BabyCode

"where the lullabies of the night cradle the dreams of tomorrow"

University of Jordan

NATURAL LANGUAGE PROCESSING (1905380)

Dr. Mohammad Abushariah

May 25, 2024

By Mira Melhem

Table of Contents

Table of Figures

1.0 Introduction:

Embark Step into the captivating world of BabyCode, a place where technology meets compassion, and the mysteries of infant cries are unraveled. This is a realm where the marvels of modern technology are harnessed to address a timeless challenge faced by parents worldwide - understanding the unique language of their baby's cries.

BabyCode is more than just a project; it's a mission to empower parents, particularly mothers, with the ability to comprehend their baby's needs. By developing a mobile application that translates the cryptic cries of infants into a language they can understand, we aim to foster stronger bonds between parents and their children and pave the way for a future where no cry goes misunderstood. Our journey began on 8th March 2024 and concluded on 25th May 2024. During this time, we navigated through various stages of the project, from understanding the needs of our users to the final implementation of our solution. This report encapsulates this journey, offering a detailed account of our project's evolution.

1.1 Need Analysis and Description

Our primary user persona is "Maya", a first-time mother with a newborn baby. Maya, like many mothers, finds herself navigating the uncharted territory of parenthood. She experiences joy and wonder, but also faces the challenge of understanding her baby's unique language of cries.

Maya often finds herself awake at odd hours, trying to soothe her crying baby. She struggles to distinguish between cries of hunger, discomfort, or tiredness. This uncertainty often leads to stress and anxiety, as she is unsure if she is correctly addressing her baby's needs. A need for a reliable tool that could help her understand her baby's cries is needed. She wants a

solution that is easy to use, non-intrusive, and can provide real-time insights. She hopes for a tool that can give her confidence and peace of mind in her motherhood journey.

From BabyCode, Maya expects a user-friendly mobile application that can accurately classify different types of cries. She also hopes for the app to be adaptable, learning and improving its accuracy over time as it 'gets to know' her baby.

1.2 Project Constraints

During the development of BabyCode, we encountered several constraints that shaped our project's trajectory:

- Limited Dataset: We began BabyCode with a small dataset of baby cries, posing a challenge for model training. To counter this, we maximized our dataset's utility through rigorous preprocessing, extracting features like MFCC, Mel Spectrogram, Chroma, and Spectral Contrast, significantly enhancing our data quality.
- Time Constraints: We had a limited timeframe to collect data and have them classified by experts. This constraint necessitated efficient planning and execution of our data collection and classification process. However, we believe that with more time and the invaluable assistance of experts in the field, we can refine our methodologies and improve our data classification.

1.3 System Environment

BabyCode is a mobile application designed with mothers in mind, simplifying the process of understanding their baby's needs. Compatible with both iOS and Android, it offers a user-friendly interface for mothers to record and analyze their baby's cries. A key feature is its real-time processing capability, which ensures prompt and attentive care by immediately processing and displaying the results as soon as a cry is detected.

1.4 Project Software and Hardware Requirements

Software:

GUI Design**:** Figma for designing the user interface of the application.[6]

Programming Language**:** Python, along with AI libraries such as Keras and Librosa for machine learning and audio processing.

Hardware:

Smartphone with a Microphone**:** The end-user would need a smartphone with a functioning microphone to record the baby's cries.

Internet Connection**:** An internet connection would be required for the initial download of the app and for receiving updates or improvements to the machine learning model.

2.0 Research Background and Related Works

Paper	Author	Task description	Dataset
"Deep Learning Assisted Neonatal Cry Classification via Support Vector Machine Models"	Ashwini K, P. M. Durai Raj Vincent, Kathiravan Srinivasan, and Chuan-Yu Chang.	Deep Learning Assisted Neonatal Cry Classification via Support Vector Machine Models	dataset of pain, hunger, and sleepiness cries was collected from the infants born in National Taiwan
"Baby Cry Classification Using Machine Learning"	P.Ithaya Rani, P.Pavan Kumar, V.Moses Immanuel, P.Tharun, and P.Rajesh.	Classifying baby cries into categories like hunger, discomfort, exhaustion, or pain using Ml.	dataset of labeled infant cry recordings
"A Review of Infant Cry Analysis and Classification"	Chunyan Ji, Thosini Bamunu Mudiyanselage, Yutong Gao, and Yi Pan	The main task of the study was to review recent research works in infant cry signal analysis and classification tasks.	The paper reviews a broad range of literatures mainly from the aspects of data acquisition
"Recent Experiments and Findings in Baby Cry Classification"	Elena-Diana Șandru, Andi Buzo, Horia Cucu, and Corneliu Burileanu	The main task of the study was to conduct various experiments on a previously developed fully automatic system that attempts to discriminate between different types of cries.	Driven by the small dimension of Dunstan database
"Premature Infant Cry Classification via Deep Convolutional Recurrent Neural Network Based on Multi-class Features"	R. Sabitha, P. Poonkodi, M. S. Kavitha, and S. Karthik.	To classify the premature infant cry signal into different categories. The sound of the target cry signal is classified into five categories.	In this study, no new data are collected or examined for analysis, so this article does not meet the definition of data sharing.
" Infant cry classification by using different deep neural network models and hand-crafted features"	Nadia Maghfira Tusty, T Basaruddin, and Adila Krisnadhi	The paper focuses on the task of classifying infant cries. The researchers used signal processing methods such as hand-crafted features or image processing methods based on the spectral image of the cry	Donate a cry corpous Github

Paper	Techniques	Performance
"Deep Learning Assisted Neonatal Cry Classification via Support Vector Machine Models"	The main techniques used in the paper include (STFT) for transforming the auditory signals into spectrogram images, a deep convolutional neural network, for feature extraction from the spectrogram images, and a SVM classifier for classifying the extracted features.	• Measured by accuracy. Highest accuracy: 88.89% with SVM-RBF.
"Baby Cry Classification Using Machine Learning"	Main technique used in the paper is the K-Nearest Neighbors (K-NN) for classification. The K-NN classifier was shown to yield considerably better results compared to other classifiers	• Accuracy • KNN precision: 76.16% • SVM precision: 42% • Naïve Bayes precision: 45%
"A Review of Infant Cry Analysis and Classification"	The paper reviews signal processing, together with traditional machine learning classifiers such as KNN, SVM, and GMM, newly developed neural network architectures such as CNN and RNN are applied in infant cry research. It discusses pre-processing, various features like MFCC, spectrogram, and fundamental frequency, and how these features can train classifiers.	• Capsule Network accuracy: 61% • SVM accuracy: 71.68% • CNN accuracy: 89% • CNN-RNN accuracy: 94.97% • fuzzy Network accuracy :97.96%
"Recent Experiments and Findings in Baby Cry Classification"	Main Techniques: The main techniques used in the paper include Gaussian Mixture Models for classifying different types of cries.	• Accuracy, no specified value.
"Premature Infant Cry Classification via Deep Convolutional Recurrent Neural Network Based on Multi-class Features"	Main Techniques: The main techniques used in the paper include the (MFCC), (BFCC), and (LPCC) features for feature extraction, and a deep convolutional recurrent neural network (DCR net) for classification.	• Accuracy: 97.27% • Recall • F1 score
" Infant cry classification by using different deep neural network	The researchers used a broad range of features, including Mel-Frequency Cepstral Coefficients (MFCC), spectrogram, and fundamental frequency.	• SVM accuracy: 87.2% • RNN accuracy 77.4 • PNN accuracy 91.0

models and hand-crafted features”	They used variety of machine learning classifiers such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Gaussian Mixture Models (GMM), as well as newly developed neural network architectures such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN)	

As a result, in our project, we utilized a Convolutional Neural Network (CNN) for classifying different types of baby cries, achieving high accuracy. In contrast, the paper "Infant cry classification by using different deep neural network models and hand-crafted features", that uses same dataset as ours, achieved impressive results in the classification of infant cries. The researchers conducted experiments with various models and features, including 1D CNN model, transfer learning, texture analysis methods, hand-crafted features, and their combinations.

They found that texture analysis methods were insufficient, while hand-crafted feature sets and spectrogram and scalogram images provided high success while we used a combanition of different feature extraction methods. Interestingly, the 1D CNN model showed lower success than traditional classifiers and transfer learning models, as for ours it achieved the highest scores.

3.0 Proposed Methodology

3.1 Pipeline of the Proposed Methodology

This Section describes the proposed project, the solution is depicted in Figure 1 and includes seven main stages: data collection, preprocessing, feature extraction, feature selection, model training, model evaluation, and Deployment.

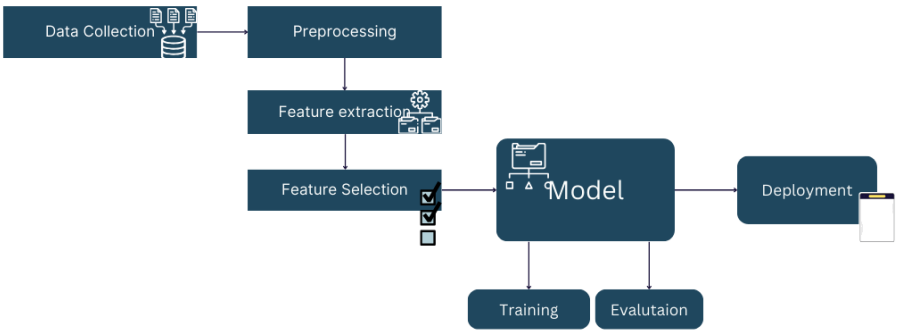

Figure 1: Pipeline of Methodology

3.2 Technical and Implementation Description

The Donate a Cry Corpus dataset from GitHub, featuring baby cries classified into five categories, was utilized. The dataset, already cleaned, required minimal preprocessing. Using the Librosa library in Python, various features were extracted, including Mel-Frequency Cepstral Coefficients (MFCC), Mel Spectrogram, Chroma, Spectral Contrast, and Tonnetz. Specific extraction settings included 40 MFCCs per sample with 1024 FFT, 160 hop (10ms @ 16kHz), 400 window (25ms), and the Hann function, alongside 12 chroma, 128 mel spectrogram, and 7 spectral contrast bands. These features were then combined for training a Convolutional Neural Network (CNN). The model was trained with a batch size of 64 and 50 steps per epoch over 100 epochs,

evaluated on accuracy and loss metrics. Finally, the model facilitated the creation of a user-friendly interface using Figma, with the application coded in Python. We developed a user-friendly graphical user interface (GUI) to bring the power of machine learning to the fingertips of users. This GUI, which is showcased in the following figures Figure:2 & Figure:3, was designed using Figma.

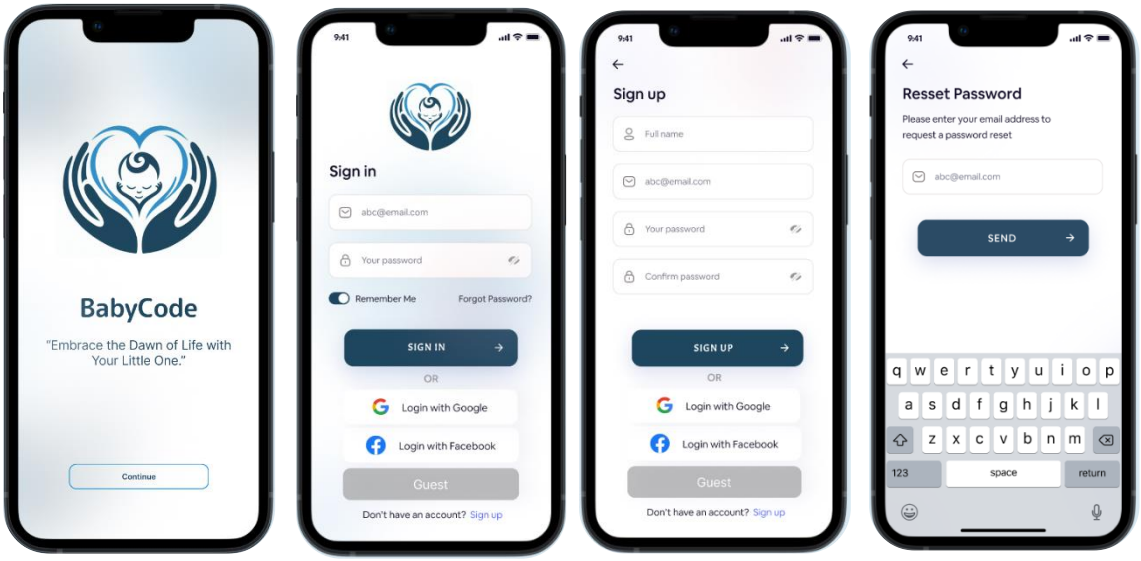

Figure 3:Starting and Log In

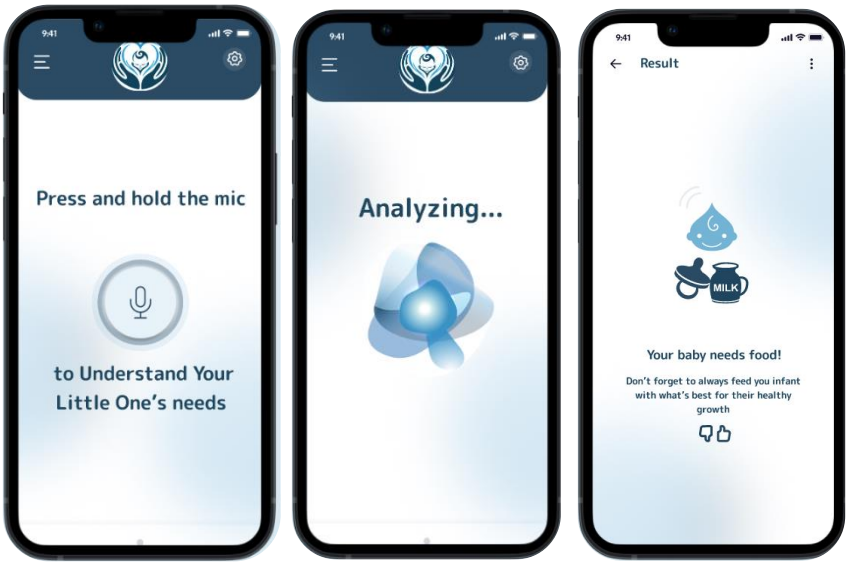

Figure 2:Predicting and Results

3.3 Dataset Description

In our project, we utilized the Donate-a-Cry Corpus dataset from GitHub, which is a rich collection of user-uploaded baby cries. Initially, the audio samples were in a uniform WAV format with a bit and sampling rate of 128 kbps and 8kHz. However, in our model, we adjusted the sampling rate to 16kHz, which, as per our study, was a better range to capture all the necessary frequencies. We ensured that the dataset only contained actual baby cries by manually removing non-cry data, such as white noise, baby chat, and adults mimicking baby cries. Afterwards, the data was divided into five classes, totaling 457 audio files. These audio files included 8 files for ‘Needs Burping’, 16 for ‘Belly Pain’, 24 for ‘Being Tired’, 27 for ‘Being Discomfort’, and 382 for ‘Being Hungry’. We noticed a potential bias in the dataset, particularly in the 'Being Hungry' category. To address this, we incorporated a max pooling layer into our model. This helped to alleviate overfitting and improve the model's ability to generalize across all categories.

3.4 Data Preprocessing

In the BabyCode project, data preprocessing was not necessary, due to the exceptional quality of the dataset used. Upon extraction from the Donate-a-Cry Corpus, the dataset was pre-cleaned and organized. By enabling a smooth transition to the feature extraction step, this greatly reduced the amount of time and resources needed. It illustrates the significance of high-quality, preprocessed datasets for machine learning projects. In addition, it’s worth noting that an important step in the BabyCode project was to determine the audio channel configuration of the dataset. A code snippet, as shown in Figure 4, was executed to ascertain whether the audio was stereo or mono. The results confirmed that the audio was indeed mono. Mono audio, having only a single channel, simplifies the process of feature extraction and ensures consistency across all audio samples in the dataset.

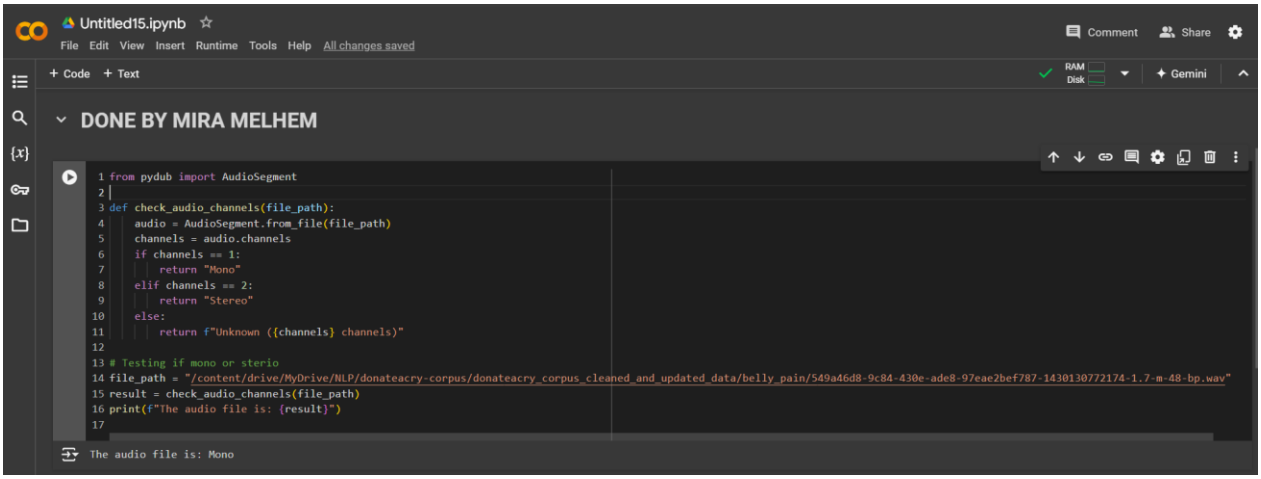

Figure 4:Test if Mono or Sterio

3.5 Features Extraction

In our project, we utilized a variety of features extracted from baby cries. These included Mel-Frequency Cepstral Coefficients (MFCCs) look at Figure 5, which provide a compact representation of the power spectrum of an audio signal, capturing the unique characteristics of different types of baby cries. We also used a Mel scale spectrogram, as shown on the right in Figure 5 which captures the perceptual content of the baby cries by mimicking the non-linear human ear perception of pitch.

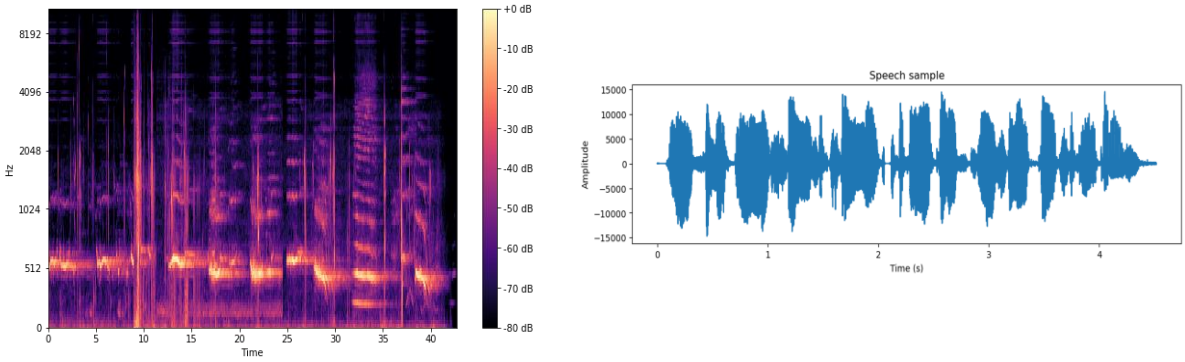

Figure 5:MFCC & Mel Spectogram examples

Chroma features were used to capture pitch-related characteristics of the cries, while spectral contrast helped distinguish between different types of cries based on their tonal quality. Lastly, we used Tonnetz features to capture certain harmonic relations in the cries. Each of these features provides a different perspective on the audio data, and together they create a more complete and informative representation of the baby cries. This comprehensive approach to feature extraction significantly contributes to the performance of our classification model.

3.6 Features Selection

In the initial stages of preparing our machine learning model, we converted the features and labels into numpy arrays for efficiency. We then encoded the categorical labels as integers using a Label Encoder for simplicity. The next critical step was feature selection, which plays a crucial role in determining the effectiveness of the model. For this purpose, we used the SelectKBest method. This method selects the 'K' best features based on a given scoring function. In our case, the scoring function was the ANOVA F-value, and 'K' was set to 10, meaning that the top 10 features were selected. Finally, we split the data into training and testing sets to commence the training process.

3.7 Features Classification

For the task of classifying features, a 1D Convolutional Neural Network (CNN) [8], was employed, leveraging its powerful capability to automatically learn spatial hierarchies in the data. Initially, the feature matrices for training and testing sets were reshaped to be compatible with the CNN input requirements, specifically converting them into 3-dimensional arrays where the third dimension represents a single channel. A Sequential model was constructed, starting with a convolutional layer using the ReLU activation function to introduce non-linearity. This was

followed by a max pooling layer with a pool size of 2 to down-sample the input representation, reducing its dimensionality and controlling overfitting. The output of the convolutional and pooling operations was then flattened into a 1D vector, which was passed through a dense (fully connected) layer with ReLU activation. The final output layer used a softmax activation function to produce probability distributions over the classes, aligning with the number of unique labels in the dataset as you can see from Figure 6.

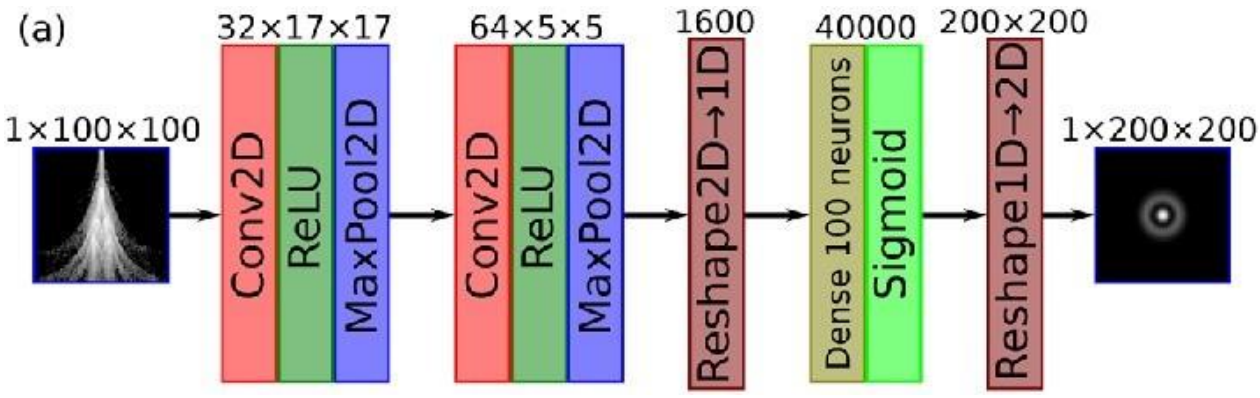

Figure 6: CNN Model [9]

The model was compiled using the Adam optimizer, which adjusts the learning rate during training. The training process was conducted with a batch size of 64, iterating over the dataset for 100 epochs, and processing 50 steps per epoch.

Data selection for Training, Validation, and Testing	**Value**
The proportion of data for training	80%
The proportion of data for validation	*20%*
Optimizer	Adam
Epochs	30
Mini Batch Size	64

4.0 Experimental Results and Analysis

4.1 Performance Measures

The CNN model then was primarily assessed using two metrics: accuracy and loss. The ability of the model to produce accurate predictions is reflected in accuracy, which gives a clear indication of the percentage of correctly classified occurrences among all instances. Furthermore, because the project is a classification task that involves multiple classes, sparse categorical cross-entropy is employed to address loss. By examining both accuracy and loss, a comprehensive and insightful assessment of the model's predictive prowess and reliability is achieved, offering a nuanced understanding of its performance and potential.

4.2 Experimental Results

As the results elegantly displayed in Table 1 below reveal, the outcomes of our scientific endeavor demonstrated that the 1D CNN model performed with an exceptional level of finesse on both the training and validation datasets. The model, a virtuoso in its field, showcased its robust fit to the training data with a staggering accuracy of 0.9513, a testament to its precision, and a remarkably low training loss of 0.1325, an indicator of its efficiency.

During the validation phase, the model achieved an accuracy of 0.8609 and a loss of 0.3185. These validation metrics illuminate the inherent challenge of ensuring a model’s generalizability beyond its training set, as they reveal a slight dip in accuracy and a rise in loss when confronted with unknown data. Refer to Table 1.

Evaluation metric	Accuracy	sparse categorical cross-entropy
Training	0.9513	0.1325
Validation	0.8609	0.3185

Table 1:Metric results

By scrutinizing both accuracy and loss, we achieve a comprehensive and insightful assessment of the model's predictive prowers and reliability. This offers a nuanced understanding of its performance. Figure 7 elegantly unveils the test results from a delicate snippet of our little ones' vocal expressions.

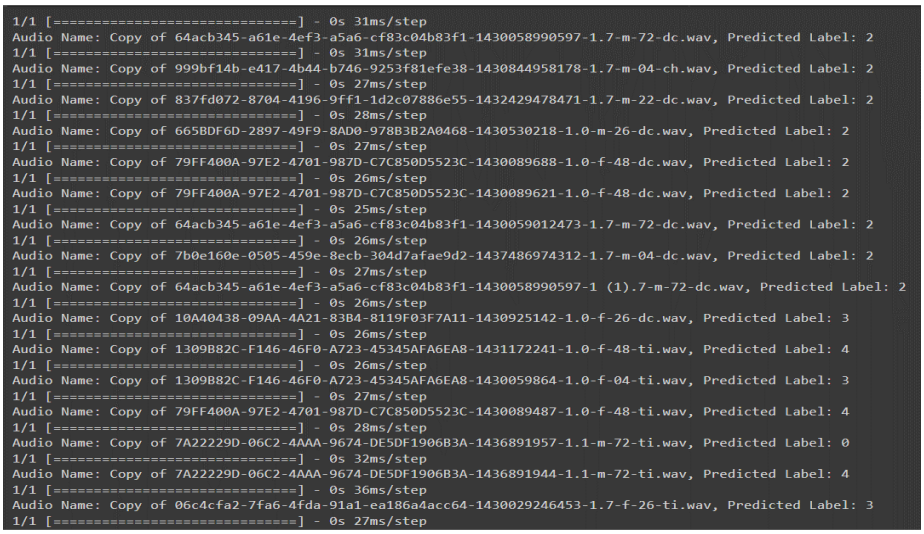

Figure 7:Test Results

5.0 Conclusions and Future Works

5.1 Strengths

Two notable strengths were key for the project, despite the small dataset, the 1D CNN model achieved high accuracy, showcasing its adaptability and robustness in making accurate predictions with limited data. Also, using meticulous preprocessing and feature extraction techniques, including MFCC and Mel Spectrogram, provided rich representations of audio signals, enabling the model to discern subtle patterns within the data. These strengths show how the project is able to overcome challenges and achieve commendable performance in classifying baby cries, making it easy for future advancements in this domain.

5.2 Weaknesses

The dataset's small size poses challenges for the model's generalization. Limited samples may lead to overfitting or reduced performance on new data. Additionally, without expert evaluation, label accuracy and reliability are uncertain, potentially impacting the model's effectiveness. Addressing these weaknesses by augmenting the dataset and seeking expert validation is crucial for improving the model's reliability and applicability in real-world scenarios.

5.3 Future Works

For future work the project will leverage larger datasets to improve the model's performance and generalization capabilities. By collecting more extensive datasets containing diverse samples of baby cries, the model would have access to a broader range of scenarios. Additionally, conducting thorough analyses on the dataset with input from domain experts would be beneficial. Experts' insights will add value and help validate the dataset labels, ensuring their accuracy and reliability.

Furthermore, building real-time application of the model, where it interacts with live data streams to provide immediate insights into a baby's condition. This could involve developing systems that continuously monitor and analyze baby cries in real-time, adjusting the algorithm's parameters dynamically based on the baby's needs and characteristics. Overall, future work in these areas has the potential to advance the field of baby cry classification and contribute to improved healthcare and childcare practices.

Conclusion

In the grand finale of the BabyCode project, the power of machine learning was harnessed to decode the language of baby cries. At the core of BabyCode lay a Convolutional Neural Network trained on carefully selected features, achieving an impressive accuracy of 0.8609 and a loss of 0.3185, affirming the model's reliability and project success. The project stands as a testament to the potential of machine learning in providing valuable insights and solutions. It illuminates the path to understanding and addressing the needs of babies, transforming the way we interpret their cries.

References

[1] K, A., Vincent, P. M. D. R., Srinivasan, K., & Chang, C. Y. (2021b, June 10). Deep Learning Assisted Neonatal Cry Classification via Support Vector Machine Models. Frontiers in Public Health. https://doi.org/10.3389/fpubh.2021.670352

[2] Rani, P. I., Kumar, P. P., Immanuel, V. M., Tharun, P., & Rajesh, P. (2022). Baby Cry Classification Using Machine Learning. International Journal of Innovative Science and Research Technology, 7(3), 677. Retrieved from https://ijisrt.com/assets/upload/files/IJISRT22MAR645.pdf

[3] Ji, C., Mudiyanselage, T. B., Gao, Y., & Pan, Y. (2021, February 5). A review of infant cry analysis and classification. EURASIP Journal on Audio, Speech, and Music Processing. https://doi.org/10.1186/s13636-021-00197-5

[4] Șandru, E. D., Buzo, A., Cucu, H., & Burileanu, C. (2018, January 1). Recent Experiments and Findings in Baby Cry Classification. Springer eBooks. https://doi.org/10.1007/978-3-319-92213-3_37

[5] Sabitha, R., Poonkodi, P., Kavitha, M. S., & Karthik, S. (2023, August 1). Premature Infant Cry Classification via Deep Convolutional Recurrent Neural Network Based on Multi-class Features. Circuits, Systems, and Signal Processing. https://doi.org/10.1007/s00034-023-02457-5

[6] Figma. (n.d.). Figma. https://www.figma.com/files/recents-and-sharing?fuid=1183095550376726809

[7] Gveres. (n.d.). GitHub - gveres/donateacry-corpus: An infant cry audio corpus that's being built through the Donate-a-cry campaign - see http://donateacry.com. GitHub. https://github.com/gveres/donateacry-corpus.

[8] Convolutional Neural Network (CNN). (n.d.). TensorFlow. https://www.tensorflow.org/tutorials/images/cnn

[9] Molodtsov, S., Volokitin, A., Ananyev, A., Medvedev, A., Stepanov, M., & Novikov, V. (2022). Restoration of the focal parameters for an extreme-power laser pulse with ponderomotively scattered proton spectra by using a neural network algorithm. ResearchGate. https://www.researchgate.net/publication/366492883_Restoration_of_the_focal_parameters_for_an_extreme-power_laser_pulse_with_ponderomotively_scattered_proton_spectra_by_using_a_neural_network_algorithm/figures?lo=1