SMART SOUND SYSTEM APPLIED FOR THE EXTENSIVE CARE OF PEOPLE WITH HEARING IMPAIRMENT

Smitha S Maganti, Sahana S, Kriti K, Shravanthi Madhugiri and Priya S

Computer Science and Engineering Department, BNMIT, Bangalore. India

ABSTRACT

We, as normal people, have access to a potent communication tool, which is sound. Although we can continuously gather, analyse, and interpret sounds thanks to our sense of hearing, it can be challenging for people with hearing impairment to perceive their surroundings through sound. Also known as PWHI (People with Hearing Impairment). Auditory/phonic impairment is one of the most prevailing sensory deficits in humans at present. Fortunately, there is room to apply a solution to this issue, given the development of technology. Our project involves capturing ambient sounds from the user's surroundings and notifying the user through a mobile application using IoT and Deep Learning. Its architecture offers sound recognition using a tool, such as a microphone, to capture sounds from the user's surroundings. These sounds are identified and categorized as ambient sounds, like a doorbell, baby cry, and dog barking; as well as emergency-related sounds, such as alarms, sirens, etc.

KEYWORDS

Deaf, PWHI, DHH, Sound detection, ResNet50, Deep Learning, Internet of Things, Mobile Application.

1. INTRODUCTION

Being privy to all the sounds around us is highly crucial. We can perceive the world around us through sound, due to our sense of hearing. Significant events that are occurring out of our sight can be signalled to us via auditory cues. Society exclusively uses sound to convey various sorts of information to each other. To give a specific example, we have alarms that sound to alert us about critical events and emergencies, and people who ring doorbells to let us know they are there. Critical information is inaccessible to the deaf and people with hearing impairment because of these societal norms (PWHI).

Technical solutions are frequently specialized for certain sounds, whereas non-technical sound awareness techniques like ocular inspection might be annoying and difficult. For instance, some deaf individuals additionally connect their doorbell to their house lights so that the lights flash when the doorbell rings. These solutions, however, focus on certain sounds. The cost and inconvenience of buying a distinct gadget for each sound can be high. Due to the uniqueness of each person's life and the noises contained within it, some sounds cannot be covered by even a large number of devices.

Our goal is to design a deep learning model that would identify sounds heard in the user's environment into the appropriate category, such as a dog barking, a baby crying, or a doorbell ringing, and then alert the user via notifications to a mobile application so they can take the appropriate actions. The dataset we have employed for the testing and training of the

DOI : 10.5121/ijasa.2022.10301

classification algorithm is Google AudioSet. The STFT (short time fourier transform) is used as a pre-processing tool to create spectrograms from the raw audio files in WAV format before further processing. The features of the spectrograms generated by the Short-Time Fourier Transform are extracted using the Log-Mel Spectrogram. The spectrograms are then uploaded to our ResNet50 deep learning model, as input for further analysis and classification of the audio.

The aim of our project is to implement the suggested methodology to accurately identify and categorize the sounds in the user's environment while also addressing the shortcomings of the existing systems, including their accuracy, frequency of repeated and persistent notifications, and error rates.

2. RELATED WORK

JoãoElison da Rosa Tavares et al. [1] proposed a system called the Apollo SignSound system. In a smart home setting, this solution encourages accessibility for PWHI and deaf individuals. The use of neural networks in Apollo SignSound's ambient danger detection represents its scientific contribution. In order to provide accessible alerts in Brazilian Sign Language, SignSound will consider the profile of the user as well, particularly the deafness degree (which, in Portuguese is LIBRAS). The risk warnings delivered to a PWHI or deaf smartphone may additionally vibrate or switch on the device's torchlight. This system was enforced as part of a dual-part device, which is the MAX4466 sensor that senses sounds. Used to record background ambient sounds and the ESP8266 Wi-Fi module and the SignSound Server to communicate with one other over HTTP. The FFT (Fast Fourier Transform) is used for data pre-processing. A neural network (NN) using the Multi-Layer Perceptron (MLP) architecture was built. After gathering 20 examples of every sort of sound, such as door knocks, dog barks, and boiling kettles, the NN was trained. The NN was modified throughout the training phase to prevent overfitting and underfitting. It was decided to use a hybrid cloud architecture, where Nueral Network training takes place on cloud while event detection takes place on local cloud, known as Fog computing. Three scenarios that simulated real-world circumstances were used to accomplish the numeric evaluation of the Apollo SignSound system. Twenty collections were run for each scenario, and the results showed a mean inclusive f-score of 0.73. The neural network might be enhanced to increase its accuracy, and this approach could be integrated with wearable gadgets like smartwatches to create personalised notifications for PWHI.

Dhruv Jain et al. [2] introduced HomeSound, an indoor sound detection architecture for DHH (Deaf and hard of hearing) users. They employed a deep CNN named VGG16, which was previously trained on 8 million YouTube videos, to build a reliable, real-time sound classification engine. Transfer learning was used to modify VGG16 for sound classification as it was primarily created for video classification. 19 typical household sounds, such as dog barking, knocking on doors, and conversation, were acquired from 6 libraries: BBC, TAU, Network Sound, TUT, Freesound and UPC. No participants reported false positives. However, false negatives (such as unsubstantiated noises like garbage disposal) and misattributions (such as a fan being mistaken for a microwave) were both problematic. These misclassification problems point to the necessity for increased system accuracy. Along with classification errors, overly persistent notifications were also a problem.

Aurora Polo-Rodriguez et al. [3] proposed the utilization of surrounding audio sensors to recognise incidents that rise from everyday tasks performed by house residents. They created a balanced dataset that was gathered and classified under carefully controlled circumstances. Convolutional network inputs were used to generate a spectrum representation of sounds, and the results for identifying ambient noises were promising. To sieve through the unprocessed audio event identification and eliminate false positives in the recognition of real-time event, using the

temporary restrictions that protoforms offered, ambiguous processing of audio event streams was introduced to IoT board. In the realm of audio recognition, CNNs and the usage of spectrograms for sound representation have been shown to produce positive outcomes in the recognition of sounds, which may be utilized for music signal analysis and ambient sound categorization. In particular, MFCC (Mel-frequency cepstral coefficient) and LM (log-Mel spectrogram) usage have been suggested for robust representation in sound categorization. A MF (Mel-frequency spectrogram) and in order to offer a deep learning model for the sound detection in the background of commonplace cases, convolutional neural networks are employed. Under controlled settings, the performance of the audio dataset for both the CNN models shows great results, with CNN + MFCC displaying on of the best results. With the Audioset dataset, however, the process of recognizing sounds of everyday activities is really poor. This is because audio snippets derived from the video on YouTube contain noise that overlaps with the other noises and audio that is generated from a variety of origins.

Chuan-Yu Chang et al. [4] proposed an anomalous sound identification system for monitoring interior sounds. Each sound frame was used to extract 24 characteristics. Then, to choose high discriminative features, SFFS (sequential floating forward selection) was used. The sounds were finally divided into six categories by the support vector machine (SVM), which consist of screams, baby cries, coughs, breaking of glass, laughs, and rings of doorbells. According to the experiment's findings, the suggested system has a high recognition rate and is capable of accurately identifying various types of anomalous sounds. Extraction of sound characteristics is by far the most crucial process in sound signal detection. The outcomes obtained were both real and imaginary values from the modification after applying the FFT (fast Fourier transform), as well as spectrum information. 15 frequency domain variables, such as intensity, variance, average, standard deviation, bandwidth, centroid, roll-off, total spectrum power, maximum, sub-band powers, valley and peak, in addition to MFCC and LPCC, depends on the fastfourier transform to be retrieved.. Peak, contrast and valley of MFCC and Octave-based features were selected to be the 4 discriminative properties using this method using the SFFS. The SVM was finally used for categorization. Exploratory outcomes depicted that this technique achieved a good precision rate of 86%.

3. PROBLEM STATEMENT

Sounds present in our surroundings play a huge role in our day-to-day lives. People who cannot perceive these sounds face a lot of hindrances in performing the simplest of tasks, like sound from a fire alarm, smoke detector, baby crying etc. The aim of our project is to implement the suggested methodology to accurately identify and categorize the sounds in the user's environment while also addressing the shortcomings of the existing systems, including their accuracy, frequency of repeated and persistent notifications, and error rates.

4. PROPOSED SOLUTION

The project's main goal is to create a DL model that would identify and categorize sounds heard in the user's environment into the appropriate category, such as a dog barking, a baby crying, or a doorbell ringing, and then alert the user via a mobile application about these sounds so they can take the appropriate actions.

4.1. Internet of Things – Smart Home

The Internet of Things layer is in charge of gathering and delivering user-generated sounds to storage for additional processing.

4.2. Server

The server layer is responsible for hosting the DL model for the classification of the sound. It checks the storage for real-time sounds, processes them, and classifies the sound to its class. It communicates the result of the classification to the Mobile Application through Rest APIs.

4.3. Mobile App Notification

The Mobile Application sends an API request to the server to check for sounds. If there are sounds in the storage and they are classified into one of the classes, the server returns the classification result back to the app. The app displays the result and gives a notification of the result to the PWHI user

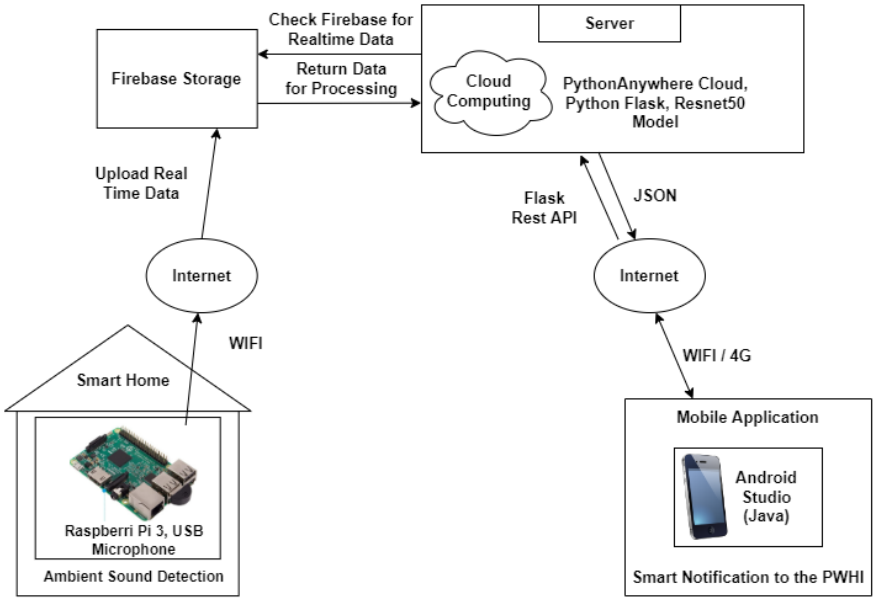

Figure 1. System Architecture

4.4. Data Flow Diagram

The user's environment serves as the source of the real-time audio in our project. After that, the sounds are uploaded and kept in Firebase Storage. Following that, the sounds are loaded onto the librosa library as required by the Python software, which is hosted on the PythonAnywhere cloud. The sounds are then transformed to spectrograms after the features of the sounds have been extracted. After that, the spectrograms are pre-processed into a model-acceptable format. The ResNet50 deep learning is then fed the spectrograms in order to classify and identify the sound. The mobile application receives the classification's outcome. The user is then informed and the information is then shown in the app. The data flow diagram is shown in fig 2.

5. BUILDING THE DEEP LEARNING MODEL

5.1. Dataset Acquisition

The dataset from FreeSound and Google AudioSet was utilized to develop and validate the categorization model. Noisy and Clean data were both collected. Baby crying, dog barking, coughing, dishes and pans, doorbell, fire alarm, smoke detector, sneeze, thunder, and water were taken into account as ambient noises. A website called FreeSound offers sound files free of background noise for every kind of ambient sound.

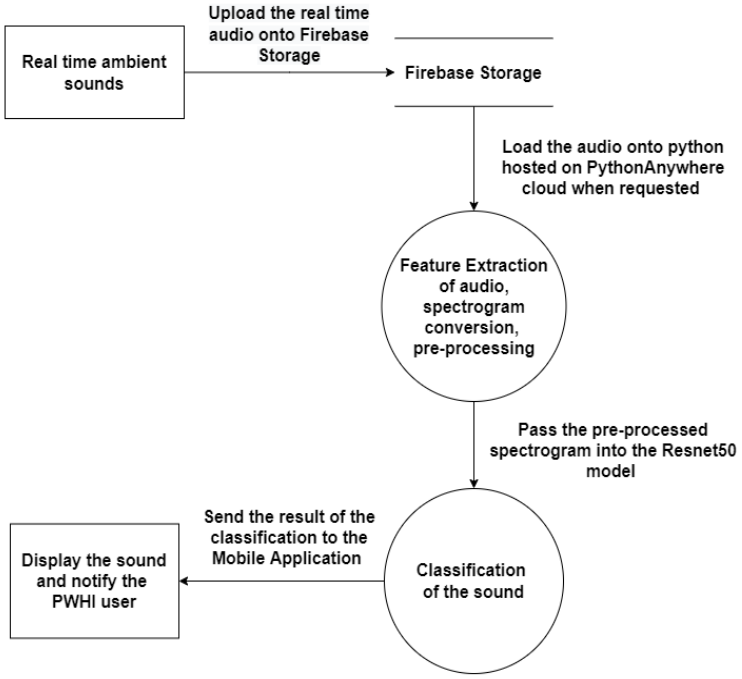

Figure 2 Data Flow Diagram

5.2. Data Augmentation

To provide the model more data to train on, we have augmented the clean dataset of each class. We have used Time Shift, Pitch Shift, and Time Stretch.

5.3. Spectrogram Conversion

The Short-time fourier transform (STFT) is used as a preprocessing tool to create spectrograms from the raw audio files in WAV format [12] before further processing [5][6][7]. The Librosa library was used to accomplish this.

5.4. Feature Extraction

The characteristics are extracted from the spectrograms produced by the Short-time fourier transform using the log-mel spectrogram [8][9]. The classification model is then used to process and label the Mel-Spectrograms as input.

5.5. Building the Pre-Trained Model

The keras library is used to build and train the pre-trained models. Transfer learning is used to train these models on our dataset. The models we trained are ResNet50 [10], VGG16, MobileNetV1, InceptionV3, InceptionResNetV2, and Xception [13].

5.6. Training and Testing the Models

The models must then be trained, validated, and tested after being built. The keras library is utilized to complete this step.

5.7. Dataset Split

We split our dataset into train, test, and validation groups in the following proportions: 80:10:10. For each class, we had about 60 validation and testing samples and about 500 training samples.

5.8. Parameters Used

1. Learning Rate (lr): 0.001
2. Optimizer: Adam Optimizer, which follows the stochastic gradient descent method to optimize the weights
3. Epochs: 15
4. Spectrogram size: 224 x 224

5.9. Test Accuracies

1. ResNet50: 100%
2. VGG16: 56.29%
3. MobileNetV1: 90.2%
4. InceptionV3: 59.5%
5. InceptionResNetV2: 71.9%
6. Xception: 100%

5.10. Outcome

ResNet50 and Xception models, as can be seen above, classified our test data with an accuracy of 100 percent. ResNet50 has the advantage over Xception in that it is lighter and faster. ResNet50 model loads the data and classifies more quickly, and it takes up less memory than Xception. Hence, we decided to identify our real-time ambient sounds using the ResNet50 model [10][11].

5.11. Converting the Model to TensorFlow Lite

After building and training the ResNet50 model, it was compressed to a smaller, more efficient ML model format called TensorFlow Lite model using TensorFlow library. The model was then hosted on the PythonAnywhere cloud to process and classify real-time data.

6. SYSTEM IMPLEMENTATION

The project has been implemented by dividing it into 3 modules:

1. Internet of Things - Collects real-time sounds.
2. Server - Pre-processes and classifies the real-time sounds.
3. Mobile Application - Displays and notifies the sounds to the users.

6.1. Internet of Things

This module of our project is implemented using 2 hardware components: Raspberry Pi 3 and USB microphone

The Raspberry Pi is used in headless setups without a monitor, keyboard, or mouse. The RaspberryPi's USB microphone can be accessed with Alsa. The recording was made using Libportaudio and arecord.

The real-time sounds are stored in Firebase Storage for further processing. The Raspberry Pi is programmed using Python to connect to Firebase Storage and uploads the real-time audio files to Firebase via the internet each time a new sound is recorded. This is accomplished by utilizing the "pryebase" library in python.

6.2. Server

PythonAnywhere cloud is used to implement this module of our project. The online services, audio pre-processing, and model classification methods were created using Python programming. ResNet50, the chosen deep learning model for classification, as well as a Python program to perform various tasks are hosted on the PythonAnywhere server. The Python software implements the web services using the Flask-RESTful framework. When an API request is made to the server and directed to this specific function, this function is called Every 10 seconds, the mobile application sends a GET request to the server via the API to route the necessary function. For each API call, the following actions are performed on the server.

1. Loading real-time audio from Firebase Storage - Real-time audio files are downloaded from Firebase Storage in the first section of the Python application. This is accomplished using the libraries "pyrebase" and "firebase admin." The 'librosa' and 'wavfile' libraries are next used to load the audio files for further processing. The audio file is removed from the Firebase Storage after it has been loaded onto the aforementioned libraries. The program ends and returns "None" if there are no audio files in the Firebase Storage.
2. Spectrogram conversion - The program's second section extracts audio features and turns them into spectrograms. This procedure makes use of the Librosa library. The required features are extracted from the audio using the Short-Time Fourier Transform, and the audio is then transformed into a Mel Spectrogram.
3. Pre-processing - The spectrogram is pre-processed in the third section of the program in accordance with the model's input. The spectrogram is preprocessed before being fed to the ResNet50 model using the 'tensorflow.keras.preprocessing.image' and 'tensorflow.keras.applications.resnet50.preprocess input' libraries.
4. Classification of sound - The pre-processed spectrogram is fed into the model in the fourth section of the program to predict the sound. The model assigns the input audio to the appropriate class according to its composition. The Keras library is used to implement this.

5. Returning the result to the Mobile Application - The program's final section sends the classification outcome back to the mobile application. The mobile app sends an API request, and once the model categorizes the sound, the result is returned to it in JSON format. The result is returned as "None" if the sound does not fit into any of the classes

To summarize, every time the mobile app makes an API call, the server retrieves the real-time input from Firebase Storage, processes it, categorizes it using the ResNet50 model, and then sends the classified data back to the mobile app.

6.3. Mobile Application for Notification

Using the Java programming language and the Android Studio IDE, the mobile application is created for devices running the Android operating system. Every 10 seconds, the app sends API GET requests to the PythonAnywhere server. To make the API requests, we used the 'volley' library. The results of the real-time sound classification are returned in JSON format by the server in response to requests. Using the "json" library, the result is extracted and put into String format. If the outcome is "None," either Firebase Storage is empty or the sound has not been assigned to any class. As a result, nothing occurs in the app. If the sound is classified into a class, the app displays that class, and the user receives a warning alerting them to a sound they should be aware of in their immediate surroundings.

The user interface of the app has the following features as explained below.

1. A dynamic sound display view for each sound. The sound's name is written in English on the card, along with a picture of the sound and a button that allows the user to delete the sound after viewing it. The same sound won't be played twice if it hasn't been deleted. This prevents the user from hearing the same notification sound again in a short period of time, which could have negative impacts.
2. A choice to alter the notification's format. Users can select either a torch light or a vibration as their notification mode. Every time the model hears a new sound, it adds it to the app's display and, depending on what the user has chosen, also adds a vibration or torch light to alert them.

Figure 3 shows how the UI of the app looks. The sounds are displayed as cards with notification settings at the bottom of the screen.

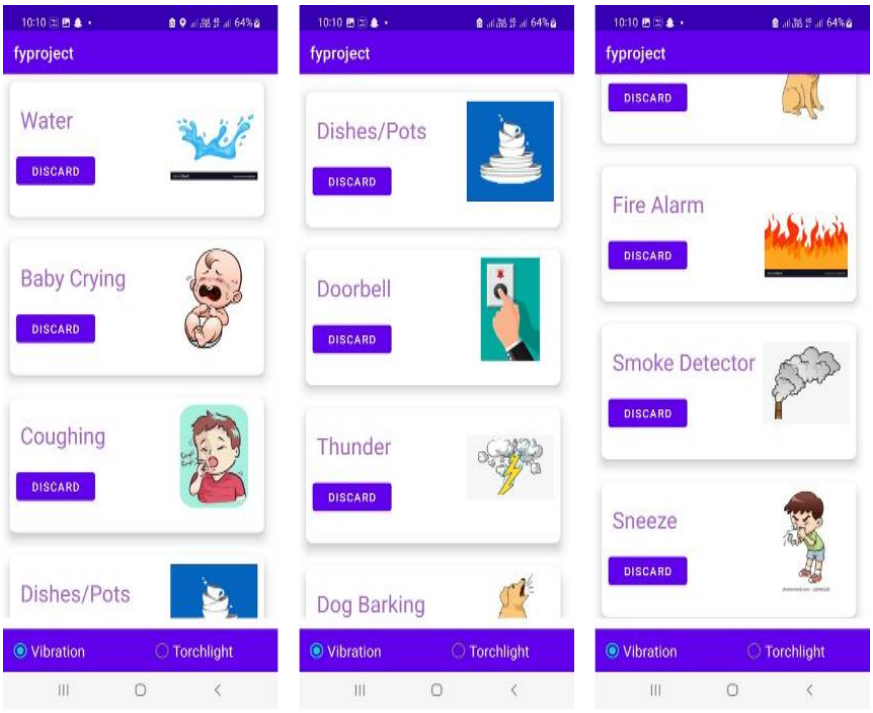

Figure 3. Mobile Application to Display the Sounds

7. TESTING AND VALIDATION

7.1. Testing Results of Different Models on Our Dataset

Table 1 displays the testing outcomes for several models based on Accuracy, Recall, Precision, and F1Score. ResNet50 and Xception architectures showed the best accuracy and F1 score out of all the models. Since ResNet50 is lighter, faster, and easily deployable as compared to Xception, we finalized ResNet50 as the architecture. Xception poses a problem when there is a memory constraint, and is slow to load the data.

Table 1. Test Result

Models	Accuracy	Recall	Precision	F1Score
ResNet50	1.0	1.0	1.0	1.0
MobileNet	0.902	0.905	0.908	0.892
InceptionResnetV2	0.719	0.697	0.766	0.712
InceptionV3	0.595	0.598	0.593	0.571
Vgg16	0.563	0.558	0.596	0.536
Xception	1.0	1.0	1.0	1.0

7.2. Confusion Matrices

The 10 classes that are chosen are represented as numbers from 0-9 in the matrices as shown in Table 2.

Table 2. Classes

Number	0	1	2	3	4	5	6	7	8	9
Class	Baby Cry	Dog Bark	Cough	Dishes and Pots	Door bell	Fire Alarm	Smoke Detector	Sneeze	Thunder	Water

Figure 6 shows the confusion matrices of all the models to visualize the accuracies.

8. RESULT AND DISCUSSIONS

The real-time sounds are recorded using Raspberry Pi in WAV format. The sounds are then classified using the ResNet50 model, and the result is sent to the Mobile Application. The app displays the sound and notifies the user through vibration or torch light. The process is shown below.

1. Recording real-time sounds using RaspberryPi Command Line Interface for a duration of 5 seconds and at 48000 Hz in WAV format. The sound being recorded is Baby Cry. This process is shown in Fig. 4.

2. The sound recorded above is processed and classified. The result is sent to the Mobile Application. The app notifies the user and displays the sound in a dynamic card view with English Text, Pictorial Representation, and a discard button to remove the sound after it is seen. It also allows the user to set the type of notifications to either vibration or torch light. This process is shown in Fig. 5.

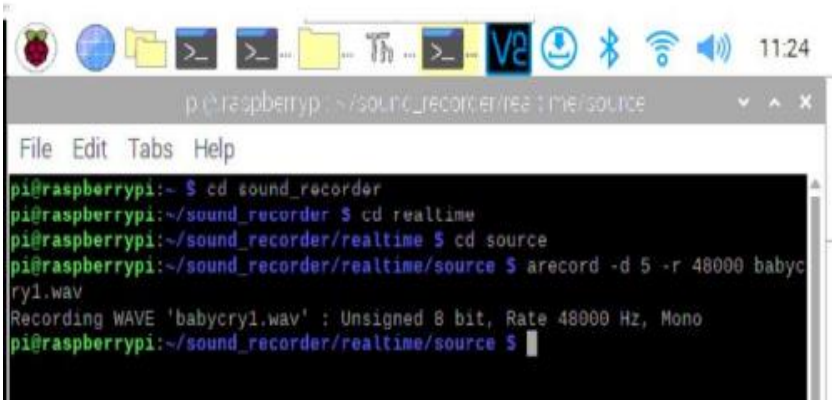

Figure 4. Recording Real-Time Baby Cry Sound Using Raspberry CLI

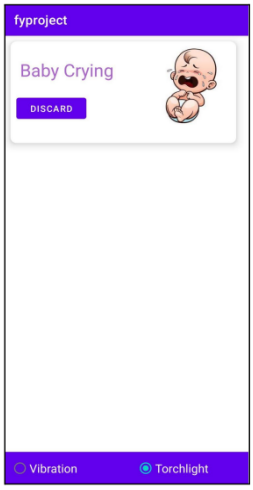

Figure 5. Baby Crying Sound Detected on the Application

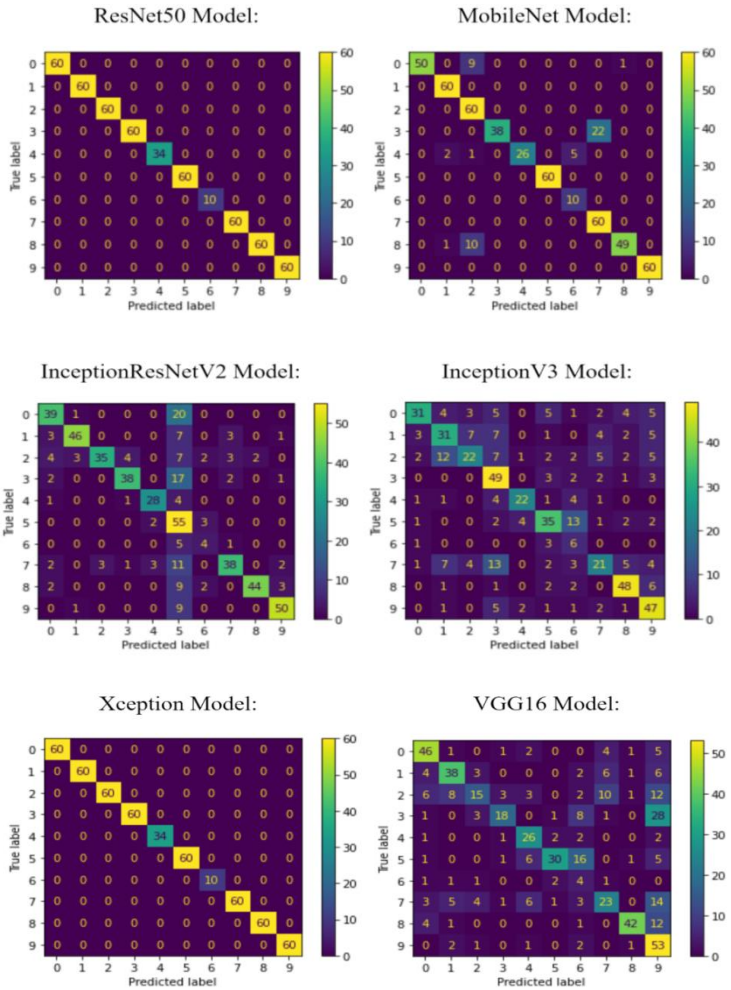

Figure 6. Confusion Matrices (True Label vs Predicted Label) of all the Models

9. CONCLUSION AND FUTURE SCOPE

The scientific contribution of our project is the detection of ambient sound for people with hearing impairment (PWHI) using neural networks with notifications in multiple modes. Our deep learning model, ResNet50, is trained to recognize 10 sounds: Baby Crying, Barking, Cough, Dishes and Pots, Doorbell, Fire Alarm, Smoke Detector, Sneeze, Thunder, and Water. The goal of our project is to detect sounds in the user's ambiance in real-time and notify the user when one of the sounds mentioned above occurs in the surroundings so that they can take the respective action. This allows them to interact with their surroundings, be aware of their environment, and carry out day-to-day tasks with ease.

When put to the test on our test dataset, the results showed that the ResNet50 architecture had an overall f1-score of 1.0. It gave the best accuracy for our training, validation, and testing dataset compared to other models that were trained, namely, VGG16, Xception, MobileNet, InceptionV3, and InceptionResNetV2. Hence, the ResNet50 model was used to classify real-time sounds from the surroundings and send the result as a notification to the users through a mobile application.

Although the ResNet50 model has shown 100% accuracy on our test dataset, it can exhibit further improvements for real-time data classification. We can use better audio recording mechanisms to improve the audio clarity and aid the classification process. In addition to this, the implementation of Indian Sign Language (ISL) can be incorporated with the mobile application for a better understanding of PWHI.

All in all, it is important to ensure that these people are guaranteed the opportunity to experience the same comfort of living in their homes as those without hearing loss.

REFERENCES

[1] J. E. da Rosa Tavares and J. L. Victória Barbosa, "Apollo SignSound: an intelligent system applied to ubiquitous healthcare of deaf people," Journal of Reliable Intelligent Environments, vol. 7, no. 2, pp. 157–170, Jun. 2021, doi: 10.1007/s40860-020-00119-w.

[2] D. Jain et al., "HomeSound: An Iterative Field Deployment of an In-Home Sound Awareness System for Deaf or Hard of Hearing Users," in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu HI USA, Apr. 2020, pp. 1–12. doi: 10.1145/3313831.3376758.S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] A. Polo-Rodriguez, J. M. VilchezChiachio, C. Paggetti, and J. Medina-Quero, "Ambient Sound Recognition of Daily Events by Means of Convolutional Neural Networks and Fuzzy Temporal Restrictions," Appl. Sci., vol. 11, no. 15, p. 6978, Jul. 2021, doi: 10.3390/app11156978.

[4] Chuan-Yu Chang and Yi-Ping Chang, "Application of abnormal sound recognition system for indoor environment," in 2013 9th International Conference on Information, Communications & Signal Processing, Tainan, Taiwan, Dec. 2013, pp. 1–5. doi: 10.1109/ICICS.2013.6782772.

[5] S. Dass, M. S. Holi, and K. S. Rajan, "A Comparative Study on FFT,STFT and WT for the Analysis of Auditory Evoked Potentials," Int. J. Eng. Res., vol. 2, no. 11, p. 6, 2013.

[6] D. D. Jayasree, "Classification of Power Quality Disturbance Signals Using FFT, STFT, Wavelet Transforms and Neural Networks - A Comparative Analysis," in International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, Tamil Nadu, India, Dec. 2007, pp. 335–340. doi: 10.1109/ICCIMA.2007.279.

[7] N. Mehala and R. Dahiya, "A Comparative Study of FFT, STFT and Wavelet Techniques for Induction Machine Fault Diagnostic Analysis," p. 6.

[8] B. Suhas et al., "Speech task based automatic classification of ALS and Parkinson's Disease and their severity using log Mel spectrograms," in 2020 International Conference on Signal Processing and

Communications (SPCOM), Bangalore, India, Jul. 2020, pp. 1–5. doi: 10.1109/SPCOM50965.2020.9179503.
[9] A. Meghanani, A. C. S., and A. G. Ramakrishnan, "An Exploration of Log-Mel Spectrogram and MFCC Features for Alzheimer's Dementia Recognition from Spontaneous Speech," in 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, Jan. 2021, pp. 670–677. doi: 10.1109/SLT48900.2021.9383491.
[10] M. Sankupellay and D. Konovalov, "Bird Call Recognition using Deep Convolutional Neural Network, ResNet-50," p. 8, 2018.
[11] C. Giuseppe, "A ResNet-50-Based Convolutional Neural Network Model for Language ID Identification from Speech Recordings," in Proceedings of the Third Workshop on Computational Typology and Multilingual NLP, Online, 2021, pp. 136–144. doi: 10.18653/v1/2021.sigtyp-1.13.
[12] Y. Kumar, M. Vyas, and S. Garg, "From Image Classification to Audio Classification," p. 8.
[13] S. Hershey et al., "CNN Architectures for Large-Scale Audio Classification," ArXiv160909430 Cs Stat, Jan. 2017, Accessed: Jan. 25, 2022.

AUTHORS

Smitha S Maganti, student at BNM Institute of Technology, pursuing B.E in Computer Science and Engineering. Enjoys coding, and is enthusiastic about making an impact and giving back to the community. Specialization in Machine Learning and Software Development and a keen interest to learn new technologies.

Sahana S, student at BNM Institute of Technology pursuing B.E in Computer Science and Engineering. Keen to solve real life problems and automate daily tasks in the field of Healthcare, Agriculture and IT using technologies like AI, ML and IoT.

Kriti K, student at BNM Institute of Technology pursuing B.E in Computer Science and Engineering. An engineer with a deep rooted passion for coding along with the intent to translate ideas into software. A major proponent for revolutionizing the world and thus equipping myself with the necessary tools and knowledge to contribute to the movement.

Shravanthi Madhugiri, student at BNM Institute of Technology, pursuing B.E. in Computer Science and Engineering. UI/UX designer who has a passion for design and a keen eye for detail.

Priya S, assistant Professor at BNM Institute of Technology, area of specialization in Network security, Data Mining and Machine Learning.