EUPHONY

User Adaptive Systems

Supervised By: Dr. Mohammad Abu Shariah

Name	ID
Maha Hayajneh	0206250
Mira Melhem	0204444
Mirna Abu Dhaim	0208285
Rosol Shrairah	0203085
Wasan Hawari	0205446

Contents

1.0 Introduction

Problem

Most headphones today are designed to block out ambient noise to allow users to immerse themselves fully in their audio content. However, completely blocking out external sounds can pose significant safety risks, especially when users are outdoors or need to be aware of specific sounds like alarms or a baby's cry. The inability of current headphones to selectively manage noise exposure creates a gap in the market that needs to be addressed.

Objective

Euphony aims to fill this gap by developing a tool that enables adaptable sound recognition on existing headphones. Utilizing environmental sensors and machine learning algorithms, the tool will provide context-aware audio management to ensure user safety and enhance the listening experience. The tool will allow users to customize sound recognition to their specific needs, making it versatile and user-friendly.

Outline

This document outlines the Euphony project, starting with a need analysis and description, followed by a discussion of project constraints, the system environment, and the software and hardware requirements. The subsequent sections will detail the design and implementation strategies, testing and validation procedures, and finally, the deployment for Euphony.

1.1) Need Analysis and Description

Need Analysis

There is a critical need for adaptive audio technology that can balance the immersion provided by noise cancellation with the necessity of remaining alert to important environmental sounds. Traditional headphones lack the capability to dynamically manage noise exposure based on the user's surroundings and specific auditory requirements. This is particularly important for individuals like mothers of young children who need to focus on their tasks while staying alert to their child's needs and potential emergencies.

Description

Euphony will utilize advanced machine learning techniques and environmental sensors to develop a context-aware audio management tool. The system will recognize sounds of baby cries. This solution will be integrated with existing headphone technology, ensuring seamless user experience without the need for new hardware.

1.2) Project Constraints

Technical Constraints

- **Hardware Compatibility:** Ensuring the tool is compatible with a wide range of existing headphones and environmental sensors.
- **Real-time Processing:** Achieving real-time sound recognition to provide immediate feedback and safety alerts.
- **Battery Life:** Managing power consumption efficiently to avoid rapid battery drain on mobile devices.

Data Constraints

- **Dataset Availability:** Limited availability of specific datasets for sound categories like baby crying, choking, and fire alarms.
- **Data Quality:** Ensuring high-quality, annotated datasets for training machine learning models to improve accuracy and reliability.

User Constraints

- **User Customization:** Providing an intuitive interface for users to customize sound recognition settings without requiring technical expertise.

- **User Privacy:** Protecting user data and ensuring privacy, especially concerning audio recordings.

1.3) System Environment

Hardware Environment

- **Environmental Sensors:** Microphones and other sensors to capture ambient sounds.
- **Mobile Devices:** Smartphones or tablets to run the application and process audio data.
- **Headphones:** Existing headphones used by the end-users.

Software Environment

- **Operating Systems:** Compatibility with major mobile operating systems like Android and iOS.

1.4 Project Software and Hardware Requirements

Software Requirements

- **PortAudio Library:** For cross-platform audio playback and recording.
- **Librosa and pyAudioProcessing Libraries:** For audio processing and feature extraction.
- **Pycaw Library:** For controlling the volume of the device.
- **Deep Learning Frameworks:** TensorFlow for developing and training the sound classification models.

Hardware Requirements

- **Smartphones/Tablets:** To run the Euphony application.
- **Headphones:** Compatible with the tool for sound recognition.
- **Microphones:** High-quality microphones for accurate sound capture and recognition.

2.0 Research Background and Related Works

2.1 Comparison of Semi-supervised deep learning algorithms for audio classification the article by Léo Cances, Etienne Labbé, and Thomas Pellegrini [1]: explores the application of semi-supervised learning (SSL) methods in audio classification tasks. They compare five recent SSL methods and evaluate them on three benchmark audio datasets. The insights gained from this study provide valuable enhancements to our project, Euphony. Specifically, understanding SSL methods can enhance our project's sound recognition component, improve accuracy with limited labeled data, and adapt to real-world scenarios by recognizing critical sounds.

2.2 Sound Event Triage: Detecting Sound Events Considering Priority of Classes [2]**:** The introduction of Sound Event Triage (SET) offers a framework for prioritizing sound events based on their significance. This method aligns closely with our project's aim of recognizing specific sounds important to the user. By incorporating the class-weighted training method proposed in the article, we can significantly enhance our tool's performance by giving greater emphasis to high-priority sounds, ultimately improving the accuracy and reliability of sound detection.

2.3 Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review [3]**:** The comprehensive review of data augmentation and deep learning methods in sound classification provides valuable insights for structuring our dataset and preprocessing data for optimal utilization by deep learning models. While our project is based on a novel idea, we can adapt methodologies outlined in this paper to achieve the highest possible accuracy and address challenges posed by the scarcity of labelled data.

2.4 Supervised Attention Multi-Scale Temporal Convolutional Network for monaural speech enhancement [4]: The proposed Supervised Attention Multi-Scale Temporal Convolutional Network (SA-MSTCN) offers robust speech enhancement, crucial for scenarios where background noise interferes with the clarity of the baby's cry. SA-MSTCN introduces techniques like Complex Compressed Spectrum (CCS) and Complex Ratio Masking (CRM) to adaptively adjust headphone volume, emphasizing the baby's cry while minimizing noise interference.

2.5 Transferable Latent of CNN-Based Selective [5]: The article proposes a CNN-based approach for filter selection in active noise control systems. By utilizing techniques discussed in the article, our project can enhance adaptive noise cancellation headphones tailored for mothers, ensuring they remain attuned to their children's needs while enjoying a personalized audio experience.

3.0 Software Requirements Document

In this chapter, we will outline the software requirements for Euphony, including the targeted users, requirements gathering techniques, functional requirements, usability and user experience goals, design principles, and user interface design.

3.1 Targeted Users

Description: Euphony is specifically designed for parents, especially mothers, who use headphones while caring for their infants and toddlers. These parents face unique challenges in balancing their need for immersion in audio content with the necessity of staying alert to their children's needs and important environmental sounds.

Rationale: Mothers often rely on headphones to create a focused environment for themselves while performing household tasks or caring for their children. However, they also need to remain vigilant to sounds like their baby crying or emergency alerts. By targeting this demographic, we aim to address their specific needs and provide a solution that enhances their listening experience while ensuring the safety and well-being of their children. Gathering feedback from this target audience will enable us to tailor Euphony to their preferences and ensure its effectiveness in meeting their needs.

3.2 Requirements Gathering and Customer Feedback Techniques

Description: To gather requirements and feedback from parents, especially mothers, who are our primary target users, we will employ various techniques tailored to their preferences and schedules.

- **Questionnaire:** We will utilize Google Forms to create a questionnaire that will be accessible to all targeted users. The questionnaire will cover the scope and dimensions of the Euphony project, addressing key aspects related to sound recognition, adaptability, and user interface preferences. The questions will be designed to gather insights into parents' experiences, needs, and expectations when using headphones while caring for their children. (Refer to Appendix A for the questionnaire.)

Rationale: The questionnaire provides a structured approach to gather feedback from a wide range of parents, allowing us to understand their diverse perspectives and requirements. By leveraging Google Forms, we ensure ease of access and participation, enabling parents to provide valuable insights that will inform the development of Euphony.

3.3 Functional Requirements

Description: The functional requirements of Euphony for parents include:

1. **Sign In / Log In:** the user will sign up for the first time, log in after creating personal account.
2. **Sound Detection:** The system must detect specific sound, such as baby cries.

3. **Real-Time Processing:** The system must process audio input in real-time to provide immediate feedback, by reduce the volume of the device, and sending a notification.
4. **Adjust the Settings:** Users must be able to customize the sound recognition settings for the preferred time of detection and volume level.
5. **Adaptive Volume Reduction:** The system must adjust the device volume levels based on the detection of baby cries.
6. **Alert Notifications:** The system provides visual and/or haptic notifications on the mobile device when baby cry sounds is detected.

Rationale: These functional requirements are essential for providing parents with a reliable and user-friendly solution that enhances their ability to listen to audio content while caring for their children.

3.4 Usability and User Experience Goals

Description: Our goals for Euphony include:

Usability Goals:

- **Efficiency:** Euphony should facilitate quick and efficient access to its features, allowing parents to adjust settings and interact with the app without unnecessary delays.

- **Effectiveness:** Euphony should accurately recognize and adapt to environmental sounds, ensuring that parents receive timely alerts and notifications when necessary.

- **Ease of Use:** Euphony's interface should be intuitive and user-friendly, enabling parents to navigate the app effortlessly and customize settings with minimal effort.

User Experience Goals:

- **Satisfying:** Euphony should provide a satisfying listening experience for parents, allowing them to enjoy audio content while remaining attentive to their child's needs.
- **Enjoyable:** Euphony should offer an enjoyable and pleasurable user experience, enhancing parents' overall satisfaction and enjoyment of the app.
- **Entertaining:** Euphony should provide entertaining features and content that captivate parents' attention and make their listening experience more enjoyable.
- **Helpful:** Euphony should be a helpful tool for parents, providing valuable assistance and support in managing their listening preferences and staying aware of their surroundings.
- **Enhancing Sociability:** Euphony should facilitate social interaction and connection among parents and their children, fostering a sense of community and support.
- **Engaging:** Euphony should engage parents actively, encouraging regular usage and interaction with the app.
- **Experiencing Flow:** Euphony should offer a seamless and immersive user experience, allowing parents to enter a state of flow where they are fully absorbed in their listening activities without distraction.

Rationale: Achieving these usability and user experience goals will enhance parents' satisfaction and trust in Euphony, ultimately leading to increased adoption and usage.

3.5 Design Principles

Overview: Design principles are essential guidelines that guide interaction designers in creating an optimal user experience. They provide a framework for decision-making and help orient designers towards considering different aspects of their designs. For Euphony, the following design principles are considered to ensure an effective and user-friendly interface:

Euphony’s Design Principles:

1. **Visibility:** Ensure that important elements and features of Euphony are clearly visible and easily accessible to users. This includes providing clear labels, icons, and navigation paths to guide users through the app.

2. **Feedback:** Provide adequate feedback to users to inform them about their actions and the system's response. Feedback mechanisms should be timely, informative, and help users understand the outcome of their interactions with Euphony.

3. **Constraints:** Implement constraints to guide users towards desired actions and prevent errors or unintended interactions. Constraints help maintain consistency and coherence in the user interface, enhancing usability and reducing cognitive load.

4. **Consistency:** Maintain consistency across Euphony's interface in terms of layout, terminology, and interaction patterns. Consistency fosters familiarity and predictability, allowing users to navigate the app more efficiently and confidently.

5. **Affordance:** Design elements in Euphony to visually and functionally suggest their purpose or potential actions to users. Affordances help users understand how to interact with different elements and features intuitively, facilitating ease of use and learnability.

Application to Euphony: These design principles will guide the development of Euphony to ensure that it provides a seamless and intuitive user experience for parents. By adhering to these principles, Euphony will offer clear visibility, informative feedback, effective constraints, consistent interactions, and intuitive affordances, enhancing usability and satisfaction for its users.

3.6 User Interface Design

Description: The user interface design of Euphony will be tailored to the preferences and needs of parents, with a focus on simplicity, clarity, and accessibility. Key considerations include:

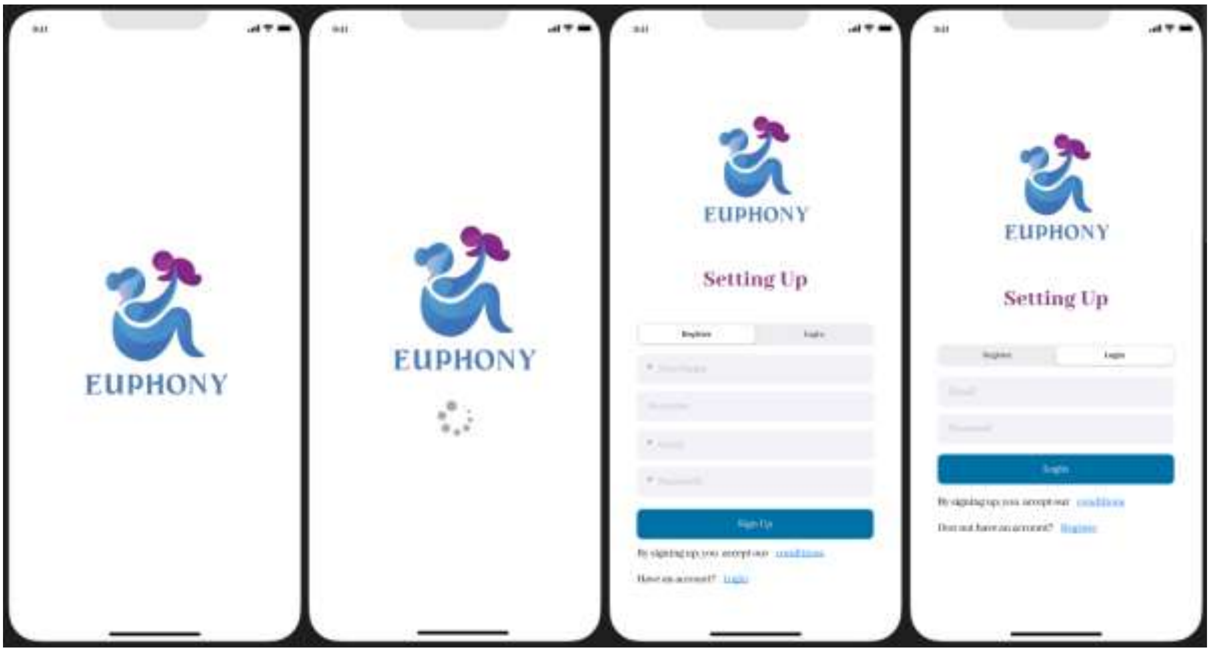


Figure 1: Euphony UI Setting Up

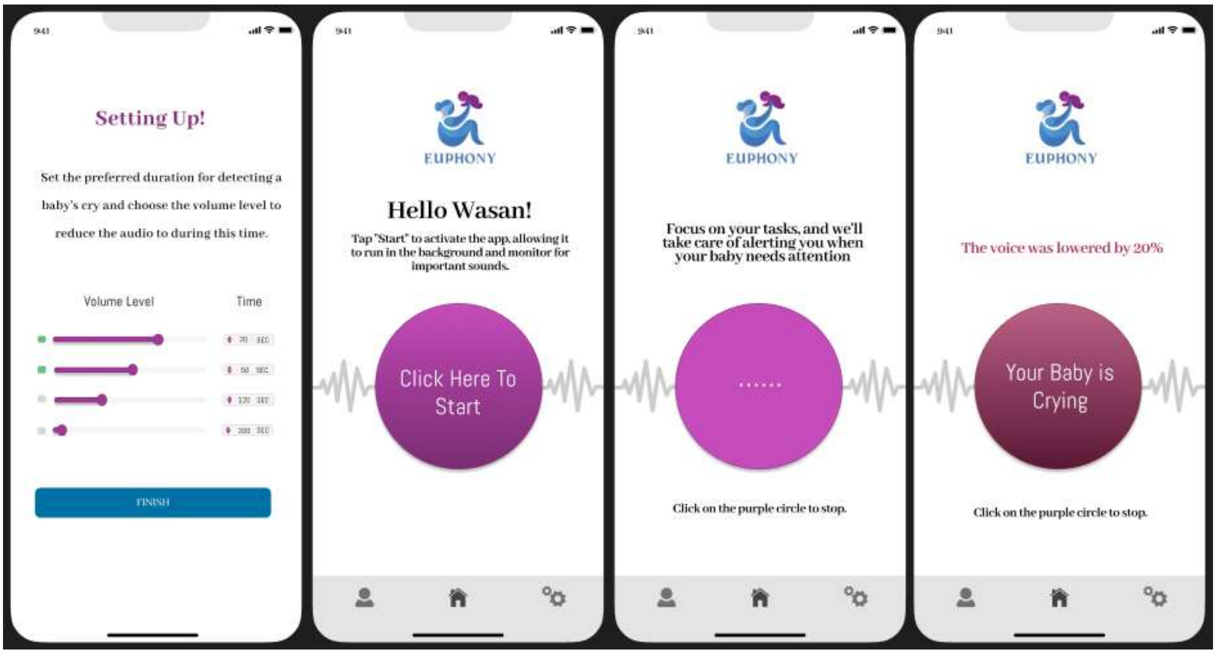


Figure 2: Home Page

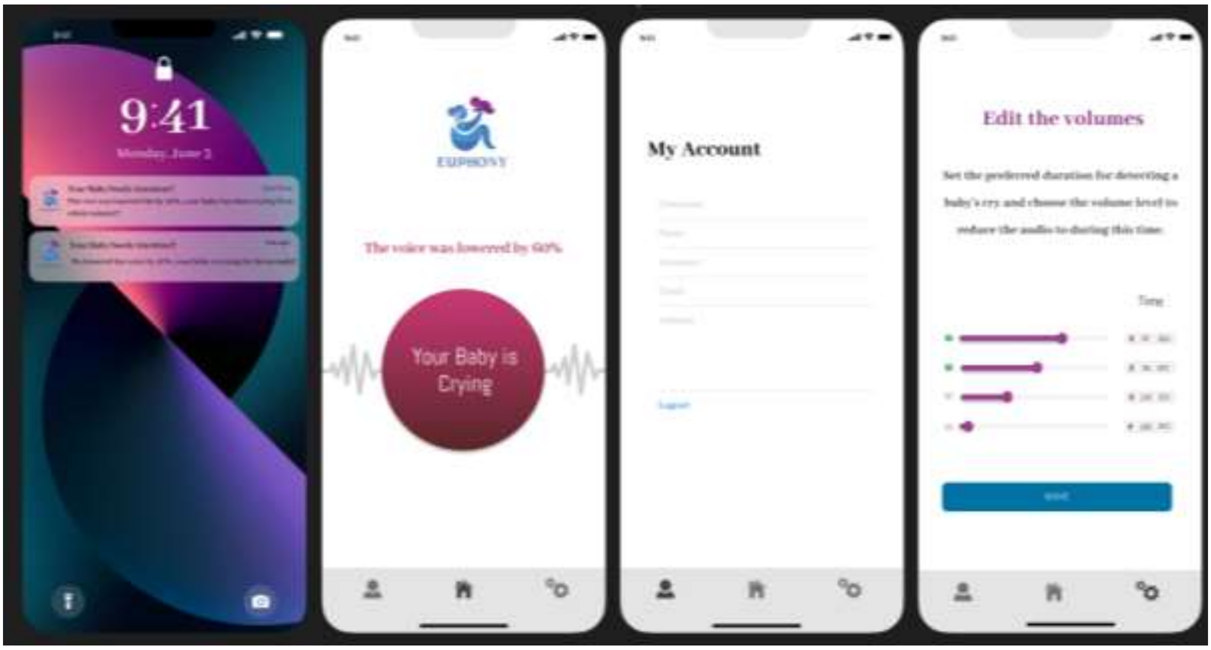
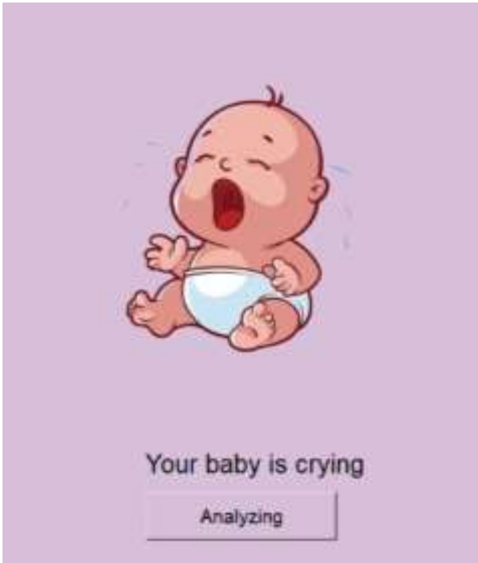


Figure 3: Notify/ Account/ Settings

Rationale: A well-designed user interface tailored to the needs of parents will enhance their satisfaction and usability of Euphony, facilitating a more enjoyable and effective listening experience while caring for their children.

3.7 Demo Implementation:

(a) Baby is not crying.



(b) Baby is crying.

Figure 4: Demo Interfaces

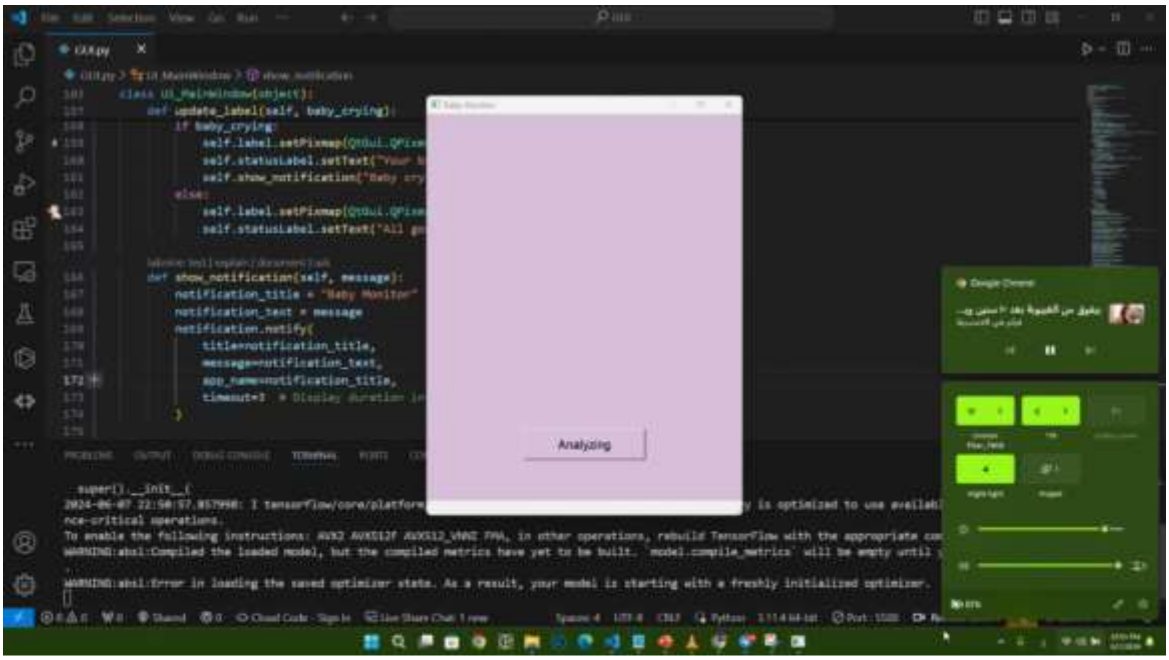


Figure 5: Sound is 100%

Figure 6; Baby Cry Detected for 5 Seconds

Figure 7: Baby Cry Detected for 10 Seconds

Figure 8: Baby Cry Detected for 15 Seconds

When implementing volume adjustments based on baby cry detection, we consider three distinct scenarios:

1) Volume Reduction (3-Second Cry):

   If a baby cry is detected for a duration of 5 seconds, we decrease the volume by 30%. This gradual reduction ensures a less disruptive adjustment while addressing the crying event see figure 6.

2) Enhanced Reduction (10-Second Cry):

   For a more prolonged cry lasting 10 seconds, we implement a 70% volume reduction. This greater adjustment aims to mitigate the impact of extended crying periods see figure 7.

3) Complete Muting (15-Second Cry):

   In cases where a baby cry persists for 15 seconds or longer, we take a more decisive approach by muting the volume entirely. This immediate action minimizes disturbances and provides a quiet environment see figure 8.

By tailoring the volume adjustments to the cry duration, we create a responsive and supportive audio environment for both the baby and caregivers.

4.0 System Implementation

4.1 Pipeline of the Proposed Methodology

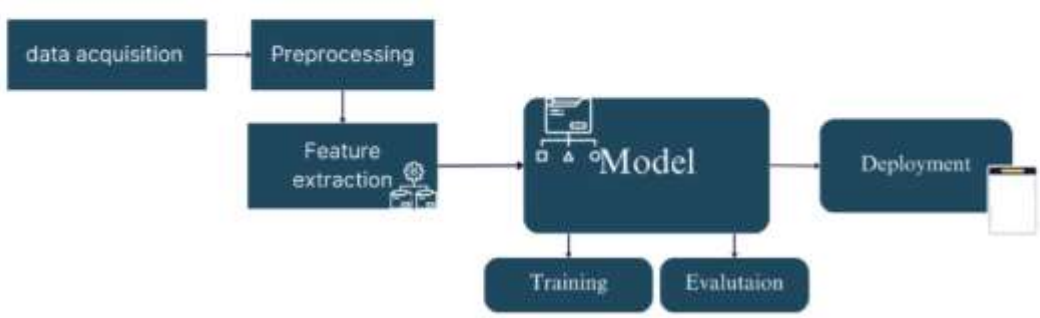


Figure 9:Pipeline of the Methodology

Our pipeline provides a comprehensive approach to the problem of classifying audio files into ‘crying baby’ and ‘others’ categories. It includes all steps from data acquisition and preprocessing to feature extraction to model training, evaluation, and prediction and finally deployment. The use of audio augmentation and noise reduction techniques in the preprocessing stage improved the model’s performance by providing it with a more diverse and cleaner dataset to learn from. The use of a CNN for the classification task leverages the power of deep learning to extract relevant features from the audio files and make accurate predictions.

4.2 Technical and Implementation Description

The technical description of the proposed methodology involves a series of steps starting from data acquisition to prediction. The audio files are then pre-processed, which includes noise reduction and audio augmentation using a Pedalboard. After preprocessing features are extracted from the audio files using Mel-frequency cepstral coefficients (MFCCs). The data is then loaded, and the labels for the ‘crying baby’ class are set to 1, and for the ‘others’ class, is set to 0. A Convolutional Neural Network (CNN) model is defined, compiled, and trained on the training data. The model’s performance is evaluated on the validation data using accuracy, loss, mean squared error (MSE), and precision as metrics. Finally, the model predicts the class of audio files in a test directory, outputting whether it predicts the audio file to be a baby crying or not.

4.3 Data About User

Our dataset comprises audio files from two categories: 'crying baby' and 'others'. The 'crying baby' data, consisting of 301audios which is sourced from a GitHub repository called 'Baby Cry Detection'[6], additionally we combined this dataset with the 'Donate-a-Cry Corpus' from GitHub [7] which consists of 752 audios. This brings our total dataset to approximately 1200 audios for the 'crying baby' data. The 'others' data, which encompasses all sounds excluding crying babies, is derived from the UrbanSound8K dataset on Kaggle [8] with 1600 audio files. These audio files undergo a series of preprocessing steps. Initially, the audio files are loaded, followed by noise reduction using the noise reduce library to minimize background noise. Subsequently, audio augmentation is performed using a Pedalboard, which manipulates the audio signal to create augmented versions of the original files, thereby enhancing the diversity of the training data. Finally, the data is loaded, and labels are assigned: '1' for the 'crying baby' class and '0' for the 'others' class. This preprocessing pipeline is flexible and modular, allowing for potential improvements and expansions based on task requirements and available resources.

4.4 Features Extraction

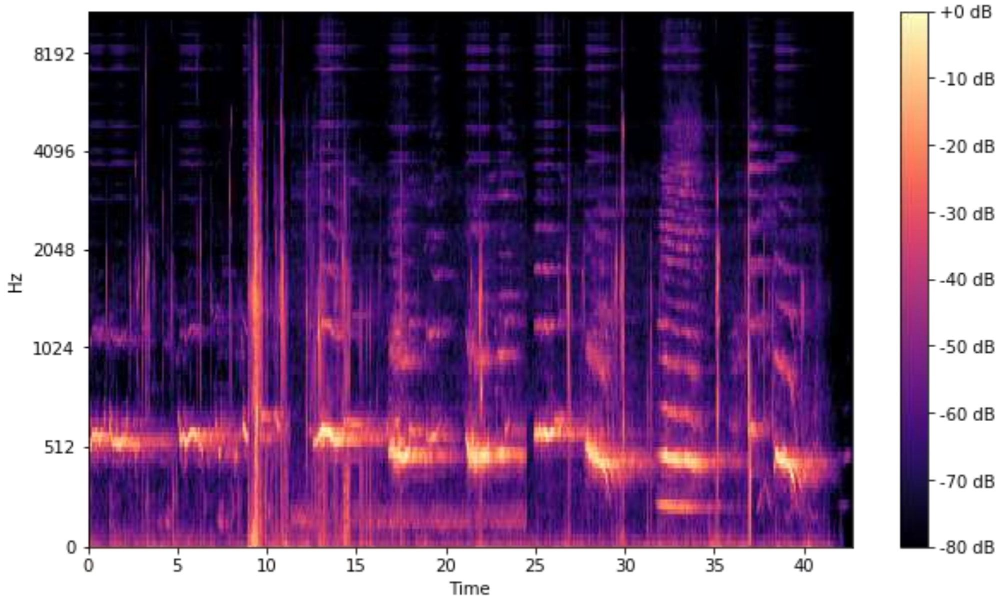


Figure 10:MFCC Method

Mel-frequency cepstral coefficients (MFCCs) was used in this project, MFCCs are extracted from each audio file using the librosa library, Figure 5 shows the MFCC of the audios, to provide a compact representation of the power spectrum of an audio signal, in essence, it transforms the audio signal into a set of coefficients that can be used for our binary classification task. To ensure uniformity in the input data regarding collecting the data from different resources, all audio files are made to have the same length so if an audio file is shorter than the maximum length, it is padded with zeros. Conversely, if it’s longer, it is truncated. This process ensures that all audio files, regardless of their original length, are transformed into a consistent format that can be fed into the machine learning model. This step is vital as it transforms raw audio files into a structured format, enabling the model to learn effectively from the data.

4.5 User Modelling

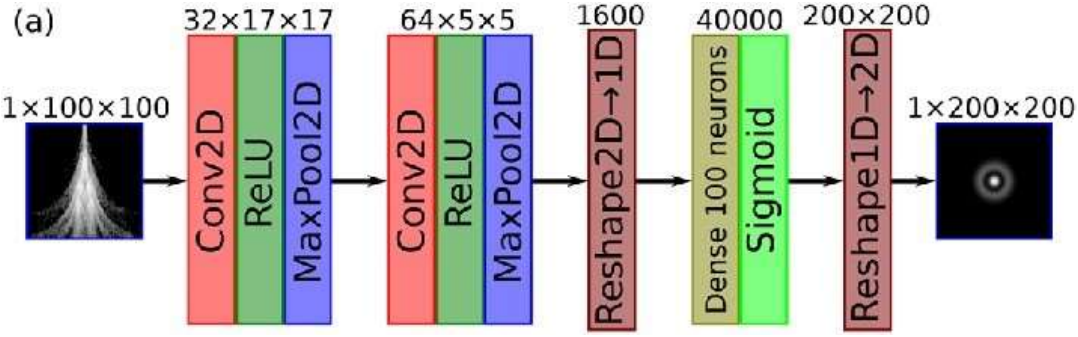


Figure 11: CNN Architecture [9]

User modeling in this project involves the application of deep learning techniques for the classification of audio files into a crying baby or anything else. The primary deep learning model used is a Convolutional Neural Network (CNN), The CNN model is defined with several layers, including Conv2D, MaxPooling2D, Flatten, and Dense layers. It uses activation functions suitable for feature extraction and which are ReLU and Sigmoid for binary classification for the last layer. The Dense layers are fully connected, meaning each neuron is linked to every neuron in the preceding layer. The model is compiled with the Adam optimizer and binary cross-entropy loss function, which are commonly used for binary classification tasks. The model is trained on the training data, and its performance is evaluated on the validation data using accuracy, mean squared error (MSE), and precision as metrics. This user modeling approach leverages the power of deep learning to classify audio files effectively.

For the compilation, we used a batch size of 32 and set the steps per epoch to 30. The model was trained for a total of 50 epochs. These numbers gave the optimal results after conducting trial and error method.

Table 1: Data selection for Training

Data selection for Training, Validation, and Testing	Value
The proportion of data for training	80%
The proportion of data for validation	20%
Optimizer	Adam
Batch Size	32
Epochs	50
Step per epoch	30

4.6 Adaptation Methods and Techniques

The adaptation methods and techniques used in this project involve a combination of audio detection, deep learning, and real-time analysis. The system uses the sounddevice library to capture the surrounding audio in real-time. This audio is then intended to be fed into a Convolutional Neural Network (CNN) model, which could be trained to distinguish between different sounds.

The system adapts its behaviour based on the output of the CNN model. If the model detects a specific sound, a baby cry in our case, the system will perform an action. For example, if a baby's cry is detected, the system will lower the volume of the audio output using Pycaw library. The degree of volume reduction is dependent on the duration of the baby's cry. If the baby cries for less than 30 seconds, the volume will be lowered by 20 percent. If the baby doesn't stop crying after 3 minutes, the system mutes the volume completely, bringing it down to zero. This approach allows the system to adapt to the situation in real-time, providing a dynamic response to the detected sound. It ensures that the audio levels are adjusted appropriately, minimizing disturbance while still allowing for necessary sounds to be heard. Subsequently, Euphony allows the user to adjust these settings according to their preferences. This makes the system highly adaptable, real-time, and reliable.

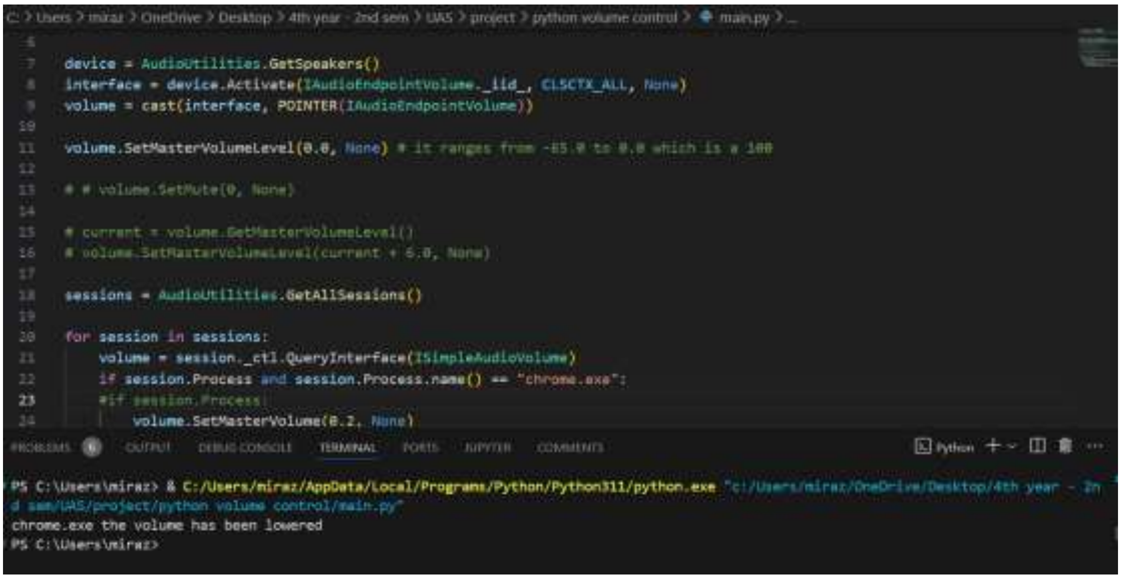


Figure 12: Adaption Code Snippet

5.0 Experimental Results and Analysis

5.1 Performance Measures

The performance of the model was evaluated using four key metrics: loss, accuracy, MSE, and precision. These metrics provide a comprehensive view of the model's performance, considering both the correctness of the predictions (accuracy and precision) and the degree to which the predictions deviate from the actual values (loss and MSE).

5.2 Experimental Results

The Convolutional Neural Network (CNN) model was trained and validated on 2800 audio files, providing it with a diverse range of data to learn from. It was then tested on 38 audio files. The results indicate that the model was able to accurately classify the majority of these files, predicting whether each audio file was a baby crying or not. This demonstrates the model's effectiveness in real-world scenarios, suggesting that it is robust, reliable, and capable of adapting to new data. However, to confirm these findings and ensure the model's generalizability, further testing on a larger and more diverse dataset would be beneficial. In the evaluation phase, the model yielded the following results:

Table 2: Validation Metrics

Validation Metrics	Value
Accuracy	0.9728
Loss Binary Cross Entropy	0.1812
MSE	0.0254
precision	0.9756

These performance measures indicate that the model performed exceptionally well on the task of classifying audio. This extensive training and validation process contributes to the model's ability to make accurate predictions, demonstrating its potential for practical applications.

5.3 Adaptation Effects

One of the key adaptation effects in our project is the dynamic volume control based on the duration of a baby's cry. The system lowers the volume by 20% if the baby cries for less than 30 seconds. If the baby doesn't stop crying after 3 minutes, the system mutes the volume completely. This shows the system's ability to adapt in real-time to the situation, providing a dynamic response to the baby's crying. Another adaptation effect is seen in the model's ability to generalize from the training data to unseen data. Despite the difference in data, the model was able to accurately classify the majority of the test audio files.

6.0 Conclusions and Future Works

6.1 Strengths

The Euphony project demonstrates several notable strengths:

- **High Accuracy:** The project achieved an impressive accuracy rate of 0.9728, indicating its reliability in recognizing and classifying relevant sounds.
- **Integrated Dataset:** By sourcing ‘crying_baby’ data from a GitHub repository and ‘others’ data from the UrbanSound8K dataset on Kaggle, Euphony benefits from a diverse and comprehensive dataset, enhancing the model's robustness and performance across various sound environments.

6.2 Weaknesses

While the project showcases significant strengths, it also faces certain limitations:

- **Limited Scope of Environmental Sound Recognition:** Currently, the model is primarily focused on recognizing baby cries and general urban sounds. It should be expanded to include critical environmental sounds such as fire alarms, breaking glass, and other urgent sounds to ensure comprehensive safety and situational awareness.
- **Dataset Dependency:** The effectiveness of the system is closely tied to the specific datasets used during training. Generalizing to other audio domains or real-world applications may require further validation and expansion of the dataset.
- **Real-world Scenario Adaptation:** While the models perform well on benchmark datasets, their performance in highly variable real-world environments with diverse and dynamic soundscapes requires further testing and optimization.

6.3 Future Works

To address the current limitations and expand the capabilities of Euphony, the following future work is proposed:

- **Expansion of Sound Recognition Capabilities:** Enhance the model to recognize a broader range of critical environmental sounds, such as fire alarms, breaking glass, and other urgent sounds, to improve overall safety and utility.
- **Dataset Enhancement:** Continue to expand and diversify the dataset by integrating more varied and representative sound samples from different sources, ensuring better generalization across different audio environments.
- **Performance Optimization:** Focus on optimizing computational efficiency to ensure real-time performance without compromising accuracy and reliability. This includes refining model architectures and exploring lightweight yet effective algorithms.
- **Broader User Base:** Expand the target user base beyond parents to include other user groups who could benefit from adaptive sound recognition, such as the elderly, individuals with hearing impairments, and professionals working in noisy environments.
- **User Feedback Integration:** Implement mechanisms for continuous user feedback to refine and improve the system's performance based on real-world usage and user experiences.

By addressing these areas, Euphony can evolve into a more versatile and comprehensive tool, enhancing its applicability and effectiveness for a broader range of users and scenarios.

6.4 Conclusion

In conclusion, Euphony represents a promising advancement in user-adaptive audio systems, leveraging cutting-edge technology to provide parents with a safer and more responsive auditory environment. By addressing its current limitations and pursuing outlined future enhancements, Euphony has the potential to become an indispensable tool for a wide range of users, ensuring critical sound detection and personalized noise management in diverse and dynamic environments.

Table of Figures

References:

[1] Cances, Léo, et al. "Comparison of Semi-Supervised Deep Learning Algorithms for Audio Classification." EURASIP Journal on Audio, Speech, and Music Processing, vol. 2022, no. 1, 19 Sept. 2022, https://doi.org/10.1186/s13636-022-00255-6.

[2] Tonami, N., & Imoto, K. (2023). Sound event triage: detecting sound events considering priority. of classes. EURASIP Journal on Audio, Speech, and Music Processing, 2023(5). https://doi.org/10.1186/s13636-022-00270-7

[3] Abayomi-Alli, O., Damaševičius, R., Qazi, A., Adedoyin-Olowe, M., & Misra, S. (2022). Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review. Electronics, 11(22), 379

[4] Zhang, Z., Zhang, L., Zhuang, X. et al. Supervised Attention Multi-Scale Temporal Convolutional Network for monaural speech enhancement. J AUDIO SPEECH MUSIC PROC. 2024, 20 (2024). https://doi.org/10.1186/s13636-024-00341-x

[5] Shi, D., Gan, W.-S., Lam, B., Luo, Z., & Shen, X. (2023). Transferable Latent of CNN-Based Selective Fixed-Filter Active Noise Control. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31, 2910–2921. https://doi.org/10.1109/taslp.2023.3261757

[6] Giulbia. (n.d.). *baby_cry_detection/data at master · giulbia/baby_cry_detection*. GitHub. https://github.com/giulbia/baby_cry_detection/tree/master/data

[7] Giulbia. (n.d.-c). *GitHub - giulbia/baby_cry_detection: Recognition of baby cry audio signal*. GitHub. https://github.com/giulbia/baby_cry_detection

[8] *UrbanSound8K*. (2020, February 4). Kaggle. https://www.kaggle.com/datasets/chrisfilo/urbansound8k?select=fold1

[9] Bukharskii, N. D., Vais, O. E., Korneev, P. A., & Bychenkov, V. Y. (2022). Restoration of the focal parameters for an extreme-power laser pulse with ponderomotively scattered proton spectra by using a neural network algorithm. *Matter and Radiation at Extremes*, *8*(1). https://doi.org/10.1063/5.0126571

Appendix A

Link for the questionnaire: https://forms.gle/wAiqs2Bysrb9koip9