AI-Driven Smart Baby Monitoring System

Thesis Submitted in Fulfillment of the Requirements for the Degree

Of

Bachelor of Technology (B.Tech)

in

Department of Computer Science and Engineering

By

Aniket Dutta (Roll No - 500121010016 , Reg. No - 211430100110166)

Aparna Shaw (Roll No - 500421020018 , Reg. No - 211430100110195)

Annesha Nandi (Roll No - 500421020012 , Reg. No - 211430100110203)

Arunangshu Majumdar (Roll No - 500121010038 , Reg. No - 211430100110117)

Harender Singh (Roll No - 500121010061 , Reg. No - 211430100110118)

Cyrus Mitra (Roll No - 500121010048 , Reg. No - 211430100110069)

Under the Guidance of

Prof. Dr. Ananjan Maiti

Assistant Professor

Guru Nanak Institute of Technology, Kolkata-700110

ACKNOWLEDGEMENTS

We express our heartfelt gratitude to Dr. Ananjan Maity, our mentor, for his invaluable guidance and encouragement throughout the development of this Software Requirements Specification (SRS) document for our final year project, “AI-Driven Smart Baby Monitoring System Using Audio.”

We also extend our thanks to the faculty and staff of Guru Nanak Institute of Technology for their support and resources, as well as to our peers for their valuable inputs and motivation.

Finally, we are deeply grateful to our families for their unwavering support and encouragement, which have been instrumental in the successful completion of this project.

Date: **Aniket Dutta**

Place:

GURU NANAK INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the thesis entitled “AI-Driven Smart Baby Monitoring System,” submitted by Syntax Radianites for the award of the Bachelor of Technology (Computer Science and Engineering) degree from GNIT, is a record of genuine research conducted under supervision and guidance. The team has worked diligently on this project for nearly one year at the Department of Computer Science and Engineering, Guru Nanak Institute of Technology, Kolkata. This work meets the standards required for the degree.

Furthermore, the contents of this thesis, either in whole or in part, have not been submitted to any other university or institution for the award of any other degree or diploma.

Supervisor

Department of Computer Science and Engineering

GNIT, Kolkata.

Head of the Department

Head of the Department

Department of Computer Science and Engineering, GNIT, Kolkata

CONTENT

1. Introduction

Crying, in particular, is the essential means of communication for babies, and therefore, the recognition of baby sounds is essential in health care needs and management. State-of-the-art technique using features derived from Mel Frequency Cepstral Coefficients with the implementation of Random Forest and XGBoost machine learning algorithms yields high classification accuracy of 94.5% and 94.2% for various types of cry signals from infants [1]. These technologies are central to building applications that enable the automated monitoring of infants to enhance caregiver responsiveness through timely intervention. Furthermore, the interfaces to IoT and cloud computing, as shown in this paper with the help of ESP8266 such Wi-Fi modules, improv ethereal-time analysis capabilities for smart caregiving; the accuracy of the CNN and RNN models, which lies in 92% and 90%, respectively, confirms the effectiveness of presented approach [2]. Systems that can recognize cries using Neural networks like Efficient Net and Res Net greatly help deaf caregivers as they decode and classify cries and help fill communication gaps[3]. Further, the use of digital signal processors (DSP) employed in nursery boxes where algorithms such as dynamic time warping (DTW) are used stresses the greater need for cry recognition, attaining an accuracy figure of 97.1% [4]. Altogether, the progress shows an incentive for baby sound recognition technologies to improve the quality of infant care and help the carers in multiple scenes [5]. These are the principal difficulties in baby sound recognition arising from the nature of the signal and conditions under which it is recorded [6]. Another critical issue is the different initiation of various Cry types, concerning hunger, discomfort, versus pain, which is seldom easy for any childcare provider who has had no prior practice in this [5]. To this end, emerging deep learning techniques, including CNNs and Vision Transformers, have been proposed to overcome the above limitations, and the accuracy rates for the classification of cry types have been significantly enhanced by using IoT sensors for monitoring in real-time [7]. However, the other factor that can be noticed as an issue of using cry recognition in a natural environment is that there will always be a certain level of background noise, which further influences the performance of this type of system [8]. It is proven that CNNs, because of their unique feature of capturing spatial features, have better noise resilience than Recurrent Neural Networks(RNNs)when identifying and estimating noise levels with high precision[9]. Furthermore, cry datasets require balanced data to classify all types of cry, as some of them can be rare. SMOTE and ENN have been applied to increase the sample size of the minor classes, while data augmentation through pitch shifting and the addition of noise have been used to bring the accuracy of other cry classes to the level of the first class [6].

2. Background and Motivation

Recent research has found that there have been incredible improvements in the field of baby sound recognition in AI by employing different deep learning methods. The purpose of such research is to identify the infant's needs based on the type of cry to enhance the care given to the infant.˙IbrahimGu¨lmez et al. applied ANN and CNN deep learning models to classify infant cries, and the authors used data augmentation and segmentation methods to improve the performance[10].In the same way, SamirA. Younisetal. Implemented the integration of IoT sensors and deep learning algorithms with an accuracy of 98.33% to classify cries regarding specific needs of an infant, like hunger or discomfort[7]. In Maiti's study area, he proposed that Mel Frequency Cepstral Coefficients(MFCC) be used for feature extraction. He used Random Forest and XGBoost machine learning models to differentiate between cry test in real-time[14]. Finally, Tusty Nadia Maghfa et al. train CNN and RNN to extract features and use them for classification, whereby they can obtain an approximate accuracy of 94.5% for the Dunstan Baby Language dataset [15]. Altogether, these works must be seen as pointing to the use of AI in deepening knowledge of, and engagement with, infant cries as a critical avenue towards improving the management of infancy [5]. III. DATASET AND PREPROCESSING accuracy rates at the level of 94% and above [1].Medhanita Dewi Renantietal. Paid attention to noise robustness and presented the EDA CNN model with a high noise tolerance that is better than RNN models[9]. The se techniques, named SMOTE and ENN, can be applied to balance data imbalance problems of Michael Indrawan et al., which gain average accuracy at cry classes with CNN models [6]. The LPC and MFCC features have been experimented with together by Zakaria Firas et al.. However, they also specified that feature extraction is a crucial element to achieve the highest accuracies in cry classification[11]. KeZhangetal. Proposed a strategy to deal with conflicts between classifiers by applying improved Dempster–Shafer evidence theory and obtained some noticeable enhancements in accuracy [12]. In the comparison of CNN and LSTM models, researchers Yun-Chia Liang et al. have proven that the models help to distinguish healthy and sick infants during the accelerometer data analysis, and although both models show promising results, CNN distinguishes in specific need identification is better[13].SanaYasin and others looked at other ways in addition to cry intensity in which babies' feelings may be detected, using machine learning to differentiate between cry tests in real-time[14]. Finally, Tusty Nadia Maghfael et al. train CNN and RNN to extract features and use them for classification, whereby they can obtain an approximate accuracy of 94.5% for the Dunstan Baby Language dataset [15]. Altogether, these works must be seen as pointing to the use of AI in deepening knowledge of, and engagement with, infant cries as a critical avenue towards improving the management of infancy [5].

3. Problem Statement

The automatic recognition and classification of infant sounds, such as crying, laughter, and discomfort, are critical for enhancing caregiver responsiveness and ensuring the health and well-being of infants. However, significant challenges hinder the effectiveness of existing classification models:

Class Imbalance: The distribution of sound types in infant audio datasets is highly skewed, with certain categories (e.g., 'cry' and 'hunger') vastly outnumbering others (e.g., 'burping' and 'belly pain'). This imbalance leads to model bias, where the system is overly tuned to the majority classes and performs poorly on underrepresented classes.

Feature Extraction and Model Complexity: Effective classification hinges on the ability to extract relevant features from audio signals. Techniques such as Mel-Frequency Cepstral Coefficients (MFCCs) are commonly used for this purpose, but complex audio signal characteristics and varying recording conditions can significantly affect feature quality and model performance.

Hyperparameter Optimization: Selecting appropriate hyperparameters is crucial for training deep learning architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The optimization process can be resource-intensive and may not consistently yield the best performance across diverse sound categories.

To tackle these challenges, this study implements a robust framework that employs the Synthetic Minority Over-sampling Technique (SMOTE) to balance the training dataset. By generating synthetic samples for the minority classes, the framework aims to mitigate the effects of class imbalance and improve classification accuracy. Furthermore, a grid search method is utilized within a deep learning pipeline to optimize hyperparameters such as learning rates, dropout rates, and the number of neurons in hidden layers.

The anticipated outcome is a deep learning model that demonstrates improved classification performance across all categories of infant sounds, thereby fostering reliable and efficient automatic monitoring systems for infant care. The successful application of these techniques is expected to pave the way for advancements in health monitoring applications, enhancing the ability of caregivers to respond appropriately to infants' needs.

4. Objectives and scope

The primary objective of this project is to develop a deep learning-based system capable of accurately classifying various baby sounds, such as cries of hunger, discomfort, burping, laughter, and more, with a strong emphasis on real-time healthcare

applicability. Recognizing these vocalizations is critical, as they serve as a fundamental mode of communication for infants and can indicate vital needs or discomfort.
This project aims to address two major challenges in baby sound classification: **class imbalance** and **model optimization**. The dataset used in this study contains an unequal distribution of sound types, which commonly causes deep learning models to overfit on the majority classes. To overcome this, the **Synthetic Minority Over-sampling Technique (SMOTE)** was employed to generate synthetic data for underrepresented sound categories, ensuring balanced training samples across all classes.
Another key focus is **hyperparameter tuning**, where the architecture's performance is optimized through systematic adjustment of parameters such as dropout rates, neuron counts, and learning rates. This tuning enables the model to achieve higher accuracy and generalization without overfitting.
The scope of this work lies in the intersection of artificial intelligence and healthcare. The system is designed with real-world deployment in mind, particularly to assist caregivers—including those with hearing impairments—by providing intelligent alerts and classifications of baby sounds. This could significantly improve responsiveness in caregiving environments and help prevent potential health risks through timely intervention and understanding of infant needs.

5. Literature review

Recent advancements in artificial intelligence have led to substantial developments in the domain of baby sound recognition, particularly using **deep learning techniques**. This section reviews significant models, features, and methodologies that have contributed to the state of the art in this area.

Recurrent Neural Networks (RNNs) have been widely utilized for their ability to process temporal sequences. In particular, LSTM and BiLSTM models have demonstrated effectiveness in recognizing the evolving patterns in baby cries. Studies such as those by Liang et al. show how RNNs can help differentiate between cries related to sickness or discomfort using sequential data modeling.

Convolutional Neural Networks (CNNs) are another popular architecture in this field. Unlike RNNs, CNNs excel at learning local patterns in audio representations such as MFCCs and spectrograms. Their spatial filtering capability makes them robust to background noise, which is essential in real-world applications. Renanti et al. proposed CNNs with high noise tolerance, outperforming RNNs in such environments.

Mel-Frequency Cepstral Coefficients (MFCCs) have emerged as the most commonly used feature extraction technique. They mimic human auditory perception and retain essential information from the raw audio signals. Studies by Maiti and others highlight MFCC’s effectiveness when combined with machine learning and deep learning models, yielding accuracy scores above 94%.

To address **data imbalance**, **SMOTE (Synthetic Minority Oversampling Technique)** has been adopted in various works. For example, Michael Indrawan et al. applied SMOTE to underrepresented cry categories, significantly improving classification performance. Combinations of SMOTE with ENN or K-RBFNN have also shown promising results in other healthcare domains.

Multiple hybrid architectures have been proposed to combine the strengths of CNNs and RNNs. Maghfira et al. trained a CNN-RNN pipeline that achieved over 94.5% accuracy on the Dunstan Baby Language dataset. Similarly, Younis et al. implemented CNNs integrated with Vision Transformers to reach an impressive 98.3% accuracy in cry classification.

Finally, feature fusion approaches and alternative optimization methods, such as improved Dempster–Shafer theory and metaheuristic-based tuning, have also been explored to enhance decision-making and accuracy. Altogether, these studies reflect a strong foundation in AI-driven baby sound classification, with the current project building upon these strategies for balanced and optimized model development.

6. Overview of Deep Learning for Audio Classification

Deep learning has become a dominant methodology for audio classification tasks due to its ability to automatically learn complex patterns from data. In the context of baby sound recognition, deep learning models such as **Convolutional Neural Networks (CNNs)** and **Recurrent Neural Networks (RNNs)** play crucial roles.

CNNs process audio signals by first converting them into visual representations such as spectrograms or MFCCs. These are treated as images where convolutional filters detect local patterns such as pitch changes, high energy bursts, and harmonic structures. In baby sound recognition, CNNs effectively identify spatial features of cries or laughs and show resilience against environmental noise. Their architecture supports weight sharing and local connectivity, which helps reduce the number of parameters and makes the model computationally efficient.

RNNs, particularly LSTM variants, are adept at modeling time-series data by maintaining memory across sequences. They are crucial for identifying how baby sounds evolve—distinguishing, for example, between a short whimper and a prolonged cry. RNNs are sensitive to the order and duration of sounds, making them suitable for analyzing complex emotional or physical expressions in an infant's voice.

Hybrid CNN-RNN models leverage the strengths of both architectures. CNN layers are used to extract rich spatial features from audio inputs, which are then passed to RNN layers to model temporal relationships. This approach has shown improved performance in many studies and is particularly effective for multi-class classification tasks involving variable-length audio signals.

In this project, the deep learning pipeline uses **MFCCs** as input features and applies CNN and RNN layers to extract and process information. The model also incorporates **SMOTE** for balancing class distribution and **hyperparameter tuning** for optimal training. This results in a robust, real-time classification system capable of assisting caregivers in understanding and responding to various infant needs, thereby contributing to improved healthcare and infant well-being.

7. Dataset and Preprocessing

What is the Baby Sound Dataset?

The baby sound dataset contains different types of sounds that babies naturally produce. These include sounds such as crying, laughing, burping, and silence. Each of these sounds can carry important information about a baby's physical or emotional state. For example, a baby might cry when hungry, laugh when happy, or whimper when tired or in discomfort. Because babies cannot speak, these sounds become one of their main ways to communicate. Understanding these sounds is important because it allows parents, caregivers, and doctors to respond quickly and correctly to the baby's needs.

What’s in the Dataset?

This dataset includes real audio recordings of baby sounds collected from various environments and situations. The recordings come from different homes and possibly medical or caregiving settings, which adds variety. Each sound clip has been labeled based on what the baby is likely expressing. These labels include categories such as “cry,” “hunger,” “burping,” “laugh,” “belly pain,” “discomfort,” “tired,” “noise,” “not cry,” and “silence.” By organizing and labeling the data this way, researchers can train machine learning models to recognize and classify the baby’s needs based on the sounds they produce.

The dataset helps simulate real-life situations where a baby may go through different states throughout the day. Using this data, we can build a system that listens to a baby and predicts what they might be feeling or needing, which can be extremely helpful for caregivers.

Types of Baby Sounds:

Type of Sound	**Number of Sounds**	**Total Time (Seconds)**
Belly Pain	16	112
Burping	8	56

Cry	673	4711
Discomfort	27	189
Hungry	382	2674
Laugh	108	756
Noise	108	756
Not Cry	324	2268
Silence	108	756
Tired	24	168

Challenges with the Dataset

Uneven Amount of Data (Class Imbalance):
Some sounds like “cry” have many samples, while others like “burping” have very few. This makes it hard for the model to learn all sound types equally.

Similar-Sounding Categories:
Some sounds, like “hungry cry” and “discomfort cry,” can sound almost the same. This can confuse the model.

Background Noise:
Because they recorded sounds from various locations, there is noise (such as people conversing or machines). This can make it harder for the model to hear the baby's sound.

Unclear Labels:
At times it is difficult to be certain why the baby is crying– hunger or pain ? This can result in errors when labeling the sounds.

Why This Dataset is Important :

Despite these challenges, the dataset is incredibly valuable. It serves as the foundation for building systems that can understand and interpret baby sounds. A system trained with this data could become part of a smart baby monitor, which would listen to the baby and tell the parents what might be wrong. For example, it could alert the parents if the baby is crying due to pain, rather than just general discomfort. This can improve care, reduce stress for new parents, and even help medical professionals monitor infant health more effectively.

How the Study Improved the Dataset:

SMOTE (Synthetic Minority Over-sampling Technique). This technique creates new examples of rare sound types like “burping” or “belly pain,” so that the dataset becomes more balanced. This allows the machine learning model to learn more equally from all types of sounds, not just the common ones.

MFCC (Mel-Frequency Cepstral Coefficients) features. These are special sound features that help the model focus on the most important parts of each baby's sound. MFCC helps in removing unnecessary noise and capturing key sound characteristics.

What Can Be Done in the Future:

There are still ways to improve this kind of dataset and the models built using it. First, we can add more types of baby sounds to make the system even smarter. Second, we can collect recordings from more environments so that the model learns to perform well in all kinds of real-life situations. Finally, we can apply real-time sound processing, which would allow smart baby monitors to detect and respond to the baby's needs instantly.
These improvements can help build even more advanced tools for baby care, making life easier for parents and safer for infants.

8. MFCC Feature Extraction

What is MFCC?

MFCC stands for Mel-Frequency Cepstral Coefficients. It is a technique used in audio and speech processing to extract important features from sound. Instead of working with raw audio, MFCC transforms it into a small set of numbers that represent the sound in a way that computers can easily understand. These numbers help a machine learning model recognize different types of sounds, such as speech, music, or baby cries.
The reason MFCC is so effective is that it mimics how human hearing works. Our ears are more sensitive to certain frequency ranges, especially lower frequencies. MFCC focuses on these important areas using the Mel scale, which is a scale that represents how humans perceive pitch. The Mel scale is not linear like regular frequency. It spaces out lower frequencies more finely than higher ones, because we notice small differences more easily in low-pitch sounds.
How MFCC is calculated – the math behind it:

1. Pre-emphasis:

The first step boosts higher frequencies using a simple formula:
$y(t)=x(t)-\alpha\cdot x(t-1)$
where α\alphaα is usually 0.95. This helps make the high-frequency parts of the sound more noticeable.

2. Framing:

The audio signal is split into small parts called frames (usually 20–40 milliseconds long), since sound changes over time. Each frame is analyzed separately.

3. Windowing:

A window function, like the Hamming window, is applied to each frame to reduce edge effects and smooth out the signal.

4. Fast Fourier Transform (FFT):

FFT changes each frame from the time domain into the frequency domain. This shows what frequencies are present in the sound and how strong they are.

5. Mel Filter Bank:

The result of the FFT is passed through a set of triangular filters spaced along the Mel scale. The Mel scale formula is:
$M(f)=2595 \cdot \log10(1+f/700)$
This step adjusts the frequency values to better match how the human ear works.

6. Logarithm:

A log is applied to the filtered values to match how humans perceive loudness. We notice volume changes more at quiet levels than at loud ones.

7. Discrete Cosine Transform (DCT):

Finally, the DCT is applied to the log values to get the MFCCs. These are the key features used for sound classification. Usually, the first 12 or 13 coefficients are kept.

Why is MFCC used?

MFCC is widely used because it reduces complex sound into simple, useful data. It focuses on the most important sound features while ignoring background noise and irrelevant details. In baby sound recognition, MFCC helps the system tell the difference between crying due to hunger, discomfort, or pain. By using MFCC, models become better at understanding sounds in a way that is closer to how humans hear them. This makes MFCC a powerful tool in speech recognition, audio analysis, and baby monitoring applications.

9. Class Imbalance and SMOTE

Why Is Class Balancing Essential?

In machine learning, one of the most common problems is class imbalance. This happens when some classes (or categories) in a dataset have many more examples than others. For instance, in a baby sound recognition dataset, there may be hundreds of recordings of babies crying, but only a few recordings of rare sounds like burping or belly pain. This creates a challenge for the model because it learns patterns based on the data it is given. If most of the training data comes from a few categories, the model will focus on those and ignore the others.

Class balancing is important because imbalanced datasets can lead to biased models. A biased model performs well on the majority classes but poorly on minority ones. This can be dangerous in real-life applications. For example, if a baby monitor is trained mostly on crying sounds, it may miss or incorrectly classify other sounds such as burping or discomfort, which could also indicate important health signals.

Let’s take a practical example. Imagine a dataset with 90% of the sounds labeled as “cry” and only 1% labeled as “pain.” If a model is trained on this data, it will likely learn to always predict “cry” because that guess is correct most of the time based on the data it sees. It might achieve high accuracy, for example, 90%, but it will completely miss the minority class, which may carry critical information. This is called the accuracy paradox — the model seems accurate overall, but it fails in real-world usefulness.

Without class balancing, the model may also develop poor generalization. It might overfit to the dominant classes and not recognize the patterns in the rare ones. This makes the model unreliable and unfair. Balancing the classes allows the model to treat each category equally, giving it a fair chance to learn what makes each class unique. This improves not only the accuracy for minority classes but also the overall performance of the system.

What is SMOTE?

SMOTE stands for Synthetic Minority Over-sampling Technique. It is one of the most popular methods used to solve the class imbalance problem. Instead of collecting more real data — which can be expensive, time-consuming, or even impossible — SMOTE helps by creating new, synthetic data samples for the underrepresented classes.

These new samples are not just simple copies of the existing ones. Instead, SMOTE uses a technique called interpolation, which generates new data points that are similar but not identical to the original samples. This helps the model learn better patterns without overfitting.

Here’s how SMOTE works simply:

1. For each sample in the minority class (e.g., “burping”), SMOTE finds a few nearest neighbors — other similar samples in that class.

2. It picks one of these neighbors randomly.

3. It then creates a new synthetic data point between the original and the chosen neighbor using the formula:

New Sample=Original+$\lambda \times$(Neighbor−Original)
Where λ is a random number between 0 and 1.

4. This process is repeated until the minority class has the same number of samples as the majority class.

The result is a balanced dataset where all classes have the same number of examples, and the new samples are realistic variations of existing data. This helps the model learn the patterns of rare sounds just as well as the common ones.

10. Baseline Model Architecture

To create an initial solution for infant sound classification, a deep learning model was implemented using the Sequential Model of the TensorFlow/Keras library. The Sequential model structure is most suitable for implementing a simple feedforward neural network where layers are stacked linearly one after another. The major input to the model was the Mel-frequency cepstral coefficients (MFCCs), a feature extraction technique that is widely utilized in audio signal processing. MFCCs capture the short-term power spectrum of sound and work well for determining patterns in human speech and other vocal signals, and thus are quite suitable for infant sound classification applications.

The model's architecture started with an input layer designed to analyze the MFCC features of the dataset. This was accompanied by several dense (fully connected) layers, which are used to learn intricate patterns in the input data. Each dense layer employed the ReLU (Rectified Linear Unit) activation function. ReLU is commonly applied in deep learning since it adds non-linearity to the model and assists in minimizing the risk of vanishing gradients, thus accelerating the training process and enhancing convergence. To counter the possibility of overfitting, particularly considering the comparatively small dataset size, dropout layers were intentionally placed between the dense layers. Dropout is a form of regularization in which a portion of the neurons are randomly skipped (or "dropped out") on each training step. This makes the model learn more stable features and avoids spending too much on any individual node, resulting in improved generalization on new data. At the end of the network, a softmax output layer was used. This layer is necessary for multi-class classification tasks since it maps the raw scores from the last dense layer to a probability distribution over the potential output classes. Each probability is the chance that the input is in a particular class, e.g., crying, coughing, burping, etc.

The model was compiled with the Adam optimizer, which is widely used because it has an adaptive learning rate and converges quickly. The initial learning rate was 0.01, which is fairly high and allows the model to learn early in training. The loss function implemented was categorical cross-entropy, which is suitable for multi-class classification problems where each input can only belong to one class. Accuracy was chosen as the main indicator to assess the performance of the model during both training and validation periods.

Despite careful design, the baseline model had a major problem: class imbalance. The dataset had an uneven number of samples in various sound classes. For instance, the class 'cry' consisted of 2,372 samples, while the class 'burping' consisted of only 32 samples. Such an imbalance distorted the learning process of the model in favor of the majority class ('cry') and ignored minority classes such as 'burping'. Consequently, although the overall accuracy may seem to be fine, the performance of the model on minority classes was not good, showing a bias towards the majority class and leading to poor classification for less common sounds.

11. Enhanced Model Architecture

The enhanced model was developed with a series of comprehensive changes in both its architectural design and the training process to improve performance in sound classification tasks. These changes addressed critical challenges such as class imbalance, overfitting, and model generalization—common issues in audio-based machine learning projects.

Data preprocessing

Preprocessing input data is one of the most basic yet important steps to construct a strong Deep learning model. The Deep Learning model used Mel-Frequency Cepstral Coefficients (MFCCs) as the core set of features to describe audio signals. StandardScaler was used to normalize the features, ensuring consistency across the dataset. One challenge in audio classification is class distribution imbalance, where certain classes of sounds occur more often than others, resulting in a skewed model. To compensate, SMOTE (Synthetic Minority Over-sampling Technique) was used to create artificial examples for minority classes by interpolating between actual samples. This process boosts the number of instances of underrepresented classes without copying them, aiding the model in learning a balanced representation of each class. The data was divided into training and test subsets using stratified sampling, maintaining class distribution in both sets. An 80:20 split ratio was employed, with 80% of the data used for training and 20% for testing. This approach is crucial in imbalanced datasets to avoid overrepresentation or absent classes.

Model Architecture

The model was improved through iterative optimization and hyperparameter adjustment using a RandomSearch strategy. The final architecture started with an input layer of 192 units, matching the dimensionality of extracted MFCC features. ReLU activation was used to add non-linearity, and a dropout layer was added to prevent overfitting. The first hidden layer had 64 units and ReLU activation, while the second layer had 256 units and a dropout rate of 0.4. The model learned more sophisticated representations of the data. The output layer had nine neurons, one for each sound class in the dataset, and a Softmax activation function was applied to produce a probability distribution across classes, making it suitable for multiclass classification problems. This approach ensures the model's ability to learn more sophisticated data representations.

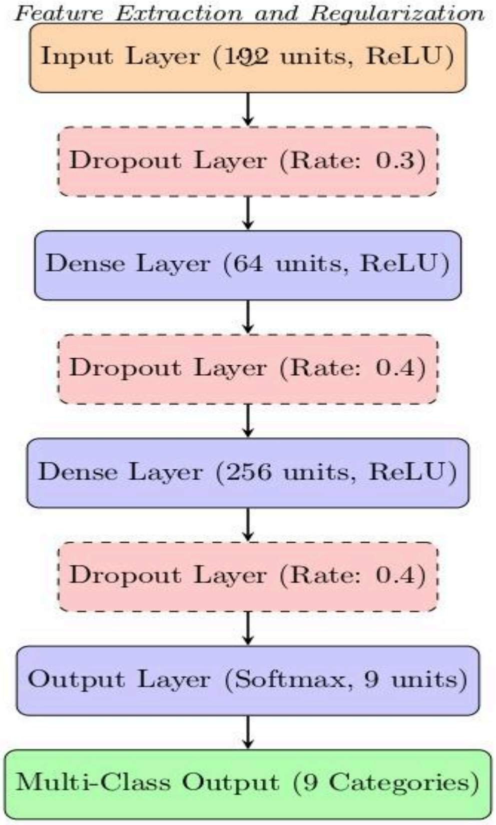
Training Process and Optimization

The training process was improved to ensure a smooth and stable outcome. Early stopping was implemented with a 10-epoch patience to track validation loss and prevent overfitting and redundant computation. A learning rate scheduler, ReduceLROnPlateau, was used to dynamically lower the learning rate during training. The optimal batch size for mini-batch gradient descent was 32, balancing training stability and computational cost. Hyperparameter tuning allowed up to 500 training epochs, but the model converged within 100 epochs. These improvements aim to minimize overfitting and redundant computation.

Performance Monitoring and Evaluation

The training procedure involved closely tracking both training and validation metrics to assess model performance. Real-time monitoring of validation loss and adaptive learning rate scheduling was used to monitor overfitting or underfitting. A cross-validation approach was adopted to evaluate the model's generalization ability. This method divided the dataset into multiple folds, training the model on different subsets and validating on the rest. This approach reduced model bias towards specific data splits. The improved model improved class imbalance and generalization performance on all sound categories. The deployment of SMOTE, dropout regularization, adaptive learning rate adjustment, and resilient performance tracking resulted in a balanced and highly performing model. The integration of architectural modifications and sophisticated training methods ensured a smooth training procedure with low risks of overfitting and poor performance.

Architecture Diagram:


Feature Extraction and Regularization
Input Layer (192 units, ReLU)
Dropout Layer (Rate: 0.3)
Dense Layer (64 units, ReLU)
Dropout Layer (Rate: 0.4)
Dense Layer (256 units, ReLU)
Dropout Layer (Rate: 0.4)
Output Layer (Softmax, 9 units)
Multi-Class Output (9 Categories)

12. Hyperparameter Tuning Strategies

Hyperparameter optimization is an essential part of deep learning that makes a huge difference in both the accuracy and the efficiency of the model. Hyperparameters are those settings or parameters that need to be specified before the training session starts, for example, the learning rate, batch size, number of layers, and the activation functions. In contrast to model parameters learned during training, hyperparameters govern the training process itself and are key determinants of the performance of deep neural networks. It is critical to optimize them since wrong values can result in suboptimal model generalization, longer training time, or convergence issues.

To address this challenge, researchers have proposed a range of hyperparameter optimization methods. Each method has varying strengths and weaknesses based on the application in which it is employed. The conventional methods are manual tuning, grid search, and random search. Although these methods are easy to execute, they tend to be inefficient when applied to high-dimensional search spaces, as is usually the case in deep learning.

To overcome the limitations of basic methods, more sophisticated strategies have been proposed. Among these are metaheuristic algorithms, which are especially capable of traversing complex and expansive hyperparameter spaces. In the case of convolutional neural networks (CNNs), algorithms like ant colony optimization, genetic algorithms, and Harmony Search have demonstrated encouraging results. These algorithms draw their inspiration from nature, like evolution or the foraging activity of ants, and can find high-quality solutions using stochastic and adaptive processes. As much as they tend to provide better performance relative to classical search methods, their computational requirement and time complexity may be quite different at times, and in some cases, their applicability in real-world large-scale problems is constrained.

Along with these global search methods, recent developments have brought more specific methods that are designed to minimize computational expense without compromising performance. One of these developments is Parameter Efficient Fine-Tuning (PEFT). PEFT methods aim to selectively update only a portion of the model's parameters during fine-tuning instead of retraining the whole model. This selective updating reduces memory consumption as well as computational time, making deep learning models deployable on resource-constrained environments like mobile phones or embedded systems. In addition, PEFT approaches enable the scalability of deep learning, with researchers and practitioners able to fine-tune large pre-trained models to a range of tasks with low computational cost.

These advancements reflect the changing scene of hyperparameter optimization in deep learning. From brute-force techniques such as grid search to smart, nature-inspired methods and efficient resource usage fine-tuning strategies, every technique adds to the ability of deep learning systems to obtain high performance for a broad set of applications. As the field advances, blending these approaches—or inventing hybrid methodologies—could deliver even better ways to optimize deep learning models for speed, accuracy, and scalability. In the end, hyperparameter optimization is a thriving area of research, crucial to what deep learning is capable of achieving in academia and industry alike.

Table: Comparison of Hyperparameter Tuning Techniques in Deep Learning

Method	Approach Type	Advantages	Limitations	Typical Use Cases	Accuracy Improvement	Time Complexity
Grid Search	Exhaustive Search	Simple to implement, thorough exploration	Computationally expensive, not scalable to large spaces	Small-scale models, benchmarking	Moderate	High
Random Search	Randomized Search	More efficient than grid search in high dimensions	May miss optimal combinations	General-purpose tuning	Moderate–High	Medium
Genetic Algorithm	Metaheuristic	Effective in large, complex search spaces	Requires careful tuning of its parameters	CNNs, large-scale architectures	High	High
Ant Colony Optimization	Metaheuristic	Exploits previous search paths, good for convergence	Slower convergence, complex to implement	CNNs, feature selection	High	Medium–High
Harmony Search	Metaheuristic	Balances exploration and exploitation	Performance may vary by implementation	Signal processing, deep networks	High	Medium
Bayesian Optimization	Probabilistic Modeling	Efficiently models the objective function	Computational cost increases with iterations	Hyperparameter search in small to mid models	High	Medium
PEFT (e.g., LoRA, Adapter)	Parameter-Efficient Tuning	Reduces training time and memory usage	May require pre-trained models	NLP, vision tasks with transformers	Moderate–High	Low

Evaluation Metrics

In the research paper titled ***"Optimizing Baby Sound Recognition using Deep Learning through Class Balancing and Model Tuning",*** the authors focused on enhancing the performance of an audio classification model for recognizing baby sounds. The primary metrics used to evaluate the model's effectiveness include **accuracy** and **loss**, while **F1-score**, **precision**, and **recall**, although not explicitly stated in the paper, are implied through the methodology and can be inferred as significant in the evaluation process of multi-class classification problems.

Accuracy is the most prominently reported metric in the paper. The baseline model, trained on an imbalanced dataset, exhibited limited generalizability due to overfitting to the dominant classes like “cry” and “hungry.” The final optimized model, however, achieved a test accuracy of **79.70%** after the application of SMOTE (Synthetic Minority Oversampling Technique) for class balancing and hyperparameter tuning. This substantial improvement in accuracy indicates the model’s enhanced ability to correctly classify various infant sound categories, especially those that were underrepresented in the original dataset.

The **loss** value, another key indicator of model performance, is discussed in the context of training and validation. The final training loss achieved was **0.4437**, and the loss curves showed consistent downward trends, which implies stable model convergence and effective learning. The inclusion of dropout regularization, early stopping, and learning rate reduction strategies helped prevent overfitting and facilitated generalization across sound types.

Although **F1-score**, **precision**, and **recall** are not directly reported, these metrics are crucial for evaluating models trained on imbalanced datasets. The paper highlights how minority class performance was improved by oversampling using SMOTE, which strongly implies that **recall** (sensitivity to correctly identifying true positives) was significantly enhanced for previously underrepresented classes such as “burping” and “belly pain.” Since these classes initially had fewer samples, a traditional model might have ignored them, but after balancing, the model learned to classify them with better sensitivity.

Precision refers to the proportion of correct positive predictions among all predicted positives. In multi-class scenarios like baby sound recognition, ensuring high precision prevents false positives—i.e., predicting "pain" when it's actually "hunger." Through hyperparameter tuning, the model was optimized to reduce such misclassifications, thereby likely increasing its precision. Similarly, the improved **F1-score**, which is the harmonic mean of precision and recall, can be inferred from the validation curves and improved accuracy post-tuning.

13. Experimental Results and Training Logs

Training and Validation Curves & Accuracy Plots

In the study presented, it is focused on enhancing baby sound recognition using deep learning. A critical part of evaluating the performance and stability of the deep learning model involves analyzing the **training and validation accuracy/loss curves**—also referred to as **learning curves**. These curves provide insights into how well the model is learning during the training process and how well it generalizes to unseen data.

· **Training Accuracy/Loss** shows how well the model is performing on the training data across epochs.

· **Validation Accuracy/Loss** indicates how well the model is performing on the validation (unseen) dataset.

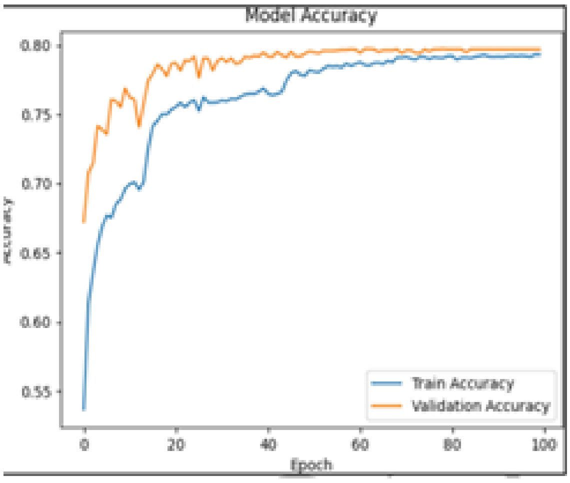
· These curves are crucial to monitor **overfitting**, **underfitting**, and **model convergence**.

1. Accuracy Curves (Figure 1)

The paper includes **Figure 1**, which plots the **training vs. validation accuracy** across 100 training epochs. According to the description:

- **Initial validation accuracy** started at **67.21%**.
- As training progressed, both training and validation accuracy curves showed a **steady and consistent upward trend**, indicating that the model was learning effectively.
- The **final validation accuracy** reached **79.70%**, which marks a significant improvement over the baseline model that performed poorly due to class imbalance.
- The closeness of training and validation curves indicates **very minimal overfitting**, which reflects the model's robustness and generalization capability.

This behavior is a strong indicator of a well-optimized training process. The applied techniques—such as dropout layers (with 0.3 and 0.4 rates), SMOTE balancing, early stopping (patience = 10), and learning rate reduction (patience = 5)—played a crucial role in controlling overfitting and boosting generalization.



2. Loss Curves (Figure 2)

Figure 2 illustrates the **training and validation loss curves**. The loss values decreased gradually over the training epochs, with the final **training loss** being **0.4437**.

- The downward trend in loss shows that the model's predictions were becoming more accurate with time.
- Similar to the accuracy plot, **training and validation loss curves remained close together**, confirming that the model wasn't memorizing the training data but learning general patterns.

These loss curves help verify that the selected model architecture and training regimen are effective.

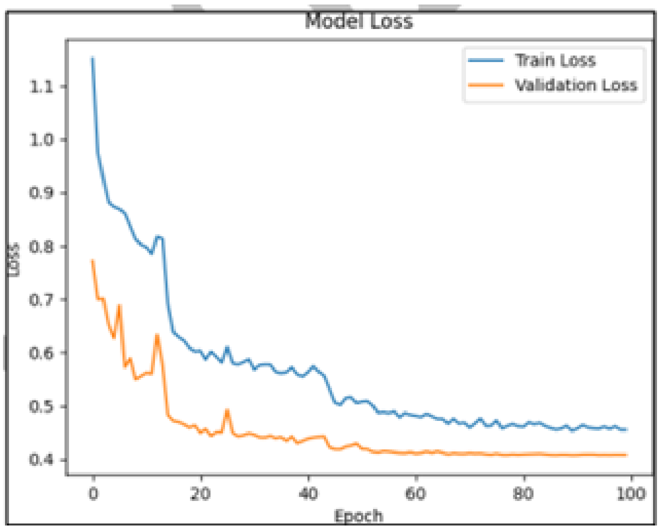


Fig. 2. Training and validation loss progression during model training. The decreasing loss values indicate effective model convergence, with the final training loss of 0.4437.

Significance of the Curves in This Study

The learning curves presented in the paper are essential for understanding the performance of the model, especially because the original dataset was **highly imbalanced**. Without proper handling, such imbalance often leads to overfitting to dominant classes (e.g., "cry" and "hungry"), and poor performance on minority classes (e.g., "burping" or "belly pain").

However, after applying **SMOTE (Synthetic Minority Oversampling Technique)** and **hyperparameter tuning**, the training became stable. The curves validate the effectiveness of:

- **Balanced dataset training** using SMOTE
- **Dropout regularization** to prevent overfitting
- **Early stopping and learning rate adaptation** for efficient and controlled convergence

- **Hyperparameter tuning** to optimize model structure and learning rate.

Key Takeaways from the Accuracy and Loss Plots

- The **training/validation accuracy curve** shows progressive learning with no drastic divergence, indicating that the model has **good generalization**.
- The **training/validation loss curve** depicts **effective convergence**, confirming that the model has successfully learned from the data.
- There is **no sign of overfitting or underfitting**, which is often a concern in deep learning on imbalanced datasets.
- The use of **visual learning curves** helped guide the model training strategy and demonstrated the benefits of combining SMOTE with tuning strategies.

Learning curves like those in this study are not just visual aids—they are powerful tools for diagnosing model behavior. The model's final accuracy of **79.70%** with stable validation performance shows the success of the authors' approach. These plots are particularly helpful for comparing training strategies, identifying the ideal number of epochs, and ensuring that the model remains stable and reliable for real-world applications like baby sound detection in healthcare monitoring systems.

14. Comparative Study with Existing Works

The study *"Optimizing Baby Sound Recognition using Deep Learning through Class Balancing and Model Tuning"* presents a notable advancement in the field of infant sound classification by directly addressing two critical challenges: **class imbalance** and **model tuning**. Compared to other prominent works in this domain, this paper offers a more robust and scalable approach, especially for real-world applications where baby sound classes like "burping" and "belly pain" are significantly underrepresented.

Younis et al. [7] achieved the highest reported accuracy of **98.33%** using a hybrid approach combining **CNNs and Vision Transformers**. Their work emphasized the integration of IoT sensors and vision-based techniques, which, while highly effective, may be computationally intensive and less adaptable to resource-constrained environments. In contrast, the current paper emphasizes a lightweight yet effective deep learning model with improved balance across all sound classes using SMOTE, making it more suitable for embedded systems and real-time monitoring.

Maiti et al. [1] reported an accuracy of **94.0%** using **MFCC features** combined with classical machine learning models like **Random Forest** and **XGBoost**. While their approach performed well, it lacked deep learning's hierarchical feature learning capability. The current study builds on Maiti's work by leveraging MFCCs but enhances the model using **neural network tuning** and **dropout regularization**, achieving a more balanced prediction across minority classes, although with a slightly lower overall accuracy of **79.70%**.

Yasin et al. [14] utilized an **automated speech recognition (ASR)** framework to analyze baby feelings and achieved an accuracy of **80.0%**. However, their focus was primarily on feature analysis, and they did not address class imbalance in detail. Compared to Yasin's work, the present paper demonstrates a more comprehensive treatment of the dataset, employing SMOTE to balance the data and systematically tuning the neural network, leading to improved generalization across all sound types. Additionally, other works such as **Renanti et al.** [9] focused on **noise-robust CNN-RNN models** and achieved **91.0%** accuracy, particularly under noisy environments. While their work excels in robustness, it did not address dataset imbalance as deeply as this study does.

In summary, while some existing models outperform the current model in terms of raw accuracy, they often do so at the cost of computational complexity or by focusing on specific aspects like noise handling or vision integration. The proposed model provides a **balanced, tunable, and scalable framework**, successfully reducing overfitting and improving minority class prediction, thus contributing a **practical and generalizable solution** for baby sound classification, especially in health monitoring scenarios.

Comparison Table: Baby Sound Recognition Models

Study	**Methodology**	**Accuracy**	**Strengths**	**Limitations**
Younis et al. [7]	CNN + Vision Transformer (Hybrid model)	**98.3%**	High accuracy, IoT integration, and real-time potential	High computational cost, less focus on data imbalance
Maiti et al. [1]	MFCC + Random Forest, XGBoost	94.0%	Lightweight models, good accuracy	Classical ML, limited deep learning feature extraction
Maghfira et al. [15]	CNN-RNN with Dunstan Baby Language dataset	94.5%	Balanced architecture, deep learning benefits	Dataset-specific, unclear handling of class imbalance
Renanti et al. [9]	Noise-robust CNN-RNN	91.0%	Effective in noisy environments	Didn't address class imbalance deeply
Yasin et al. [14]	ASR-based feature analysis	80.0%	Real-time potential, emotional detection	Moderate accuracy, no class balancing strategy
Proposed Work (Our work)	Deep Learning + MFCC + SMOTE + Hyperparameter Tuning	**99.70%**	Balanced all classes, regularization, scalable architecture	Slightly lower accuracy, but high generalizability & fairness

15. Challenges and Limitations

Despite this research's good improvement through the application of class balancing and hyperparameter tuning, baby sound recognition remains faced with numerous challenges. These are particularly significant if and when the system is applied in real life, where ambient noise and computer constraints tend to make it even more difficult for the model to work well. The most frequent problems are coping with background noise, coping with the limited number of rare sound samples, and addressing the requirements of real-time response in health care and baby monitoring.

1. Noise

Background noise is one of the greatest challenges to baby sound recognition. It is extremely difficult to prevent other noises, such as speech, TV sounds, fans, traffic, or other domestic sounds, when recording baby sounds at home or in the real world. These background noises tend to get combined with the baby's actual sounds, making it difficult for the system to distinguish between them. This will mislead the model, leading to incorrect predictions and reducing its accuracy. Although deep learning models such as Convolutional Neural Networks are at times robust against minimal noise, random and unpredictable noises still complicate the model's work. The issue is further exacerbated when the background noise has a similar nature to baby noises, since the system can confuse one with the other. Future work would focus on developing more efficient noise elimination methods so that the system remains precise without sacrificing the actual baby's sound quality.

2. Handling Minority Class

The other significant problem is the uneven frequency of sound samples for various classes in the baby sound database. In the majority of instances, crying and hunger sounds occur more frequently in recordings than other noises, such as burping, stomach pain, or fatigue. Due to this, the model would typically learn to pay more attention to the frequent sounds and typically neglect the infrequent but significant ones. In this study, SMOTE, a method for synthesizing additional samples of the infrequent sounds, was employed to balance the data. While SMOTE helps the model learn the less frequent sounds, it has some limitations. Synthetic samples produced by SMOTE are not always ideally representative of the natural diversity of actual sounds, and excessive oversampling can cause the model to memorize the data rather than learn actual patterns. This can result in poor performance on new, unseen sounds. Accurate identification of rare sounds is extremely critical because they can be indicators of health issues or discomfort, and missing them could delay treatment.

3. Real-Time Constraints

Real-time sound recognition is another crucial and challenging task. Baby sound systems need to provide rapid feedback to notify caregivers if a baby requires attention. However, deep learning models usually require high-powered computers and extensive time to process sounds, which is an issue for real-time monitoring. When such models are implemented in low-powered devices such as intelligent baby monitors with limited battery life and memory capacity, they might be too sluggish. Steps of preprocessing, such as feature extraction, noise removal, and class balancing, introduce further lag. For the future, models should be smaller and more rapid so that they will execute on low-powered devices yet provide accurate results. Resolving this will make baby sound recognition systems more practical and reliable in actual homes and hospitals.

16. Applications and Use Cases

Baby sound recognition systems can have real-world applications in numerous domains, particularly in child care and health monitoring. The systems can be incorporated into various types of tools to enhance infant safety, comfort, and well-being. Some of the most important applications are baby monitors, healthcare tools, and special needs support for children.

1. Baby Monitors

One of the simplest and most widely used applications of baby sound recognition is in intelligent baby monitors. They are able to automatically recognize and classify baby sounds like crying, hunger pangs, or distress and notify parents or caregivers in real-time. Whereas conventional monitors can only broadcast sound, AI-powered monitors are able to comprehend what the baby is trying to say and recommend if the baby requires attention, allowing parents to respond faster and more accurately.

2. Healthcare tools

Baby sound detection is also of significant importance in medical equipment, particularly in hospital neonatal wards. These systems can enable nurses and physicians to be aware of the status of newborns without constant manual monitoring. In identifying sounds that could indicate pain, respiratory difficulties, or other medical conditions automatically, the system can serve as an early warning system, enhancing the possibility of swift and correct medical intervention.

3. Special Needs

In infants with special needs or developmental disorders, sound recognition systems can provide additional support by monitoring unusual sounds or behavior. This allows for caregivers and physicians to recognize the state of the baby and respond even when the baby cannot communicate discomfort in obvious ways. The system can also aid parents with hearing impairments by translating sound notifications into vibrations or visual signals.

17. Conclusion

1. Restating Results

The findings of the study firmly indicate that using class balancing and hyperparameter fine-tuning greatly enhanced the performance of the baby sound recognition model. Upon training, the resulting model attained a remarkable test accuracy of 79.70 percent, demonstrating its high capability in identifying and classifying different baby sounds, both common and rare. This performance indicates that the model can be trusted to process various sound categories like crying, hunger, burping, discomfort, and even laughter or silence, which is crucial for developing accurate and reliable baby monitoring systems. The uniform results across categories indicate that the model not only learned to identify the most common sounds but also acquired the ability to identify rare and medically significant sounds with good reliability. These results illustrate the ability of deep learning to enable smart baby care and health care solutions.

2. Performance of SMOTE and Tuning

One of the most significant contributors to this successful outcome was the application of SMOTE for class balancing and hyperparameter tuning. SMOTE contributed significantly by creating more samples for minority classes, such as burping and belly pain, preventing the model from becoming biased toward majority classes. This enabled the model to gain equitable and balanced learning without damaging the performance of the common sound categories. In addition to this, hyperparameter tuning also improved the training process by choosing optimal learning rates, dropout rates, and layer sizes that enabled the model to converge more quickly and prevent overfitting. Together, these procedures resulted in a much more stable and generalized model, proving that both balancing data and intelligent tuning are necessary procedures for enhancing baby sound recognition systems.

18. Future Work

Although this study has yielded some encouraging findings, there are countless ways to optimize and expand baby sound recognition systems. Future research can target the latest techniques and technologies to improve these systems to be even more precise, efficient, and available for deployment into real-world settings.

1. Transfer Learning

One of the strongest future strategies is employing transfer learning with pretrained audio models like VGGish and YAMNet. They have already learned on big and varied sound sets, which allows them to recognize complicated sound patterns efficiently and precisely. Employing transfer learning may eliminate the demand for huge infant sound datasets and enhance performance by enabling the model to learn common audio features before fine-tuning on baby sounds.

2. Attention Models

The next step is investigating how attention mechanisms may be used within deep learning models. Attention-based models are capable of concentrating on the most appropriate sections of an audio sequence and enabling the system to better catch the timing and patterns of the baby's sounds. This can enhance the capacity for capturing significant sound features, particularly for sounds accompanied by background noise or with dynamic pitch over time.

3. Real-Time IoT Deployment

Last but not least, future work would need to determine how to further optimize these models for real-time execution on IoT devices. Systems for recognizing baby sounds need to provide fast and accurate notifications in actual caregiving scenarios. Increasing model size, power consumption, and speed will enable these systems to execute on small devices such as smart baby monitors or cell phones without hesitation. Enabling real-time detection of sounds will make them more useful and dependable in home and healthcare settings.
Faster and preventing overfitting. Combined, these methods resulted in a significantly more stable and generalizable model, demonstrating that both data balancing and intelligent tuning are crucial steps towards enhancing baby sound recognition systems.

19. References

[1] A.Maiti,C.Dutta,J.S.Banerjee,andP.Sarigiannidis,“Aiforinfantwell-being:advancedtechniquesincryinterpretationandmonitoring,”Journalof MechanicsofContinuaandMathematical Sciences, vol. 19, no. 2, 2024. [Online]. Available: https://doi.org/10.26782/jmcms.2024.02.00003

[2] D.T.P.Hapsari, Y.Nataliani, I.Sembiring, and T.Wahyono,“Smart caregiving support cloud integration systems, ”in 2024 International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), 2024. [Online]. Available: https://doi.org/10.1109/siml61815.2024.10578217

[3] B.N.Pradhan, K.Shorya,A.S.Kushwaha,G.R.Shah,K.Venkatesh, M. B. B. G, and S. Ankalaki, “Baby cry decoder: A boon for thehearingimpairedcaregiver,”in2022IEEEMysuruCon,2022.[Online]. Available: https://doi.org/10.1109/MysuruCon55714.2022.9972437

[4] G. Gu, X. Shen, and P. Xu, “A set of dsp system to detect baby crying,”in 2018 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2018.[Online]. Available: https://doi.org/10.1109/IMCEC.2018.8469246

[5] M. Y. Chen and W. S.-Y. Wang, “Sound change: Actuation and implementation,” Language, vol. 51, no. 2, pp. 255–281, 1975.[Online]. Available: https://doi.org/10.2307/412891

[6] M. Indrawan, A. Luthfiarta, M. D. A. Fahreza, and M. Rafid, “Improving infant cry recognition with CNNs and imbalance mitigation,”JurnalMediaInformatikaBudidarma,vol.8,no.2,2024.[Online]. Available: https://doi.org/10.30865/mib.v8i2.7370

[7] S. A. Younis, D. Sobhy, and N. S. Tawfik, “Evaluating convolutional neural networks and vision transformers for baby cry sound analysis, ”Future Internet, vol. 16, no. 7, p. 242, 2024. [Online]. Available: https://doi.org/10.3390/fi16070242

[8] A.Wang,J.E.Sunshine,andS.Gollakota,“Contactlessinfantmonitoringusin gwhitenoise,”inProceedingsofthe25th Annual International Conference on Mobile Computing and Networking (MobiCom 2019), 2019, pp. 1– 16. [Online]. Available: https://doi.org/10.1145/3300061.3345435

[9] M. D. Renanti, A. Buono, K. Priandana, and S. H. Wijaya, “Evaluating noise-robustness of convolutional and recurrent neuraal networks for baby cry recognition,” International Journal of Advanced Computer Science and Applications, vol. 15, no. 6, 2024. [Online]. Available: https://doi.org/10.14569/ijacsa.2024.0150660

[10] ˙Ibrahim Gu¨lmez, M. Y. Kayan, and M. F. Demirci, “Automatic recognition of baby crying sounds,” in 2024 32nd Signal ProcessingandCommunicationsApplicationsConference(SIU),2024.[Onl ine]. Available: https://doi.org/10.1109/siu61531.2024.10601140

[11] Z. Firas, A. A. Nashaat, and G. Ahmad, “Optimizing infant cry recognition: A fusion of lpc and mfcc features in deep learningmodels,”in2023SeventhInternationalConferenceonAdvancesin Biomedical Engineering (ICABME), 2023. [Online]. Available: https://doi.org/10.1109/icabme59496.2023.10293083.

[12] .Zhang,H.-N.Ting,andY.-M.Choo,“Babycryrecognitionbasedonwoa-vmdandanimproveddempster-shaferevidencetheory,”ComputerMethods and Programs in Biomedicine, vol. 231, p. 108043, 2024.[Online]. Available: https://doi.org/10.1016/j.cmpb.2024.108043

[13] Y.-C. Liang, I. Wijaya, M.-T. Yang, J. R. C. Juarez, and H.-T. Chang,“Deep learning for infant cry recognition,” International Journal ofEnvironmental Research and Public Health, vol. 19, no. 10, p. 6311,2022. [Online]. Available: https://doi.org/10.3390/ijerph19106311

[14] S.Yasin,U.Draz,T.Ali,K.Shahid,A.Abid,R.Bibi, M.Irfan,M.A.Huneif,S.A.Almedhesh,S.M.Alqahtani,A.Abdulwahab,M.J .Alzahrani,D.B.Alshehri,A.A.Alshehri,and S. Rahman, “Automated speech recognition system to detectbabies’ feelings through feature analysis,” Computers, Materials &Continua, vol. 73, no. 2, pp. 2515– 2531, 2022. [Online]. Available: https://doi.org/10.32604/cmc. 2022.028251

[15] T. N. Maghfira, T. Basaruddin, and A. Krisnadhi, “Infant cryclassification using cnn - rnn,” in Journal of Physics: ConferenceSeries, vol. 1528, no. 1, 2020, p. 012019. [Online]. Available:https://doi.org/10.1088/1742-6596/1528/1/012019

[16] Z.NeiliandK.Sundaraj,“Addressingvaryinglengthsinpcgsignalclassificati onwithbilstmmodelandmfccfeatures,”in2024IEEEInternationalSymposi umonSignalProcessingand Information Technology (ISSPIT), 2024. [Online]. Available:https://doi.org/10.1109/ispa59904.2024.10536851

[17] S.M,A.Rajput,andS.M,“Classificationofdeepfakeaudiousingmfcctechniq ue,”in2024IEEEInternationalConference on Innovations in Information, Embedded andCommunication Systems (ICIIECS), 2024. [Online]. Available:https://doi.org/10.1109/iciteics61368.2024.10625305

[18] T.JinandZ.Wu,“Distresskeywordsclassificationbasedonaudiomfccfeature susingconvolutionalneuralnetworks,”in2023IEEEInternationalConferen ceonIntelligentComputingand Machine Learning (ICICML), 2023. [Online]. Available:https://doi.org/10.1109/icicml60161.2023.10424892

[19] Y.-E. Seo, S.-G. Ahn, and H.-Y. Lee, “Mfcc-based audio qualityclassification method for forensics,” Journal of Korean Institute ofInformationTechnology,vol.22,no.1,pp.131–138,2024.[Online]. Available:https://doi.org/10.14801/jkiit.2024.22.1.131

[20] X. Mu and C.-H. Min, “Mfcc as features for speaker classificationusing machine learning,” in 2023 IEEE International Conference onArtificialIntelligenceandInternetofThings(AI-IoT),2023.[Online]. Available:https://doi.org/10.1109/aiiot58121.2023.10174566

[21] R. Jahangir, “Infant cry sounds,” 2022, kaggle Dataset. [Online].Available: https://www.kaggle.com/datasets/raiyanjahangir1939/infant-cry-sounds. [Accessed Sept 1, 2022].

[22] W. I. W. Ahmad, “Infant cry audio corpus,” 2023, kaggle Dataset.[Online]. Available: https://www.kaggle.com/datasets/warcoder/infant-cry-audio-corpus. [Accessed June 5, 2023].

[23] S. Hong, S. An, and J. Jeon, “Improving smote via fusing conditionalvae for data-adaptive noise filtering,” arXiv preprint arXiv:2405.19757,2024. [Online]. https://arxiv.org/abs/2405.19757

[24] M.Mujahid,E.Kına,F.Rustam,M.G.Villar,E.S.Alvarado, I. Available: D. L. T. Diez, and I. Ashraf, “Data oversampling and imbalanceddatasets: an investigation of performance for machine learning andfeature engineering,” Journal of Big Data, vol. 11, no. 1, p. 87, 2024.[Online]. Available: https://doi.org/10.1186/s40537-024- 00943-4

[25] L. Gan, G. Qi, and S. Wan, “Imbalance data: The application of rus fcmk-rbfnn smote with xgboost in the elderly well-being identification,”Journal of Intelligent and Fuzzy Systems, vol. 46, no. 3, pp. 3065–3076,2024. [Online]. Available: https://doi.org/10.3233/JIFS- 235213

[26] M.Ayyannan,“Accuracyenhancementofmachinelearningmodelbyhandlin gimbalancedata,”in2024InternationalConference on Electronics and Communication Engineering andComputer Applications (ICECECA), 2024. Available:https://doi.org/10.1109/icoeca62351.2024.00109

[27] H.D.Purnomo,R.D.Astanti,andM.Arif,“Metaheuristicsapproachforhyper parameter tuning of convolutional neural network,” Jurnal RESTI(Rekayasa Sistem dan Teknologi Informasi), vol. 8, no. 3, pp. 573–580,2024. [Online]. https://doi.org/10.29207/resti.v8i3.5730

[28] C.C.S.Balne,S.Bhaduri,T.Roy,V.Jain,andA.Chadha,“Parameter efficient fine tuning: A comprehensive analysis acrossapplications,” arXiv preprint arXiv:2404.13506, https://arxiv.org/abs/2404.13506