SoundScape: Real-Time 3D Sound Localization and Classification with Sensory Substitution for the Deaf and Hard of Hearing

Eugene Choi, Grade 11, eugenechoi2004@gmail.com
Raffu Khondaker: Grade 11, rkhondaker2017@gmail.com
Irfan Nafi: Grade 11, irfan.nafi716@gmail.com
Teacher: John Zacharias, Lab Technology Teacher at TJHSST, jbzacharias@fcps.edu
School: Thomas Jefferson High School for Science and Technology (TJHSST), Alexandria VA
Category: Robotics and Intelligent Machines

Current devices geared towards the deaf and hard of hearing, such as hearing aids, struggle to localize and transmit sounds to those with severe hearing impairments. Advanced devices, like cochlear implants, are invasive and cost $30,000 to $50,000. Devices that classify sounds, such as home alert systems, are not suited for mobile use and only recognize a limited number of noises, such as alarms and doorbells. The purpose of this project was to classify, localize, and transmit both environmental sounds and human speech to those with hearing impairments through a low-cost device worn around the user's neck. We performed sound localization with the SRP-PHAT-HSDA algorithm which uses a directivity model and the Time Difference of Arrival of incoming audio captured by a 6-microphone array. This allowed us to isolate and stream up to 4 independent audio sources to a remote server for real-time sound classification. We classified audio using an ensemble learning network utilizing stacking, where 4 deep-learning models, trained on the ESC-50 dataset, combined their outputs to produce a final prediction. The sounds were transmitted to the wearer through sensory substitution, where vibrations allowed them to feel the varying amplitudes and frequencies of sounds. Our device accurately predicted the direction (degrees) of up to 4 speakers playing simultaneously with 15.49 RMSE. The ensemble learning network also outperformed human accuracy (81.3%) by about 12%. Our results show that sound localization can be performed with a cost-efficient microphone array while classifying and transmitting audio through touch.

Statement of Outside Assistance

This project was conceived and developed entirely by our team. The project began in August of 2020 and was inspired by Eugene's grandfather in Korea, who is deaf in his right ear and hard of hearing in the other. Eugene noticed that his grandfather struggled with localizing and distinguishing between sounds despite wearing hearing aids. After further research, our team realized most hearing aids fail to accurately localize sound, with many lower-cost hearing aids unable to separate sounds. Due to COVID-19, our team spent the entirety of this project at home. We accessed our school's Linux Workstations remotely to conduct the majority of the machine learning training. Our supervisor, Dr. Zacharias, gave us feedback when we pitched our original design, assisted us when accessing our school's resources, and also reviewed this paper. However, our team of three conducted the entirety of the research process independently. This project was not a continuation of any existing research.

Table of Contents

I. Introduction

According to the World Health Organization, over 466 million people suffer from hearing loss, and that number is projected to rise to 900 million by 2050 [18]. Hearing loss is not a unique condition and can happen to anybody as one in two elderly citizens over age 75 suffer from hearing loss [15]. Popular assistive devices that aid those with hearing impairments include hearing aids, cochlear implants, and phone applications. Hearing aids wrap around the user's ear and amplify surrounding noises but have several drawbacks. They are too expensive and can cost upwards of $7,000, with many insurance plans not covering their costs. This price is unaffordable for many hearing-impaired citizens as the majority are the elderly who have a yearly income of $25,000 or less [25]. Additionally, hearing aids amplify unwanted background noises. The human auditory system can subconsciously filter out background noise from noisy environments such as restaurants and bars. On the other hand, hearing aids amplify all noises rather than just the ones the user wants to focus on causing the user to have a hard time distinguishing individual voices in crowded areas. Finally, most hearing aids are unable to localize sound, which is essential for situational awareness. It helps us separate conversations in crowded areas, avoid incoming threats (such as a speeding car), and react to our names being called.

When hearing loss becomes too severe, hearing aids are insufficient and users have to turn towards cochlear implants. Cochlear implants are listening devices surgically inserted in the head to stimulate the auditory nerves, but they can cost 30,000 to 50,000 and do not restore normal hearing [26]. Additionally, due to their invasive nature, they can cause complications such as constant ringing, infections, and internal fluid leaks. An alternative to cochlear implants is phone applications that transcribe speech to text and classify sounds. However, phone microphones have a limited range, cannot perform sound localization, and such apps do not classify sounds.

Our engineering goal was to design a device that would provide a greater degree of safety to anyone with any degree of hearing loss. We sought to do this by conveying essential auditory information to the user, so the directionality, what made the sound, as well as pitch and amplitude. This way, we hope to give people with hearing impairments a greater degree of safety. We sought to do this in a comfortable, wearable device that operates in real-time, and under $100. We broke down our task into four specific goals:

1. Track and separate up to 4 audio sources.
2. Classify each separated sound source as either 1 of 50 environmental sounds or perform speech recognition if a human voice is detected.
3. Convey other audio data such as directionality, amplitude, and frequency to the hearing impaired user.
4. (Constraints) Have the device be wearable, operate in real-time, and cost under $100.

Our solution is SoundScape, the first assistive listening to separate, localize, and classify multiple sound sources for anyone with hearing loss. We developed a haptic design to be worn around the user's neck which vibrates in relation to the direction, amplitude, and frequency of a sound source (for our prototypes, we switched out the motors with LEDs for the sake of live demonstration). We developed both a mobile and WatchOS app which displays the classifications for the sound source along with any speech detections. We do all of this at an

affordable price of $60 which is much lower than the assistive hearing devices that are priced in the thousands.

II. Materials, Methods, and Procedures

A. Design and Prototype

Existing haptic devices for the deaf do not incorporate directional audio or multiple independent audio sources. Neosensory Buzz, for example, is a bracelet that vibrates according to general environmental audio instead of separating it into distinct audio sources. This means that if simultaneous audio sources are playing at once, Buzz will vibrate according to their combined audio, which decreases the user's understanding of the audio sources. Furthermore, this device costs $300 per year to use.

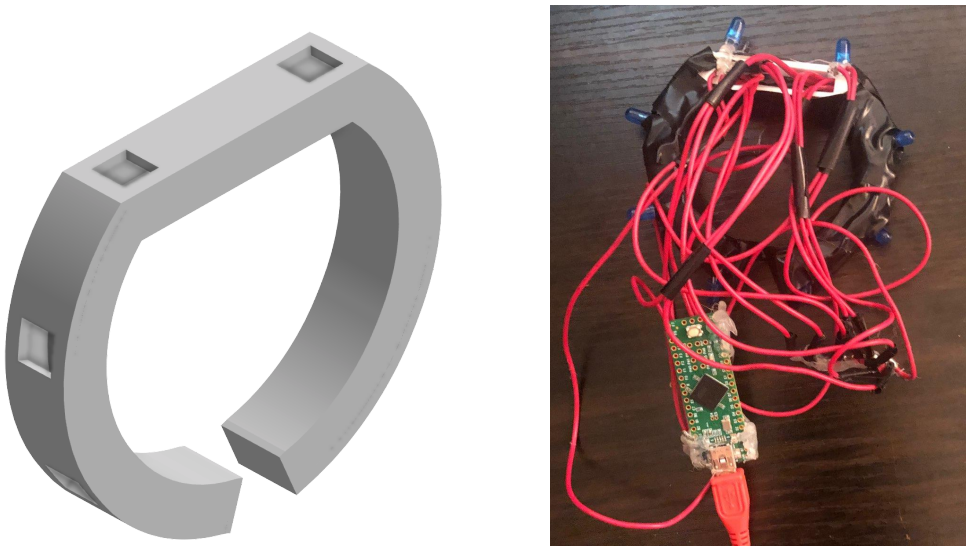

Figure 1. 3D CAD model of the bracelet on the left with the physical prototype and electronics on the right. The LEDs were used in place of vibration motors to display the intensity of a source.

Our initial directional vibrational prototype was a bracelet that vibrated towards the direction of sound sources. We have eight LEDs (Light Emitting Diodes) placed equally around the surface to cover an equal angular region. The intensity of the LEDs is controlled through PWM analog ports on the Teensy 2.0 microcontroller.

For sound localization to occur, however, the microphone array to be in the same plane as the bracelet (Section II.B). This is not the case for daily use, as the wrist is constantly moving. Furthermore, the small size of the bracelet means that a limited number of motors can fit, so we can not have the motor pairs we use to convey frequency. This design would not be sufficient for day-to-day use since this essential information, so we decided to create a new design.

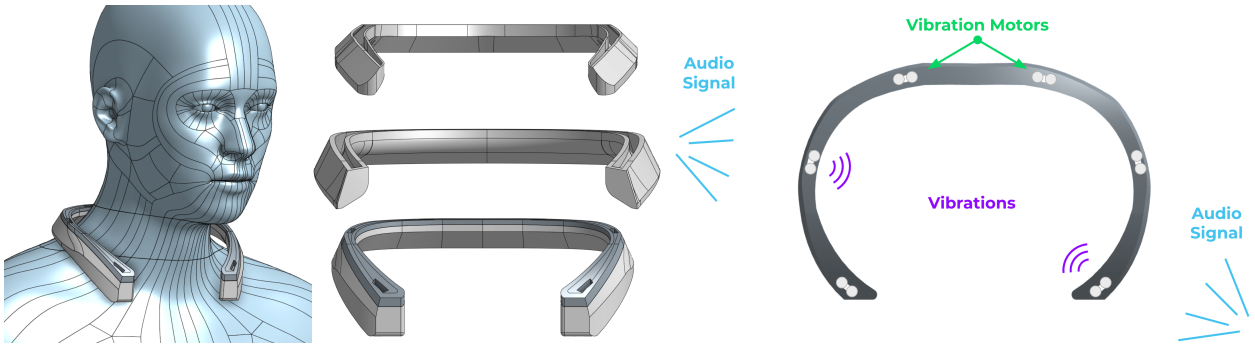

Figure 2. The image on the left shows the model fitting around a person's shoulders. The center image is the three prototype iterations we went through, with the right image showing the motors on the model and the motor directionality.

We had a total of four iterations on the neck-based design, with the first 3 shown in Figure 2. These iterations were used to determine the best design that maximized contact with the neck. For the third prototype, we embedded the electronics, this time with a total of twelve LEDs spread out to form six pairs. The same principle of directionality extends to this design, where the LEDs closest to the sound source light up. Furthermore, due to the additional space, we were able to have motor pairs for every spot, allowing us to convey two bands of frequency. Within each pair, the motor on the right vibrates for low frequencies and the motor on the left vibrates for high ones. The threshold frequency between high and low for the motor pairs is 640 Hertz, which we determined by analyzing the median frequency from the ESC-50 dataset. In the 3rd prototype iteration for the neck design, we used two Arduino Nanos to control 12 LEDs through PWM, since the two microcontrollers combined have a total of 12 analog out ports.

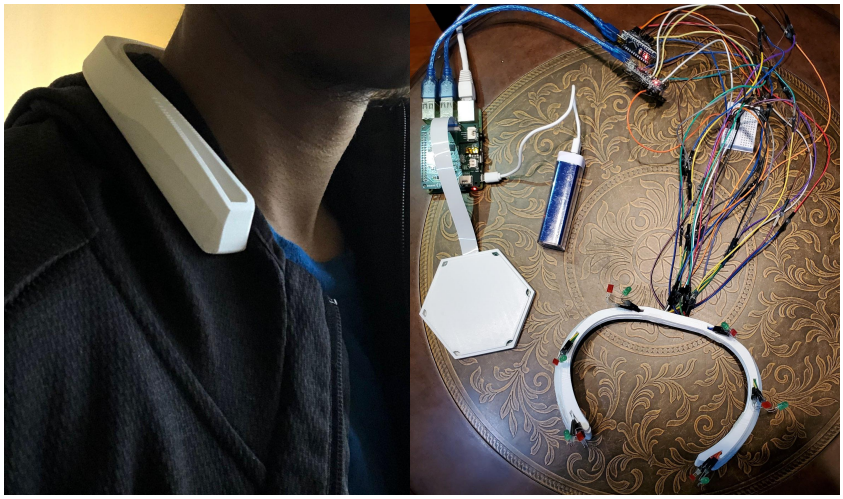

Figure 3. The image on the left is the 3rd prototype being worn around the neck. The image on the right shows the full circuitry of the prototype, with the microphone array and Raspberry Pi.

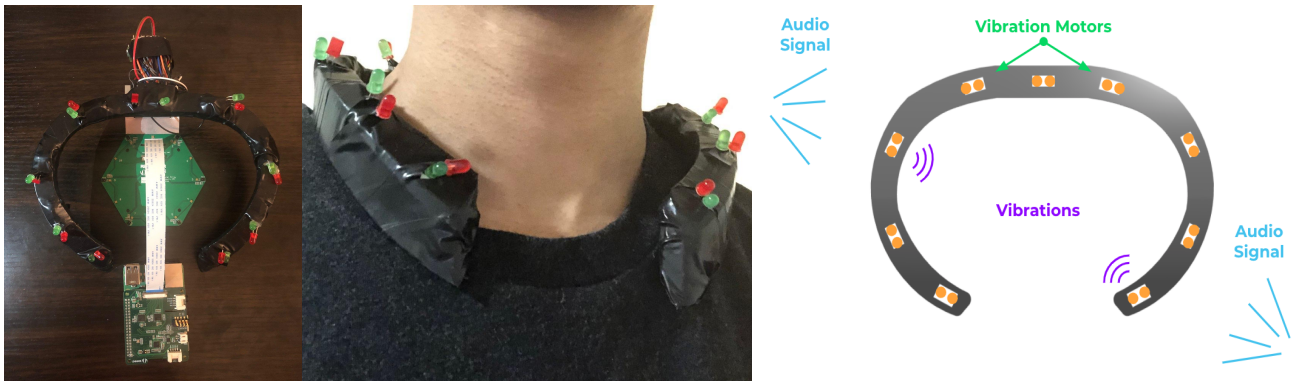

Figure 4. The fourth and current neck design iteration. The leftmost image shows the compressed design with the microphone array in the same plane. The middle image shows the design wrapping around the neck with the rightmost image showing the directionality principle with 9 motor pairs.

The electronics in the third neck design iteration were very sparse and not practical for wearable purposes. To mitigate this issue, we used a more compact microcontroller system, by using a Teensy 4.0 in our fourth and current iteration. This board is smaller than an individual Arduino Nano while having 31 analog ports, with 19 on the front side. To relay directionality, we have nine pairs of vibration motors spread evenly throughout the device, as shown in Figure 4. The localization and separation software is run on a Raspberry Pi 3B+ and the motors are connected to a Teensy 4.0. The Teensy is connected via USB to the Raspberry Pi, which sends instructions on the specific motors to vibrate and at what intensity through the USB Serial connection. The Teensy controls the intensity through PWM (Pulse Width Modulation) ports. The audio is captured by a 6-microphone array connected to the Raspberry Pi and can be placed anywhere on the user's upper body. The cost of all these electronics is about $60.

B. Sound Localization

Sound source localization is the process of identifying and isolating individual sound sources in the environment. The ability was evolved by vertebrates 200 million years ago through the middle ear which allowed for binaural hearing. This was essential to locate prey and predators in 3D space and was thus key to their survival. Binaural hearing works due to interaural time difference (ITD), the time it takes for sound to travel between the two ears. The human auditory system subconsciously detects the ITD and uses visual context clues to not only determine the location of the sound but also track it as it moves [14].

We sought to localize sounds in a computerized environment by implementing the open embedded auction system (ODAS) and by using a 6-microphone array. Using this model, we can track and locate multiple sound sources simultaneously. ODAS first makes generalized predictions for each sound source by using the Steered Response Power with Phase Transform, or SRP-PHAT. Similar to a human, SRP-PHAT calculates the ITDs between adjacent microphones through Generalized Cross Correlation with Phase Transform or GCC-PHAT. This relies on a similar principle to binaural hearing, which relies on the fact that sound from the right reaches the microphone on the right sooner than the one on the left. The first run of SRP-PHAT generates a list of candidate sources, each with a broad area. We then pinpoint the final sound

sources by running SRP-PHAT on the candidate regions, leaving us with up to four distinct sound sources [2]. This way, we can mitigate the interference from multiple sound sources while being computationally efficient.

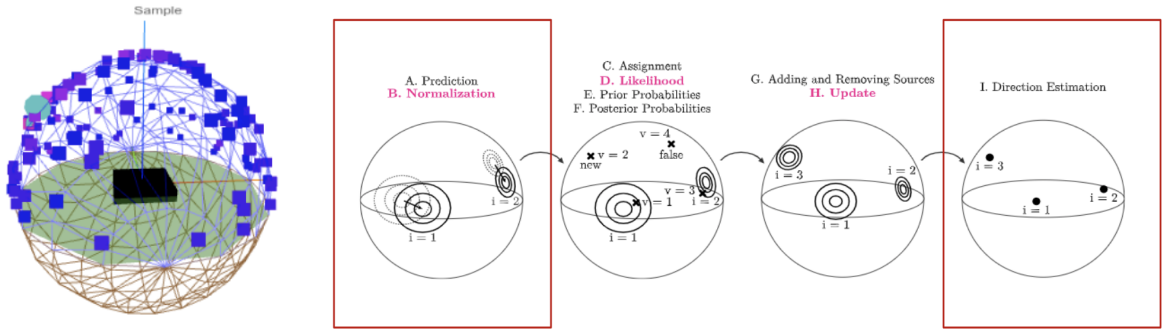

Figure 5. Diagram of how SRP-PHAT model pinpoints sound sources. Using outputs of GCC-PHAT, the model generates a list of potential sources (on the left). It then generates a broader region of each candidate and runs GCC-PHAT on each sub-region to generate the final list of sound source directions.

C. Sound Separation

For sound separation, we use the convolutive Geometric Sound Separation, or GSS, which combines multiple cross-power minimizations from source separations with geometric linear constraints to separate the sound channels. One technique that GSS uses is beamforming. When sound from a source arrives at microphones, they do so at different times since the mics are positioned apart from each other. For different sound sources positioned at different places, each mic's time difference will also be different. If you want to hear sound from a particular direction, in our case the direction is provided by ODAS's sound localization, you can make that sound source from that direction constructively interfere, with other sources destructively interfering, leaving only the sound source you want. This is demonstrated in Figure 6.

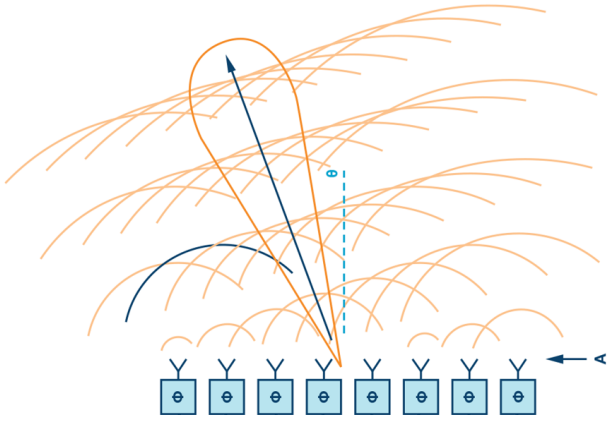

Figure 6. Example of beamforming at work with 2 different sound sources.

D. Sensory Substitution and Noise Reduction

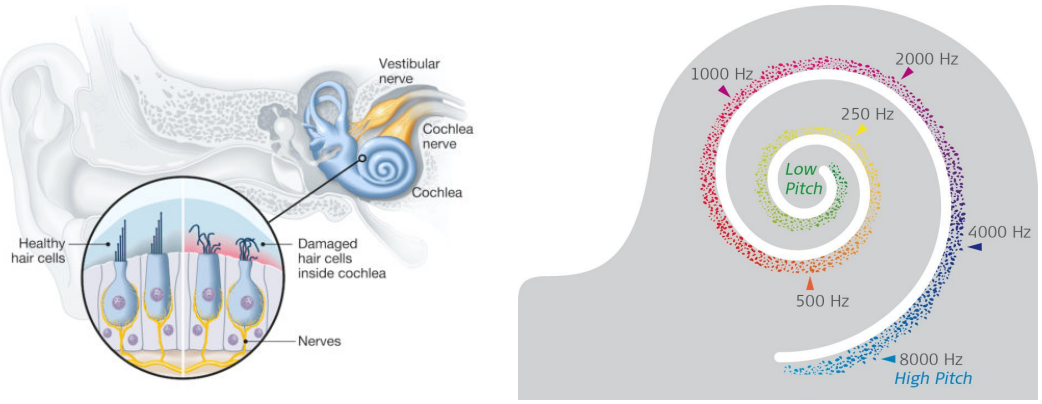

Figure 7. On the left is a Visual Representation of damaged and healthy hair cells. On the right is a representation of what audio frequencies correlate with distinct regions of the cochlea

As we explain in Sections II.E-I, we perform sound classifications to identify what a sound source is. But certain audio features cannot be translated into text such as music. To address the issue of communicating sounds to those with hearing impairments, we drew inspiration from the human auditory system itself.

Sound waves enter the ear and are recognized by the hair cells on the snail shaped cochlea. The hair cells near the wide end of the cochlea detect higher-pitched sounds such as a baby crying while the cochlea closer to the center recognizes lower-pitched sounds as represented in Figure 7. And the intensity the hair cells vibrate in is directly correlated to the amplitude of the audio. The vibrations from the hair cell are then translated into electrical firings in our brain. For the deaf and hard of hearing these hair cells are damaged and are unable to transmit the nature of the audio as shown in Figure 7. Through sensory substitution, we could convey the same information with haptic motors on the neck.

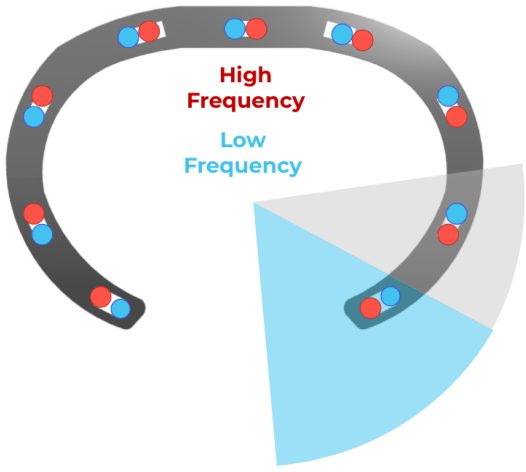

Figure 8. On the left, Frequency-Amplitude-Time graphs showing frequency extraction from background noise and its removal from audio. The diagram on the right shows the location of high and low-frequency motors along with the area coverage for 2 motors.

Instead of vibrating hair cells, our device uses coin vibration motors placed around the user's neck. While our vibration motors are not as sensitive as human hair cells, because of the neuroplasticity of human brains users can learn and recognize sounds from the vibrations within just a couple of minutes of training. In each pair of motors, we implement sensory substitution by determining the intensity of the vibrations by the amplitude of the sound. Within each pair of motors, the right motor vibrates in cases of high-frequency sounds, while the left motor vibrates for low-frequency sounds, as shown in Figure 8. We distinguish between high and low by analyzing the median frequency for audio in the ESC-50 datasets. In each audio clip, we took one data point as the weighted frequency in every 0.18 seconds of audio, where the weights are the amplitudes for the frequencies. Then for each audio file, we calculated the median weighted frequency. Finally, we averaged the median weighted frequency across all audio files to reach our frequency threshold of 640 HZ. This distinction of high frequency and low frequency has proven useful for lip-reading, as shown by MIT's Sensory Communication Group [23]. Certain consonants have identical lip formations that are impossible to distinguish based on sight alone, such as "p" and "b." However, the additional information of frequencies through vibrations reported a boost in lip reading performance as users were able to associate "p"s with higher frequencies and "b"s with lower frequencies.

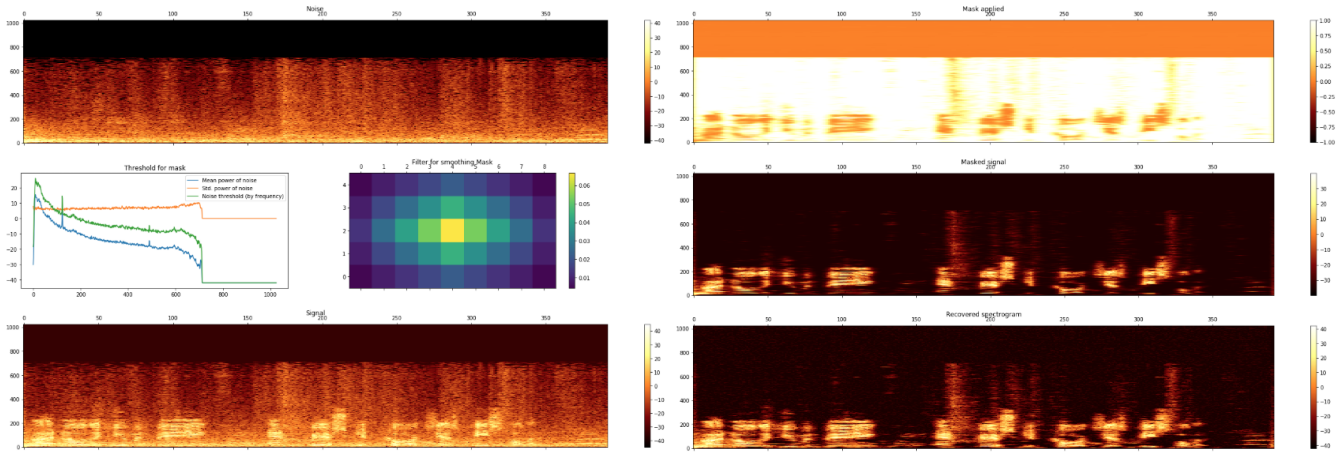

Figure 9. Frequency-Amplitude-Time graphs showing frequency extraction from background noise and its removal from audio.

When implementing sensory substitution, we realized that the device continued to vibrate due to the background noise's amplitude. To remove the background noise, we use a noise reduction algorithm that records an audio clip of the background noise. To determine if the audio clip is background noise, we look at the amplitude variance over the audio. If the variance is under an average standard deviation of 0.5, it is used as background noise. Then a Fourier Transform (FT) is calculated over the noise audio clip to extract the frequency-amplitude data, as shown in the first graph. Standard deviation and the mean for each frequency level are calculated to determine the variance for sound intensity in decibels to be removed. Then an FT is calculated over the stream or incoming audio as shown in the second graph. A mask composed of all the thresholds previously calculated is determined and sent through a Gaussian filter so that the mask is smoothed out, removing any sharp frequency thresholds. Finally, the mask is applied to the FT of the signal and produces the final output. The full system is summarized in Figure 9. Finally, to ensure noise reduction in varying environments, the background noise recording is changed if the

average amplitude of the user's environment changes for an extended period. This triggers a re-recording of the background noise which is then applied to the incoming audio.

E. Environmental Sound Classification

The Environment Sound Classification dataset (ESC-50) was used in training the models for environmental sound classification. The dataset consists of 2,000 labeled audio clips organized into 50 classes, with there being 40 recordings per class [6]. The classes are further split into more general categories: animal sounds, natural soundscapes, water sounds, human (non-speech) sounds, interior/domestic sounds, and exterior/urban noises. Each recording, which was sampled from www.freesound.org, is 5 seconds long, sampled at a uniform rate of 44.1kHz, and is monophonic, or contains 1 audio source per sample. The dataset is split into five predetermined folds and we evaluated each model through cross-validation across the 5 folds. The accuracy of the human auditory system in classifying this dataset was also approximated on this dataset. Each participant was exposed to one of the 2,000 audio recordings and was tasked with classifying the sound under one of the 50 classes. By the end of this test, the researchers gathered 4,000 human predictions. The average accuracy attained by this test was 81.3%. However, this number was treated as a rough approximation as this test lacked a formal experimental setup.

There have been several ways to represent audio samples with the simplest representation being the raw waveform. While models with 1-D convolution such as EnvNet[16] have been used to classify raw audio, representing audio through a variation of the spectrogram yield better accuracy. These variations include Log-Spectrograms, MFCCs, and Gammatone-Spectrograms. We decided on converting the audio recordings of the ESC-50 dataset to the Mel-Spectrograms as [17] and [4] have shown that this variation of the spectrogram is best suited for feature extraction for our database. A spectrogram is a visualization of the spectrum of frequencies of a signal as it varies with time. Since the frequency varies with time, a Fourier transform is calculated on overlapping segments of the signal. A Mel Spectrogram transforms that frequency to the Mel Scale or a scale based on how humans perceive sound, shown in Figure 10. The Mel Spectrogram was computed across three channels as explained by [4] with window sizes of 25ms (milliseconds), 50ms, and 100ms, and hop lengths of 10ms, 25ms, and 50ms, respectively. Computing the spectrograms in this way allowed us to account for a wider range of frequency and time information.

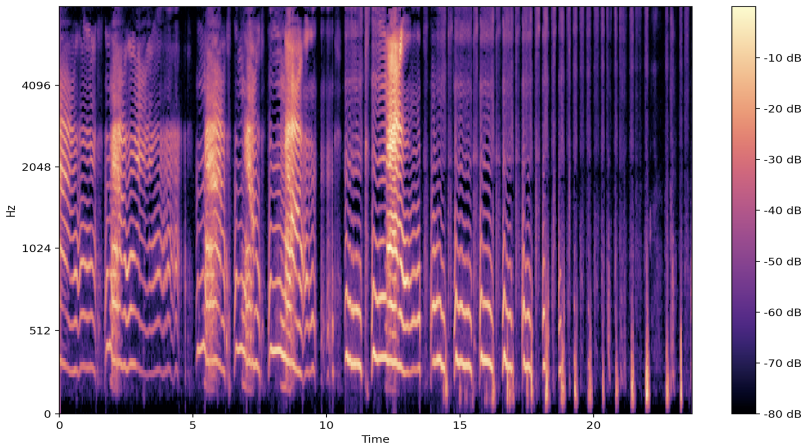

Figure 10. Mel-frequency Spectrogram of an audio clip of a baby crying from the ESC-50.

To increase the amount of training data and to reduce overfitting, we performed data augmentation or deforming the original sound files without losing what the original sound was. We did this in 3 ways, as proposed by [3] with some minor changes. Our first augmentation technique is a time stretch. This randomly stretches or compresses the audio sample by a factor of either 0.75, 0.9, 1.15, 1.25. Our second augmentation technique is shifting the pitch of the audio file. This randomly raises or lowers the pitch by a factor of −3.5, −2.5, 2.5, 3.5. Our final technique is overlaying an audio clip with background noise. Here we mix the audio clip with 1 of 3 recordings, each of different scenes: street traffic, street people, and the park. We verified that none of these recordings contained any of 50 sound classes.

We used four base models trained on the augmented ESC-50 dataset. These base-models were run together in a meta-classifier with each base-model using different techniques. ResNet [19], or Residual Network, is a network of stacked residual blocks, with each block having two 3x3 convolutional layers and 2 output channels followed by a layer performing batch normalization and a ReLu activation function. This architecture makes use of an identity shortcut connection that skips two convolution operations and passes the input through a final ReLu activation function. This addresses the vanishing gradient problem, where the gradient can become infinitely small after back-propagation to earlier layers resulting in a subpar performance. DenseNet [20], or Dense Convolutional Network, is a network where every layer has direct access to the gradients from the loss function. Similar to feed-forward networks, each layer is connected to the following layer such that the inputs of a layer are the feature-maps of all of the previous layers. DenseNets are split into DenseBlocks which consist of feature maps with the same dimensions but different filters or transition layers. CNN10 [21] is a convolutional neural network of 10 layers with 4 convolutional layers. Each convolutional block consists of 2 convolutional layers with kernel sizes of 3 x 3 with batch normalization applied between each block. ReLU's nonlinearity increased the speed and stability of training. TALnet [22], or Tagging and Localization network, performs audio tagging and localization (we use this network strictly for classification as our localization method is more recent and accurate). TALnet is a convolutional and recurrent neural network that consists of 10 convolutional layers with 5 max-pooling layers in between. This feeds into a Gated Recurrent Network (GRU) followed by a final, fully connected layer.

F. System Summary

The final system integrates sound localization, separation, classification, and sensory substitution into one cohesive flow. The process starts with a 6-microphone array capturing audio from the environment. The data is processed on the Raspberry Pi, which performs sound localization and separation through ODAS and GSS, respectively. The 3D coordinates from sound localization are sent to the Firebase Real-time database. The raw audio data from separate audio sources are streamed to a remote server, where we run sound classification, noise reduction, and natural language processing (NLP) in parallel on each source.

The audio sources are streamed in chunks of 8192 bits (.18 seconds) of raw audio data for real-time environmental sound and speech classification. The server receives these chunks in a queue where the audio is classified cumulatively as either environmental sounds or as text. If a human voice is detected, the audio is converted to text through NLP using the Google Cloud Speech API. Otherwise, the audio is classified through the environmental sound classification model, labeling it as one of 50 sound classes. The outputs of both NLP and sound classification, for each source, are stored in our Real-time Firebase Database. This information is displayed on

the app, with each audio source labeled with classification and its predicted direction. The final process run on the server is the Noise Reduction algorithm that reduces background noise. It then extracts amplitude and frequency and streams this data back to the Raspberry Pi and then to the motors. The motors vibrate in correspondence with amplitude, frequency, and direction around the user's shoulders and neck. This streaming of data from the Pi to the server and back has a latency under 50 milliseconds and updating the app has a latency of 30 milliseconds.

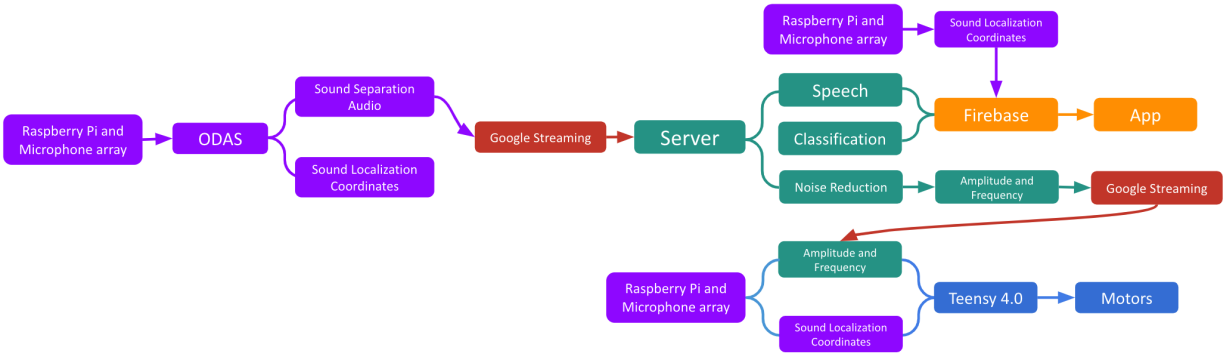

Figure 11. Full system diagram with entire device network.

Figure 12. The Mobile app displaying the classification and speech as well as the locations of the distinct sound sources are shown on the left while the watch app with the sound classification is shown on the right.

III. Testing

A. Deep Ensemble Network

To further enhance the overall accuracy of our system, we employed an ensemble learning technique called stacked generalization or stacking. Stacking combines the predictions from different machine learning models trained on the same dataset [23] as shown in Figure 13. We did this by training a meta-classifier that uses logistic-regression on how to best use the predictions from each base-model. Our base models were Densenet, CNN10, TalnetV3, and Resnet18, each explained in Section II part I.

To train and evaluate the accuracies of both our base-models and meta-classifiers, we employ a variation of 5-fold cross-validation which is illustrated in Figure 13. The ESC50 dataset is split into 5 sections, or folds, with each of the audio samples randomly placed into a fold. During training, one of the five folds was labeled as the testing fold while the model trained on the remaining four-folds. The model's accuracy was then evaluated on the isolated testing fold with each prediction made on the testing fold, the out-of-fold predictions, saved for later use. This process was repeated for the remaining 4 folds, with the final classification accuracy of a base-model being the average of the mean accuracies of each testing fold. Once each base-model was trained and evaluated, we trained the meta-classifier with the out-of-fold predictions.

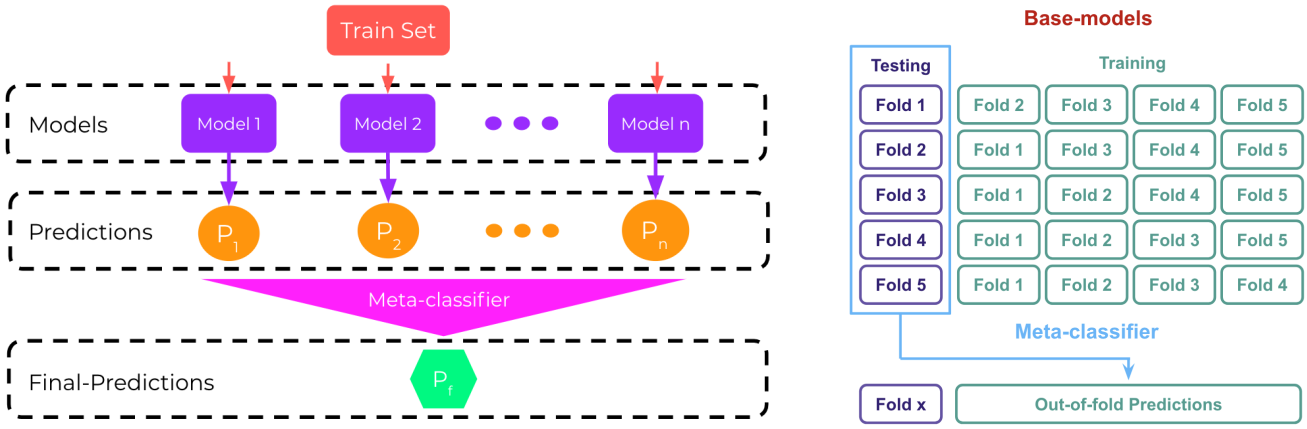

Figure 13. The diagram on the left shows how stacking meta-classifiers are developed. The diagram on the right lays out 5-fold cross-validation training with the base-models and training the meta-classifier on the out-of-fold predictions.

B. Sound Localization Test

To test the accuracy of the sound localization algorithm, we placed the microphone array of the SoundScape in the center of the room encircled by 6 equidistant speakers each 5 meters away from the center. The configuration is shown in Figure 14. We then played white noise from each of the sound sources at a time. While this happened, the sound localization algorithm predicted and recorded the location of the sound sources in degrees. We repeated this process, but with different combinations of 2, 3, then finally 4 sound sources playing white noise simultaneously. For example, when we tested the algorithm on 2 audio sources, we played audio from every possible combination of 2 speakers for 6 audio sources, or 6 choose 2. In total, we tested on a total of 56 speaker configurations. After we collected our data, we calculated the root mean squared error between the recorded degree values and the actual degree values for all combinations.

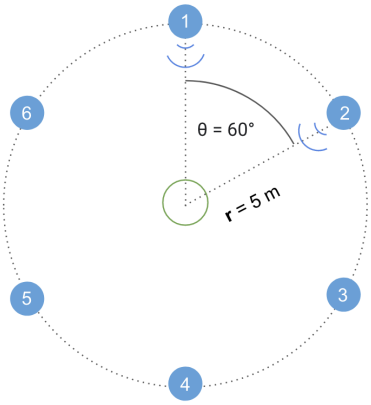

Figure 14. The diagram depicts the speaker configuration. In the center (green) is the six microphone array surrounded by the six speakers (blue) numbered.

IV. Results

Figure 15 displays the results for the sound localization test. We displayed the Root Mean squared Errors (RMSE), in degrees, of the predicted vs. actual sound directions. The dashed lines represent the average root-mean-squared error for each audio source trial. Each point represents one of the 56 different speaker configurations mentioned above. The RMSEs were 3.61, 5.82, 10.15, and 15.49 for 1, 2, 3, and 4 audio sources, respectively. Figure 15 shows that the degree error increased as more audio sources were added. Figure 16 shows the accuracies of the base-models and meta-classifiers. Each classifier outperformed human accuracy of about 81% on the ESC50 dataset with the meta-classifier outperforming the other models, at 93.4% accuracy.

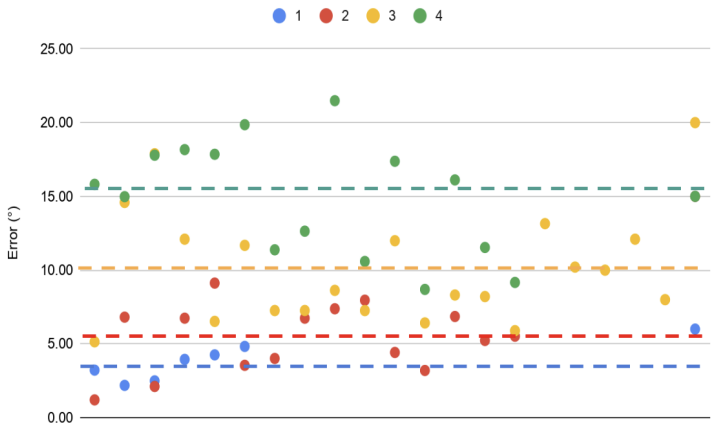

Figure 15. Error, in degrees, of predicted vs actual sound directionality for each number of sound sources (1 through 4) represented by the dashed lines. Each data point represents a speaker configuration.

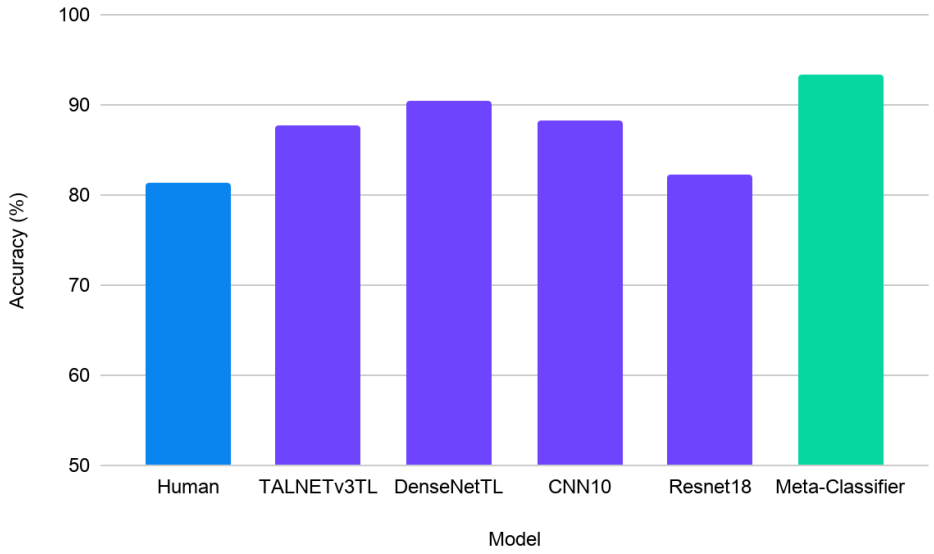

Figure 16. Accuracies of base-models and Meta-classifier on ESC-50 compared to human classification accuracy. The meta-classifier had the highest accuracy. *TL indicates transfer learning.

V. Conclusion and Discussion

The goal of this project was to create a wearable device under $100 for people who have progressively lost their hearing. Our solution was SoundScape, the first assistive listening device that conveys essential auditory information to anyone with any degree of hearing loss non-invasively. Unlike hearing aids, our device can localize sounds, identify what the sound is, and, most importantly, conveys sound to people who are deaf. We did all of this under $60.

In the sound classification system, the Meta-classifier, as expected, had the highest accuracy and could still classify sounds in real-time through a queue of audio chunks. However, the meta-classifier was computationally expensive as it ran the 4 base-models simultaneously. Another obstacle in sound classification was the limitation of training data as the ESC-50 dataset only included 40 audio samples per class. This is due to the high cost of manually annotating sound samples and ensuring that each sound sample is monophonic. From a practical standpoint, future research should also try to optimize the base-models to run locally on the phone, removing the need for an external server. The sound localization algorithm tracked different sound sources live and the sound separation allowed for the classification of the audio sources in real-time, although the localization became less accurate as more sound sources were added as seen in Figure 15. This is because adding audio sources increases audio interference, further complicating sound separation. For sensory substitution, we created a wearable design worn around the neck and shoulders that conveys both the feeling and frequency of sound through vibrations. However, frequency interpretation is limited since there are only 2 motors per pair which only convey 2 frequency groups. We can improve this by adding more motors in each motor group and decreasing their size.

Our current design is suited for active outdoor use with a range of 5 meters. As stated, the device can track and localize audio of up to 4 sound sources and notify the user of direction and other sound data through a vibration on the neck. As of now, the microphone array must be

attached to the neck-design externally either on a backpack, purse, or attached to the user's clothing. To make the device more stream-lined, we plan to design a flexible PCB with microphones so that we can embed the microphone array within the outer case as opposed to using an external microphone array. The audio and speech classifications are made in real-time and can classify up to 4 separate sound sources. This information is currently displayed on a mobile app along with an Apple Watch app for better accessibility. We hope that this research advances into a low-cost wearable device on the market to assist the Deaf and Hard of Hearing in their day-to-day lives.

Figure 17. The current prototype, with 9 motor pairs and compact circuitry through the Teensy 4.0 microcontroller.

VI. Bibliography

[1] Massachusetts Institute of Technology. (2009, March 2). New Devices Aid Deaf People By Translating Sound Waves To Vibrations. ScienceDaily. Retrieved December 21, 2020 from www.sciencedaily.com/releases/2009/02/090227112311.htm

[2] Grondin, F., & Michaud, F. (2018). Lightweight and optimized sound source localization and tracking methods for open and closed microphone array configurations. ArXiv:1812.00115 [Cs, Eess]. http://arxiv.org/abs/1812.00115

[3] Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters, 24(3), 279–283. https://doi.org/10.1109/LSP.2017.2657381

[4] Palanisamy, K., Singhania, D., & Yao, A. (2020). Rethinking cnn models for audio classification. ArXiv:2007.11154 [Cs, Eess]. http://arxiv.org/abs/2007.11154

[5] Arnault, A., Hanssens, B., & Riche, N. (2020). Urban Sound Classification: Striving towards a fair comparison. ArXiv:2010.11805 [Cs, Eess]. http://arxiv.org/abs/2010.11805

[6] Piczak, K. J. (2015). Esc: Dataset for environmental sound classification. Proceedings of the 23rd ACM International Conference on Multimedia, 1015–1018. https://doi.org/10.1145/2733373.2806390

[7] Guzhov, A., Raue, F., Hees, J., & Dengel, A. (2020). Esresnet: Environmental sound classification based on visual domain models. ArXiv:2004.07301 [Cs, Eess]. http://arxiv.org/abs/2004.07301

[8] Adapa, S. (2019). Urban sound tagging using convolutional neural networks. ArXiv:1909.12699 [Cs, Eess]. http://arxiv.org/abs/1909.12699

[9] Grondin, F., Létourneau, D., Ferland, F., Rousseau, V., & Michaud, F. (2013). The ManyEars open framework: Microphone array open software and open hardware system for robotic applications. Autonomous Robots, 34(3), 217–232. https://doi.org/10.1007/s10514-012-9316-x

[10] Frechette, M., Letourneau, D., Valin, J.-M., & Michaud, F. (2012). Integration of sound source localization and separation to improve Dialogue Management on a robot. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. doi:10.1109/iros.2012.6385565

[11] Introlab/odas. (2021). [C]. IntRoLab. https://github.com/introlab/odas (Original work published 2017)

[12] Sainburg, T. (2021). Timsainb/noisereduce [Jupyter Notebook]. https://github.com/timsainb/noisereduce (Original work published 2019)

[13] Phased array beamforming ics simplify antenna design | analog devices. (n.d.). Retrieved January 30, 2021, from https://www.analog.com/en/analog-dialogue/articles/phased-array-beamforming-ics-simplify-antenna-design.html

[14] Risoud, M., Hanson, J.-N., Gauvrit, F., Renard, C., Lemesre, P.-E., Bonne, N.-X., & Vincent, C. (2018). Sound source localization. European Annals of Otorhinolaryngology, Head and Neck Diseases, 135(4), 259–264. https://doi.org/10.1016/j.anorl.2018.04.009

[15] "Hearing Loss: A Common Problem for Older Adults." National Institute on Aging, http://www.nia.nih.gov/health/hearing-loss-common-problem-older-adults. Accessed 15 Feb. 2021.

[16] Y. Tokozume and T. Harada, "Learning environmental sounds with end-to-end convolutional neural network," in 2017 IEEE International Conference on Acoustics,

Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 2721–2725.
[17] M. Huzaifah, “Comparison of time-frequency representations for environmental sound classification using convolutional neural networks,” 2017.
[18] Deafness. https://www.who.int/news-room/facts-in-pictures/detail/deafness. Accessed 15 Feb. 2021.
[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[21] Kong, Qiuqiang, et al, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition." arXiv preprint arXiv:1912.10211 (2019).
[22] Yun Wang, "Polyphonic sound event detection with weak labeling", PhD thesis, Carnegie Mellon University, Oct. 2018.
[23] L. Nanni, Y. M. Costa, R. L. Aguiar, R. B. Mangolin, S. Brahnam, and C. N. Silla, “Ensemble of convolutional neural networks to improve animal audio classification,” EURASIP Journal on Audio, Speech, and Music Processing, vol. 2020, no. 1, pp. 1–14, 2020.
[24] “Good Vibrations.” MIT News | Massachusetts Institute of Technology, https://news.mit.edu/2009/deaf-touch-0226. Accessed 16 Feb. 2021.
[25] Income of Today’s Older Adults | Pension Rights Center. 6 Jan. 2011, https://www.pensionrights.org/publications/statistic/income-today%E2%80%99s-older-adults.
[26] “Cochlear Implant: Cost, Pros, Cons, Risks, How It Works.” Healthline, 27 Feb. 2020, https://www.healthline.com/health/cochlear-implant.
[27] Gatehouse, S., & Noble, W. (2004). The speech, spatial and qualities of hearing scale(Ssq). International Journal of Audiology, 43(2), 85–99. https://doi.org/10.1080/14992020400050014