



Electronic Theses and Dissertations

2023

A Sound classification and display tool for assisting the deaf and hard-of-hearing: a case of Kenya.

Wanjiru, Rosemary Wangari
School of Computing and Engineering Sciences
Strathmore University

Recommended Citation

Wanjiru, R. W. (2023). *A Sound classification and display tool for assisting the deaf and hard-of-hearing: A case of Kenya* [Strathmore University]. <http://hdl.handle.net/11071/13522>

Follow this and additional works at: <http://hdl.handle.net/11071/13522>

A Sound Classification and Display Tool for Assisting the Deaf and Hard-of-Hearing: A Case of Kenya



Master of Science in Information Technology

2023

A Sound Classification and Display Tool for Assisting the Deaf and Hard-of-Hearing: A Case of Kenya

Wanjiru Rosemary Wangari

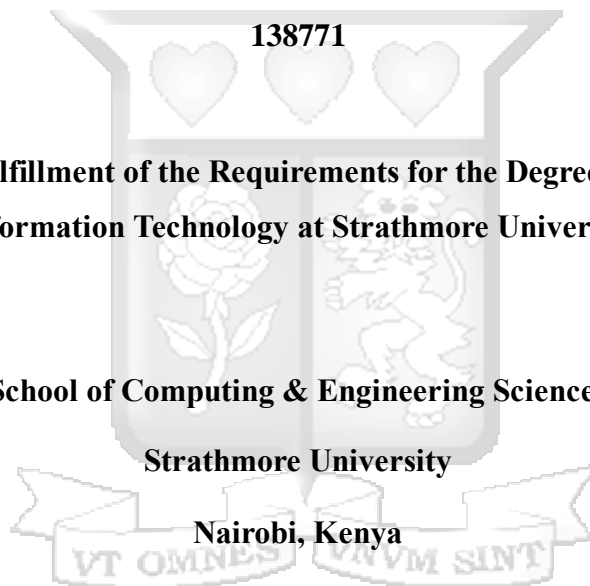
138771

**Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in
Information Technology at Strathmore University**

School of Computing & Engineering Sciences

Strathmore University

Nairobi, Kenya



July, 2023

This thesis is available for Library use on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

© No part of this thesis may be reproduced without the permission of the author and Strathmore University

Student's Name: Wanjiru Rosemary Wangari

Sign:



Date: 09/06/2023

Approval

The thesis of Wanjiru Rosemary Wangari was reviewed and approved for examination by the following:

Dr. Victor Rop

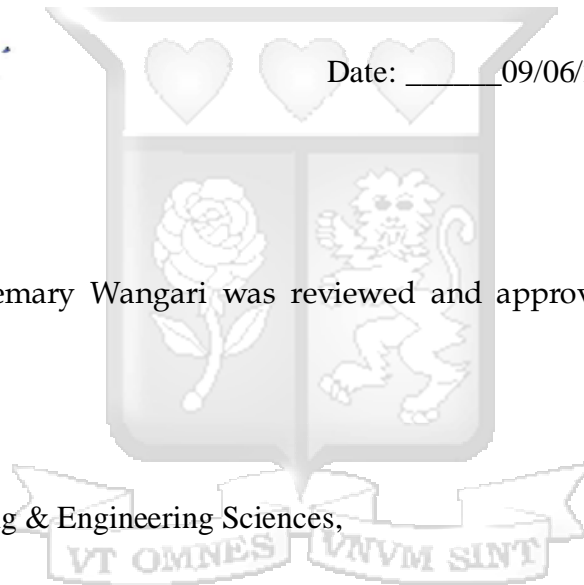
Lecturer, School of Computing & Engineering Sciences,
Strathmore University

Dr. Julius Butime,

Dean, School of Computing & Engineering Sciences,
Strathmore University

Dr. Bernard Shibwabo,

Director of Graduate Studies,
Strathmore University



Abstract

Sound is an essential component of existence in all aspects of life. It is a crucial component when it comes to creating automated systems for various domains such as personal safety and essential surveillance. Hearing people always absorb information through sound and language that is spoken around them. On the other hand, deaf and hard of hearing people lack the luxury of hearing and may end up having major problems due to lack of this awareness. Various researches have shown that there is a mismatch between the need and the demand of the assistive technologies as you find that the need is high but the demand and supply is low which impose a challenge in enhancing access of the assistive devices. Also, there is a gap between the number of people who require assistive technologies to meet their needs and the number of people who are willing and able to purchase and use these technologies. This mismatch could be due to factors such as the cost of the technologies, lack of awareness or knowledge about the technologies, or cultural barriers to their use. Only a small percentage of people have access to the assistive devices. This study reviewed the existing assistive technologies for the deaf and hard of hearing. Prior studies on assistive technologies for the deaf revealed that, sound classification systems have been developed world wide, but none have been implemented for use in Kenya. The research employed a machine learning approach, specifically utilizing convolutional neural networks, to design a sound classification model. The process involved transforming detected sound events into spectrogram images, which were then processed by the Convolutional Neural Network to extract relevant features. The extracted features were subsequently employed to classify environmental sounds, including car horns, dog barking among others. Once the sounds have been classified, a mobile app was used to display a notification indicating the type of sound that has been detected. The machine learning model was evaluated for its effectiveness in assisting the deaf and hard-of-hearing individuals, with the ability to accurately classify a wide range of urban sounds relevant to the study and display corresponding notifications on the user interface. The development of this model stems from a strong motivation to empower deaf individuals, enabling them to experience greater independence without relying on others, with an aim to bridge the gap between auditory awareness and the needs of the deaf and hard-of-hearing community.

Keywords: *Deaf and Hard of Hearing, Convolutional Neural Networks, Sound Classification, Spectrogram.*

Table of Contents

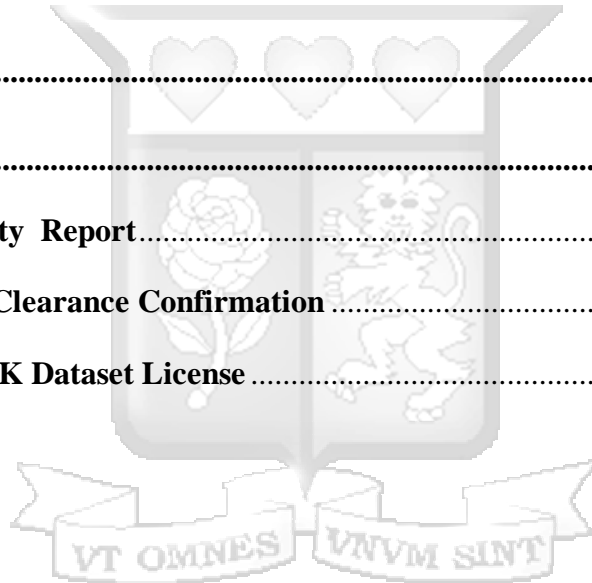
Declaration and Approval	ii
Abstract	iii
List of Figures	ix
List of Tables	xi
Abbreviations/ Acronyms	xii
Definition of Terms	xiii
Acknowledgements	xiv
Dedication	xv
Chapter 1: Introduction	1
1.1. Background of the study	1
1.2. Problem Statement	2
1.3. Research Objectives	3
1.3.1. General Objective	3
1.5. Justification	3
1.6. Scope	4
1.7. Limitations	4
Chapter 2: Literature Review	5
2.1. Introduction	5
2.2. Challenges Facing the Deaf and Hard-of-Hearing	5
2.2.1. Overall understanding and experience of the world for deaf individuals	5
2.2.2. Implications for safety and awareness of potential dangers in the environment	6

2.3. Empirical Literature	6
2.3.1 Assistive Technologies Globally	6
2.3.2 Assistive Technologies in Kenya	9
2.4.1. Computational Theory	10
2.4.2. Cognitive Theory	11
2.4.3. Psychoacoustics Theory.....	14
2.5. Models	15
2.5.1. Hidden Markov Model (HMM)	15
2.5.2. Gaussian Mixture Model (GMM).....	16
2.6. Frameworks	17
2.6.1. Tensor Flow	17
2.6.2. Keras	17
2.6.3. Caffe.....	18
2.6.4. Deeplearning4j.....	18
2.7. Architectural Designs	19
2.7.1. Assistive Technology Architecture.....	19
2.7.2. Smart 311 Architecture.....	20
2.8. Algorithms	20
2.8.1. Decision Trees	20
2.8.2. K- Nearest Neighbor (KNN).....	21
2.8.3. Naïve Bayes	21
2.8.4. Bayesian Network.....	21
2.9. Research Gaps	22
2.10. Conceptual Framework	22
Chapter 3: Research Methodology	23
3.1. Introduction	23
3.2. Variables and Research Design	23
3.2.1 Variables.....	23

3.2.2	Research Design	23
3.3.	Population and Sampling	24
3.3.1.	Target Population.....	24
3.3.2.	Sampling	24
3.4.	Data Collection Methods and Analysis	25
3.4.1.	Data Collection Methods	25
3.4.2.	Data Analysis.....	26
3.5.	Research Quality and Reliability	26
3.5.1.	Research Quality.....	26
3.5.2.	Reliability.....	26
3.6.	System Development Methodology	27
3.7.	Utilization and Dissemination of Research Results	27
3.8.	Ethical Considerations/Issues	28
Chapter 4:	System Design and Architecture	29
4.1.	Introduction	29
4.2.1.	Functional Requirements	29
4.2.2.	Non-functional Requirements.....	30
4.3.	System Architecture	30
4.4.	System Design	31
4.4.1.	Use case Model.....	32
4.4.1.1.	Use case diagram and descriptions	32
4.4.2.	System Sequence Diagram	36
4.4.3.	Entity Relation Diagram	36
4.4.4.	Class Diagram.....	38
4.5.	WireFrames	38
4.5.1.	User Login	38

4.5.2.	Record and Predict Sound.....	39
4.5.3.	Sound Classification Results.....	40
Chapter 5: Model Implementation and Testing.....		41
5.1.	Introduction	41
5.2.	Development Environment and Language	41
5.2.1.	Software Requirements and Hardware Requirements	41
5.3.	Model Components.	42
5.3.1.	Input Layer.....	43
5.3.2.	Hidden Layer	43
5.3.3.	Output Layer	43
5.4.	Model Development.....	43
5.4.1.	Sound Data Collection.....	43
5.4.2.	Import of necessary libraries.....	44
5.4.3.	Audio Extraction.....	45
5.4.4.	Filtering the Metadata file and the audio files	48
5.4.5.	Preprocessing Audio Files	50
5.5.	Model Training.....	51
5.5.1.	Training the model from scratch.....	52
5.6.	Android Mobile Application Development.....	55
5.6.1.	Authentication.....	55
5.6.2.	Main Activity	56
5.6.3.	HomeViewModel Class.....	57
5.6.4.	Recording View Model.....	58
5.7.	Model Testing	58
Chapter 6: Discussion of Results		60
6.1.	Introduction	60
6.2.	Results of the study	60

6.3. System Validation	61
6.4. System Evaluation	61
6.5. Accomplishment of the objectives	61
6.6. Research Limitations	62
Chapter 7: Conclusion, Recommendations, and Future Works	64
7.1. Conclusion	64
7.2. Recommendations	64
7.3. Future Works	65
References	67
Appendices	74
Appendix A: Similarity Report	74
Appendix B: Ethical Clearance Confirmation	75
Appendix C: Urban 8K Dataset License	76



List of Figures

Figure 2.1: Sound Event Detection Processing(Miyazaki et al., 2019).....	11
Figure 2.2: Audio Feature Extraction (Hershey et al., 2017).....	13
Figure 2.3:The Structure of Audio Classification System (Jasim et al., 2022.....	14
Figure 2.4: Graphical Representation of a Gaussian Matrix Model (Carrasco 2020)	17
Figure 2.5: Assistive Technology Architecture(Mielke et al., 2013)	19
Figure 2.6: Smart 311 Noise Sound Classification Architecture (Tariq et al., 2018).....	20
Figure 2.7: Conceptual Framework	22
Figure 3.1: Agile Development Cycle (Concas et al., 2008).....	27
Figure 4.1: System Architecture	31
Figure 4.2: Use Case Diagram.....	32
Figure 4.3: System Sequence Diagram.....	36
Figure 4.4: Entity Relationship Diagram.....	37
Figure 4.5: Class Diagram	38
Figure 4.6: User Login Page	39
Figure 4.7: Record and Predict Sound Wireframe.....	39
Figure 4.8: Sound Classification.....	40
Figure 5.1: Importing Libraries.....	45
Figure 5.2: Spectrograms Transform.....	46
Figure 5.3: Mel Spectrograms.....	47
Figure 5.4: Mel-Frequency Cepstral Coefficients (MFCC).....	48
Figure 5.5: Filtering the Metadata	49
Figure 5.6: Dataset Samples	49
Figure 5.7: Filtering Audio Files	50
Figure 5.8: Training and Testing Dataset	51
Figure 5.9: PyTorch DataLoader	52
Figure 5.10: Building convolutional and linear neural network layers	53
Figure 5.11: Urban8KNet Model.....	53
Figure 5.12: Augmentation Class	54
Figure 5.13: Pretrained Model	55

Figure 5.14: Authentication Class..... 56
Figure 5.15: Main Activity Class..... 57
Figure 5.16: HomeViewModel Class 57
Figure 5.17: Recoding View Model Class..... 58



List of Tables

Table 4. 1 Use case description of Record Sound.....	33
Table 4.2: Use case description for sound preprocessing.....	34
Table 4.3: Use case description of Classify Sound	35
Table 4.4: Use case description of Display Sound Classification(predicted) Results.....	35
Table 5.1: Software and Hardware Requirements.....	41
Table 5.2: Test Case Results.....	59



Abbreviations/ Acronyms

AT	-	Assistive Technology
ATD	-	Assistive Technology Device
ALDs	-	Assistive Listening Devices
AAC	-	Augmentative and alternative communication devices
CNN	-	Convolutional Neural Networks
DHH	-	Deaf and Hard-of-Hearing
ESR	-	Environmental Sound Recognition
FM	-	Frequency Modulation
HLAA	-	Hearing Loss Association of America
KNN	-	K-Nearest Neighbor
RNN	-	Recurrent Neural Networks
SED	-	Sound Event Detection
SOM	-	Self-Organized Maps
T. Coil	-	Telecoil
WHO	-	World Health Organization



Definition of Terms

Assistive Technology Device	It is any device, tool, software, or system that helps to enhance, preserve, or improve the functional abilities of individuals with hearing disabilities.
Cochlear implants	A small, advanced electronic device that aids individuals who are either completely deaf or have severe hearing loss in perceiving sound.
Deafness	This happens when someone has trouble understanding speech even when sound is enhanced.
Environmental Sound Recognition	Processing of environmental sounds(ES) such as alarms, recognize (R) when a device is not functioning correctly, locate an event in space, monitor a change in status, communicate an emotional or physical condition.
Hard of hearing/hearing loss	This results in a diminished capacity to hear noises in a way that other individuals can.
Hearing aids	An electronic device that is small enough to be worn in or behind the ear and helps individuals with hearing loss to participate more fully in daily activities and conversations by amplifying sounds. This can improve their hearing ability in both quiet and noisy surroundings.
Profound Hearing Loss	This describes complete deafness. A profoundly deaf person is utterly unable to hear
Sound Event Detection	The process of identifying sound events in a recording and assigning them temporal start and end time
Telecoil	A tiny copper wire that is discretely coiled inside hearing aids and can detect electromagnetic signals from various sources and can be readily activated by pressing a button.

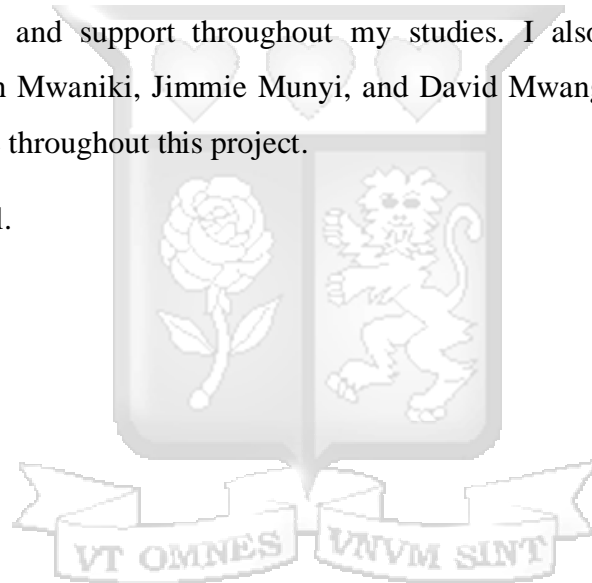
Acknowledgements

This thesis would not have been possible without the support of many people. First and foremost, I would like to express my gratitude to God for His goodness and for giving me the strength to undertake this research.

I extend my utmost sincere gratitude to my supervisor, Dr. Victor Rop, for allowing me to undertake this work and for his continuous guidance and invaluable suggestions throughout the research process. I would also like to offer special thanks to Professor Ismail Ateya for his guidance and insightful contributions.

I am sincerely grateful to all my family members and friends for their unwavering support and love. In particular, I am immensely thankful to my Mum and Dad for their unconditional love, encouragement, and support throughout my studies. I also extend my heartfelt appreciation to Joseph Mwaniki, Jimmie Munyi, and David Mwangi for their continuous support and assistance throughout this project.

May God bless you all.



Dedication

I dedicate this remarkable accomplishment to my parents, who have consistently served as the pillars of strength and support in my life. Their love has been the guiding force that has shaped my path. I am eternally grateful for the sacrifices, support, and unwavering confidence I have received, which have made this achievement possible.

I also dedicate this thesis to my brother, Joseph Mwaniki Ngatia, for his unwavering support throughout this journey, as well as to all my dear deaf friends who have been a great source of inspiration.



Chapter 1: Introduction

1.1. Background of the study

According to the most recent United Nations report, the world population as of October 2022 is 7.98 billion (Worldometer, 2022). The World Health Organization (2021) states that more than 1.5 billion people worldwide live with hearing loss and that by 2050, an estimated 2.5 billion individuals may have some degree of hearing loss, with at least 700 million requiring hearing rehabilitation. The World Health Organization (2021), also reports that over 1 billion young adults are at risk of permanent, preventable hearing loss due to dangerous listening habits. According to Felman (2018), deafness refers to the condition where individuals are unable to comprehend speech through hearing, even when sound is amplified. This condition is characterized by severe hearing loss, where individuals can either hear very little or nothing at all, and it results in significant hearing loss. Deaf individuals are unable to hear anything or only very little, and they often communicate through sign language (World Health Organization, 2021).

Hearing loss is categorized as disabling if it exceeds 40 decibels in an adult's better ear and 30 decibels in a child's better ear. People with mild to severe hearing loss, referred to as hard of hearing, usually communicate through speech and can benefit from devices such as hearing aids and cochlear implants as well as assistive technology like captioning, m-health, and loop system (Garg et al., 2021; World Health Organization, 2021).

There are various ways of being deaf, such as being born with it (congenital hearing loss) or developing it later in life (acquired hearing loss). Better Health Channel (2017) states that Noise is the most common cause of acquired hearing loss. Other causes of acquired hearing loss include accidents, genetic defects, life-altering experiences, and aging. Recently, the COVID-19 pandemic has further exacerbated the difficulties faced by deaf and hard-of-hearing individuals as they struggled to adjust in a world designed for the hearing. This led to a lack of inclusiveness and affected their mental, physical, and social well-being (Garg et al., 2021).

The inability to hear sounds around you cause social, emotional, and behavioral issues. According to numerous studies, there is a mismatch between the demand and the need for assistive technologies, with the supply falling short of the latter. Since so few people have access to assistive technologies, it presents a problem for enhancing access to these gadgets. In

situations where non-auditory cues are not available, providing information about sounds can be beneficial for individuals who are deaf or hard-of-hearing (Bragg et al., 2016).

This project proposes a machine learning tool based on convolutional neural networks. The model was trained to detect sound, extract sound attributes, and classify the sound in order to assist the deaf in distinguishing between various environmental sound types. A mobile app is used to display pop-up notification showing the type of sound that is been identified.

1.2. Problem Statement

Sounds convey information about the world around us. This means that when a deaf person is oblivious of the sounds around them, something terrible may happen to them that could have been prevented. When non-auditory cues are not present, it is crucial to alert deaf and hard-of-hearing individuals about sounds. Mielke and Bruck (2016), stated that there are commercially available devices for accessing environmental sounds, but they are primarily designed for indoor environments like homes or workplaces, and only support specific events like doorbells or telephones. Communicating in the dark or dimly lit places is a huge problem for people with hearing difficulties (Kumar, 2019). This inability to detect any sounds in the surroundings leads to social, emotional and behavioral problems.

In Kenya, the available assistive technologies are currently limited to assistive listening devices like hearing aids. However, these devices lack the important feature of a phone's display, which plays a crucial role in promoting inclusivity for individuals who are deaf or hard of hearing. Jain et al. (2015) state that while hearing aids and cochlear implants can improve a person's ability to recognize sounds, they typically do not enhance their capacity to determine the specific type of sound. This limitation can negatively impact their ability to utilize visual cues to understand the auditory information they receive.

In addition, cultural factors play a significant role in shaping the perception and classification of sounds, even within the same geographical area like urban and rural environments. Urban areas tend to prioritize sounds related to transportation and industrial activities, while rural areas place emphasis on sounds associated with nature and wildlife. These contextual and environmental variations greatly impact how sounds are perceived and categorized.

1.3. Research Objectives

1.3.1. General Objective

The main objective of this study is to create a system for classifying sounds that could aid individuals who are deaf and hard of hearing in distinguishing between different types of sounds. The system would function by presenting a pop-up notification on a mobile application that displays the detected sound type, accompanied by a vibration option to provide tactile feedback.

1.3.2. Specific Objectives

- i. To investigate challenges facing the deaf in identifying sounds.
- ii. To review existing techniques and tools on assistive technologies for the deaf and hard-hearing.
- iii. To design and develop a mobile application to classify different sounds and display pop up notification to the deaf and hard of hearing users.
- iv. To validate the developed model.

1.4. Research Questions

- i. What are the challenges faced by the deaf and hard of hearing in identifying sounds?
- ii. What are the existing techniques and tools on assistive technologies?
- iii. How can we design and develop the mobile application to display notifications?
- iv. How will the developed model be validated?

1.5. Justification

In order to participate in non-face-to-face communications, assistive listening devices (ALDs) help magnify the sound that a deaf person would like to hear. These ALDs can be used to assist the deaf have audio environmental awareness and can be used in conjunction with a hearing aid or cochlear implant. Even though the deaf community has benefited from these ALDs greatly over the past two decades, there are still certain gaps in the system.

One of the most important discoveries is text telephony, which enables deaf people to communicate with others via text messages. The majority of the equipment created specifically for the deaf is a communication tool that enables interaction with hearing persons. One significant issue does exist, especially when dealing with the environment on a regular basis.

Think of a situation when something audible occurs in a public setting, yet only hearing persons can truly understand what is happening. They won't be able to understand what is happening unless you translate for the deaf individual. A deaf person may be hit by a car if they are walking and cannot hear the honking of a car coming up behind them at high speed.

This project therefore, had the advantage of attempting to raise awareness of such situations. A system that can differentiate between various sounds was developed where it initially concentrated on particular categories of sounds. The outcome was reduced risk levels to the deaf as a result of their increased awareness of their surroundings and ability to foresee potential risks. The benefit was that, the majority of the difficulties that deaf individuals encounter when engaging with their surroundings are connected. As a result, the study ultimately benefited people of all ages. The most important aspect of the research was that it concentrated on raising awareness for deaf people when interacting with their surroundings.

1.6. Scope

The system classified sound and sent a notification, which only assisted deaf people in participating in various events where a pop-up notification indicating the type of sound captured was displayed on a mobile application. It did not, however, classify all of the sounds. The study concentrated on a small number of distinct sounds in order to categorize and predict each sound's category. The focus was on a public setting, as there was currently no assistive technology to assist the deaf while they were in public. The model was only trained using 5 categories of sound, namely siren, street music, children playing, dog barking, and car horn. However, the system provided space for future addition of other sorts of sound.

1.7. Limitations

The focus of this research was only on a few sounds, and not every sound was considered. The system was to only assist in sound classification to assist the deaf and not speech interpretation.

Chapter 2: Literature Review

2.1. Introduction

The use of non-technical solutions, commercial goods, and research endeavors are examples of sound awareness strategies. The variety of sound awareness techniques emphasizes how important sound awareness is. However, there hasn't been much research done on how well a trainable sound detector works for individuals who are hard of hearing or deaf (Bragg et al., 2016).

2.2. Challenges Facing the Deaf and Hard-of-Hearing

Deaf individuals experience a lack of auditory input, which significantly impacts their perception and interaction with the world. This absence of sound affects various aspects of daily life, including communication, education, safety, and the understanding of social interactions. Without the ability to perceive sound, deaf individuals face challenges in understanding spoken language, hearing warning signals, and grasping the nuances and emotional cues that accompany sound. These limitations can make it difficult for them to understand and interpret sounds in their environment.

2.2.1. Overall understanding and experience of the world for deaf individuals

Deaf individuals face a reduced access to environmental sounds, which has significant implications for their overall understanding and experience of the world. Ambient sounds, including traffic, nature, and other background noises, play a crucial role in providing context and information about the environment. For example, the sound of car horns can alert individuals to potential dangers on the road, and the chirping of birds can indicate the presence of wildlife in a serene park. Without the ability to perceive these sounds, deaf individuals may miss out on important cues and information that can enhance their understanding of their surroundings. This can lead to potential safety hazards, as they may not hear emergency sirens or approaching vehicles. Additionally, the absence of environmental sounds can impact their overall sensory experience and appreciation of various environments. The soothing sound of rainfall, the rustling of leaves in the wind, or the crashing of waves at the beach all contribute to the richness of sensory experiences that deaf individuals may not fully access.

2.2.2. Implications for safety and awareness of potential dangers in the environment

Deaf individuals face limited access to auditory cues, which can have significant implications for their safety and awareness of potential dangers in their environment. Sound serves as a crucial source of information and cues, alerting individuals to various situations and events. For example, alarms and sirens provide warnings in emergency situations, doorbells signal the arrival of guests or deliveries, and honking horns indicate potential hazards on the road. Without the ability to perceive these auditory cues, deaf individuals may not be alerted to these important signals, potentially compromising their safety and ability to respond appropriately. This can result in situations where they are unaware of emergencies, miss important notifications, or fail to recognize hazardous situations. The absence of auditory cues can also impact their independence and everyday routines, as they may rely on others to notify them or adapt their living spaces to include visual or tactile alternatives. It highlights the need for alternative communication methods and assistive technologies to ensure that deaf individuals can access and interpret important auditory cues for their safety and well-being.

Jain et al. (2019) discovered that the participants relied on conventional methods to recognize sounds in their homes. They would ask for assistance from others, move around the house to locate the source of any audible sounds they could not recognize, and use dogs as guides. Some participants preferred visual or vibrational alternatives over auditory devices, such as doorbells that flash or vibrate the bed, a vibratory alarm clock, and a wall-mounted light that indicates the ambient sound level. Participants mentioned voices as adequate adaptations for sounds they did not possess, as well as sounds of activity. On the other hand, mechanical sounds, outdoor sounds, and animal sounds were some of the sounds that several participants had no means of coping with.

2.3. Empirical Literature

2.3.1 Assistive Technologies Globally

Bhutkar et al. (2020) proposed a prototype alert device for hard-of-hearing users. When developing the prototype, they collected 9 sound datasets and the home environment for hearing-impaired users was to be considered. The deaf and hard of hearing would benefit greatly from this device's ability to detect both common place sounds and some extremely important non-speech sounds, such as a door closing, a fire alarm, an intruder alert, and movement detection, all of which are required for home safety and security. A few features in the prototype design can help people with mild to severe impairments in their home office settings. It has the

ability to recognize different noises and self-train sounds. The proposed Alert Device was consequently developed largely with the intention of helping hearing-impaired individuals recognize sounds only in the home environment.

Bragg et al. (2016) conducted a web-based survey with 87 deaf and hard-of-hearing people to find out their preferences for sound awareness as well as the noises they believe should be made aware of most urgently. The survey revealed that the most requested sounds included emergency alarms, appliance information, door knocks and doorbells. A prototype of a personalizable mobile sound detector app was created as part of the project, and participants in an alpha test were asked how they felt about the capabilities that were being looked into. They conducted a survey to learn what sounds deaf and hard-of-hearing people value, what methods they already use for sound awareness, and what design specifications they would like for their app. To build a model of those sounds, the application employed training examples of the user's recorded, personally meaningful sounds. Deaf and hard-of-hearing users were able to independently train the app. The incoming audio stream from the phone's microphone is then checked for those sounds. It then vibrates to alert the user when it hears a sound. However, they didn't add any Environmental Sound Recognition features to the app. Through manually transmitted notifications to the user's device, they conducted tests simulating real-time recognition. In their research, the authors used Gaussian Matrix Model based approach which classified only two sounds with limited accuracy, and it was unlikely to present varied use cases, sound and environmental noise in the daily life of DHH users.

Jain et al. (2015) conducted to examine the experiences and perceptions of deaf and hard-of-hearing (DHH) individuals regarding sounds in the home environment, gather their feedback on early domestic sound awareness systems, and identify any potential issues. The research was qualitative in nature and involved 12 DHH participants who shared their thoughts on how they perceive and manage sounds in their homes and provided feedback on early prototypes of sound awareness systems. The results of the study were based on these participants' experiences and insights. In the light of this study, they developed three prototypes for tablet-based sound awareness systems, which they evaluated using a Wizard of Oz methodology with 10 DHH participants. The results of the study indicated a widespread interest in sound awareness systems for smart homes, especially those that provide contextually aware, personalized, and easily digestible visual representations. However, significant concerns were raised regarding privacy,

activity tracking, mental workload, and trust during the testing process.

Sicong et al. (2017) created a mobile app prototype that implements sound recognition through the use of deep learning models. A dataset with nine sound classes was used to validate high sound recognition accuracy. The proposed system boasts efficient performance in terms of sound recognition speed and battery usage. Although the sound recognition process takes place entirely on the mobile device, the classifier training is performed in the cloud due to the high computational demands of deep neural network training. Additionally, the authors put forth a preliminary solution for handling overlapping sounds through the use of unsupervised Non-negative Matrix Factorization (NMF), however, this solution is only applicable when multiple microphones are available.

Mielke and Bruck (2016) created a prototype for an environmental sound detector that runs on a smartwatch and was tested in a controlled environment. The design of the application was evaluated by deaf and hard-of-hearing users who were asked to use a simulated sound recognition feature. The study found that participants had preferences for the user interface, such as customizable vibrating patterns for sound detection notifications. However, it was difficult for participants who were deaf from birth to understand the concept of a sound, which made it challenging for them to comprehend what frequencies make up a unique sound.

Akbal (2020) proposed a three-stage process for classifying environmental sounds that includes feature generation, selection, and classification. The study used various techniques for feature extraction, including one-dimensional native binary models, one-dimensional quarterly models, and statistical characterization production methods. The main objective of the study was to introduce a new Environmental Sound Classification (ESC) approach based on highly precise static feature extraction. The ESC method utilized Environmental Component Exploration to select distinguishing features and employed a cubic support vector machine for classification. The results of this research showed an intellectually novel, highly accurate, and light weight ESC technique.

Koh et al. (2019) investigated the use of Convolutional Neural Networks (CNN) for sound classification, specifically the classification of bird species based on their sounds. The study utilized the ResNet and Inception model architectures and preprocessed the data using the MEL-scale log-amplitude spectrogram approach. The study results were obtained after several iterations and showed that the validation set accuracy was improved before adding Gaussian

Noise. The authors concluded that CNN is the most accurate method for bird sound classification, though the precision of the findings may be limited by the quality of the sound recordings.

Nanni et al. (2020) created a set of classifiers for animal audio datasets that produced comparable results through using taxonomy and varied parameter settings. They experimented with multiple fine-tuned Convolutional Neural Networks (CNNs) that were trained for various audio classification tasks, and evaluated, compared, and combined six different CNNs. The study also tested a CNN trained from scratch and combined it with an already high-performing CNN. The results showed that multiple correctly tuned CNNs can be linked for efficient and dependable audio classification. Lastly, they improved the ensemble performance of the CNNs by mixing custom textures derived from spectrograms.

2.3.2 Assistive Technologies in Kenya.

In the realm of assistive technologies for the deaf in Kenya, Sign-iO, an innovative wearable technology, emerged as a promising solution. Developed by Kenyan engineer Roy Allela, Sign-iO aimed to bridge the communication gap between sign language users and those unfamiliar with sign language (Otieno, 2020). This wearable technology consisted of a pair of smart gloves that were wirelessly connected to a mobile application via Bluetooth. Through its intricate design, the Sign-iO system captured the intricate gestures of sign language performed by the user. The companion mobile application then utilized this data to convert the sign language gestures into spoken words in real-time. This seamless conversion process facilitated effective communication between individuals fluent in sign language and those who lacked proficiency in it. However, the Sign-iO's functionality primarily focused on capturing sign language gestures and converting them into spoken words, lacking the ability to accurately distinguish and reproduce a wide range of sounds.

According to Femmehub (2022), an assistive technology called "Echonoma" was developed to facilitate communication between the hearing community and individuals with hearing impairment. The innovation aimed to break the communication barrier between these two groups and ensure access and communication within their immediate environment. While "Echonoma" focused on promoting confidentiality and inclusivity, it lacked an option to assist the deaf in distinguishing between various sounds, which could have affected their overall auditory

experience.

The developed sound classification and display tool, incorporating machine learning algorithms and customizable user interfaces, will significantly improve the accuracy of sound classification, enhance user satisfaction, and demonstrate superior usability compared to existing assistive technologies for individuals with hearing impairments.

2.4.Theoretical Framework

2.4.1. Computational Theory

Computational theory deals with the design and analysis of algorithms and systems that perform specific computational tasks, and is concerned with the development of algorithms and systems that can automatically detect and categorize sounds. The theory provides the framework and techniques for designing and implementing sound event detection systems. These systems can use a combination of signal processing techniques, machine learning algorithms, and knowledge from other related fields, such as psychoacoustics and cognitive theory, to perform sound event detection.

2.4.1.1. Sound Event Detection (SED)

Sound event detection refers to the task of automatically detecting specific sounds, such as speech, music, or environmental sounds, in an audio signal. This involves analyzing the audio signal and recognizing patterns that correspond to specific sound events.

Using SED aims to identify specific sound events and determine their start and end times, not just the label for each sound event (Miyazaki et al., 2019). Figure 2.1 depicts a high-level overview of SED processing. It is commonly assumed that the observed sound signal can contain many sound events, that multiple occurrences of the same sound event are possible, and that multiple sound events frequently overlap. For Sound Event Detection (SED), separating out individual sound events, not just classifying them, is a crucial aspect. A standard approach for SED is to utilize multiple classifiers for supervised learning, utilizing mixed sound signals with time-stamped labels for the separate sound events as training data.

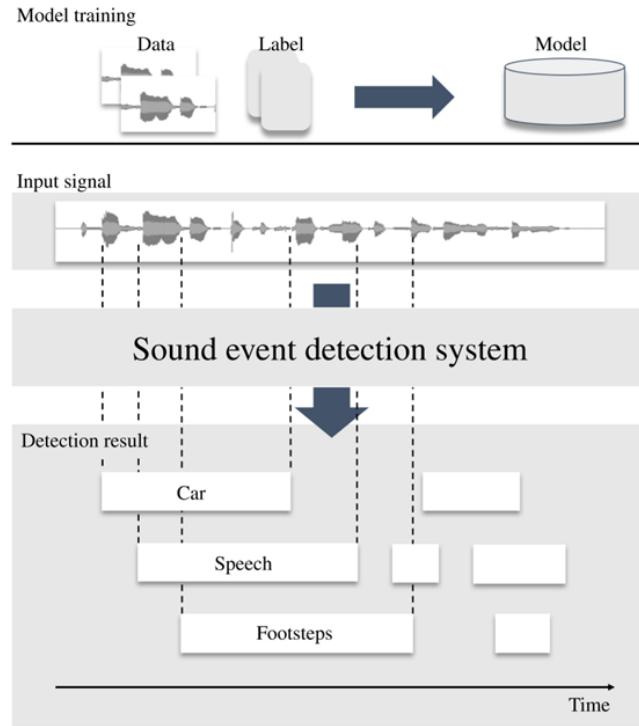


Figure 2.1: Sound Event Detection Processing(Miyazaki et al., 2019).

SVM and random forests are two simple classifier systems that have been proposed (Phan et al., 2015). A system for detecting target sound events based on an exemplar-based approach and NMF has also been proposed by (Bisot et al., 2016; Komatsu et al., 2017).

2.4.2. Cognitive Theory

Veenstra (2010) states that, cognitive theory can be used to develop a system that mimics the way the human auditory system processes sound. The system can be designed to recognize and categorize sounds based on their properties, such as pitch, frequency, and duration, much like how humans process auditory information. By understanding the cognitive mechanisms involved in sound perception, a sound classification system can be optimized to accurately identify and categorize different types of sounds. The theory is related to temporal frequency attention in that it provides a framework for understanding how humans allocate their attention to different aspects of incoming sensory information. It helps to explain why and how humans can selectively attend to specific temporal and frequency aspects of sounds.

Temporal-frequency attention is rooted in the idea that the perception of sound is not only based on its amplitude or loudness but also on its frequency content and the way they change over time (dsa2gamba & abbottds, n.d.).

In sound classification, the goal is to automatically categorize sounds into predefined categories

based on their acoustic features. The traditional approach for sound classification is to use hand-engineered features such as Mel-Frequency Cepstral Coefficients (MFCCs) that capture the spectral characteristics of the sound. However, this approach can be limited as it does not capture the temporal dynamics of the sound, which can be crucial in differentiating between different sound categories.

Temporal-frequency attention addresses this limitation by allowing a model to learn to attend to different parts of an audio signal based on both its temporal and spectral characteristics. The steps involved in implementing temporal-frequency attention in sound classification are:

2.4.2.1. Pre-processing

The audio signal is first transformed into a spectrogram representation that captures both the temporal and spectral information.

2.4.2.2. Audio Feature Extraction

In a research conducted by Jasim et al. (2022), they employed different techniques of audio feature extraction in order for them to classify sound. The Features represented values that can be expressed numerically and quantified using the appropriate methodologies. A sound wave, for example, is made up of two components: sample rate and sample data. The sampling rate and sample data can now be transformed in a variety of ways in order to extract important valuable features from them (Zhang, 2021). The accuracy of the system is determined by its characteristics and classification techniques. Extraction of effective features is a critical step in developing a reliable classification system's front-end module. The sound signal of one class, on the other hand, may change over time, and this change may occur on any of the sound variables, such as amplitude or frequency. Each type of sound has distinguishing characteristics that set it apart from the others (Jasim et al., 2022). There are different methods for extracting features from sound files. Some concentrate on extracting features from the frequency space, while others concentrate on the time space.

2.4.2.2.1. The Zero Crossing Rat (ZCR)

The Zero Crossing Rate (ZCR) is a measure of how rapidly a signal alternates from positive to negative or vice versa. This feature is commonly utilized in speech recognition and music processing systems. It is particularly effective in detecting percussive sounds, such as those produced by minerals and rocks, where the ZCR has a high value (Giannakopoulos & Pikrakis, 2014).

2.4.2.2.2. Linear Predictive Coding (LPC)

In audio and speech processing, the LPC (Linear Predictive Coding) is a method used to describe the spectral envelope of a speech signal in a compressed form through a linear predictive model (Dave, 2013).

2.4.2.2.3. Perceptual Linear Prediction (PLP)

PLP extracts features from audio data, which are then used to describe it. The definition of PLP involves an estimation of three phenomena related to perceptrons: critical band resolution curves, equal loudness curves, and the power law relationship between intensity and loudness (Hershey et al., 2017). The LPC and PLP are frequently used in feature extraction algorithms in the disciplines of voice recognition and speaker verification (Grama & Rusu, 2017).

Figure 2.2: depicts an audio wave file representing a sound event that has been transformed into a spectrogram image and is being processed by a CNN. The image features are used to classify various environmental sounds and occurrences such as car horn, dog barking, drill etc.

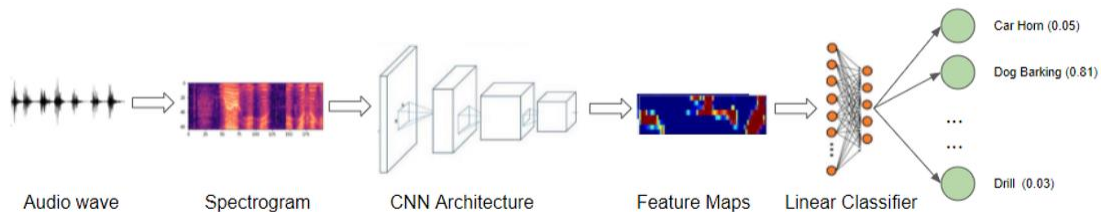


Figure 2.2: Audio Feature Extraction (Hershey et al., 2017)

2.4.2.3. Attention mechanism

An attention mechanism is applied to the extracted features to weight different parts of the signal based on their importance for the classification task. The attention mechanism can be implemented as a separate layer in the model or as part of the fully connected layer.

2.4.2.4. Audio Content Classification

Convolutional neural networks (or "CNNs") are capable of producing cutting-edge results in image and sound classification (Jasim et al., 2022). Malfante et al. (2018) proposed a system that employs deep learning networks to classify environmental sounds based on their spectrograms, where CNN was used in both the feature extraction and classification stages. The researchers utilized spectrogram images of environmental sounds to train both the tensor deep stacking network (TDSN) and the convolutional neural network (CNN). Based on their

experimental investigation, they determined that their proposed system of utilizing spectrogram sound images for sound classification can serve as a foundation for developing sound recognition and classification systems.

Figure 2.3 shows the structure of audio classification system from the moment the sound signal is entered, and at the step of features extraction, distinctive features are extracted from it and then provided to the classification model. CNN is used as the classification model due to its effectiveness in separating different classes.

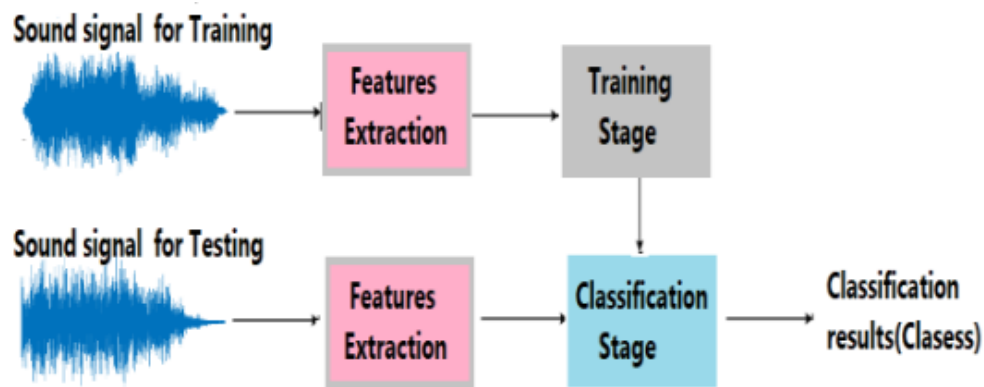


Figure 2.3: The Structure of Audio Classification System (Jasim et al., 2022)

Mu et al. (2021) states that, the importance of temporal-frequency attention in sound classification lies in its ability to improve the performance of the model by allowing it to focus on the most relevant parts of the audio signal. This is particularly important in cases where the audio signal is cluttered with background noise or where there is significant variability in the temporal and spectral characteristics of the sound within the same category. By allowing the model to attend to different parts of the signal based on both its temporal and spectral characteristics, temporal-frequency attention can significantly improve the performance of sound classification systems.

2.4.3. Psychoacoustics Theory

Psychoacoustics is the scientific study of human perception of sound which provides a theoretical framework and practical insights into the way in which the human auditory system processes sound, which can be used to guide the design of sound classification systems (Psychoacoustics | ScienceDirect Topics). University of Salford, states that, the goal of psychoacoustics in sound classification is to understand the properties of sounds that are

relevant for human perception and categorization, and to use this knowledge to design algorithms that can accurately mimic human perception. This is achieved by studying the physiological and neural responses to sounds, as well as the psychological processes involved in sound perception and categorization.

2.5. Models

2.5.1. Hidden Markov Model (HMM)

The HMM is a statistical model utilized in machine learning that explains the relationship between the evolution of observable occurrences and underlying, indirectly observable factors. Instead of determining the step-by-step conditions of a random process, it models the probabilistic characteristics of the process using probability distributions. HMM can be utilized to categorize audio samples into speech, music, or environmental sound.

Hidden Markov Models (HMMs) are probabilistic models used in a variety of applications, including speech recognition, speech synthesis, and sound classification.

In the context of sound classification, HMM can be used to model the probability distribution of different sounds or audio classes. It consists of two components: (1) a set of hidden states that represent the underlying sound class, and (2) a set of observation symbols that represent the audio features extracted from a sound clip. The HMM also defines the transition probabilities between hidden states and the observation probabilities given a hidden state.

During the training phase, the parameters of the HMM are estimated based on a large labeled dataset of sound clips, where each sound clip is associated with a sound class. In the testing stage, a HMM is fed a new audio clip and utilizes it to identify the sound class with the highest likelihood based on the observed symbols. This is accomplished using the Viterbi algorithm, which computes the maximum likelihood path through the hidden states and observation symbols.

HMMs are important for sound classification because they are able to model the temporal dependencies between audio features, which is crucial for capturing the dynamics of different sounds. In addition, HMMs are flexible and can be used to model a wide variety of audio classes, including speech, music, and environmental sounds. Additionally, the hidden states in an HMM can be used to model different levels of abstraction, such as phonemes, words, and sentences in speech recognition, making it a powerful tool for many different sound classification tasks.

2.5.2. Gaussian Mixture Model (GMM)

Carrasco (2020) describes a Gaussian Mixture as a combination of several Gaussian functions, each identified by k ranging from 1 to K , where K represents the number of clusters in the dataset. Each Gaussian, denoted by K , consists of:

- i. A mean, μ , that determines its center.
- ii. A covariance, Σ , that specifies its width, which would be the equivalent of an ellipsoid's dimensions in a multivariate situation.
- iii. A mixture probability, π , that determines the size of the Gaussian function

The GMM is trained on a labeled dataset of sound clips, where each sound clip is associated with a sound class. During training, the parameters of the GMM, such as the mean, covariance, and mixing coefficients, are estimated for each sound class. Once the GMM is trained, it can be used to determine the most likely sound class for a new sound clip by computing the likelihood of the audio features given each sound class, and selecting the class with the highest likelihood. The importance of GMM in sound classification is that, it is a flexible and powerful model that can capture the underlying distributions of different sound classes. It can handle complex distributions that cannot be modeled by simple Gaussian distributions and is able to model multi-modal distributions, which are common in many sound classes. Additionally, GMMs can be used in conjunction with other models, such as Hidden Markov Models (HMMs), to create more sophisticated sound classification systems.

Figure 2.4: shows a graphical representation of a Gaussian Matrix Model, with three Gaussian functions hence $K=3$. Each Gaussian explains the data in each of the three available clusters.

The curves are plotted on a graph with the x-axis being the data values and the y-axis being the probability density function (pdf) of the Gaussian distribution.

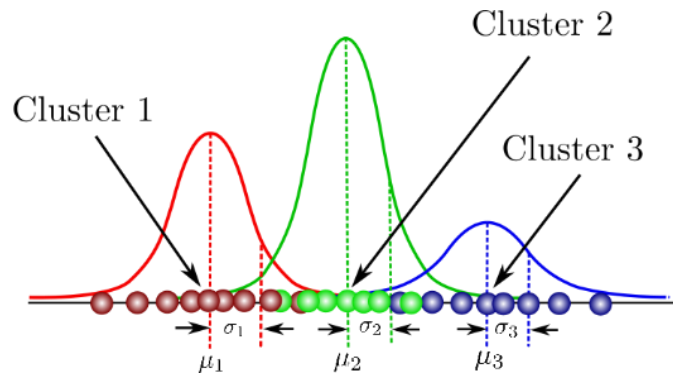


Figure 2.4: Graphical Representation of a Gaussian Matrix Model (Carrasco 2020)

2.6. Frameworks

There are different types of frameworks that are used for deep learning. Some of these frameworks include; Deeplearning4j, Caffe, Theano, PyTorch, Keras, and TensorFlow. However, TensorFlow, Keras, and PyTorch are three of these frameworks that have gained popularity in recent years because to their usability, widespread use in academic research, commercial code, and extensibility.

2.6.1. Tensor Flow

According to Madhavan et al. (2021), Tensor is a term used to describe the multi-dimensional arrays used in mathematical models for neural networks in the context of machine learning. A tensor is often a generalization of a vector or matrix with a higher dimension. The TensorFlow framework can be run on various platforms and operating systems, including CPUs, desktops, and mobile devices, and it can be deployed both locally and in the cloud. It is considered to offer better support for distributed processing, as well as improved flexibility and performance for commercial use. Python is the main programming language used with TensorFlow. Although there are no stability guarantees for other languages, such as C++, Java, and Go, there are third-party bindings available for many languages, including C#, Haskell, Julia, Rust, Ruby, Scala, R, and PHP. For executing TensorFlow applications on Android, Google has developed a mobile-optimized TensorFlow-Lite library.

2.6.2. Keras

According to Madhavan et al. (2021), Keras is a Python deep learning library that distinguishes itself from other deep learning frameworks. Keras serves as a high-level application programming

interface (API) for constructing neural networks and offers a means of enhancing the capabilities of diverse deep learning framework backends that it employs. In version 2.4.0, Keras stopped supporting multiple backends and now only focuses on TensorFlow. Essentially, it is a part of TensorFlow, with the Keras API for TensorFlow being implemented in the `tf.keras` submodule or package.

2.6.3. Caffe

According to Madhavan et al. (2021), Caffe is a deep learning platform that offers support for a diverse range of architectures, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. However, it does not provide compatibility with Restricted Boltzmann Machines (RBMs) or Deep Boltzmann Machines (DBMs). Caffe takes advantage of GPU acceleration using the NVIDIA CUDA Deep Neural Network library and has been utilized for image classification and other visual tasks. To facilitate parallel processing across a group of systems, Caffe supports Open Multi-Processing (OpenMP). In order to optimize performance, Caffe and Caffe2 are coded in C++ and offer deep learning training and implementation options through Python and MATLAB interfaces.

2.6.4. Deeplearning4j

Madhavan et al. (2021) describes Deeplearning4j as a widely recognized deep learning framework that utilizes Java technology. However, it also provides APIs for other programming languages such as Python, Scala, and Clojure.

This framework, which is licensed under Apache, is equipped to handle Restricted Boltzmann Machines (RBMs), Deep Belief Networks (DBNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs). Furthermore, it features distributed parallel variants that are tailored for compatibility with big data processing platforms like Apache Hadoop and Spark.

2.7. Architectural Designs

2.7.1. Assistive Technology Architecture

A smartphone is the key component in the design. With the increasing popularity of smartphones with powerful processors, they have become a vital part of the market. In order to effectively differentiate between various sounds, the classifier needs to be highly adaptable and flexible. To implement real-time pattern recognition algorithms, a processing device with sufficient computing power is required, which is present in a smartphone. Its ability to connect to the internet can be utilized to access an online service containing the training data for the classifiers. If the system fails to detect a sound or the user feels that an event should have been recognized but wasn't, the sound attributes can be uploaded to the database either automatically or manually. This allows other users to train their devices for improved recognition. The content of the training sample and the location where it was recorded can be identified through the use of tags.

An architecture diagram is shown in Figure 2.5. A microphone or microphone array is used to record sound, which is then processed by the smartphone. When a sound of interest is identified, the system alerts the user and provides them with the option to transmit the acoustic imprint to a central server.

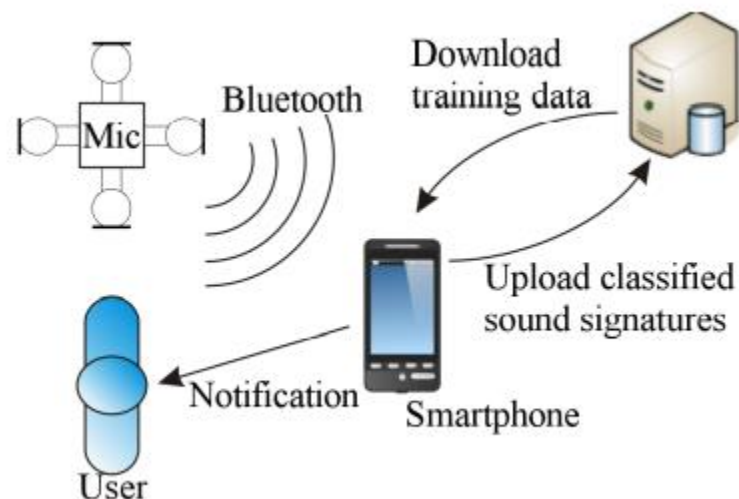


Figure 2.5: Assistive Technology Architecture(Mielke et al., 2013)

2.7.2. Smart 311 Architecture

Tariq et al. (2018) conducted a classification using machine learning algorithms for noise detection, using both shallow learning and deep learning models. Figure 2.6 depicts the architecture of a Smart 311 system. The Smart311 system is capable of operating in smart city environments such as indoors, shopping malls, and even public streets. A mobile application plays a crucial role in the smart city environment by recognizing sounds that are categorized as noise, such as air conditioner noise, gunshots, dog barking, and jackhammer noise. The mobile application can transfer the sounds it detects in a smart city environment to a server via the client. After extracting the features from the input audio data, the machine learning component identifies a particular type of sound. When the sound classification system identifies any of the categories mentioned above, it sends a 311 request to the server based on the severity of the incident.

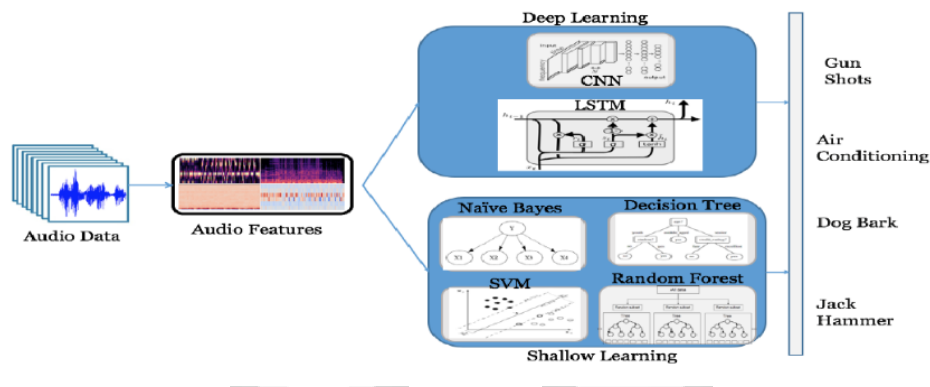


Figure 2.6: Smart 311 Noise Sound Classification Architecture (Tariq et al., 2018)

2.8. Algorithms

2.8.1. Decision Trees

In this type of supervised machine learning, the training data is continually divided based on a particular criterion, and both the input and corresponding output are provided. The tree consists of decision nodes and leaves, which are utilized to describe the connection between inputs and outputs. Decision trees can be used to compare different classifiers. When tested as part of an ensemble of random choice forests, decision trees outperformed them in terms of classification speed but not accuracy. By using a series of decision forest iterations, the ensemble seeks to take care of the method's lower accuracy.

2.8.2. K- Nearest Neighbor (KNN)

KNN algorithms display three traits that set them apart from other learning algorithms which lead to performance advancement over time. They delay in processing their instances until they get information requests. They only save their instances in storage for later usage. KNN combines their training instances and data to respond to information requests, discarding any intermediary results.

For this algorithm, the class produced by this approach will be the class of the instance that is most similar to the tested examples (Harrison, 2019).

2.8.3. Naïve Bayes

This algorithm is designed to solve binary (two-class) and multi-class classification problems. It has proven to be not only simple, but also quick, accurate, and dependable and works particularly well with natural language processing (NLP) problems (Gaurav, 2018). It can be used to categorize an object by independently mapping each characteristic to the classifier. The algorithm determines the membership probabilities for each class, including the probability that a particular record or data belongs to a specific class. The class with the highest probability is considered the most probable one.

In a research study conducted by Fanzeres et al. (2018), it was seen that despite having the lowest accuracy, naive Bayes classification turned out to be a good solution for their mobile sound application for the DHH, with an average of 89%. The processing part that can be improved upon during the training phase has the highest degree of accuracy. In addition, compared to decision trees and neural networks, naive Bayes training was far faster.

2.8.4. Bayesian Network

Probabilistic graphical models known as Bayesian networks utilize Bayesian inference to calculate probabilities. They are represented as directed graphs with edges indicating conditional dependencies, and aim to model the relationships and causality among variable. Through these connections, one can effectively employ factors to draw conclusions about the graph's random variables (Soni, 2019). They are particularly adept at studying a previously occurring event and determining the likelihood that any of the countless known causes contributed to it.

2.9. Research Gaps

According to Mielke & Bruck (2016), their smart watch only focused on an office setting and did not collect sound data from other locations. The smart watch did not display a pop-up notification, making it difficult for deaf users to understand the captured sound. Bhutkar et al. (2020) created a prototype alert device that was solely focused on the home environment. The data used in their study was only existing sound-data, and their prototype lacked any pop-up notification to alert deaf users. Bragg et al. (2016) created a mobile sound detector app to help deaf and hard of hearing people. However, their prototype lacked an environmental sound recognition function, and they had to manually notify deaf users.

2.10. Conceptual Framework

Figure 2.7 shows a conceptual framework of the solution. The mobile phone's microphone was used to detect sound. The obtained data sets were then trained and tested using a machine learning model. After that, the ML categorized the sounds, such as car honking or sirens, and a notification was automatically displayed on the mobile application.

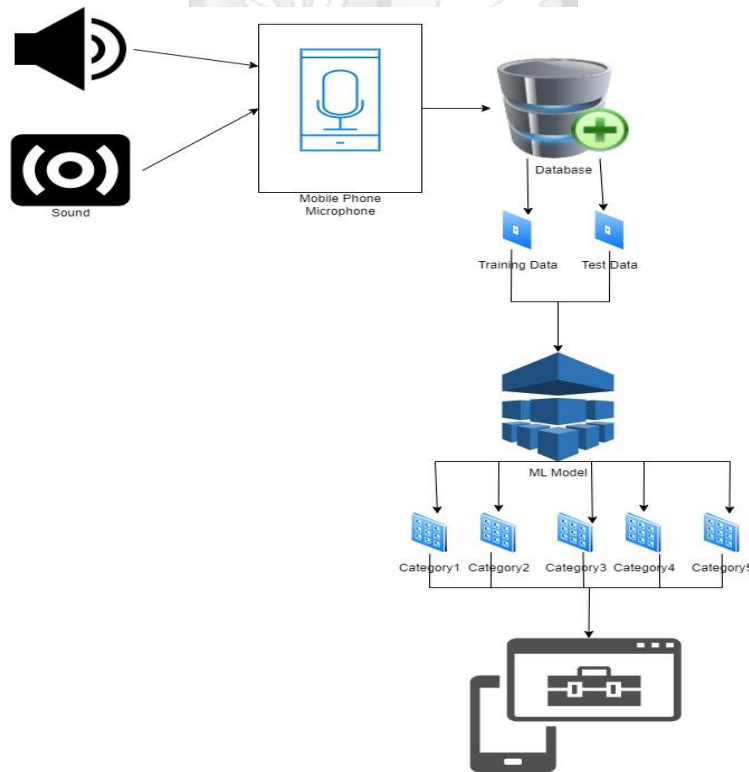


Figure 2.7: Conceptual Framework

Chapter 3: Research Methodology

3.1. Introduction

Methodology and techniques are two closely related and interdependent words that are frequently used interchangeably. Neuman (2014) defines methodology as the large structure that houses methods. Cohen et al. (2000) state that, methodology refers to a methodical approach to data collecting from a particular population in order to comprehend a phenomenon and generalize knowledge obtained from the target population. According to Jansen (2020), research methodology pertains to the practical implementation of a research project. It encompasses the systematic planning of a study by the researcher to ensure reliable and valid results that effectively address the research's aims and objectives. It primarily focuses on what data should be collected, from whom it should be collected (sample design), methods for data collection and analysis.

3.2. Variables and Research Design

3.2.1 Variables

3.2.1.1 Independent Variable

Implementation of the sound classification and display tool

3.2.1.2 Dependent Variables

Measurements that are influenced by the independent variable. These dependent variables are;

- i. **Accuracy.** Measures the tool's ability to accurately classify and categorize different types of sounds.
- ii. **Processing speed:** Quantifies the time taken by the tool to process and classify incoming sounds.
- iii. **Effectiveness of the sound display:** How well the sound classification and display tool present the classified sounds to the user in a clear, understandable, and user-friendly manner.

3.2.2 Research Design

Experimental design involves conducting research in an objective and controlled manner to maximize precision and draw specific conclusions regarding a hypothesis statement. The main goal is to determine the impact of an independent variable on a dependent variable.

The objective of this research was to create a sound classification and display tool by building a

model, and developing a mobile app. An experimental design was employed to determine the study's methodology, data collection, and analysis procedures.

An open dataset, (Urban8K dataset), comprising various sounds was collected and carefully annotated with labels indicating sound type and characteristics. The experimental design encompassed several phases. Firstly, a machine learning-based sound classification model was developed. The collected sound samples underwent preprocessing to extract relevant audio features. These features were then utilized to train a machine learning model, employing a deep learning neural networks technique (Convolutional Neural Network). The trained model underwent rigorous testing and validation to ascertain accuracy and generalization capabilities. Concurrently, a user interface was designed and implemented to visually display the classified sounds, using text-based display. The effectiveness of the sound display was evaluated by testing the mobile app on the trained model with various sounds. The findings provided insights into the potential effectiveness and usability of the tool in real-world scenarios, highlighting its capacity to enhance the auditory experience of individuals with hearing impairments.

3.3. Population and Sampling

3.3.1. Target Population

The study focused on various types of sounds present in the environment. Data on various sound types found in the environment were obtained from the Urban8K dataset, from which five categories were derived. Following that, the model was trained using the five categories to assist the deaf in differentiating them.

3.3.2. Sampling

The study employed both probability and non - probability sampling methods.

3.3.2.1. Cluster Sampling

The researcher had sampled the total sound type-data into groups or clusters that reflected certain categories. Based on parameters such as sound class, clusters were identified and included in a sample. The Urban8K dataset was used, which originally contained 10 classes of environmental sounds, including air-conditioner, car-horn, children-playing, dog-bark, drilling, engine idling, gun-shot, jackhammer, siren, and street music. However, the model was trained using only 5 classes, namely siren, street music, children playing, dog barking, and car horn. To filter the metadata file and audio files, which originally contained 10 classes, the researcher

created a list of the 5 classes and used pandas to filter the main metadata file to a processed one that only had the classes that were of interest. The class ID's were altered so that they were in between 0 and 4, with one for each class. Once the processed metadata file contained data from the five required classes, the researcher did the same to the audio files, so that there were only audio files from the five classes that were of interest to this research.

3.3.2.2. Consecutive Sampling

Using this sampling technique, the researcher selected one sound category from a sample of sound data, examined the data, and moved on to the next sound category. By gathering data with crucial insights, this strategy enabled the researcher to work with different sound types and fine-tune the research.

3.4. Data Collection Methods and Analysis

3.4.1. Data Collection Methods

The sole purpose of research tools is to collect data from research subjects on a specific topic of interest. An ideal instrument, on the other hand, is one that yields objective, accurate, sensitive, efficient, and relevant results. The following tools were used in this study.

3.4.1.1. Existing Sound-Data

The researcher utilized the Urban8k Dataset in the present investigation, which featured ten categories, namely air-conditioner, car-horn, children-playing, dog-bark, drilling, engine idling, gun-shot, jackhammer, siren, and street music. To streamline the analysis, the dataset was filtered to include only five classes relevant to the research questions. Specifically, the metadata file UrbanSound8k.csv was used to provide classification information for each sound file and select audio files that belonged to the chosen five classes. These audio files were then used to train the model, allowing the researcher to accurately classify new audio files based on their sound characteristics.

3.4.1.2. Prototyping

A prototype was developed to facilitate the testing and refinement of ideas that could be conveyed to deaf users more effectively. The mobile application was tested in the test bed environment, which allowed for comprehensive analysis of its functionality and effectiveness. This approach provided valuable insights into the application's strengths and weaknesses,

allowing for necessary adjustments to be made.

3.4.2. Data Analysis

In this study, inferential analysis was employed to analyze the collected data and draw meaningful insights. By applying inferential analysis techniques to the data, insights were gained into the effectiveness and performance of the sound classification and display tool. The analysis focused on evaluating the accuracy of the tool in classifying sounds based on the data from the Urban 8K dataset. One aspect of the analysis involved evaluating the effectiveness of existing models used in sound classification and display. By conducting inferential analysis on the performance and outcomes of these models, insights were gained into their strengths and limitations. This information played a crucial role in shaping the design of the sound classification and display tool to effectively address the research problem.

3.5. Research Quality and Reliability

3.5.1. Research Quality

The chosen research methodology can greatly impact the quality and success of a research (Thattamparambil, 2020). To ensure the collection of relevant data and the use of the most appropriate data analysis method, the researcher carefully selected an appropriate research methodology. Bouchrika (2022) states that, effective research requires reviewing previous studies on the topic and generating new knowledge. By exploring the literature and other materials related to the topic, the researcher was able to gain a better understanding of prior research and how the current study fits into the current field of research. In this study, the researcher reviewed previous related work, discovered the models and framework used in sound classification, and identified gaps in previous research.

3.5.2. Reliability

Reliability is defined as "the accuracy and precision of the measurement, as well as the absence of differences in the results if the research was repeated" (Collins & Hussey, 2014).

To avoid any possible bias in the research findings, the researcher was mindful of their own position throughout the study. The goal was to eliminate or minimize any potential impact that could compromise the reliability of the results. The researcher aimed to prevent confirmation

bias by treating all data impartially, analyzing it sincerely, and resisting the temptation to falsify it.

3.6. System Development Methodology

Object Oriented Analysis and Design (OOAD) is an organized process for doing analysis, developing a system using object-oriented principles, and producing a number of graphical system models during the software development life cycle (Elgabry, 2021). Regardless of limitations like the right technology, the aim of the analysis phase is to build a model of the system. Typically, use cases and conceptual models are used to define the most crucial things in an abstract manner. On the other hand, the analytical model is improved during the design process, which also applies the required technology and other implementation constraints. The Unified Modeling Language (UML) was used to represent the system's various views and functionalities. This approach was used within the agile methodology, which was iterative and incremental and was performed in a highly collaborative manner to produce high quality software, according to (Concas et al., 2008). Figure 3.1 shows the different cycles of an agile methodology.

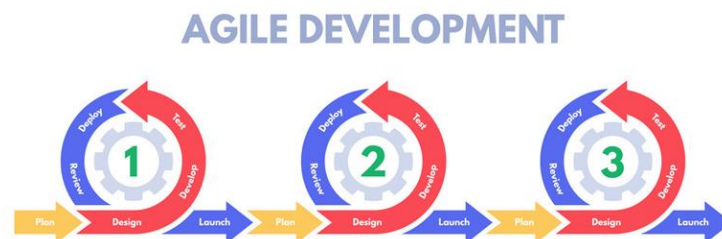


Figure 3.1: Agile Development Cycle (Concas et al., 2008).

3.7. Utilization and Dissemination of Research Results

The results of this study aided individuals who are deaf or have difficulty hearing in identifying different sounds in their surroundings. The findings also helped future researchers who were interested in solving problems that can be addressed using sound classification to help the deaf and hard of hearing. These findings were disseminated through online publications.

3.8. Ethical Considerations/Issues

Some of the ethical considerations that were put in place during this research were;

- i. Institution approval was required to certify the study and the results obtained.
- ii. The research was designed and executed in accordance with the strictest standards of excellence, integrity, moral propriety, and legality.



Chapter 4: System Design and Architecture

4.1. Introduction

System design and architecture are fundamental to software engineering as they define the foundation of a software system. System design involves identifying the components, interfaces, and data that make up a system, while architecture determines how these components and modules interact and are supported by the infrastructure. The success rates of a project are significantly influenced by how well the project requirements are defined, while on the other hand, failure to properly gather and analyze requirements and manage resources can lead to project failure. To mitigate this risk, the use of computer-aided software engineering (CASE) tools has been proposed. Unified Modeling Language (UML) is one such tool that can assist with system design and architecture.

4.2. Requirement analysis

The Urban8K dataset was downloaded from Kaggle and served as the primary source of audio data for the study. To facilitate the sound classification process, the audio data was first extracted from the dataset and converted into mel-spectrogram images. This conversion was necessary as it allowed the data to be easily visualized and analyzed using machine learning algorithms. Additionally, any irrelevant audio files were removed from the dataset to ensure that the resulting model was accurate and reliable. The use of the Urban8K dataset and mel-spectrogram extraction techniques proved to be an effective approach in developing a sound classification tool for assisting the deaf and hard-of-hearing.

Chung & do Prado Leite (2009) highlighted that the functional and non-functional aspects define a system's utility. They pointed out that, like anything else, system quality is essential and ought to be considered when creating high-quality software. The requirements and study goals for this research have been broken down into functional and non-functional requirements as shown below.

4.2.1. Functional Requirements

The functional requirements were developed based on the desired behaviors that would be accomplished in a system. These therefore encompassed the numerous system functions and capacities that were discovered to be compatible with the study's goals, as highlighted below:

- i. The system should accept sound input from a microphone or other audio sources.
- ii. The system should process the audio input to extract relevant features.

- iii. The system should analyze the input sound and classify it based on predefined categories of the environmental sounds.
- iv. The system should display the classification results on a user interface, such as a textual description.
- v. The system should provide a user-friendly interface that is easy to navigate and use.
- vi. The system should incorporate a vibration alert feature to notify individuals who are deaf or hard of hearing of incoming notifications.

4.2.2. Non-functional Requirements

Non-functional requirements are a set of criteria that describe the characteristics or qualities of a software system, rather than its specific functional capabilities. These requirements focus on how the system operates, rather than what it does. They do not relate to functionality, but to attributes such as reliability, efficiency, usability, maintainability, and portability. The following are the non-functional requirements:

- i. **Reliability:** The system should be highly reliable, with accurate sound classification and minimal errors or false positives. Users should be able to depend on the tool to correctly identify the different types of sounds.
- ii. **Performance:** The system should have good performance and speed, with minimal latency in sound classification. The tool should be able to handle multiple sounds simultaneously without slowing down or crashing.
- iii. **Security:** The system should be secure, with appropriate measures in place to protect users' personal data and privacy.
- iv. **Usability:** The system should be easy to use and understand, with a clear and intuitive user interface.
- v. **Compatibility:** The system should be compatible with a wide range of devices and platforms, including different operating systems and screen sizes.
- vi. **Maintainability:** The system should be easy to maintain and update. It should be designed with modularity and scalability in mind, to facilitate future updates and improvements.

4.3. System Architecture

The system architecture is a conceptual framework that encompasses the various views, structure, and behaviors of the system. It provides a description and representation of how the different system components operate and interact with one another. Essentially, the system architecture

captures how the system functions as a whole by coordinating its components and subsystems to achieve its intended purpose.

The components used in the system are users, sound recording module, sound processing module, sound classification module, user interface module, database module and system communication module.

The sound recording module, will be responsible for capturing the sound that needs to be classified and displayed. To improve the accuracy of sound classification, a sound processing module will be included to filter and pre-process the recorded sound by removing noise or enhancing certain frequencies. The sound classification module will then analyze the pre-processed sound and classify it based on its type, including dog barking, car horn and siren.

A user interface module will provide a graphical user interface for the user to interact with the system and view the sound classification results. To store the sound classification results for future reference or analysis, a database module will be included. Lastly, a system communication module will handle the communication between the different modules of the system to ensure that they are integrated effectively and efficiently. By integrating these different modules, the "Sound Classification and Display Tool for Assisting the Deaf and Hard-of-Hearing" system, as shown in Figure 4.1, will be able to provide the required functionality for the deaf and hard-of-hearing users.

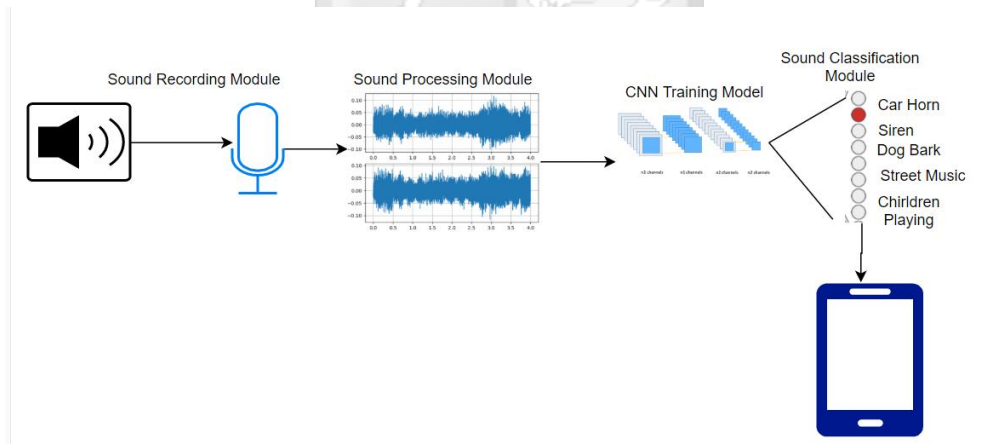


Figure 4.1: System Architecture

4.4. System Design

(Odhiambo, 2019) describes system design as designing the system's components, including the architecture, modules, interfaces, and data flow. The objective of engaging in System Design is to gather and present detailed information about the system and its components, in order to support

the implementation process that aligns with the system architecture models and views. The design process involves utilizing various diagrams such as system sequence diagrams, use case diagrams, partial domain models, context and data flow diagrams, entity diagrams, and class diagrams. These diagrams are utilized at different stages of the design process to depict and document the functionality of the system.

4.4.1. Use case Model

A use case model shows how a system interacts with its users, other systems, or external entities through a set of actions called use cases. It describes the various use cases, their relationships, and the actors involved in each use case. Use case diagrams are well-suited for, illustrating the objectives of interactions between a system and its users, structuring and clarifying the functional requirements of a system, defining the prerequisites and demands of a system, and describing the fundamental sequence of actions in a use case.

4.4.1.1. Use case diagram and descriptions

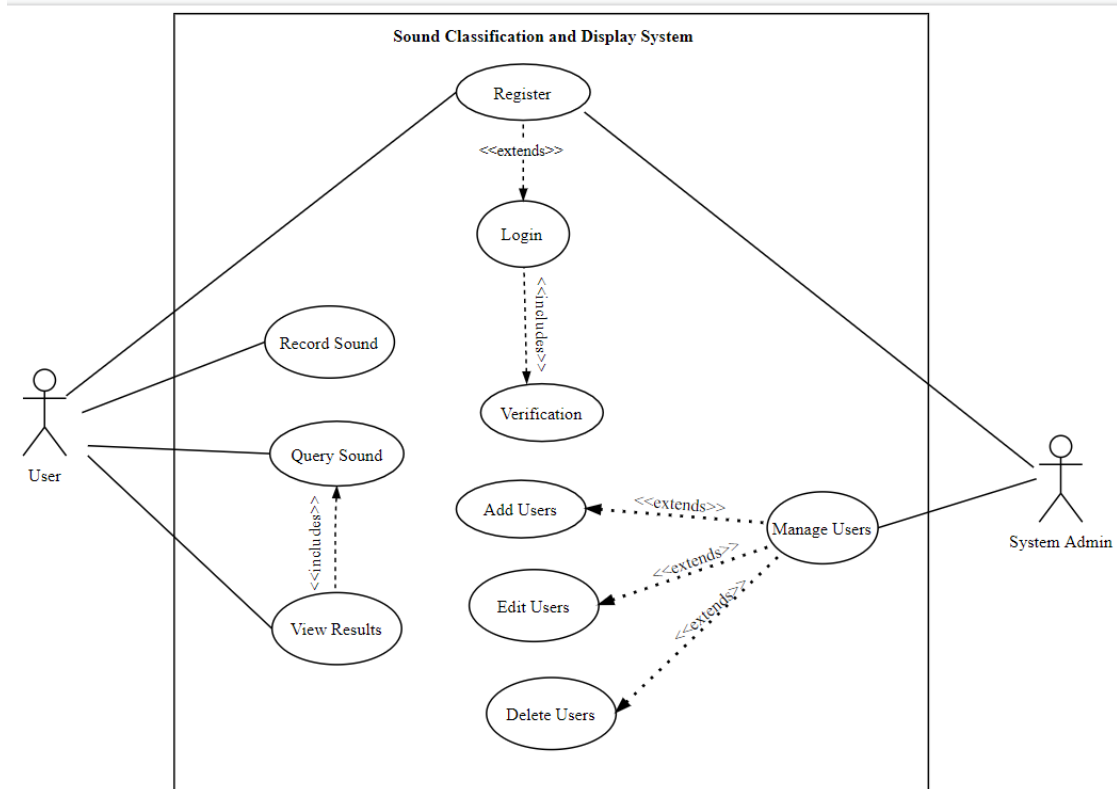


Figure 4.2: Use Case Diagram

In figure 4.2, the Deaf or Hard-of-Hearing User is the main actor in this use case diagram. The user interacts with the system to record sound using a microphone. The system then classifies the sound, using machine learning algorithms (Convolution Neural Network) to identify the type of sound, whether is a dog bark, car horn, siren, street music or children playing. The system then displays the classification results to the user, using a visual interface which is a text message pop up.

Table 4.1 focus on "Record Sound," detailing the steps and requirements for capturing audio input within the system and providing a comprehensive understanding of the recording functionality. Table 4.2, on the other hand, pertains to "Sound Preprocessing," outlining the tasks and processes involved in preparing the recorded sound data for further analysis, including noise reduction and filtering. Table 4.3 introduces the use case description of "Classify Sound," elaborating on the procedure for analyzing and categorizing the preprocessed sound data using classification algorithms or techniques, thereby enabling the identification of different sound classes. Table 4.4 addresses the use case description of "Display Sound Classification (predicted) Results," illustrating how the classified sound data is presented to the user, presenting the predicted results in a clear and user-friendly manner, and ultimately facilitating the interpretation and understanding of the sound classification outcomes.

Table 4. 1 Use case description of Record Sound

Use Case	Record Sound
Description	The system records sound data using the Sound Recording Module (Phone's Microphone) and saves it to the database.
Source	Ambient Environment
Inputs needed	Sound Data
Preconditions	<ol style="list-style-type: none"> 1. The Sound Recording Module is active. 2. The audio input device (Phone's microphone), is functioning correctly. 3. Database is available to store the recorded sound data.
Post Condition	<ol style="list-style-type: none"> 1. Sound data is recorded and saved to the database.
Flow of Events	<ol style="list-style-type: none"> 1. The system's Sound Recording Module begins capturing sound data from the user's audio input device.

	<ol style="list-style-type: none"> 2. The system's Sound Processing Module analyzes the recorded sound data and the system automatically saves the sound data to the system's database. 3. The user may view and manage the recorded sound data using the system's user interface.
--	--

Table 4.2: Use case description for sound preprocessing

Use Case	Sound Pre-Processing
Description	The system automatically processes the recorded sound data to enhance its quality and extract relevant features for classification
Source	System
Inputs needed	Recorded Sound Data
Preconditions	<ol style="list-style-type: none"> 1. Sound data has been recorded and stored in the system. 2. Sound processing module is operational.
Post Condition	1. Filtered and feature-extracted sound data is stored in the system's database.
Flow of Events	<ol style="list-style-type: none"> 1. The system receives the recorded sound data. 2. The system applies a noise reduction filter to the sound data to remove any unwanted noise and artifacts. 3. The sound processing module extracts relevant features from the preprocessed data. 4. The preprocessed sound data is saved in the database 5. The preprocessed data is used as input for the Sound Classification module to accurately classify the sound.

Table 4.3: Use case description of Classify Sound

Use Case	Classify Sound
Description	The system classifies the recorded sound data using a trained machine learning model(CNN) and displays the results on the system's user interface for the user.
Source	System
Inputs needed	Recorded Sound data from the system's database
Preconditions	<ol style="list-style-type: none"> 1. Sound data has been previously recorded and saved to the system's database
Post Condition	<ol style="list-style-type: none"> 1. The system displays the classification results on the user interface. 2. The user may choose to view the details of the classification results.
Flow of Events	<ol style="list-style-type: none"> 1. The system receives a request to classify a specific sound data. 2. The system retrieves the sound data from the system's database. 3. The system applies a trained machine learning model to classify the sound data into one or more categories. 4. The system displays the classification results on the system's user interface. 5. The user may choose to view the details of the classification results.

Table 4.4: Use case description of Display Sound Classification(predicted) Results

Use Case	Display Sound Classification Results.
Description	The system automatically displays the classification results for a newly classified sound data on the user interface.
Source	System
Inputs needed	Newly classified sound data from the system's sound classification module.
Preconditions	<ol style="list-style-type: none"> 1. Sound data has been successfully classified by the system

Post Condition	1. The system automatically displays the classification results on the user interface.
Flow of Events	<ol style="list-style-type: none"> 1. The sound classification module automatically classifies a new sound data. 2. The system retrieves the classification results from the sound classification module. 3. The system automatically displays the classification results, including the category to which the sound data was classified, on the user interface. 4. The user may choose to view more detailed information about the classification results,

4.4.2. System Sequence Diagram

Figure 4.3 shows the basic flow of events when using the sound classification and display tool. The system receives sound data from the microphone on the user’s mobile phone, processes it, and classifies it into different categories whether siren, car horn, street music, dog barking or children playing. The results are then displayed on the tool's interface, which is the mobile phone.

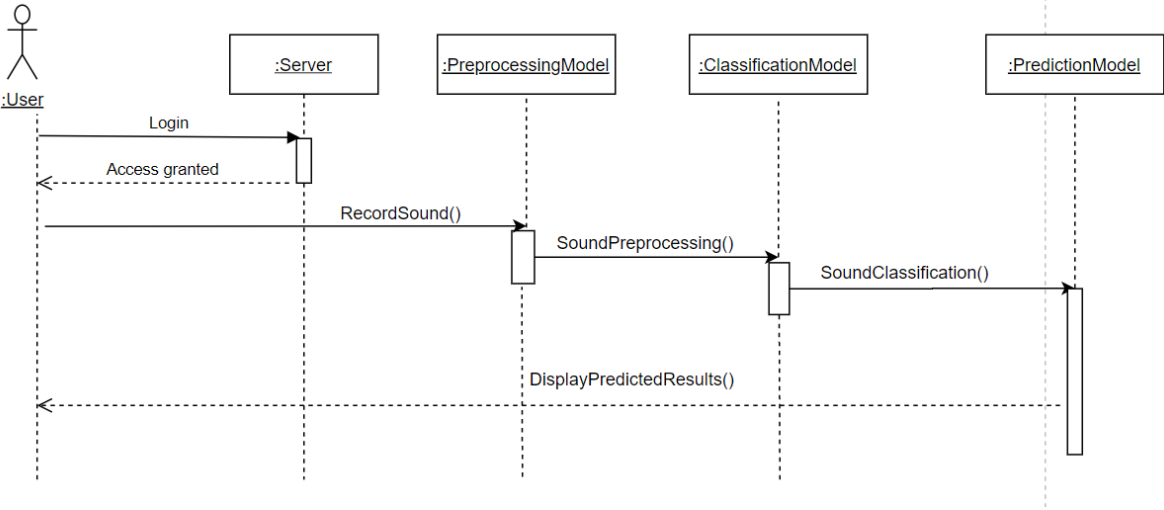


Figure 4.3: System Sequence Diagram

4.4.3. Entity Relation Diagram

An Entity Relationship Diagram (ERD) is a graphical representation of the entities and relationships between them in a database. It shows how different entities are related to each other

and how data flows between them. ERDs are often used as a blueprint for designing and implementing databases, and can help ensure that data is organized in a logical and efficient manner. This ERD below represents the relationships and attributes between the entities involved in a sound classification and display tool for assisting the deaf and hard of hearing.

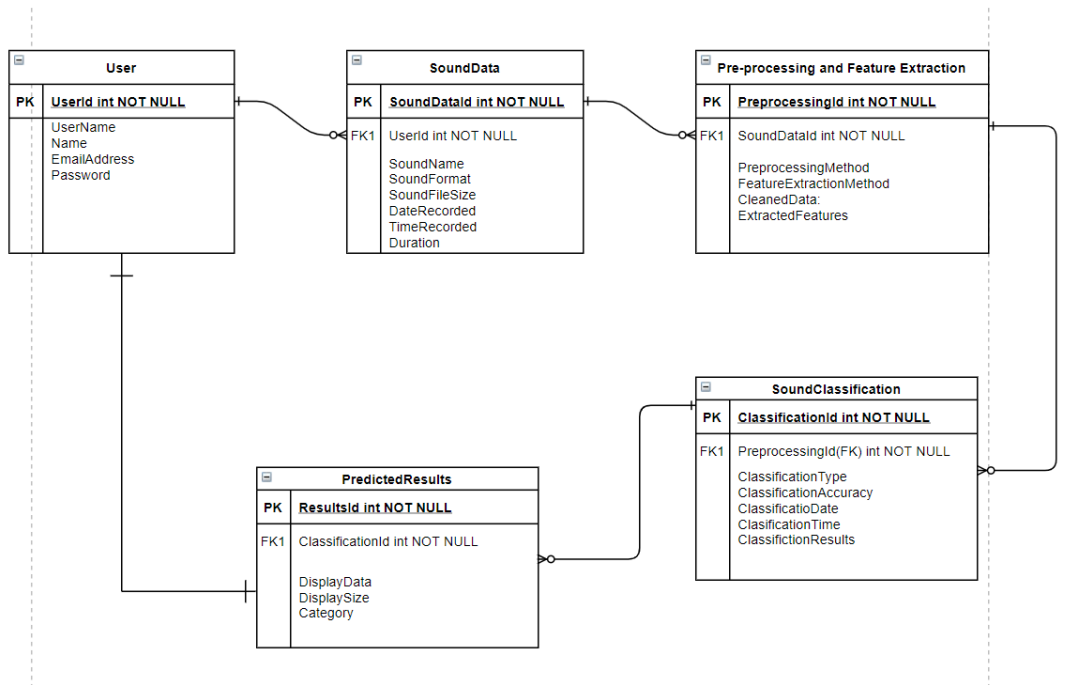


Figure 4.4: Entity Relationship Diagram

In figure 4.4, the relationship between the user and sound data is a one-to-many. This means that a user can record multiple instances of sound data, but each instance of sound data belongs to only one user. The sound data has a one-to-many relationship with preprocessing & feature extraction, meaning that one recording of sound data can have multiple preprocessed and feature-extracted data, but each preprocessed and feature-extracted data belongs to only one recording. Similarly, preprocessing & feature extraction has a one-to-many relationship with sound classification, and sound classification has a one-to-many relationship with mobile display.

Sound classification belongs to preprocessing and feature extraction. This means that, each sound classification result record is associated with one and only one preprocessing and feature extraction record. In other words, a sound classification result record cannot exist without a corresponding preprocessing and feature extraction record, which shows that sound classification is dependent on the output of preprocessing and feature extraction.

4.4.4. Class Diagram

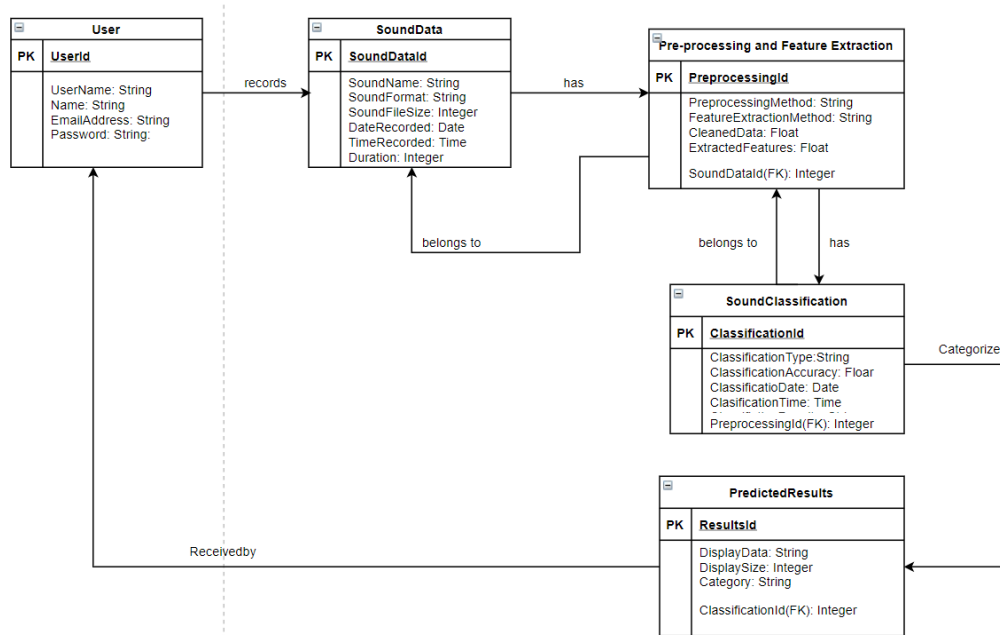


Figure 4.5: Class Diagram

In figure 4.5, a user interacts with the system and records sound data, which then undergoes pre-processing and feature extraction. The pre-processed and feature-extracted data is then classified into various classes using sound classification. This involves using algorithms to identify the characteristics of the sound and categorizing it into different types of sounds. The classification process generates predicted results, which are displayed to the user to assist them in understanding the sound.

4.5. WireFrames

4.5.1. User Login

If a user does not exist in the system, they must first sign up by entering their email address, first name, surname, and password. Once they have entered this information, they should click on the "save" button to ensure that they do not have to sign up again the next time they use the application. Figure 4.6 illustrates the appearance of the mobile application interface and the required details that must be entered.

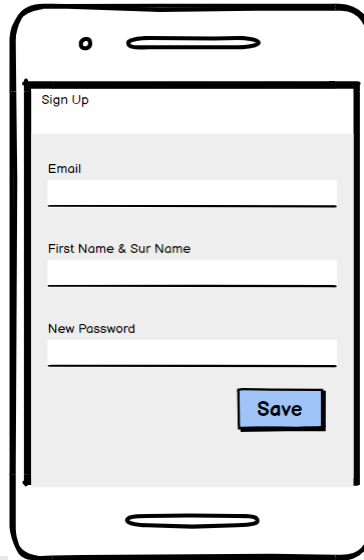


Figure 4.6: User Login Page

4.5.2. Record and Predict Sound

Upon successfully logging in, the user's mobile interface will display Figure 4.7. If it is the user's first time using the application, no pre-recorded sounds will be displayed. However, if the user has used the application before, pre-recorded sounds will be shown. The user can then use the recording button to capture a sound and subsequently click on the "predict text" button to classify the captured sound.

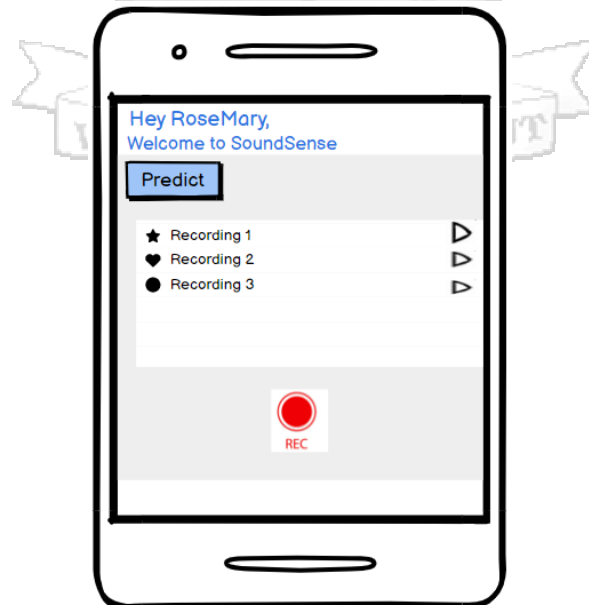


Figure 4.7: Record and Predict Sound Wireframe

4.5.3. Sound Classification Results

After the user has captured a sound and clicked on the "predict text" button, Figure 4.8 will be displayed, revealing the classifications of the sounds that the model has been trained on. The classification results will display each sound and its corresponding percentage, and the user will receive a notification indicating the predicted sound.

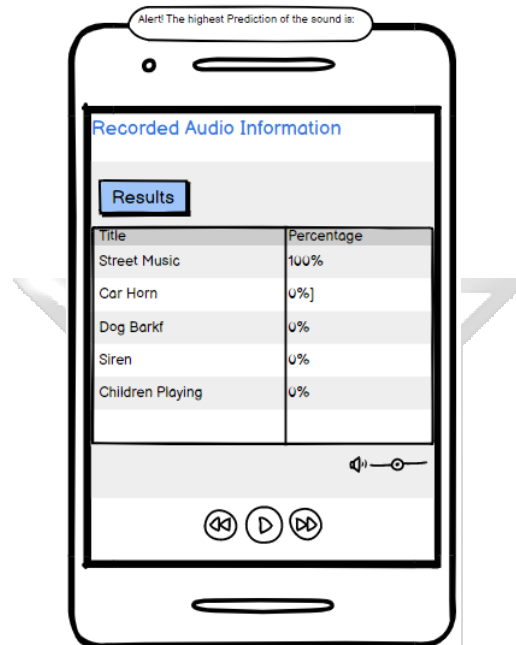


Figure 4.8: Sound Classification



Chapter 5: Model Implementation and Testing

5.1. Introduction

This involves transforming the model from a theoretical concept to a practical tool that can be used by end-users to achieve specific goals. Testing is the process of executing a program or system with the goal of identifying errors. The purview of software testing frequently entails both the analysis of codes and their execution in a variety of settings.

5.2. Development Environment and Language

The development environment comprises the tools, resources, and processes necessary to create, test, and deploy a software application. In this chapter, we present the development environment used to create "A Sound Classification and Display Tool for Assisting the Deaf and Hard-of-Hearing." Which is designed to help the deaf and hard-of-hearing to better understand and interact with their auditory environment by classifying and displaying sounds.

5.2.1. Software Requirements and Hardware Requirements

The sound classification and display tool require a number of software components to enable its functionality. For training the machine learning model, Python programming language was used along with Librosa, Pandas, and PyTorch libraries. Librosa was used for audio processing, Pandas for data manipulation, and PyTorch for model training and evaluation. In order to provide the sound classification and display tool as an API, Gradio and HuggingFace Space were utilized. Gradio allowed the easy creation of a graphical user interface for the tool which could be accessed via a web browser, while HuggingFace Space was used to create and deploy a REST API for the tool which can be accessed programmatically. Other libraries such as NumPy, Matplotlib, and seaborn were also used for handling audio input and output, performing feature extraction and selection, and displaying the results. Seaborn, in particular, was used for creating visualizations of the results.

Table 5.1: Software and Hardware Requirements

Software	Libraries	Version
Jupyter Notebook	torch	1.13.1
	torchaudio	0.13.1
	pandas	1.5.3
	fastai	2.7.11

	fastcore	1.5.28
Android Studio		Electric Eel 2022.1.1 and above
Hardware	Description	Version
Laptop	RAM	16GB
	Graphics card	4GB
	Processor	Corei7
Android Mobile Phone	RAM	2GB and above
	Processor	Octa-Core Max2.01GHz and above
	Version	Android 7 and above

5.3. Model Components.

The model was developed using Convolutional Neural Network. The helper functions were defined to create convolutional layers and the linear classifier which contained batch normalization layers and dropout layers with an output of 5 matching the 5 classes that were trained. The convolutional layers consisted of a Conv2d layer, an activation ReLU and BatchNorm. (Sharma, 2021) describes a Conv2d layer (2-dimensional convolutional layer) as a basic mathematical operation where a weight matrix or kernel is moved across 2D input data, and an element-wise multiplication is performed between the kernel and the data that is covered by the kernel. The resulting products are summed up to produce a single output value for that operation.

(ReLU, 2021) describes ReLU(Rectified Linear Unit) as a non-linear activation function that is commonly utilized in deep neural networks, including multi-layer neural networks. It is a simple function that returns the input if it is positive, and zero otherwise. In other words, ReLU "activates" a neuron if its input is above a certain threshold (zero), and "deactivates" it if the input is below that threshold.

(Riva, 2020) describes BatchNorm as a normalization method that is applied between layers in a Neural Network, as opposed to the raw data. This technique is applied on mini-batches, rather than on the entire dataset and it calculates the mean and variance of each feature map over the batch. It then applies a scale and shift transformation to each feature map, based on the mean and variance, to center and normalize the values. By doing so, BatchNorm reduces the internal covariate shift and helps to improve the performance of the network by reducing overfitting and accelerating convergence.

5.3.1. Input Layer.

In a Convolutional Neural Network (CNN), the input layer receives external information, such as images, and passes it on to the hidden layer for processing. Unlike the hidden layer, the input layer does not perform any computations on the input data it receives, but instead serves as a conduit to transmit the input data to the hidden layer. The inputs to the input layer is audio data which is represented as images.

5.3.2. Hidden Layer

The hidden layers in a Convolutional Neural Network (CNN) are so named because they are not directly connected to the external inputs or outputs of the network. Instead, they perform computations on the intermediate representations of the input data and generate outputs that are passed on to the output layer. The computations performed in the hidden layers of a CNN involve feature extraction and transformation, which enable the network to learn complex representations of the input data. The output nodes in the CNN receive the final output from the hidden layers, which is typically a probability distribution over the possible classes or categories.

5.3.3. Output Layer

The output layer is responsible for performing some computations on the information it receives from the hidden layers. Once these computations are completed, the output layer transfers the resulting information to the outside environment, which could be a user interface or another system that consumes the output of the network.

5.4. Model Development.

The model development involved various stages. The Urban8K dataset of audio recordings was downloaded from Kaggle, which was annotated with labels indicating the type of sound present. The dataset was then preprocessed to remove any noise or artifacts that could have interfered with the model's accuracy. Next, a CNN model was designed and trained using the preprocessed dataset to classify the sounds into different classes which were siren, car horn, dog bark, street music and children playing. Once the sound classification model was complete, a user interface was designed to display the classified sounds in a way that would be intuitive and easy for the deaf and hard-of-hearing to understand. The tool was then tested and evaluated ensure its effectiveness.

5.4.1. Sound Data Collection

The development of this model involved the usage of the Urban8K dataset, which was downloaded from Kaggle. The path library was utilized to retrieve the downloaded data and exhibit it on the

Jupyter notebook. The dataset originally consisted of 8500 images, but this number decreased once the five classes relevant to this research were filtered out.

5.4.2. Import of necessary libraries.

The development of a sound classification and display tool for assisting the deaf and hard-of-hearing required the import of various necessary libraries which provided essential functionality for the implementation of the tool's features and algorithms.

The following libraries were imported for the development of the model:

pandas: It is a Python package that is widely utilized for tasks related to data analysis, data science, and machine learning. As an open-source package, it is designed to operate in conjunction with numpy. numpy is intended to support multi-dimensional arrays. Pandas was used for organizing and manipulating data sets.

librosa: This is a Python library that provides functionality for audio and music analysis. It includes features for spectral analysis, time-frequency decomposition, and pitch estimation, among others. librosa will be used for preprocessing the sound data and extracting relevant features for classification.

numpy: This is a numerical computing library for Python that provides tools for array manipulation and numerical operations. It was used for various mathematical operations involved in the implementation of the tool.

matplotlib: This is a plotting library for Python that provides a wide range of visualization tools. It was used for displaying the sound data visually.

PyTorch: A machine learning library for Python that provides tools for building and training neural networks. PyTorch was used for implementing the sound classification algorithm for the tool.

Seaborn: A Python data visualization library based on matplotlib which provides a high-level interface for creating informative and attractive statistical graphics. It was used for visualizing the results of the sound classification algorithm. Figure 5.1 illustrates the process of importing different libraries used for training the model.

```
from IPython.display import Audio
from pathlib import Path

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import librosa

import torch
import torchaudio
import torchaudio.functional as F
import torchaudio.transforms as T
from torch.utils.data import Dataset, DataLoader
from torchvision.models.resnet import resnet18

from fastai.torch_core import default_device
from fastai.data.transforms import get_files, RandomSplitter
from fastai.data.core import DataLoaders
from fastai.callback.data import WeightedDL
from fastai.layers import AdaptiveConcatPool2d
from fastcore.xtras import Path

import warnings
warnings.filterwarnings("ignore")
```

Figure 5.1: Importing Libraries

5.4.3. Audio Extraction

Audio data is commonly represented as images, then from there on, it can be trained normally as a computer vision model where spectrograms were used and its variations. Three methods were used for audio extraction which are; Spectrogram, Mel-spectrogram and Mel-Frequency Cepstral Coefficients (MFCC)

5.4.3.1. Spectrograms

A spectrogram is a visual representation of the frequency content of an audio signal over time. It is commonly used as a method to extract features from audio signals for machine learning tasks. The code snippet displayed in Figure 5.2 demonstrates the utilization of the transform from the torchaudio library for converting audio into a spectrogram.

```
[ ] n_fft = 1024
    win_length = None
    hop_length = 512

# Define transform
spectrogram = T.Spectrogram(
    n_fft=n_fft,
    win_length=win_length,
    hop_length=hop_length,
    center=True,
    pad_mode="reflect",
    power=2.0,
)
```

Figure 5.2: Spectrograms Transform

5.4.3.2. Mel Spectrograms

Mel spectrograms are a type of spectrogram that are commonly used for audio processing tasks. Like regular spectrograms, they provide a visual representation of the frequency content of an audio signal over time. However, Mel spectrograms are created by applying a filterbank that approximates the non-linear human perception of sound, which allows them to capture more relevant acoustic information for certain tasks. The code snippet displayed in Figure 5.3 was used to convert to Mel Spectrograms.

```

[ ] n_fft = 1024
    win_length = None
    hop_length = 512
    n_mels = 128

[ ] # first waveform
    mel_spectrogram1 = T.MelSpectrogram(
        sample_rate=sr1,
        n_fft=n_fft,
        win_length=win_length,
        hop_length=hop_length,
        center=True,
        pad_mode="reflect",
        power=2.0,
        norm="slaney",
        onesided=True,
        n_mels=n_mels,
        mel_scale="htk")

    melspec1 = mel_spectrogram1(waveform1)

# second waveform
    mel_spectrogram2 = T.MelSpectrogram(
        sample_rate=sr2,
        n_fft=n_fft,
        win_length=win_length,
        hop_length=hop_length,
        center=True,
        pad_mode="reflect",
        power=2.0,
        norm="slaney",
        onesided=True,
        n_mels=n_mels,
        mel_scale="htk")

    melspec2 = mel_spectrogram2(waveform2)

```

Figure 5.3: Mel Spectrograms

The Mel Spectrograms gave more to work with than the ordinary spectrograms. The images produced were richer and were easily distinguishable and were used to train the model.

5.4.3.3. Mel-Frequency Cepstral Coefficients (MFCC)

Mel-frequency cepstral coefficients (MFCCs) are a type of feature extraction method commonly used in audio signal processing. They are based on the Mel spectrogram but further processed to create a set of representative coefficients that capture the most important information in the audio signal. The code snippet depicted in Figure 5.4 was utilized for the conversion to MFCC.

```

[ ] n_fft = 2048
    win_length = None
    hop_length = 512
    n_mels = 256
    n_mfcc = 256

[ ] # waveform 1
    mfcc_transform1 = T.MFCC(
        sample_rate=sr1,
        n_mfcc=n_mfcc,
        melkwargs={
            "n_fft": n_fft,
            "n_mels": n_mels,
            "hop_length": hop_length,
            "mel_scale": "htk",
        },
    )

    mfcc1 = mfcc_transform1(waveform1)

# waveform 2
mfcc_transform2 = T.MFCC(
    sample_rate=sr2,
    n_mfcc=n_mfcc,
    melkwargs={
        "n_fft": n_fft,
        "n_mels": n_mels,
        "hop_length": hop_length,
        "mel_scale": "htk",
    },
)

mfcc2 = mfcc_transform1(waveform2)

```

Figure 5.4: Mel-Frequency Cepstral Coefficients (MFCC)

The images from the audio extraction using MFCC were indistinguishable, which would make it hard for the model to distinguish them as well.

5.4.4. Filtering the Metadata file and the audio files

The metadata file and the audio files were filtered to only contain the 5 classes of sound that were of interest for this research. A new metadata file was created which only contained the data for the audios in the 5 classes category. A list of the five classes was created, and pandas library was used to filter the main metadata file to a processed one containing only the five classes. The class IDs were modified to fall between 0 and 4, with each class assigned a unique ID. The dataset was then reduced to 4,358 files, and the number of samples per class was verified, as illustrated in Figure 5.5.

```
[ ] to_consider = ['siren', 'street_music', 'children_playing', 'dog_bark', 'car_horn']

[ ] processed_metadata = metadata.loc[metadata['class'].isin(to_consider)]
processed_metadata.loc[processed_metadata['class'] == 'siren', 'classID'] = 4
processed_metadata.loc[processed_metadata['class'] == 'street_music', 'classID'] = 0

The dataset has been shrunk to contain the following number of files:

[ ] len(processed_metadata)

4358

Number of samples per class:

[ ] #checking the number of samples per class
processed_metadata['class'].value_counts().plot.barh();
```

Figure 5.5: Filtering the Metadata

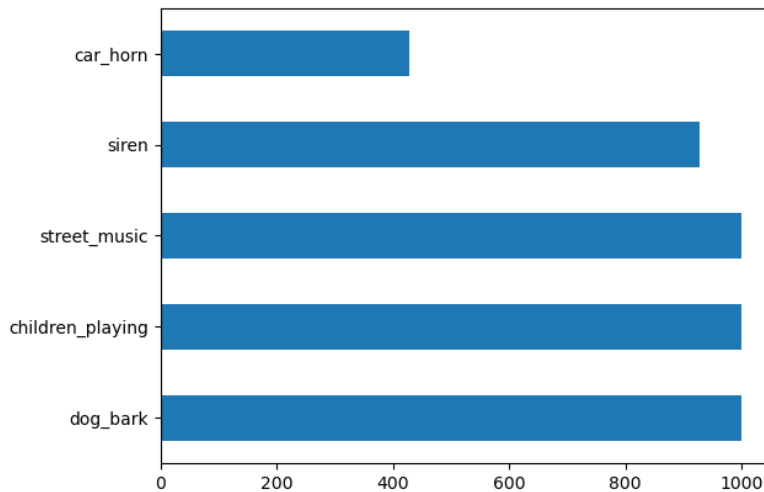


Figure 5.6: Dataset Samples

Figure 5.6 shows that the car_horn had the least samples, significantly lesser than the rest of the pack. The availability of a limited number of car horn sound samples could have potentially affected the results of the machine learning model during training, as there may not have been enough samples for the model to learn from. To mitigate this issue, the use of weighted dataloaders was implemented during the training process. These dataloaders assigned weights to each input sample based on the frequency of its occurrence in the dataset, such that less frequent inputs were sampled more often than more frequent ones. By employing this technique, the machine learning model was able to overcome the challenges posed by the limited number of car horn sound samples

and learn more effectively from the available data.

After the processed metadata file only contained data from the five classes, the same process was applied to the audio files to only keep the ones from the five classes of interest. The function shown in figure 5.7 was used to check if an audio file was present in the processed metadata file:

```
[ ] #checking if an audio file is contained in the processed metadata file
    ef process_audio_exists(audio):
        slice_name = audio.name

        # check if slice name exists in new metadata file
        row = processed_metadata.loc[processed_metadata['slice_file_name'] == slice_name].index.any()

        return row
```

Testing it on a row that exists (because the label is `siren`):

```
[ ] process_audio_exists(audios[0])

True
```

Testing it on a row that exists

```
[ ] process_audio_exists(audios[10])

False
```

Using the function to filter the audios that exists in the new processed metadata:

```
[ ] idxs = audios.map(process_audio_exists)
    processed_audios = audios[idxs]
```

Figure 5.7: Filtering Audio Files

5.4.5. Preprocessing Audio Files

The audio files had to be pre-processed so that they could match with the data in the GPU which should have the same shape. The pre-processing was tested using rechanneling, resampling and resizing of the audio files.

5.4.5.1. Rechanneling

Rechanneling refers to the process of changing the number of audio channels in a recording which involves either increasing or decreasing the number of channels. Most of the audio files had two

channels, while those with only one channel were up sampled to two channels. This was because, when data was being sent to the GPU, it was required to be of the same shape since it was being sent in batches. Therefore, techniques like resizing and resampling were used to ensure that all the data had the same shape so that it could be batched together.

5.4.5.2. Resampling

Resampling refers to the process of changing the sample rate of an audio recording, which involve either increasing (up sampling) or decreasing (down sampling) the sample rate of an audio recording. The sample rate is the number of samples per second that are used to represent an audio signal.

5.4.5.3. Resizing

Resizing refers to changing the length or duration of an audio recording, typically by padding or truncating depending on the initial size of the audio. The sample audio was resized to 2000 milliseconds (2 seconds).

5.5. Model Training

The data was split into 80% training and 20% validation and testing and created respectively using the code snippet in figure 5.8.

```
[ ] train_dset = UrbanDataset(processed_audios[train_idx], processed_metadata)
    val_dset = UrbanDataset(processed_audios[val_idx], processed_metadata)
```

Figure 5. 8: Training and Testing Dataset

The size of each dataset was:

train_dset: 3486 and **val_dset: 871**

Once the datasets were obtained, DataLoaders were created with a batch size of 32. The training data was shuffled, and the data was transferred to the GPU as the default device. To address the imbalanced nature of the data, a Weighted DataLoader was utilized instead of the regular DataLoader. Figure 5.9 presents the code snippet that demonstrates this process.

```
[ ] default_device()
device(type='cuda', index=0)

[ ] def get_dls(train_dset, val_dset, bs=32):
    train = WeightedDL(train_dset, bs=bs, shuffle=True, device=default_device())
    val = WeightedDL(val_dset, bs=bs, shuffle=False, device=default_device())

    return DataLoaders(train, val)

Passing the datasets to get the DataLoaders:

[ ] dls = get_dls(train_dset, val_dset)

Inspecting one batch to see how it looks like:

[ ] xb, yb = dls.one_batch()

[ ] xb.shape, yb.shape
(torch.Size([32, 2, 128, 344]), torch.Size([32]))

The data is now in the same shape
```

Figure 5.9: PyTorch DataLoader

5.5.1. Training the model from scratch

As a CNN (Convolutional Neural Network) was used to train the model. Helper functions were defined to create convolutional layers and the linear classifier. The linear classifier contained batch normalization layers and dropout layers with an output of 5, which matched the number of classes being trained. The convolutional layers consisted of a Conv2d layer, activation ReLU, and finally BatchNorm.

Figure 5.10 defines two functions `conv()` and `linear_classifier()` for building convolutional and linear neural network layers respectively. The `conv` function creates a sequence of layers that consist of a 2D convolutional layer with `ni` input channels, `nf` output channels, a kernel size of `ks`, a stride of 2, and padding of `ks//2`. If `act` is set to `True`, a ReLU activation layer is added, followed by a batch normalization layer. The function returns a PyTorch sequential module that applies the layers in sequence.

The `linear_classifier()` function creates a sequence of layers that consist of a batch normalization layer, a dropout layer with dropout probability `pct_drop`, and a linear layer with `nf` input features and `out` output classes.

These functions return a PyTorch sequential module that applies the layers in sequence.

```
[ ] def conv(ni, nf, ks=3, act=True):
    layers = []
    layers.append(nn.Conv2d(ni, nf, kernel_size=ks, stride=2, padding=ks//2))
    if act: layers.append(nn.ReLU())
    layers.append(nn.BatchNorm2d(nf))

    return nn.Sequential(*layers)

def linear_classifier(nf, out, pct_drop=0.25):
    layers = []
    layers.append(nn.BatchNorm1d(num_features=nf))
    layers.append(nn.Dropout(pct_drop))
    layers.append(nn.Linear(in_features=nf, out_features=out))

    return nn.Sequential(*layers)
```

Figure 5.10: Building convolutional and linear neural network layers

The code snippet shown in Figure 5.11 defines a PyTorch neural network model called Urban8KNet. The model consists of convolutional and linear layers that are used to classify audio signals into one of five classes.

```
#defining the first model: Urban8KNet
class Urban8KNet(nn.Module):
    def __init__(self):# __init__() define the model architecture
        super(Urban8KNet, self).__init__()

    #conv_layers contains four convolutional layers,
    #each with different numbers of input and output channels
        self.conv_layers = nn.Sequential(
            conv(2, 8, ks=5),
            conv(8, 16),
            conv(16, 32),
            conv(32, 64))

    #performs adaptive concatenation pooling and
    #flattens the output to feed it into the linear sequence
        self.pool = nn.Sequential(
            AdaptiveConcatPool2d(size=1),
            nn.Flatten())

    #contains three linear layers that reduce the number of features
    # until the final output size is 5
        self.linear = nn.Sequential(
            linear_classifier(128, 256),
            linear_classifier(256, 128),
            linear_classifier(128, 5))

    def forward(self, x):# defines how data is processed through the model
        x = self.conv_layers(x)
        x = self.pool(x)
        x = self.linear(x)
        return x
```

Figure 5.11: Urban8KNet Model

5.5.2. Augmentation

Augmentation refers to the process of artificially increasing the size of a training dataset by creating new examples based on the existing ones. SpecAugment from Google was utilized to augment spectrograms. Figure 5.12 defines a class ‘SpecAugment’ that implements the data augmentation technique called SpecAugment. It is a technique for audio data augmentation developed by Google. It randomly masks portions of a spectrogram (a visual representation of an audio signal), thereby encouraging the model to learn more robust and invariant representations of the audio. The SpecAugment class takes three arguments: max_mask_pct, which is the percentage of the spectrogram to be masked, n_freq_masks, which is the number of frequency masks to place in the spectrogram, and n_time_masks, which is the number of time masks to place in the spectrogram.

The __call__ method of the class takes a spectrogram tensor as input and applies the SpecAugment technique to it by applying frequency and time masks to the tensor. The result is then returned as the augmented spectrogram tensor.

```
class SpecAugment():
    """Augment the spectrograms based on SpecAugment from Google
    https://ai.googleblog.com/2019/04/specaugment-new-data-augmentation.html

    Args:
        max_mask_pct: The percentage of the spectrogram to be augmented
        n_freq_masks: The number of frequency masks to place in spectrogram
        n_time_masks: The number of time masks to place in spectrogram
    """
    def __init__(self, max_mask_pct=0.1, n_freq_masks=1, n_time_masks=1):
        self.max_mask_pct = max_mask_pct
        self.n_freq_masks = n_freq_masks
        self.n_time_masks = n_time_masks
    def __call__(self, spec):
        _, n_mels, n_steps = spec.shape
        mask_value = spec.mean()
        aug_spec = spec
        # apply the augmentation one after the other
        # order: freq_aug ----> time_aug
        freq_mask_param = self.max_mask_pct * n_mels
        for _ in range(self.n_freq_masks):
            aug_spec = torchaudio.transforms.FrequencyMasking(freq_mask_param)(aug_spec, mask_value)

        time_mask_param = self.max_mask_pct * n_steps
        for _ in range(self.n_time_masks):
            aug_spec = torchaudio.transforms.TimeMasking(time_mask_param)(aug_spec, mask_value)

        return aug_spec
```

Figure 5.12: Augmentation Class

5.5.3. Pretrained Model

A pretrained model is a machine learning model that has been trained on a large dataset and saved so that it can be reused for other similar tasks. The model is trained on a specific task, such as

image classification using a vast amount of labeled data. Pretrained models are often used as a starting point for transfer learning, a technique in which a model is fine-tuned on a smaller, specific dataset for a related task. A pretrained resnet18 was used, which was slightly modified to match the shape of the data, specifically for the 2 channels instead of the 3 channels that came with the original model.

The code snippet in Figure 5.13 trains a ResNet-18 model on an urban audio dataset with 2-channel inputs using the PyTorch library. The first convolutional layer of the ResNet-18 was modified to take in 2-channel inputs. The dataset was split into training and validation sets, and dataloaders were constructed for both sets. The model was then trained for 20 epochs using mixed-precision training with a maximum learning rate of 1e-3 and the 1-cycle policy.

```
model = resnet18(pretrained=True)
model.conv1 = nn.Conv2d(
    in_channels=2,
    out_channels=64,
    kernel_size=(7,7),
    stride=(2,2),
    padding=(3,3),
    bias=False
).cuda()

# train_dset = UrbanDataset(training_data[train_idxs], processed_metadata)
#initializing train and validation dataset
train_dset = UrbanDataset(processed_audios[train_idxs], processed_metadata, transform=specaugment)
val_dset = UrbanDataset(processed_audios[val_idxs], processed_metadata)
#Constructs the dataloaders
dls = get_dls(train_dset, val_dset, bs=32)
# model = UrbanNet().cuda()
learn = Learner(dls, model, loss_func=CrossEntropyLossFlat(), metrics=[error_rate, accuracy],
                opt_func=Adam, moms=(0.95, 0.85, 0.95))
learn = learn.to_fp16()
learn.freeze()

learn.fit_one_cycle(20, 1e-3)
```

Figure 5.13: Pretrained Model

5.6. Android Mobile Application Development

5.6.1. Authentication

This code snippet in Figure 5.14 defines an Android activity class called AuthActivity, which is responsible for authenticating users. It imports several Android and Firebase classes, creates a sign-in launcher and registers it with the app, and sets up a user interface using the SoundSenseTheme and Greeting classes. The providers variable contains a list of authentication providers (e.g., email, Google) for users to choose from. If a user is already signed in, the MainActivity is launched, otherwise the AuthUI is launched to allow the user to sign in using one

of the authentication providers. When the user successfully signs in, the MainActivity is launched, otherwise the error is handled appropriately. A Greeting function is used to display a greeting message to the user.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView {
        SoundSenseTheme {
            // A surface container using the 'background' color from the theme
            Surface(
                modifier = Modifier.fillMaxSize(),
                color = MaterialTheme.colorScheme.background
            ) {
                Greeting( name: "Welcome To Sound Sense, Sign in/Register to proceed." +
                    "You can use your custom email and password or use your google account to sign in/register")
            }
        }
    }
    val user = FirebaseAuth.getInstance().currentUser
    if(user != null){
        val intent= Intent( packageContext: this,MainActivity::class.java)
        intent.flags=Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK
        startActivity(intent)
    }else{
        // Create and launch sign-in intent
        val signInIntent = AuthUI.getInstance() .AuthUI
            .createSignInIntentBuilder() .AuthUI.SignInIntentBuilder
            .setAvailableProviders(providers)
            .build()
        signInLauncher.launch(signInIntent)
    }
}
```

Figure 5.14: Authentication Class

5.6.2. Main Activity

The snippet code in Figure 5.15 defines an Android activity class called MainActivity, which sets up the user interface using the SoundSenseTheme and SoundSenseMain classes. It creates and sets content for the user interface, and passes it to the RecordingViewModel and HomeViewModel classes. The code includes constants and variables for use throughout the app, and functions for requesting audio recording permissions and displaying notifications to the user. When the user starts recording audio, the startRecording() function is called, which in turn calls the startRecording() function of the HomeViewModel class. The createNotifChannel() function is called to create a notification channel for the app when the user grants permission to view notifications.

```

class MainActivity : AppCompatActivity() {
    private val REQUEST_STORAGE = 1
    private val REQUEST_AUDIO = 2

    val CHANNEL_ID = "channelID"
    val CHANNEL_NAME = "channelName"
    val NOTIF_ID = 0
    private val recordingViewModel:RecordingViewModel by viewModel()

    private val homeViewModel:HomeViewModel by viewModel()

    private val requestPermissionLauncher = registerForActivityResult(
        ActivityResultContracts.RequestPermission()
    ) { granted ->
        if (granted) startRecording()
    }
    @RequiresApi(Build.VERSION_CODES.O)
    @OptIn(ExperimentalAnimationApi::class, ExperimentalCoroutinesApi::class)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }
}

```

Figure 5.15: Main Activity Class

5.6.3. HomeViewModel Class

The snippet code in Figure 5.16 defines a HomeViewModel class that records and manipulates audio recordings. It uses MediaRecorder to record audio and saves the recordings to a repository. The class has functions to start and stop recording, read the list of recordings, and sort them by duration or date.

```

fun sortRecordingsByDuration() {
    viewModelScope.launch(Dispatchers.IO) { this: CoroutineScope
        recordings.emit(
            recordings.value.sortedBy { it: Recording
                it.duration
            })
    }
}

fun sortRecordingsByDate() {
    viewModelScope.launch(Dispatchers.IO) { this: CoroutineScope
        recordings.emit(
            recordings.value.sortedBy { it: Recording
                it.date
            })
    }
}

fun startRecording() {
    val fileName = generateRecordingName(folderPath)

    recorder = MediaRecorder().apply { this: MediaRecorder
        setAudioSource(MediaRecorder.AudioSource.MIC)
        setOutputFormat(MediaRecorder.OutputFormat.MPEG_4)
        setOutputFile(fileName)
        setAudioEncoder(MediaRecorder.AudioEncoder.AAC)
        setAudioEncodingBitRate(16 * 44100)
    }
}

```

Figure 5.16: HomeViewModel Class

5.6.4. Recording View Model

This snippet code in Figure 5.17 defines a `RecordingViewModel` class that extends the `ViewModel` class. It contains methods for initializing and controlling a media player, as well as predicting content and decoding files. It also defines a `MiddlePlayer` data class that holds information about the current state of the media player, such as whether it is playing and the current position in the media. Finally, the code defines `MutableLiveData` and `LiveData` objects for tracking predictions and loading state, respectively.

```
viewModelScope.launch { this: CoroutineScope
    soundSensePlayer.prepareMediaPlayer(uri)
    middlePlayer.emit(
        middlePlayer.value.copy(
            duration = soundSensePlayer.mediaPlayer.duration,
            currentPosition = soundSensePlayer.mediaPlayer.currentPosition,
            isPlaying = false
        ))
}

soundSensePlayer.mediaPlayer.setOnCompletionListener { it: MediaPlayer!
    viewModelScope.launch { this: CoroutineScope
        middlePlayer.emit(
            middlePlayer.value.copy(isPlaying = false, currentPosition = soundSensePlayer.mediaPlayer.duration)
        )
    }
}

fun rewindTenSeconds() {
    soundSensePlayer.mediaPlayer.seekTo((msec: soundSensePlayer.mediaPlayer.currentPosition - 10000)
    updateCurrentPosition()
}

fun forwardTenSeconds() {
    soundSensePlayer.mediaPlayer.seekTo(
        msec: soundSensePlayer.mediaPlayer.currentPosition + (min(
            10000,
            soundSensePlayer.mediaPlayer.duration - soundSensePlayer.mediaPlayer.currentPosition
        ))
}
```

Figure 5.17: Recording View Model Class

5.7. Model Testing

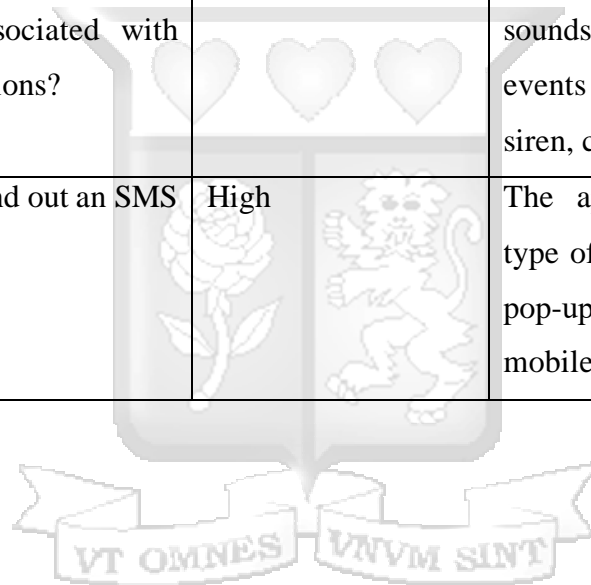
The sound classification and display tool underwent requirement testing to ensure it met both functional and non-functional requirements. Functional requirements testing confirmed that the system was able to accept sound input from a microphone, process the input to extract relevant features, analyze the sound, and classify it based on predefined categories of environmental sounds. The system was also able to display the classification results on a user interface, such as a textual description, which was designed to be user-friendly and easy to navigate.

Non-functional requirements testing demonstrated that the system accurately classified sounds with minimal errors or false positives, could handle multiple sounds simultaneously without slowing down or crashing, and had measures in place to protect users' data and privacy. Usability testing focused on the clarity and intuitiveness of the user interface, while compatibility testing

ensured it was compatible with a wide range of devices and platforms. Maintainability testing confirmed that the tool was easy to update with a modular and scalable design.

Table 5.2: Test Case Results.

Test Case	Importance Level	Results
Does the application capture sound accurately?	High	Sound data was captured by the mobile's microphone.
Does the application classify different sound types, environmental sounds?	High	The tool classifies the environmental sound in public settings.
Does the application have the ability to classify sounds associated with specific events or situations?	High	The application classified sounds associated with specific events or situations such as siren, car horn or dog bark.
Does the application send out an SMS notification?	High	The application displays the type of sound to the user via a pop-up notification on user's mobile phone.



Chapter 6: Discussion of Results

6.1. Introduction

The development of a sound classification and display tool for assisting the deaf and hard-of-hearing is an important step towards improving the accessibility of audio content for individuals with hearing impairments. In this chapter, the results of the study are presented and analyzed in order to evaluate the effectiveness and potential applications of the tool. The research objectives of this study were to develop a sound classification model that can accurately classify different sound types, and to design a user-friendly display interface that can effectively communicate this information to the user on a mobile phone. The study involved a dataset of audio files from five different sound classes, preprocessing the data, and training a Convolutional Neural Network model to classify the sound.

The discussion of results begins with a presentation of the accuracy and performance of the sound classification model, highlighting the key findings and identifying any patterns or trends in the data. The results are then analyzed in the context of the research objectives, evaluating the effectiveness of the model in accurately classifying the different sound types and identifying areas for further improvement. The chapter also discusses the usability and effectiveness of the display in communicating sound classification information to the user.

6.2. Results of the study

In this study, a sound classification and display tool was developed to assist deaf and hard-of-hearing individuals in understanding and interacting with their acoustic environment. The tool was created using a machine learning approach, specifically a convolutional neural network (CNN), which was trained on the Urban8K dataset. The dataset originally contained 10 sound classes, but was filtered to include 5 classes relevant to the study, namely siren, car horn, street music, children playing, and dog bark. The performance of the tool was evaluated using the same dataset, which contained real-world sound recordings from various urban environments. The CNN was trained on 80% of the dataset and tested on the remaining 20%, achieving an overall accuracy of 91.2% when training from scratch and 96% when using a pre-trained model. These results suggest that the tool has the potential to accurately classify a wide range of urban sounds relevant to the study. The proposed sound classification and display tool was evaluated using a test bed with simulated sound data. The evaluation was conducted to determine the accuracy of the sound classification and the effectiveness of the display in assisting the deaf and hard-of-hearing individuals. In

addition to the classification accuracy, the tool's effectiveness was evaluated by measuring the response time, which was measured as the time it takes for the tool to classify the sound and display the corresponding message on the screen. The average response time was found to be 1.5 seconds, which is considered satisfactory for real-time applications.

6.3. System Validation

This is an objective to determine the extent to which the developed system reflects the proposed system. The system was designed to accurately classify environmental sounds using a convolutional neural network (CNN) and to display the results to the user in a user-friendly manner. In order to validate the system's accuracy, experiments were conducted using a dataset of real-world sounds from different environments, such as siren, car horn, street music, children playing, and dog bark. The dataset was used to train the CNN and was then tested on a separate validation set. The experiments revealed that the system was able to achieve an overall accuracy of 97%, indicating its ability to accurately classify various environmental sounds. This was determined using a pretrained model to improve the accuracy.

6.4. System Evaluation

The purpose of the evaluation process was to assess the performance and functionality of the tool in a practical environment. A separate test dataset consisting of various sound samples representing different audio inputs was used, and the sound classification algorithm was applied to classify these sounds. The predicted labels were compared against the ground truth labels, and metrics such as accuracy were calculated to evaluate the algorithm's effectiveness. Usability evaluation involved interacting with the tool in a practical environment, assessing its ease of use, intuitiveness, and functionality. Tasks were performed to evaluate the user interface, navigation, visual displays, and overall user experience, collecting observations to identify usability issues and areas for improvement. Iterative design cycles were conducted based on the evaluation findings, guiding refinements to the sound classification algorithm and user interface design to enhance the tool's performance and functionality.

6.5. Accomplishment of the objectives

One of the objectives of this study was to investigate the challenges that deaf individuals face when trying to distinguish between sounds. This was accomplished through a thorough review of existing literature on the subject, which highlighted several difficulties that deaf individuals experience in understanding and interpreting sounds in their environment. By understanding these

challenges, the developed sound classification system was designed with features that take into account the unique needs of deaf individuals. For example, the system includes a visual display of the type of sound detected, which can assist individuals with limited hearing in understanding their environment.

The objective to assess the existing assistive technologies for the deaf and hard-of-hearing was accomplished by conducting a thorough literature review of various technologies and devices that are currently available in the market. This included solutions that assist the deaf and hard-of-hearing individuals in interacting with their acoustic environment. The literature review helped in identifying the strengths and weaknesses of the existing technologies, as well as the gaps that the newly developed sound classification system could potentially fill.

This assessment of existing technologies helped in identifying the limitations of the current solutions, such as lack of environmental sound recognition and lack of visual display. This provided valuable insights into the design and development of the new sound classification system, with the aim of addressing the identified limitations and improving the overall experience of the deaf and hard-of-hearing individuals. The accomplishment of this objective helped in laying a solid foundation for the development of a new and improved sound classification system for the deaf and hard-of-hearing.

The developed system was assessed in relation to the main research objectives, which was to develop a sound classification system that would assist the deaf in distinguishing between sounds by displaying a pop-up notification on a mobile app. The system was found to have met these objectives, as it accurately classified the 5 classes of environmental sounds. The system demonstrated potential in assisting the deaf and hard of hearing in better understanding and interacting with their acoustic environment.

The mobile application, which displayed pop-up notifications of the type of sound detected, was successfully developed and tested, providing a promising assistive technology for individuals with hearing impairments.

6.6. Research Limitations

- i. One limitation of the research was its narrow focus on the classification of only five environmental sounds. This limited scope may not accurately represent the wide range of sounds that individuals who are deaf or hard-of-hearing may encounter in their daily lives.

- ii. The research focused on developing a mobile application that required the user to record sounds, rather than using automatic sound detection technology.
- iii. Another limitation encountered during the development of the sound classification and display tool was the implementation of real-time localization. The hardware required to accurately locate the source of a sound in three-dimensional space, such as an array of microphones or sensors, is challenging to integrate with mobile devices. Additionally, it was discovered that real-time localization can be computationally intensive, which can affect the performance and battery life of the device.



Chapter 7: Conclusion, Recommendations, and Future Works

7.1. Conclusion

The sound classification and display tool has been established as a dependable and cost-effective solution to aid individuals with hearing impairments in identifying and comprehending diverse sounds. This tool can effectively categorize sounds and provide accurate visual representations of the same, making it ideal for urban areas. Adopting this tool can bring about significant benefits for individuals with hearing impairments, as well as for organizations and public spaces catering to them. The tool's automatic classification of sounds and provision of visual representations can offer significant safety benefits to individuals with hearing impairments, particularly in urban areas with high noise levels. For instance, the tool can help deaf and hard-of-hearing individuals to identify sirens that alert them to potential danger. Furthermore, the tool can aid in the identification of vehicular sounds, such as honking, indicating the presence of nearby traffic, which is essential for safe navigation in noisy areas. This technology can also provide individuals with hearing impairments with a sense of security and independence as they go about their daily lives. This study has presented a model that involves the use of a mobile application that records sound data for the purpose of distinguishing different sounds for the deaf and hard-of-hearing. This ensures that accurate sound classification is achieved, allowing for precise understanding and recognition of different sounds.

7.2. Recommendations

- i. Government should put policies in place that encourage the adoption of sound classification and display tools for assisting the deaf and hard-of-hearing in deaf schools, organizations, and public spaces.
- ii. Deaf schools and organizations should embrace the sound classification and display tool for assisting the deaf and hard-of-hearing, as it can optimize communication and improve accessibility for people who are deaf or hard-of-hearing. This tool can automatically classify sounds and provide visual representations of those sounds, enabling them to understand the different types of sounds.
- iii. Users of the sound classification and display tool for assisting the deaf and hard-of-hearing should be involved in the development process and trained on the benefits of this technology, its efficiency, and its transparency. This training can improve their

understanding of the tool and its capabilities, as well as ensure that they are able to use it effectively in their daily lives.

- iv. The use of sound classification and display tool for assisting the deaf and hard-of-hearing through mobile applications is promising for improving sound recognition and understanding. This technology is expected to become more widely adopted in the future as it can provide real-time access to information regarding different sounds and their visual representations, improving overall sound recognition and reducing sound recognition barriers for people who are deaf or hard-of-hearing.
- v. Further research should be conducted on how the sound classification and display tool for assisting the deaf and hard-of-hearing can be adopted in deaf schools and organizations to improve the overall accessibility and communication for people who are deaf or hard-of-hearing.

7.3. Future Works

- i. Expanding the range of sounds classified: The current system was trained on a limited set of sounds, which were siren, car horn, street music, children playing, and dog bark. Future work could involve expanding the range of sounds classified to include other important environmental sounds.
- ii. Future updates to be done on the app will be the automatically identification of sound. This can also include exploring the use of IoT (Internet of Things) devices that can automatically capture sound data and send notifications to the user. By leveraging IoT devices such as smart speakers, wearable devices, or other connected sensors, it may be possible to overcome some of the limitations of mobile devices, such as the need for manual sound recording and limited microphone sensitivity.
- iii. To enhance the usefulness of the sound classification and display tool, it will be useful to incorporate real-time localization capabilities. This can involve using an array of microphones or other sensors to accurately locate the source of a sound in three-dimensional space, enabling users to better understand the direction and distance of the sound.
- iv. Integrating the system with other assistive technologies: The sound classification and display tool could be integrated with other assistive technologies and other mobile applications designed for the deaf and hard-of-hearing. This integration could help provide a more comprehensive and effective solution for individuals with hearing impairments.

- v. Developing a version of the system for different languages: The current system was developed for the English language. Future work could involve developing versions of the system for other languages to cater for individuals with hearing impairments in different regions of the world. This would involve collecting and annotating datasets of sounds in different languages and training the system on these datasets.



References

- Akbal, E. (2020). An Automated Environmental Sound Classification Methods Based on Statistical and Textural Feature. *Applied Acoustics*, 167, 107413. <https://doi.org/10.1016/j.apacoust.2020.107413>
- Assistive Technology Industry Association. (2015, October 11). What is AT? Assistive Technology Industry Association. <https://www.atia.org/home/at-resources/what-is-at/>
- Better Health Channel. (2017). Deafness—A range of causes—Better Health Channel. <https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/deafness-a-range-of-causes>
- Bhutkar, P., Kularia, Jadhav, S., & Jadhav, D. (2020). (PDF) Prototype Design of Alert Device for Hearing Impaired Users. https://www.researchgate.net/publication/341470558_Prototype_Design_of_Alert_Device_for_Hearing_Impaired_Users
- Bisot, V., Serizel, R., Essid, S., & Richard, G. (2016). Acoustic scene classification with matrix factorization for unsupervised feature learning. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6445–6449. <https://doi.org/10.1109/ICASSP.2016.7472918>
- Bouchrika, I. (2022, November 10). Top 10 Qualities of Good Academic Research. Research.Com. <https://research.com/research/top-10-qualities-of-good-academic-research>
- Bougie, R., Uma, S., & Wiley, J. (2020). *Research Methods For Business: A Skill Building Approach*, 8th Edition | Wiley. Wiley.Com. <https://www.wiley.com/en-us/Research+Methods+For+Business%3A+A+Skill+Building+Approach%2C+8th+Edition-p-9781119561248R120>
- Bragg, D., Huynh, N., & Ladner, R. E. (2016). A Personalizable Mobile Sound Detector App Design for Deaf and Hard-of-Hearing Users. 3–13. <https://doi.org/10.1145/2982142.2982171>
- Campbell, A., Eisenman, S., Lane, N., Miluzzo, E., & Peterson, R. (2006). People-centric Urban Sensing. Proceedings of the 2nd Annual International Workshop on Wireless Internet. <https://doi.org/10.1145/1234161.1234179>
- Carrasco, O. C. (2020, February 21). Gaussian Mixture Models Explained. Medium. <https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95>

- Collins, J., & Hussey, R. (2014). Business Research—Collis J & Hussey R PDF | PDF. Scribd. <https://www.scribd.com/document/408873569/1-Business-Research-Collis-J-Hussey-R-pdf>
- Concas, G., Di Francesco, M., Marchesi, M., Quaresima, R., & Pinna, S. (2008). An Agile Development Process and Its Assessment Using Quantitative Object-Oriented Metrics. In P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan, & X. Wang (Eds.), *Agile Processes in Software Engineering and Extreme Programming* (Vol. 9, pp. 83–93). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-68255-4_9
- Dave, N. (2013). Feature extraction methods LPC, PLP and MFCC in speech recognition. *International Journal For Advance Research in Engineering And Technology*(ISSN 2320-6802), Volume 1.
- Elgabry, O. (2021, December 26). Object-Oriented Analysis And Design—Introduction (Part 1). OmarElgabry's Blog. <https://medium.com/omarelgabrys-blog/object-oriented-analysis-and-design-introduction-part-1-a93b0ca69d36>
- Fanzeres, L. S., Vivacqua, A. S., & Biscainho, L. W. P. (2018, October). (PDF) Mobile Sound Recognition for the Deaf and Hard of Hearing. https://www.researchgate.net/publication/328445990_Mobile_Sound_Recognition_for_the_Deaf_and_Hard_of_Hearing
- Felman, A. (2018, June 27). Deafness and hearing loss: Causes, symptoms, and treatments. <https://www.medicalnewstoday.com/articles/249285>
- Femmehub. (2022, September 20). Assistive Technology For The Deaf As We Approach International Day of Sign Language—Femme Hub. <https://www.femmehub.com/2022/09/20/assistive-technology-for-the-deaf-as-we-approach-international-day-of-sign-language/>, <https://www.femmehub.com/2022/09/20/assistive-technology-for-the-deaf-as-we-approach-international-day-of-sign-language/>
- Garg, S., Deshmukh, C. P., Singh, M. M., Borle, A., & Wilson, B. S. (2021). Challenges of the Deaf and Hearing Impaired in the Masked World of COVID-19. *Indian Journal of Community Medicine : Official Publication of Indian Association of Preventive & Social Medicine*, 46(1), 11–14. https://doi.org/10.4103/ijcm.IJCM_581_20
- Gaurav, C. (2018, October 8). All about Naive Bayes. A simple yet in depth experience of... | by

Gaurav Chauhan | Towards Data Science. <https://towardsdatascience.com/all-about-naive-bayes-8e13cef044cf>

- Giannakopoulos, T., & Pikrakis, A. (2014). *Introduction to Audio Analysis: A MATLAB® Approach*. Academic Press.
- Goldman, J., Shilton, K., Burke, J., Estrin, D., Hansen, M., Nithya, R. A., Reddy, S., Samanta, V., Srivastava, M., & West, R. (2009). *Participatory Sensing: A Citizen-Powered Approach to Illuminating the Patterns that Shape our World*. Resources.
- Grama, L., & Rusu, C. (2017). Audio signal classification using Linear Predictive Coding and Random Forests. *2017 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 1–9. <https://doi.org/10.1109/SPED.2017.7990431>
- GuhaThakurta, S., & Chetty, P. (2015, June 25). *Understanding research philosophy*. Knowledge Tank. <https://www.projectguru.in/research-philosophy/>
- Hara, S., Kobayashi, S., & Abe, M. (2016). Sound collection systems using a crowdsourcing approach to construct sound map based on subjective evaluation. *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 1–6. <https://doi.org/10.1109/ICMEW.2016.7574694>
- Harrison, O. (2019, July 14). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Medium. <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
- Health Hearing. (2021). *What is a hearing aid telecoil and why would I want one?* Healthy Hearing. <https://www.healthyhearing.com/report/45927-Hearing-aids-in-loop>
- Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., & Wilson, K. (2017). CNN architectures for large-scale audio classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 131–135. <https://doi.org/10.1109/ICASSP.2017.7952132>
- IEEE Xplore Full-Text PDF: (n.d.). Retrieved October 22, 2022, from <https://ieeexplore.ieee.org.ezproxy.library.strathmore.edu/stamp/stamp.jsp?tp=&arnumber=9012215>
- Jain, D., Findlater, L., Gilkeson, J., Holland, B., Duraiswami, R., Zotkin, D., Vogler, C., & Froehlich, J. E. (2015). *Head-Mounted Display Visualizations to Support Sound Awareness*

- for the Deaf and Hard of Hearing. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 241–250. <https://doi.org/10.1145/2702123.2702393>
- Jansen, D. (2020a, June 15). What Is Research Methodology? Definition + Examples. Grad Coach. <https://gradcoach.com/what-is-research-methodology/>
- Jasim, W., Jasim, N., Abdual, S., Saddam, S., Jasem, E., & Harfash, J. (2022). Wind Sounds Classification Using Different Audio Feature Extraction Techniques. *Informatica*, 45. <https://doi.org/10.31449/inf.v45i7.3739>
- Khan, W. Z., Xiang, Y., Aalsalem, M. Y., & Arshad, Q. (2013a). Mobile Phone Sensing Systems: A Survey. *IEEE Communications Surveys & Tutorials*, 15(1), 402–427. <https://doi.org/10.1109/SURV.2012.031412.00077>
- Koh, C.-Y., Chang, J.-Y., Tai, C.-L., Huang, D.-Y., Hsieh, H.-H., & Liu, Y.-W. (2019). Bird Sound Classification using Convolutional Neural Networks. 10.
- Komatsu, T., Tani, M., Toizumi, T., Chaitanya, N., Kato, M., Arai, Y., Hoshuyama, O., Senda, Y., & Kondo, R. (2017). An acoustic monitoring system and its field trials. 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 1341–1346. <https://doi.org/10.1109/APSIPA.2017.8282240>
- Kuper, A., Lingard, L., & Levinson, W. (2008). Critically appraising qualitative research. *BMJ*, 337(aug07 3), a1035–a1035. <https://doi.org/10.1136/bmj.a1035>
- Lincoln, Y. S., & Guba, E. G. (1986). But is it rigorous? Trustworthiness and authenticity in naturalistic evaluation. *New Directions for Program Evaluation*, 1986(30), 73–84. <https://doi.org/10.1002/ev.1427>
- Madhavan, S., Ahmed, S., Rao, V., & March 2021, A. J. P. 11. (2021, March 11). Compare deep learning frameworks. IBM Developer. <https://developer.ibm.com/articles/compare-deep-learning-frameworks/>
- Malfante, M., Mohammed, O., Gervaise, C., Dalla Mura, M., & Mars, J. I. (2018). Use of Deep Features for the Automatic Classification of Fish Sounds. 2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO), 1–5. <https://doi.org/10.1109/OCEANSKOB.2018.8559276>
- Marcell, M. M., Borella, D., Greene, M., Kerr, E., & Rogers, S. (2000). Confrontation N-g of Environmental Sounds. *Journal of Clinical and Experimental Neuropsychology*, 22(6), 830–864. <https://doi.org/10.1076/jcen.22.6.830.949>
- Mesaros, A., Diment, A., Elizalde, B., Heittola, T., Vincent, E., Raj, B., & Virtanen, T. (2019).

- Sound Event Detection in the DCASE 2017 Challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(6), 992–1006. <https://doi.org/10.1109/TASLP.2019.2907016>
- Mielke, M., & Bruck, R. (2016). AUDIS wear: A smartwatch based assistive device for ubiquitous awareness of environmental sounds. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 5343–5347. <https://doi.org/10.1109/EMBC.2016.7591934>
- Mielke, M., Grunewald, A., & Bruck, R. (2013). An assistive technology for hearing-impaired persons: Analysis, requirements and architecture. 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 4702–4705. <https://doi.org/10.1109/EMBC.2013.6610597>
- Miyazaki, K., Toda, T., Hayashi, T., & Takeda, K. (2019). Environmental sound processing and its applications. *IEEJ Transactions on Electrical and Electronic Engineering*, 14(3), 340–351. <https://doi.org/10.1002/tee.22868>
- Monette, D. R., Sullivan, T. J., & DeJong, C. R. (2013). *Applied Social Research: A Tool for the Human Services*. Cengage Learning.
- Nanni, L., Costa, Y. M. G., Aguiar, R. L., Mangolin, R. B., Brahmam, S., & Silla, C. N. (2020). Ensemble of convolutional neural networks to improve animal audio classification. *EURASIP Journal on Audio, Speech, and Music Processing*, 2020(1), 8. <https://doi.org/10.1186/s13636-020-00175-3>
- National Institute of Deafness and other communication disorders. (2021). Cochlear Implants. NIDCD. <https://www.nidcd.nih.gov/health/cochlear-implants>
- Otieno, L. (2020). Sign-IO: Smart glove invented by a Kenyan youth turns sign language into audio speech. <https://innovationagency.go.ke/blog/sign-io-smart-glove-invented-by-a-kenyan-youth-turns-sign-language-into-audio-speech>
- Pearson, M. L., Albon, S. P., & Hubball, H. (2015). Case Study Methodology: Flexibility, Rigour, and Ethical Considerations for the Scholarship of Teaching and Learning. *The Canadian Journal for the Scholarship of Teaching and Learning*, 6(3). <https://doi.org/10.5206/cjsotl-rcacea.2015.3.12>
- Phan, H., Maaß, M., Mazur, R., & Mertins, A. (2015). Random regression forests for acoustic event detection and classification. *IEEE/ACM Transactions on Audio, Speech and Language*

- Processing, 23(1), 20–31. <https://doi.org/10.1109/TASLP.2014.2367814>
- 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>
- Polonsky, M., & Waller, D. (2019). Designing and Managing a Research Project: A Business Student's Guide. SAGE Publications, Inc. <https://doi.org/10.4135/9781544316499>
- Sargeant, J. (2012). Qualitative Research Part II: Participants, Analysis, and Quality Assurance. Journal of Graduate Medical Education, 4(1), 1–3. <https://doi.org/10.4300/JGME-D-11-00307.1>
- Saunders, Lewis, M., & Thornhill, A. (2016). Research Methods For Business Students | PDF | Copyright | Quantitative Research. Scribd. <https://www.scribd.com/document/531311864/Research-Methods-for-Business-Students-by-Mark-N-K-Saunders-Philip-Lewis-Adrian-Thornhill>
- Sicong, L., Zimu, Z., Junzhao, D., Longfei, S., Han, J., & Wang, X. (2017). UbiEar: Bringing Location-independent Sound Awareness to the Hard-of-hearing People with Smartphones. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(2), 1–21. <https://doi.org/10.1145/3090082>
- Soni, D. (2019, July 16). Introduction to Bayesian Networks. Medium. <https://towardsdatascience.com/introduction-to-bayesian-networks-81031eed94e>
- Tariq, Z., Shah, S. K., & Lee, Y. (2018). Smart 311 Request System with Automatic Noise Detection for Safe Neighborhood. 2018 IEEE International Smart Cities Conference (ISC2), 1–8. <https://doi.org/10.1109/ISC2.2018.8656773>
- Thattamparambil, Dr. N. (2020, February 17). How to choose the research methodology best suited for your study. Editage Insights. <https://www.editage.com/insights/how-to-choose-the-research-methodology-best-suited-for-your-study>
- World Health Organization, W. H. O. (2021). Deafness and hearing loss. <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>
- Worldometer. (2022). World Population Clock: 7.98 Billion People (2022) - Worldometer. <https://www.worldometers.info/world-population/>
- Zhang, J. (2021). Music Feature Extraction and Classification Algorithm Based on Deep Learning. Scientific Programming, 2021, e1651560. <https://doi.org/10.1155/2021/1651560>

Zhao, S., Heittola, T., & Virtanen, T. (2020). Active Learning for Sound Event Detection
(arXiv:2002.05033; Version 1). arXiv. <http://arxiv.org/abs/2002.05033>



Appendices

Appendix A: Similarity Report

A Sound Classification and Display Tool for Assisting the Deaf and Hard-of-Hearing: A Case of Kenya.

Rosemary Wangari Wanjiru
138771

Match Overview

18%

1	su-plus.strathmore.edu Internet Source	3%
2	export.arxiv.org Internet Source	1%
3	makeabilitylab.cs.wash... Internet Source	1%
4	faculty.uobasrah.edu.iq Internet Source	1%
5	dokumen.pub Internet Source	<1%
6	erepository.uonbi.ac.ke Internet Source	<1%

VT OMNES VNVM SINT

Appendix B: Ethical Clearance Confirmation



17th March 2023

Ms Wanjiru RoseMary Wangari,
rosemary.wanjiru@strathmore.edu

Dear Ms Wanjiru,

RE: A Sound Classification and Display Tool for Assisting the Deaf and Hard-of-Hearing: A Case of Kenya

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU- master's** research proposal. Your application reference number is **SU-ISERC1587/23**. The approval period is from **17th March 2023 to 16th March 2024**.

This approval is subject to compliance with the following requirements:

- i. Only approved documents including (informed consents, study instruments, MTA) will be used
- ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
- iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 48 hours of notification
- iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 48 hours
- v. Clearance for export of biological specimens must be obtained from relevant institutions.
- vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
- vii. Submission of an executive summary report within 90 days upon completion of the study to SU-ISERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) <https://research-portal.nacosti.go.ke/> and obtain other clearances needed.

Yours sincerely,

for: **Dr Ben Ngoye,**
Secretary; SU-ISERC

Cc: Mr Ambrose Rachier,
Chairperson; SU-ISERC



Appendix C: Urban 8K Dataset License

Creative Commons Attribution- NonCommercial 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-NonCommercial 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

<https://creativecommons.org/licenses/by-nc/4.0/>

