE. 2018, pp. 373–378.

[7] Karol J Piczak. "Environmental sound classification with convolutional neural networks". In: *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)*. IEEE. 2015, pp. 1–6.

[8] Ohini Kafui Toffa and Max Mignotte. "Environmental sound classification using local binary pattern and audio features collaboration". In: *IEEE Trans. on Multimedia* 23 (2020), pp. 3978–3985.

[9] Khine Zar Thwe and Nu War. "Environmental sound classification based on time-frequency representation". In: *2017 18th IEEE/ACIS Intl. Conf. on Software Engg., Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*. IEEE. 2017, pp. 251–255.

[10] Divya K V, Yashmitha R, Tejal Lalji Rangani, and Anushka Sen. "IoT based Divyang Assistant Technology: Your Hearing Support". In: *2022 Intl. Conf. on Electronics and Renewable Systems (ICEARS)*. 2022, pp. 474–481.

[11] Leonardo A Fanzeres, Adriana S Vivacqua, and Luiz WP Biscainho. "Mobile Sound Recognition for the Deaf and Hard of Hearing". In: (2018).

[12] Dhruv Jain, Khoa Huynh Anh Nguyen, Steven M. Goodman, Rachel Grossman-Kahn, Hung Ngo, Aditya Kusupati, Ruofei Du, Alex Olwal, Leah Findlater, and Jon E. Froehlich. "ProtoSound: A Personalized and Scalable Sound Recognition System for Deaf and Hard-of-Hearing Users". In: *CHI Conference on Human Factors in Computing Systems*. ACM, 2022.

[13] Dhruv Jain, Hung Ngo, Pratyush Patel, Steven Goodman, Leah Findlater, and Jon Froehlich. "SoundWatch: Exploring smartwatch-based deep learning approaches to support sound awareness for deaf and hard of hearing users". In: *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. 2020, pp. 1–13.

[14] Benjamin M. Gorman. "VisAural: A wearable sound-localisation device for people with impaired hearing". English. In: *ASSETS'14*. 16th International ACM SIGACCESS Conference on Computers and Accessibility , ASSETS 2014 ; Conference date: 20-10-2014 Through 22-10-2014. Association for Computing Machinery, 2014, pp. 337–338.

[15] Tara Matthews, Janette Fong, and Jennifer Mankoff. "Visualizing non-speech sounds for the deaf". In: *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*. 2005, pp. 52–59.

[16] Sicong Liu, Zimu Zhou, Junzhao Du, Longfei Shangguan, Han Jun, and Xin Wang. "UbiEar: Bringing Location-independent Sound Awareness to the Hard-of-hearing People with Smartphones". en. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.2 (2017), p. 17. ISSN: 2474-9567.

[17] Zohaib Mushtaq and Shun-Feng Su. "Efficient classification of environmental sounds through multiple features aggregation and data enhancement techniques for spectrogram images". In: *Symmetry* 12.11 (2020), p. 1822.

[18] Nicolae-Catalin Ristea, Radu Tudor Ionescu, and Fahad Shahbaz Khan. "SepTr: Separable Transformer for Audio Spectrogram Processing". In: (2022).

[19] Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. "Efficient training of audio transformers with patchout". In: (2021).

[20] Irene Martín-Morató, Maximo Cobos, and Francesc J Ferri. "Adaptive distance-based pooling in convolutional neural networks for audio event classification". In: *IEEE/ACM Trans. on Audio, Speech, and Language Processing* 28 (2020), pp. 1925–1935.

[21] Zhichao Zhang, Shugong Xu, Shunqing Zhang, Tianhao Qiao, and Shan Cao. "Learning attentive representations for environmental sound classification". In: *IEEE Access* 7 (2019), pp. 130327–130339.

[22] David Elliott, Carlos E Otero, Steven Wyatt, and Evan Martino. "Tiny transformers for environmental sound classification at the edge". In: (2021).

[23] Steven Wyatt, David Elliott, Akshay Aravamudan, Carlos E. Otero, Luis D. Otero, Georgios C. Anagnostopoulos, Anthony O. Smith, Adrian M. Peter, Wesley Jones, Steven Leung, and Eric Lam. "Environmental sound classification with tiny transformers in noisy edge environments". In: *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*. IEEE. 2021, pp. 309–314.

[24] George L Vairaktarakis. "Optimization tools for design and marketing of new/improved products using the house of quality". In: *Journal of operations management* 17.6 (1999), pp. 645–663.

[25] Gritstone Studios Ltd. *Around Sound Application*. Version 3.2.1. 2023.

[26] Siddharth Sigtia, Adam M Stark, Sacha Krstulović, and Mark D Plumbley. "Automatic environmental sound recognition: Performance versus computational cost". In: *IEEE/ACM Trans. on Audio, Speech, and Language Processing* 24.11 (2016), pp. 2096–2107.

[27] Michele Valenti, Stefano Squartini, Aleksandr Diment, Giambattista Parascandolo, and Tuomas Virtanen. "A convolutional neural network approach for acoustic scene classification". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 1547–1554.

[28] Like Xue and Feng Su. "Auditory scene classification with deep belief network". In: *MultiMedia Modeling: 21st Intl. Conf., MMM 2015, Sydney, NSW, Australia, January 5-7, 2015, Proceedings, Part I 21*. Springer. 2015, pp. 348–359.

[29] Turgut Özseven. "A Review of infant cry recognition and classification based on computer-aided diagnoses". In: *2022 Intl. Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE. 2022, pp. 1–11.

[30] Lawrence R Rabiner. *Digital processing of speech signals*. Pearson Education India, 1978.

[31] Z Yanlei, O Shifeng, and G Ying. "Improved Wiener filter algorithm for speech enhancement". In: *Autom. Control. Intell. Syst* 7.3 (2019), p. 92.

[32] Sen M Kuo and Dennis R Morgan. "Active noise control". In: *Springer handbook of speech processing* (2008), pp. 1001–1018.

[33] Tayyab Zafar, Khurram Kamal, Senthan Mathavan, Ghulam Hussain, Mohammed Alkahtani, Fahad M Alqahtani, and Mohamed K Aboudaif. "A Hybrid Approach for Noise Reduction in Acoustic Signal of Machining Process Using Neural Networks and ARMA Model". In: *Sensors* 21.23 (2021), p. 8023.

[34] Stavros Ntalampiras, Ilyas Potamitis, and Nikos Fakotakis. "Automatic recognition of urban soundscenes". In: *New directions in intelligent interactive multimedia* (2008), pp. 147–153.

[35] Carla Schroder. *The book of Audacity: Record, edit, mix, and master with the free audio editor*. No Starch Press, 2011.

[36] Chaoyi Wang, Yaozhe Song, Haolong Liu, Huawei Liu, Jianpo Liu, Baoqing Li, and Xiaobing Yuan. "Real-Time Vehicle Sound Detection System Based on Depthwise Separable Convolution Neural Network and Spectrogram Augmentation". In: *Remote Sensing* 14.19 (2022), p. 4848.

[37] Jizuo Li, Jiajun Yuan, Hansong Wang, Shijian Liu, Qianyu Guo, Yi Ma, Yongfu Li, Liebin Zhao, and Guoxing Wang. "LungAttn: advanced lung sound classification using attention mechanism with dual TQWT and triple STFT spectrogram". In: *Physiological Measurement* 42.10 (2021), p. 105006.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[39] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. "Audio set: An ontology and human-labeled dataset for audio events". In: *2017 IEEE Intl. Conf. on acoustics, speech and signal processing (ICASSP)*. IEEE. 2017, pp. 776–780.

[40] Sercan O Arik, Markus Kliegl, Rewon Child, Joel Hestness, Andrew Gibiansky, Chris Fougner, Ryan Prenger, and Adam Coates. "Convolutional recurrent neural networks for small-footprint keyword spotting". In: (2017).

[41] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

[42] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[43] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks". In: *Towards Data Sci* 6.12 (2017), pp. 310–316.

[44] Lucas Martin Wisniewski, Jean-Michel Bec, Guillaume Boguszewski, and Abdoulaye Gamatié. "Hardware Solutions for Low-Power Smart Edge Computing". In: *Journal of Low Power Electronics and Applications* 12.4 (2022), p. 61.

[45] J. Salamon, C. Jacoby, and J. P. Bello. "A Dataset and Taxonomy for Urban Sound Research". In: *22nd ACM Intl. Conf. on Multimedia (ACM-MM'14)*. Orlando, FL, USA, 2014, pp. 1041–1044.

[46] Liangzhen Lai, Naveen Suda, and Vikas Chandra. "CMSIS-NN: Efficient neural network kernels for ARM Cortex-M CPUs". In: *arXiv:1801.06601* (2018).

[47] Annamaria Mesaros, Aleksandr Diment, Benjamin Elizalde, Toni Heittola, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. "Sound event detection in the DCASE 2017 challenge". In: *IEEE/ACM Trans. on Audio, Speech, and Language Processing* 27.6 (2019), pp. 992–1006.

[48] Annamaria Mesaros, Toni Heittola, Tuomas Virtanen, and Mark D Plumbley. "Sound event detection: A tutorial". In: *IEEE Signal Processing Magazine* 38.5 (2021), pp. 67–83.

AUTHORS

Velmurugan S is pursuing his Master of Science (MS) at the Indian Institute of Technology Madras. His research interests include assistive technology and rehabilitation, with a current focus on EdgeML on wearable devices. He has published several pieces of work in the field of wearables and sensor networks and has a patent (424295) for his invention entitled "Process for harvesting energy using floor tiles with a piezoelectric sensor."

Prof. Anil Prabhakar has been with the faculty at the Dept. of Electrical Engineering, IIT-Madras since 2002, and is engaged across multiple laboratories that work on quantum technologies, fiber lasers and assistive devices. He is the Founder-Director of Yali Mobility and Enability Foundation, companies that focus on rehabilitation engineering. He has over 50 research publications, has co-authored a book on spin waves, a few book chapters, and has 18 patents on a wide range of devices in areas of photonics, magnonics and assistive devices.