Arab American University

Faculty of Graduate Studies

Department of Natural, Engineering & Technology Sciences

Master Program in Computer Science

Utilizing Deep Learning Models to Identify and Classify Various Types of Infant Cries.

Asma Jawdat Mohammad Shawabkeh

201720327

Supervision Committee:

Dr. Mouth Sabha

Dr. Ahmad M. Hasnah

Dr. Anas Samara

This Thesis Was Submitted in Partial Fulfillment of the Requirements for the Master Degree in Computer Science.

Palestine, 7/2025

Arab American University

Faculty of Graduate Studies

Department of Natural, Engineering & Technology Sciences

Master Program in Computer Science

Thesis Approval

Utilizing Deep Learning Models to Identify and Classify Various Types of Infant Cries.

Asma Jawdat Mohammad Shawabkeh

201720327

This thesis was defended successfully on 27 /07/2025 and approved by:

Thesis Committee Members:

Name	Title	Signature
1. Dr. Mouth Sabha	Main Supervisor	.....
2. Dr. Ahmad Hasasna	Member of Supervision Committee	.....
3. Dr. Anas Samara	Member of Supervision Committee	.....

Palestine, 7/2025

Declaration

I declare that, except where explicit reference is made to the contribution of others, this thesis is substantially my own work and has not been submitted for any other degree at the Arab American University or any other institution.

Student Name: Asma' Jawdat Mohammad Shawabkeh

Student ID: 201720327

Signature: Asma Shawabkeh

Date of Submitting the Final Version of the Thesis: 20/8/2025

Acknowledgments

I thank myself, the fighter and the ambitious

I also thank everyone who supported me in my educational career, especially the professors of the Arab American University

I thank my father, the first and strongest supporter

I thank my dear husband who bore with me the burdens of life in order to provide the appropriate environment to complete my studies

I thank Dr. Alaa El-Din Hoji and Dr. Mohamed Zidan for their help in classifying the sound of the infant's crying

I would like to express my sincere gratitude to the Arab American University and the College of Graduate Studies for the modern and supportive facilities and equipment they provided, which greatly assisted me in learning and conducting research effectively. I also extend my special thanks to the program coordinator and the program committee for the valuable guidance and assistance they provided throughout my studies, which contributed significantly to the successful completion of my academic projects.

Utilizing Deep Learning Models to Identify and Classify Various Types of Infant Cries.

Asma Jawdat Mohammad Shawabkeh

Dr. Mouth Sabha

Dr. Ahmad M. Hasnah

Dr. Anas Samara

Abstract

Current research emphasizes the development of effective models to aid specialists in the efficient classification of behavioral conditions, one of these fields is infant crying. Infant crying is a critical indicator of a child's health, functioning as a non-verbal communication method that conveys their needs, understanding the cause of the cry of an infant could be quite a challenging task, especially for parents and caregivers. Therefore, this problem becomes an area of research for researchers.

Deep learning utilization is increased in healthcare, such as private clinics and hospitals. The utilization of deep learning methodologies can improve earlier diagnosis; it enhances their ability to diagnose the causes of the infant crying without subjecting the infant to greater suffering. To those seeking to improve diagnostic tools for classification of infant cries, this study provides important insights into the factors of cry features that correlate lightly with cry function and function condition. The capability of deep learning techniques applied in this work for boosting diagnostic precision and results in pediatric healthcare is checked.

Techniques for converting audio to images using MFCC has been employed and then classified the resulting images with different deep-learning techniques (VGG16, DenseNet,

ResNet, and GoogLeNet). Appropriate matrices techniques were used to evaluate and compare the performance of these models.

Deep learning techniques have been examined in the classification of infant cries, utilizing data gathered from local care and global dataset platforms such as Kaggle. And then, a comprehensive analysis of the performance of various algorithms to identify the most suitable approaches for accurately classifying infant cries in diverse data environments is performed.

According to the result, ResNet and DenseNet performed better than the others, with ResNet reaching higher accuracy on global dataset and DenseNet following. The effectiveness of VGG-16, on the contrary, was lowest, reflecting that each algorithm must be evaluated on its own effectiveness relative to the dataset it is used with. With local data, ResNet has high accuracy and proves its reliability and robustness on this type of data.

in addition, the study highlights the importance of architectural improvements in enhancing model performance. The incorporation of a Random Forest (RFF) layer with Dense layers resulted in significant enhancements in model stability and accuracy. The architectural modifications suggest that hybrid approaches can significantly improve the model's capacity to learn complex patterns, which is essential for applications in medical imaging and diagnostics.

By the above, it follows that we can aid service providers in recognizing accurately and fast the real cause of an infant’s cry using the latest deep learning techniques. Also, A model can be generated and distributed to these centers of the future to benefit from it.

Keywords: Infant, deep learning, RFF, MFCC, classification

Table of Contents

List of Tables

List of Figures

List of Definitions of Abbreviations

ANNs	Artificial Neural Network
CNNs	Convolution Nural Network
DL	Deep Learning
ES	Ensemble
FN	False-Negative
FP	False-Positive
LM	Levenberg-Marquardt
LMS	Learning Management System
LSTM	Long Short-Term Memory network
ML	Machine Learning
MLP	Multi-Layer Perceptron
MLT	Machine Learning Techniques
MSE	Mean Squared Error
NNs	Neural Network
RFF	Random Fourier Feature
RNN	Recurrent Neural Network
TN	True-Negative
TP	True- Positive
LFCC	Linear Frequency Cepstral
BFCC	Bark Frequency Cepstral coefficients
GFCC	Gammatone Frequency Cepstral coefficients

Chapter One

1.1. Introduction

Crying is the most common way in which neonates and young infants can express their wants, adverse conditions, and feelings to the caregiver. Such cries may signify hunger, pain, and tiredness or maybe a call for attention or a call for a diaper change[1]. Numerous published research indicate that rates of baby crying differ across countries. The rhythms, sounds, and intensity of newborn crying fluctuate based on the underlying cause. The Dunstan system, created by Priscilla Dunstan, identifies five distinct styles of crying that newborns utilize to communicate their requirements. The infant produces a “Nah” sound when hungry, “Owh” sound when fatigued, a “Eh” sound when requiring burping, a “Heh” sound when uncomfortable (needing to be cleansed or experiencing temperature discomfort), and a “Eairh” sound when suffering from colic. It is noteworthy that any statistics from the World Health Organization (WHO) that correlate with patterns of infant crying could not be found. Abuse of crying causes unmet needs, ongoing discomfort for the infant, and stress for caregivers, which emphasizes the need for a good crying identification system[2] .

In the past, understanding the meaning of infant crying was based on the perception and knowledge of parents, caregivers, and healthcare providers. It has been found that parental identification of the cause of an infant’s crying lies in about 60-70%, depending on the previous experience and knowledge of grandparents and the community. Although these are somewhat subjective methods, they have been shown to be very effective and, at the same time, can be quite

inaccurate [3].This is because when different people hear a cry, they may interpret it in different ways and, therefore, act in different and sometimes wrong ways. This inconsistency can lead to less than adequate care and stress for both the infant and the caregiver [4]. In light of these problems, science has sought to devise better and less arbitrary ways to classify and interpret infant cries. The most viable solution to this problem lies in the use of artificial intelligence (AI) with a focus on deep learning (DL) [5].

Thanks to scientific advancements and the wisdom passed down from grandparents; we can now use techniques to assist new parents and carers in responding more quickly and efficiently to their children's cries. For example, when a baby is hungry, he may be sucking his finger, and when he is colicky, he may pull his hands and feet. When he wants to sleep, he rubs his eyes. However, when he needs to change his diaper, his cries are more consistent.

The challenge lies in analyzing the sounds emitted by children and then classifying those using techniques capable of categorizing such data. To move forward, we studied the latest methods for converting sound into signals or images from which audio features can be extracted. Among the prominent methods for converting audio signals is LFCC, which uses linear Frequency division without simulating the human ear to find sound features. It is also used to distribute energy evenly across the Frequency range, making it ideal for academic applications. BFCC uses the Barco scale, like MFCC, but with a greater range. GFCC is the best method for analyzing crying baby sounds because it simulates the human cochlea, which is sensitive to sound frequencies, especially in unstable environments like high noise, where very subtle sound changes occur. MFCC, the most prevalent method for analyzing crying baby sounds, yields good findings in most

circumstances. Although less accurate than GFCC, it is better for classifying sobbing sounds. Our study used MFCC, which is more frequent in health and medical systems, to analyze sound.

By analyzing sound using MFCC technology, various techniques can be implemented, including machine learning and deep learning algorithms. Artificial intelligence systems can utilize these methods to learn from data, recognize patterns, and make informed decisions. Several learning techniques have been proposed in the literature for classification tasks, such as Support Vector Machines (SVMs), K-nearest Neighbors (KNN), and Random Forest [6]. For more complex and nonlinear cases, deep learning offers a suitable solution due to its ability to model intricate patterns. Deep learning, a subset of machine learning, employs neural networks to enable data-driven learning. It is particularly effective for handling large datasets and modeling complex relationships, making it ideal for applications like image and speech recognition. Among the most popular deep learning architectures are Convolutional Neural Networks (CNNs) and Dense Neural Networks, each tailored to excel in specific types of tasks[7].

These AI techniques are very useful in the context of infant cry classification. Since cries of infants are available in large datasets, the AI models can be trained to classify the cries based on the acoustic features present in them [8]. This entails the breakdown of the cry signal into different components including the pitch, duration, intensity and spectral energy density. This is usually done through the feature extraction step where the different characteristics of the audio signal are selected and measured. These features are then fed into the AI model where the model gets trained to recognize certain patterns as belonging to types of cries [9].

Thus, the task can be solved using machine learning models like SVM and random forests, but they often demand much time and effort for feature engineering and are less capable of

handling highly complicated patterns. Deep learning models, however, can learn the features of the data in a hierarchical manner, therefore, identify specific features of the data that would be overlooked by other methods; that means CNNs and other deep learning models automatically find important data trends without having to do any feature engineering by hand. They use many layers to handle data. The top layers find simple features (like edges) and the lower layers find complex patterns (like faces or objects). This hierarchical learning takes into account aspects that more traditional models might miss [10].

In our works, CNNs can be employed to process the spectrograms of infant cries and these are graphical representations of the Frequency content of the given audio signal over time[10]. These spectrograms are then fed to CNNs, which through convolutional filters, learn about the features and pattern that is associated with different cry types. RNNs, specifically LSTM, can be utilized to map the cries and the temporal order of the sounds as well as the changes in the sounds to certain needs or ailments [11].

For this work, we converted infant cry sounds to spectrogram images using the Mel Frequency Cepstral Coefficients (MFCC) idea, which can be understood and classified by deep learning techniques. FCC is an extraction feature method used for audio and speech processing that extracts the most meaningful features of sound in terms of Frequency components. We attempt to leverage the tool of spectrograms, to convert the infant cries into a format that deep learning algorithms can ingest faster, in order to better inform the model of the possibilities for advanced recognition in the audio data. This changed the model to focus on sound characteristics that are important such as pitch, tone, and duration, making it more accurate when classification. In this context, the use of MFCC serves as a strong technique that functions by bridging the gap between

audio signals and the visual representation of data, making a contributory input to the performance of the deep learning models in the task to classify infant cries based on their different emotional or physical need [12].

These DL techniques, when coupled with the infant cry classification systems can help in making the identification of cry more accurate and reliable. These systems can help parents, caregivers, and healthcare givers to have a good understanding of the needs of infants and take an appropriate action on them. For instance, a mobile application which has the feature of cry recognition using artificial intelligence can give recommendations to the parents on what is likely to be wrong with the baby and what the parents should do. In a clinical environment such systems would assist pediatricians and nurses in determining the state of babies especially when they are unable to communicate.

There is great potential for AI in classifying infant cries. As AI technologies continue to develop, subsequent generations of models will likely be of greater complexity and accuracy, detect an even broader array of cry types, and offer insights into a baby's health and development [12]. Geographical diversity contributes significantly to the enhanced accuracy of predictions. Future works can aim to fill in gaps by creating systems with more diverse data by doing the following: First, large datasets must be polished and collected from various sources to develop better systems; Second, examination of nonstandard approaches to the conversion of audio signals into images, which potentially can reveal different features than conventional MFCCs and Third, use of high-level deep learning methods to classify and differentiate between cries (e.g., hunger, pain, discomfort) with different acoustic parameters. Such developments could greatly improve care for infants and relief for carers.

1.2. Problem Description

Infants communicate their needs and discomfort mainly by crying; thus, it is important to be able to interpret and respond to these cues. However, deciphering the reason for an infant's cries remains one of the biggest challenges parents and carers face. That is especially common for new parents, who may be unable to tell the difference between a babies's crying and hunger, pain, tiredness, or something else. Misinterpretation of infant cries can lead to unmet needs, prolonged discomfort for the baby, and increased stress and anxiety for carers.

The key issue for this study is to classify infant cries using AI methods with high accuracy and reliability. This entails several important questions. The crying of infants is not always high or low-pitched and not always long or short in its presentation [13]. This problem occurs when clear patterns of need or condition related to activity levels are found. In addition, finding many labeled data sets of infant cries from different sources and areas for training AI models is a problem. Acquiring and labeling large amounts of cry data is a tedious and time-consuming process, thus limiting the amount of data that may be used to train a model. Furthermore, it is important to note that cry sounds are not easy to classify since there is a need for advanced algorithms that can recognize the complex patterns in the data, and this requires the use of ML and DL models. It is also important to note that the feature extraction methods used in the models to identify cry sounds are also very important to the success of the models [3].

In addition, there is the issue of developing a system that teaches how to perform well in any infant and any given environment. Some factors that may influence the performance of the AI models include differences in the recording environment, interference, and differences in the

crying patterns of infants. The system must function well in real-life situations, especially after its development. This research aims to meet these challenges through the identification and enhancement of AI-based approaches for the classification of infant cries. Through DL, the research intends to develop effective models to help understand the cause of the cries made by infants and offer ready recommendations to the carers. In the long run, the aim is to design a robust, flexible, and intuitive system that can be implemented in a wide range of care contexts, ranging from homes to hospitals to enhance infant care and decrease the stress levels of carers.

1.3. Research Questions

What ways make it most convenient to convert infant cry sounds into images that work well with deep learning models? This research attempts to convert audio signals, naMely infant cries, into visual representations including spectrograms or any other image format that encodes some salient characteristics of the sound. The study focuses finding technique that make processing and classifying uses of deep learning algorithms easier.

To what extent are cry types recognizable by designing model based on the acoustic parameters of cries? This question aims at designing and comparing the performance of different deep learning models in identification of the cause of infant crying. The emphasis is to calculate the accuracy and loss performance of these models in correctly categorizing the cry. The main concern is the accuracy, and loss of these models in their ability to correctly categories the type of cry.

1.4. Research Objectives

This research aims to achieve several objectives focusing on the local dataset and its implications:

This work seeks to use well known method for mapping infant cry sounds to realizable images for use in deep learning models depending on previous works. More specifically, the project will explore the issue of converting audio signals (e.g., infant cries), into visual representations (e.g., spectrograms or other images), which retain the important characteristics of the sound.

The performance of various types of DL models will be evaluated to detect and classify different classes of infant cries with their acoustic parameters. This does compare DL techniques to which can be more efficient, more specific, more sensitive, and ultimately more accurate at determining reasons for crying. Through this research, we aim to investigate which model is most successful in classifying infant cries as well as how these models can be used to enhance cry classification systems in real practical applications.

1.5. Contribution

This research is effective in making several contributions to infant care and deep learning. Firstly, it uses two types of datasets locally and globally. Also, the study intends to augment these algorithms' ability to perform and classify infant cries by using techniques, such as Mel-Frequency Cepstral Coefficients (MFCC). In this work, we increase the understanding of how audio signals can be transformed into an understandable format by using suitable techniques, thus making it easier to determine whether a cry relates to the infant’s physical or emotional needs.

Second, this study serves as a contribution to the cry classification machine learning and deep learning model development, through the comparison of some machine learning techniques as well as deep learning techniques in terms of performance regarding their ability in detecting and categorizing cry type. The research analyzes these models on ways of classifying kicks of pain, hunger, or discomfort such as occurrences of hunger, pain or discomfort. The results will enrich previous work by analyzing cry patterns through the extraction of key acoustic features to identify the most suitable approaches for cry classification. In short, this research is intended to improve diagnostic systems in infant care, providing new ways to identify infant cries more precisely and reliably.

Chapter Two

Background and Literature Review

2.1. Introduction

The study of infant cry classification research has advanced greatly, because of the critical need to recognize and address infant signals. Conventionally, the previous studies have used their perception of the cry of the baby which has been developed over time through experience and intuition. However, the recent improvement in artificial intelligence has given more credible and reasonable methods of identifying these cries [14].

Other studies have also attempted to categorize infant cries using classical machine learning methods. Newborn babies' crying is recorded and analyzed; audio information such as pitch and loudness is extracted. The classical deep learning algorithm then is used to decide why the baby is crying. Among these algorithms, the most commonly used is the "decision tree," a set of simple rules based on questions asked in sequence resulting in a probable explanation for the crying of an infant, such as "Was the baby's crying loud? and then another question, "Was the crying ongoing for an extended period of time?"[15]. 'Nearest neighbor algorithms (K-NN)' were also applied which compares a new cry to one of previously classified as cry and the method assumes that similar sounds have the same reason. This is an intuitive algorithm. It converges to the true LBP distributions in case of rich training data. We also apply the "support vector machine (SVM)" algorithms, that is, intelligent algorithms that draw a line discriminating both the types of cries considering their characteristics, discriminate whether or not new sounds belong to cries

passing through the line. SVM algorithm is good for complex data and accurate results, so it is appropriate for this application [16] [17].

Early research utilized basic acoustic features and machine learning algorithms like SVM and k-NN, but these methods struggled with the complexity of audio signals [15]. Recent developments in deep learning, such as “Convolutional Neural Networks (CNNs)” for spectral analysis and “Recurrent Neural Networks (RNNs)”, particularly “Long Short-Term Memory networks (LSTMs)”, have improved the ability to identify temporal patterns in infant cries [17].

The implementation of deep learning to the problem of infant cry classification has proved to enhance accuracy and reliability . Research has been done using various data sets, including data collected from different populations as well as different settings to help increase the applicability of the DL models [13].Such approaches have stressed the need for feature extraction and data expansion strategies to overcome difficulties that include a lack of labeled data and variation in cry patterns [18].

Besides, the demand for the application of DL-based cry classification systems in practical settings has grown [19]. Research has attempted to investigate intuitive interfaces and applications supporting the comprehension and reaction to the cries of infants in real-time by caregivers. Such systems are designed to alleviate stress among caregivers, improve infant care, and may also assist in the identification of health problems early on through repeated and accurate cry analysis [20].

Therefore, the initial work provided the basis for categorizing cry signals through the acoustic features with the help of traditional machine learning methods, and advances in deep learning have opened a new chapter of more precise and smart cry recognition. This review

provides the background for future studies on the enhancement of DL models for real-life applications in infant care and the continuous enhancement of caregiving practices.

2.2. Significance of Infant Cry Classification in Caregiving and Health Monitoring

Classification of infant cries is important in nurturing and identifying possible sickness of the infants, which is why it is necessary to know and differentiate such cries[21]. Cries which babies make may be different and depend on the child's requirements, discomfort, and age. The first attempts in the field were devoted to the identification of the different types of crying which can be differentiated based on the reasons as hunger, pain, discomfort, and tiredness and thus help the caregivers to understand what it means [22].

Categorical distinction of cry types has conventionally been based on acoustic properties including pitch, intensity, duration and dynamics of cry patterns. Research has established that every cry has different properties which can be noticed if one analyses the acoustic properties of each type of cry. Such hunger cries are presented with brief intervals between the cries and a rising tone while pain cries may have sudden, sharp, and loud bursts[23].

The need to correctly categories the cry of the infant is because it has a direct bearing on the management of the child. Incorrect interpretation of cries may result in delayed response to the infant's needs, increased crying of the infant and stress for the caregivers. Hence, it is vital to create methods that can be dependable, for instance, the use of artificial intelligence to classify the patients [16].

2.3. Advancements in Infant Cry Classification: Integrating Machine Learning and Deep Learning

2.3.1. Traditional Machine Learning Approaches (SVM, k-NN, Random Forest)

In the early research work on infant cry classification, basic acoustic features and ML algorithms like SVM and k-NN were used, but these procedures were limited in their capability to overcome the complexity of the audio signal [15]. More recently, deep learning models as "convolutional Neural networks (CNNs)" have been used to analyze audio spectra, and "recurrent neural networks (RNNs)" and their improved version, “Long Short-Term Memory networks (LSTMs) “ , have been used to more effectively identify temporal patterns of infant cries [24].

Recent developments in artificial intelligence, specifically with machine learning and deep learning methodologies have enhanced cry classification these methodologies leverage large repositories of cry samples and machine learning algorithms to develop models which can classify different types of cry patterns with high levels of accuracy. The initial capability of ML algorithms like SVM and "Random Forest" was to classify the data according to the labelled data. Yet, more complex models like CNNs and RNNs have been suggested as better alternatives to models that are capable of extracting more complicated features from cry signals [25].

Initial studies on infant cry classification relied primarily on simple machine learning techniques, such as the k-NN algorithm and SVM [26] . These methods were based on the

measurement of basic acoustic parameters of the voice, such as pitch, intensity, and duration, to differentiate cries as hunger, discomfort, pain, tiredness, and so on. While they produced promising results during that time, these methods had a few problems with regard to richness and variability of crying sounds, and were not very practical in actual application [15].

The cry recordings in [27] were pre-classified by need (such as discomfort, hunger, pain), and the classification was performed by SVM, k-NN, Naive Bayes, and Random Forest. It was observed that MFCC features provided higher classification accuracy compared to LFCC, and SVM and Random Forest provided higher performance compared to the other classifiers. The combination of MFCC-SVM had the best performance, and it is shown that it would be a candidate for real usage in automatic infant cry recognition systems.

2.3.2. Deep Learning Models (CNNs, RNNs, LSTMs)

Deep learning techniques have made crying classification systems more efficient in the last few years. "Convolutional neural networks (CNNs)" were particularly useful for spectrogram analysis of crying signals because they are capable of identifying spatial patterns of Frequency content for different crying types. CNNs possess better feature extraction from data and achieve more accurate classification compared to other machine learning techniques.

"Recurrent neural networks (RNNs)" also improved the temporal modeling of scream sequences, particularly extensions of long short-term memory (LSTM) networks. These networks are highly effective at recognizing patterns in scream signal sequences, which is important when decoding the context of scream signals. Accordingly, the transition from traditional machine learning approaches to deep learning approaches for infant cry classification has significantly

boosted the studies and applications of this field. Backed by "convolutional neural networks (CNNs)", regression neural networks (RNNs), and other deep learning technologies, researchers are still seeking new approaches and applications to make current AI systems more precise, efficient, and easy to use in infant care for improved infant health across the world.

In the paper [28] the researcher relied on “convolutional neural networks (CNNs)” to extract features of crying signals from an audio context, enabling the learning of the time-Frequency features required to identify different types of crying. This was followed by the use of “recurrent neural networks (RNNs)” and “long-short-term memory (LSTM)” networks, where temporal relationships are learned within the features. As mentioned earlier, CNNs operate through spatial hierarchies, while RNNs, particularly LSTMs in this case, can identify sequential patterns and are therefore a good model for infant cry classification. The cry signals in the database are diverse, and the performance metrics of the proposed model consist of precision, recall, accuracy, and F1 score.

The scientific paper [29] sought to analyze the effectiveness of CNN and RNN in identifying the type of cry that an infant produces: hunger cry, pain cry, or tired cry. Itch cries are one of the ways that a baby can convey its needs, but the problem with identifying specific signals from the general number of sounds is that all the sounds are very similar. This paper aims at developing an efficient model that can assist in determining the type of cry made by an infant so as to enable the caregiver/parent to meet the needs of the infant.

While the paper [30] Focus on the classification of infant cry and the DNN models and hand-crafted features used in the classification of the cry of the infant. Another problem which the study also seeks to address is the pain of identifying the different cries typical of infants. In this

way, the authors offer the chance to enhance the cry classification systems' performance and reliability by employing both combined DNN models and hand-designed features.

2.3.3. Hybrid systems

In paper [31] proposes a combined deep learning approach that combines "convolutional neural networks (CNNs)" and "recurrent neural networks (RNNs)" for classifying infant cries into categories as hunger, pain, and discomfort. The CNN is utilized to uncover spatial features from the spectral representations of crying audio signals, while the RNN (LSTM) discerns temporal dependencies between the crying sequences. The study reveals that the combined architecture is more accurate than single models, and that the fusion of spatial and temporal information greatly enhances the system to properly analyze different types of infant cries. This approach is promising for real-time pediatric care applications and early diagnosis.

In the paper [32], the researcher relied on "convolutional neural networks (CNNs)" to extract features of crying signals from an audio context, enabling the learning of the time-Frequency features required to identify different types of crying. This was followed by the use of "recurrent neural networks (RNNs)" and long-short-term memory (LSTM) networks, where temporal relationships are learned within the features. As mentioned earlier, CNNs operate through spatial hierarchies, while RNNs, particularly LSTMs in this case, can identify sequential patterns and are therefore a good model for infant cry classification. The cry signals in the database are diverse, and the performance metrics of the proposed model consist of precision, recall, accuracy, and F1 score.

The approach used in the research paper [14] involves creating a massive library of potential cries that an infant could produce, with each cry corresponding to a specific wish. In this work, features are extracted with the help of "convolutional neural networks (CNNs)", due to their ability to capture the characteristics of acoustic crying signals in the time-Frequency domain. CNNs, "recurrent neural networks (RNNs)", and long-short-term memory (LSTM) are then used to learn the temporal relationships of the features. These layers, both CNNs and LSTMs, have the advantage of enabling the model to analyze the spatial and temporal patterns of the cries to be classified. The results indicate that the proposed CNN-LSTM model has superior classification capabilities for infant cries compared to other machine learning models. The model was validated using accuracy, recall, precision, and F1 score. The authors would like to emphasize the practical value of the current research by focusing on the development of instant cry recognition systems, which may be useful for parents and healthcare workers. As for future work, the plan is to increase the sample size and apply the proposed model in practical life to improve its results.

The paper [33] with the aim of ascertaining how deep learning is used in the classification of the different kinds of cries that babies produce. Relevance of the study can be drawn from the common knowledge that babies use cries to communicate their needs like hunger, pain, or discomfort. It would be of tremendous benefit to parents and caregivers if they are able to classify cries accordingly so that the needs of the infants are taken care of soon enough. The study is related to the problem of cry types of sound homophony and proposes a deep learning solution to the problem.

The approach is based on the creation of a vast database of the possible cries a baby can make, and each cry was correlated with a specific need. Feature extraction is achieved with the

help of "Convolutional Neural Networks (CNN) "in this study since they possess the capability of capturing the features of the cry audio signals in the time Frequency domain. Then the "convolutional neural networks (CNN)", "Recurrent Neural Networks (RNN)" and the "Long Short-Term Memory (LSTM)" networks are applied in a sequence so as to learn the temporal relations of the features. This network of layers both CNN and LSTM have their utility because they enable the model to have the capability of analyzing the spatial and temporal patterns of the cries to be classified.

The results obtained exhibit that the proposed CNN-LSTM model has excellent classification potential of infant cry compared to other machine learning models. The model is validated using accuracy, recall, precision and F1-score. The authors would like to point out the applicative utility of the current study by emphasizing the development of real-time cry recognition systems that can be beneficial for parents and health experts. For the upcoming study, it is planned to take a higher sample size, as well as apply the proposed model to practice to improve the model outcomes.

The approach followed in the paper [34]involves creating a huge library of possible cries that an infant could produce, each of which corresponds to a specific wish. In this work, features are extracted with the help of “Convolutional Neural Networks (CNNs)” due to their ability to capture the characteristics of acoustic crying signals in the time-Frequency domain. CNNs, “Recurrent Neural Networks (RNNs)”, and Long Short-Term Memory (LSTM) are then used to learn the temporal relationships of the features. This multilayer network, whether CNN or LSTM, has its advantages in enabling the model to analyze the spatial and temporal patterns of the cries to be classified. The results indicate that the proposed CNN-LSTM model has superior

classification capabilities for infant cries compared to other machine learning models. The model was validated using accuracy, recall, precision, and F1 score. The authors would like to emphasize the practical value of the current research by focusing on the development of real-time cry recognition systems, which may be useful for parents and healthcare workers. For future work, the plan is to increase the sample size and apply the proposed model in real-world settings to improve its results.

The assessment and classification of infant cry is an important field of study which aims at tending to the needs of the infants through their cry. In review[20], several methods and new advancements are highlighted especially on the use of machine learning algorithms. For one of the discussed approaches, it is based on the most popular techniques, naMely CNNs and RNNs for cry classification. The CNNs are employed to extract the challenging features from cry signals and then the time-Frequency representation, and spatial dependencies are examined. These features are then fed into the RNNs as LSTM useful in the extraction of temporal dependencies important in the differentiation of cries as hungry, painful or uncomfortable cries.

The review also identifies the data acquisition and preparation, and the role of cry databases for creating the models with enough variability and the label quality. These requirements raise the issue of instability in cry patterns that has to be solved by the methods used in the study, primarily because of the age of the subjects, their illnesses, and noise in the environment. Evaluation of such models involves the use of parameters like accuracy, recall rates, precision and F1-score which depicted the possibility in classifying differences in cry patterns.

Similarly, the review also acknowledges the agenda of the field as it goes on to establish how it has the possibility of integrating systems that will help monitor the infants in real-time and

wearable devices that will be able to interpret the cry of the infant and offer the right response in the right time. These can contain ideas on childcare by providing the caregiver with the relevant information at the right time when the infant requires it in order to reduce tension and stress on the right way to handle the child. Therefore, it can be stated that the use of CNNs and RNNs in analyzing infant cries is a major contribution in enhancing pediatric health and parental care of infants with the help of the best ML algorithms that can work well with the baby's communication system.

2.3.4. Transformer-Based Approaches

The study proposes [35] a new deep learning approach to classify infant cries using a hybrid architecture that integrates excitation-squeeze residual networks (SE-ResNet) with a transformer encoder. The goal is to accurately classify infant cries into three main emotional states: hunger, pain, and discomfort, which are key indicators of infant needs and well-being. A robust dataset was constructed from three publicly available corpus, naMely Corpus, Chilanto, and ESC-50. The researchers also used 5,204 audio samples extracted from YouTube, with a sample size of 2,080 expressing hunger, 1,760 expressing discomfort, and 1,364 expressing pain. The researchers then extracted features using MFCC and its derivatives. The SE-ResNet network was used to enhance the capacity of the convolutional layers through adaptive recalibration of each channel, while the transformer encoder captures long-term dependencies and sequential relationships within the audio data. In conclusion, the SE-ResNet-Transformer model provides a robust framework for infant cry recognition, with promising implications for real-time infant monitoring systems. Its classification

accuracy is 93%, outperforming traditional CNET, which can help caregivers better understand and respond to infants' needs.

Table 2.1 presents a comparative analysis of several recent studies focused on infant cry recognition using deep learning techniques. Each study employs different preprocessing methods, algorithms, and network architectures to address various objectives as classification of cry types and diagnosis of health conditions. For instance, some study implemented MFCC feature extraction combined with MLP and CNN models, resulting in improved classification accuracy. While another study leveraged CNN-RNN frameworks to enhance sequence analysis and achieved high accuracy in distinguishing cry types. And while several studies incorporated advanced preprocessing techniques like signal de noising, noise reduction, and feature engineering, combined with algorithms as LSTM, GAN, and other DNN variants, to either improve recognition efficiency or enable health diagnostics. Overall, the table highlights the evolving focus from basic cry classification to more complex diagnostic and pattern recognition tasks, with recent trends emphasizing hybrid architectures and refined preprocessing for improved performance.

TABLE 2.3.1: COMPARISONS OF PREVIOUS STUDIES

Paper Title	Focus	Preprocessing	Algorithms	Date	Results
Infant Cry Classification by MFCC Feature Extraction with	Classification using MFCC features	MFCC extraction	MLP, CNN	2023	Improved accuracy using MFCC and CNN

MLP and CNN Structures					
Infant Cry Classification Using CNN-RNN	Classification using CNN-RNN	CNN for feature extraction, RNN for sequence analysis	CNN, RNN	2020	average classification accuracy up to 94.97%
Deep Learning Systems for Automatic Diagnosis of Infant Cry Signals	Automatic diagnosis of health conditions	Signal de noising, feature extraction	CNN, LSTM, GAN	2022	Accurate diagnosis of health conditions from cries
Deep Learning for Infant Cry Recognition	Recognition of cry patterns	Feature extraction, noise reduction	CNN, LSTM	2022	Effective pattern recognition in cry signals

Infant Cry Classification by Using Different Deep Neural Network Models and Hand-Crafted Features	Classification using various DNN models	Feature engineering	DNN variants	2023	Comparative study of DNN models and hand-crafted features
Baby Cry Recognition Using Deep Neural Networks	Recognition of cry types	Feature extraction, noise filtering	CNN, RNN, LSTM	2018	Efficient recognition of cry types
Review of Infant Cry Analysis and Classification	Review and synthesis of methodologies		Various methodologies	2021	Summary of methodologies and future directions
A Comparative Study of Machine Learning Methods for Baby Cry Detection Using MFCC Features"	Traditional ML (SVM, k-NN)	MFCC extraction	SVM, k-NN	2024	High accuracy with MFCC + SVM

"Hybrid CNN-RNN for Cry Classification"	Hybrid Systems (CNN-RNN-LSTM)	Feature engineering	CNN-RNN	2024	
"SE-ResNet-Transformer for Cry Recognition"	Transformers	Transformers	SE-esNet, Transformer	2024	93% accuracy

2.4. Knowledge gaps in previously worked methods

By reviewing the previous literature on infant cry classification, several knowledge gaps were identified that emerged from the effectiveness of these models in practical applications. Most notable among these gaps is that most studies relied on traditional models such as SVM and k-NN, which demonstrated limited ability to handle the extreme complexity of infant cry acoustic signals, particularly in extracting the deep and fine-grained patterns needed to distinguish between different types of cries. These studies also lacked integration of temporal and spatial analysis of the signals, as they did not use deep learning models such as CNN and LSTM, which have greater capabilities for processing audio data in both spectral and temporal forms. Furthermore, most research was limited to limited or non-diverse data, affecting the generalizability of the results across different environments and contexts.

2.5. Our contribution

While previous research has explored the use of "deep learning" models and MFCC features to classify infant crying, this research makes several new contributions that distinguish it from previous work. First, it incorporates hybrid architectural optimization by integrating random Fourier transform (RFF) features with dense layers, greatly improving the stability and classification accuracy of deep learning models. Second, in contrast to most studies which have employed single-source datasets, this study shows the model's effectiveness in two different datasets: a global dataset and a locally collected from pediatric hospitals setting, which guarantees generalizability across environmental settings. Moreover, we provide an extensive comparison among four of the most prominent CNN architectures (VGG16, DenseNet, ResNet, and GoogleNet) for this task under identical pre-processing settings, revealing more significantly while comparing the relative advantages of those CNNs for classifying cry. Combined, these factors contribute to the utility and trustworthiness of AI-driven infant care applications.

2.6. Summary

The data utilized in this study for the classification of infant cries were sourced from both global datasets and locally collected data from healthcare centers. The global datasets comprise a diverse range of cry recordings, capturing variability across different populations, environments, and conditions under which the cries were recorded. These recordings have been meticulously labeled by experts in infant cry analysis to categorize distinct cry types, such as hunger, pain, discomfort, fatigue, and others.

To further enhance the robustness of the dataset, additional recordings were incorporated to reflect cry pattern variations influenced by regional and cultural differences. This enrichment ensures that the model is exposed to a broad spectrum of acoustic features, making it more adaptable to real-world diversity.

All data collection efforts adhered strictly to ethical standards, including obtaining informed consent and ensuring the anonymity of all subjects involved. The resulting dataset serves as a valuable resource for training and evaluating deep learning models aimed at accurately classifying infant cries based on key acoustic characteristics such as pitch, duration, intensity, and Frequency spectrum.

Ultimately, this research seeks to leverage these datasets to advance the development of AI systems that support caregivers in better interpreting and responding to the needs of infants, thereby contributing to improved infant care and well-being.

Chapter 3

Methodology

3.1. Introduction

The methodology employed in infant cry classification plays a crucial role in ensuring the reliability and validity of cry analysis. This section outlines the comprehensive approach taken, including data acquisition, preprocessing, annotation, and analysis, alongside the integration of machine learning and deep learning models for classification. Key methodological stages involve selecting appropriate datasets, applying preprocessing techniques such as feature extraction and normalization, and leveraging expert annotations from specialists in infant behavior. Additionally, advanced AI techniques are utilized to enhance classification accuracy. This methodological framework is designed to establish a systematic and effective approach to infant cry categorization, facilitating tiMely and appropriate care while enabling early identification of potential health complications.

3.2. Proposed Classification Model

The developed model for infant cry classification uses the state-of-the-art machine learning and deep learning methodologies for categorizing the cry of newborn baby into various classes

based on the acoustic signals. The model was developed based on dataset collected from both global and local resources, with preprocessing steps including noise filtering.

In the next phase, the audio signal was transformed into MFCC (Mel-Frequency Cepstral Coefficients) to facilitate feature extraction and enable classification. We build a model of deep learning to classify MFCC images in an efficient way, in terms of accuracy and computation.

The first step of the model is to accept input images of shape (224, 224, 3) and resize them to (64, 64, 1) for speed of training. Normalization is applied to standardize pixel values then two convolutional layers (64 and 128) with ReLU activation, to extract relevant features by using filters of increasing size. Max-pooling layer reduces spatial dimension of feature maps while dropping out layer mitigates overfitting by randomly disabling neuron.

The architecture transitions through a flatten layer, preparing the data for fully connected layers, which are further enhanced using random Fourier features to approximate non-linear mappings. Compiled with the Adam optimizer, the model uses categorical cross-entropy for multi-class classification. Evaluation is based on accuracy and loss, demonstrating its capacity to efficiently capture and process complex image data. Figure 3.2.1 illustrates the detailed architecture of the proposed approach.

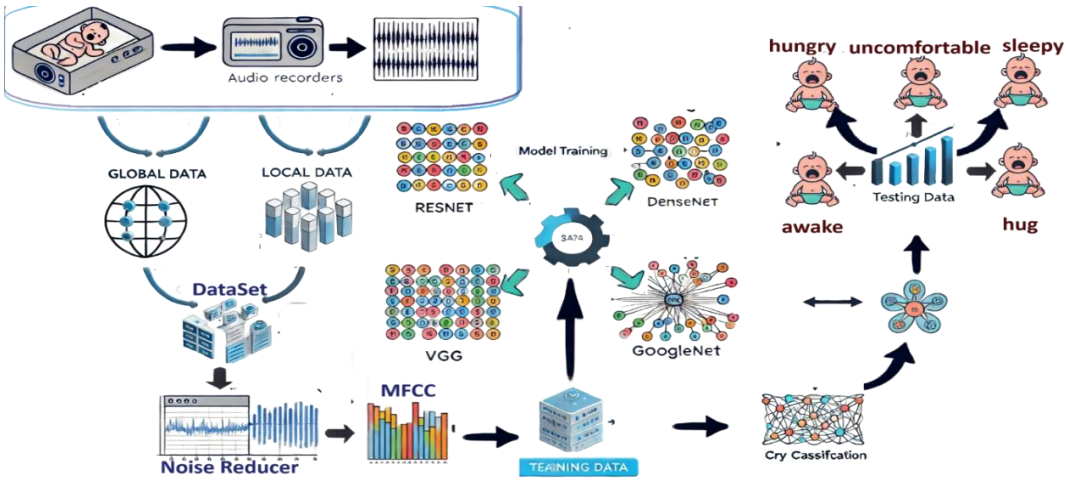


Figure 3.2.1: The proposed model.

In more details, the architecture of the model accepts images of dimension (224, 224, 3), representing a pixel resolution of 224x224 and three-color channels (RGB). What follows is a resizing layer that compresses their dimensions to (64, 64, 1), which increases speed during the training process by simplifying computational complexity. Next a normalization layer normalizes pixel values, which supports stabilization and speeding up of training.

To generate spectrograms, the model utilizes a function that takes an audio file path as input and outputs a spectrogram image stored in the designated directory. Using the Librosa package, the function loads the audio, computes the Mel spectrogram, and converts the power spectrogram to a decibel scale for improved visualization.

Using MFCC, the create_pngs_from_wavs function automatically creates PNG images from multiple audio files in a specified folder. The program converts all .wav audio files in the input folder to spectrogram images in the output folder. This function is internally called the "generate spectrogram" function to generate all the spectrogram images. The advanced models of Convolutional Neural Networks (CNN), naMely VGG16, DenseNet, ResNet, and GoogleNet ,

were used to evaluate and compare the performance of each model individually. Each model was trained on the same dataset and processed in the same way to ensure consistency in testing conditions. The performance of each model was then measured using specific evaluation metrics such as accuracy and loss.

The advanced models of Convolutional Neural Networks (CNN), naMely VGG16, DenseNet, ResNet, and GoogleNet , were used to evaluate and compare the performance of each model individually. Each model was trained on the same dataset and processed in the same way to ensure consistency in testing conditions. The performance of each model was then measured using specific evaluation metrics such as accuracy and loss.

The objective of this approach is to find out which of these models works better for the classification task, and each model uses a different architectural design that enables it to have unique processing of data. To determine the comparative complexity of each model, its sensitivity to training data and generality to new test data, we have compared these four models in a detailed and accurate way.

3.3. Dataset Description and Selection

This work used two datasets, one global and the other local. The global dataset was obtained from Kaggle and developed from audio files stored on GitHub. We used this dataset to analyze different infant crying patterns and train models to classify infant crying. The dataset used, named donateacry_courpus [54], contains 457 audio files, divided into five categories as follows:

Table 3.3.1: dataset description in a clear tabular format

Infant cry Type	pain	Burping	discomfort	Hunger	Sleepiness
No. audio files	16	8	27	382	24

These audio recordings serve as the primary data for training and testing machine learning and deep learning models.

We also used a local dataset recorded from daycare centers. It contains 75 audio files that were classified by experts in the field of neonatology and preterm care. These files were divided into five categories to match the global dataset on which the models were trained, with each category containing 15 audio files. In the classification process, the process relies heavily on the features extracted from these recordings. Several forms of preprocessing are applied to ensure the audio data is suitable for analysis and model training. These include noise removal to remove background interference, amplitude normalization to normalize sound levels, and feature extraction to identify the most prominent acoustic features.

The size, diversity, and quality of the dataset are critical factors in determining the overall effectiveness of classification models. A large and varied dataset enhances the model's ability to generalize across different types of infant cries, significantly improving performance in real-world applications.

Equally important is the data quality—ensuring that the dataset is free from distortion and irrelevant noise. Poor-quality data can hinder the model's ability to learn underlying patterns, potentially reducing classification accuracy.

Therefore, in this proposed system, the dataset must be carefully curated and preprocessed to maintain integrity, maximize learning potential, and produce reliable and robust classification outcomes.

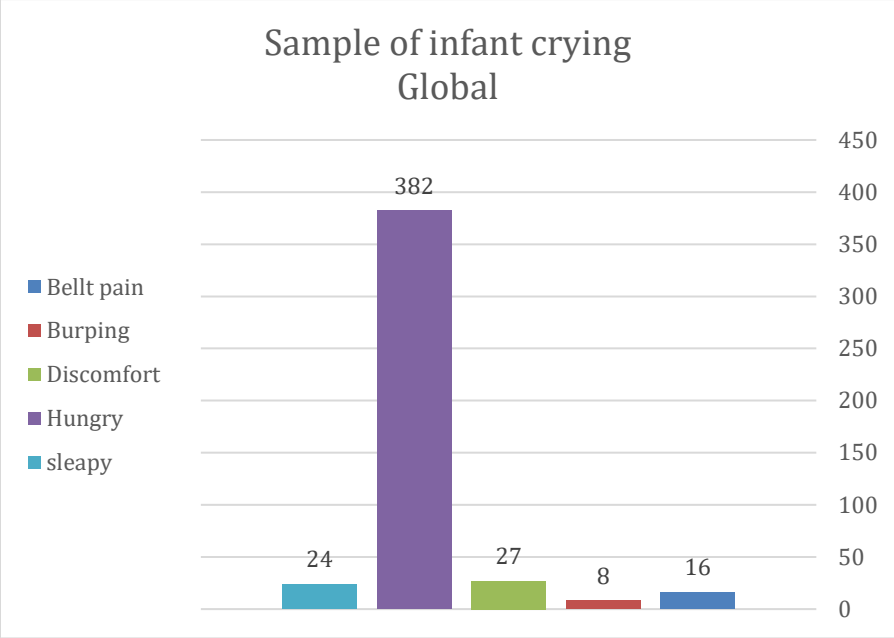


Figure 3.3.1: Sample of infant crying Global

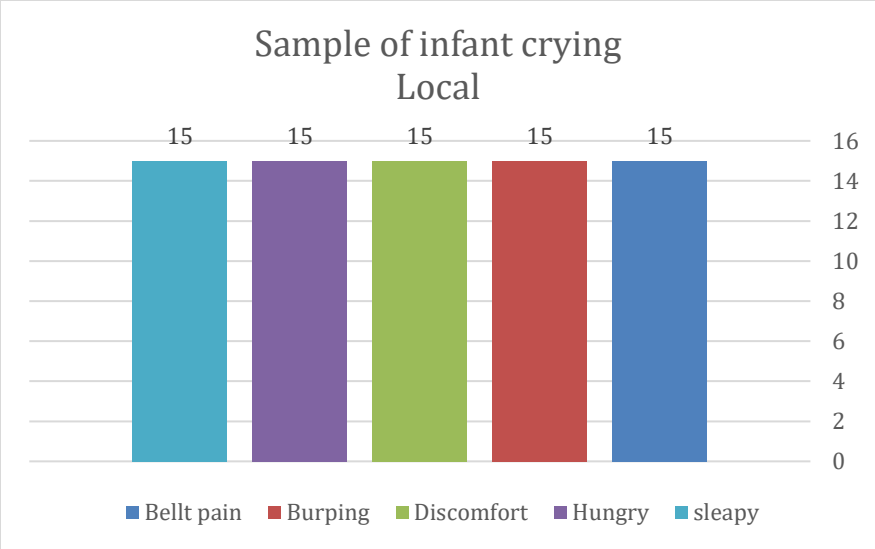


Figure 3.3.2: Sample of infant crying Local

3.4. Data Preprocessing

A set of preprocessing techniques was applied to ensure the quality of the audio data used in infant cry classification. We first cleaned the data by removing corrupted or incomplete audio files. We then used specialized audio editing tools, such as Catchpoint and CapCut, to remove background noise and enhance the audio signal. We then segmented the audio signal into short segments by analyzing time intervals in the audio waveform to identify the boundaries of individual crying episodes, without relying on advanced algorithms specifically identified by name. Techniques based on analyzing changes in audio signal intensity (energy-based segmentation) were used to facilitate the extraction of precise audio features. This was followed by an annotation process by neonatology experts to classify each segment according to the type of cry according to the five categories mentioned above. This goal was to create good, candidate data

for use in supervised learning models. We relied on the MFCC technique to extract the Frequency characteristics of the sound through a series of steps that will be detailed later. This process resulted in a two-dimensional matrix representing the values of Frequency and temporal changes. Then it was converted to spectral images using the Librosa library. Then the pixel values of these images were normalized and their dimensions changed to (64*64*1) to be used as inputs for deep learning algorithms CNN and deal with them. Here are the details:

3.4.1. Cleaning

The following are the steps that are crucial to carry out in order to prepare the dataset for infant cry classification in order to make it fit for analysis and training of a model. In the first stage, all the files that have some issues like incomplete or corrupted ones and can hinder the quality of the data are first eliminated. This step assists in data cleaning where some recordings may have been recorded wrongly or there may be some technical problems that may make the recording useless during collection or storage.

Second, methods of noise suppression are used in order to eliminate unwanted background noise and other sounds that might hide the acoustic characteristics of the infant's cries. This involves applying some filters using programs naMely Catchpoint and CapCut which helps in the removal of noise by only allowing the cry signals to pass through thus improving the cleanness of the dataset.

Furthermore, data cleaning may include the procedures of making sure that all cry recordings are in the same format, or sampled at the same rate, for instance. Standardizing

guarantees having the most efficient means of analyzing the data, and reducing the possibilities of other variations occurring because of differences in recorder settings.

However, cleaning the dataset is a prerequisite to infant cry classification process in particular, which ensures that the data is clean and it does not contain any missing values and incorrect values, which assist to train the machine learning models on infant cry data to accurately classify infant cry with the help of the acoustic characteristics.

3.4.2. Segmentation

In the framework of infant cry classification segmentation is a strategy which splits the cry recordings into a series of smaller chunks of data where each chunk is associated with a single cry event or instance. This is very important in order to analyze and annotate the cry patterns in a more effective manner.

From the above, it can be seen that cry recordings can be divided into segments which will assist us to differentiate between the different cry instances based on the gaps in the acoustic waveform. This segmentation helps in identifying and categorizing each cry event so that it can be useful in classifying a cry as hunger, pain, or discomfort.

Also, segmented data permits us to determine the acoustic features of each segment of cry in detail. Factors including pitch, amplitude, and spectral power are needed in order to design a machine-learning model that is capable of identifying different types of cries in relation to the sounds they make. Hence, these features must be divided into various cry instances; this makes it easier for the model to identify and differentiate the various types of cries.

So, segmentation is useful for the sorting and preparation of the material to be used by us as it allows them to have labeled cry segments that can be used to train the classification models. In this manner, the models can learn and generalize from the segmented data in the most efficient manner and later on assist the caregivers in comprehending the cry of the baby.

3.4.3. Data annotation:

The process of data annotation for infant cry classification involves the act of assigning labels on the cry recordings depending on the type of cry that is being made for instance hunger, pain, discomfort, or fatigue. This annotation is important in supervised learning where machine learning trains to map extracted features of cry signals into the respective categories of cry signals.

Infant cry tags were performed by two experts in infant behavior or healthcare providers with an understanding of crying patterns: one from the Neonatal Care Unit at Al-Isra Specialized Hospital and the other a pediatrician, neonatologist, and neonatal nurse practitioner working in government healthcare centers. They listened to the audio recordings and classified them based on their knowledge and experience. Each audio recording was independently reviewed by two experts, and cry types (e.g., hunger, discomfort, pain, burping, and sleepiness) were classified based on acoustic characteristics and context. To ensure accuracy and minimize subjective bias, a standardized guideline was used to outline the criteria for classifying each cry type. In cases of disagreement between classifiers, observations of the infant's behavior and body movements were used to inform the final decision. As mentioned above, annotations help make sure that the given dataset has correct and valid labels without which models cannot be trained to effectively identify and analyze infant cries in real life.

Also, data annotation entails the issue of consistency and quality in the course of labeling. Annotators also use guidelines and criteria to label the data thus reducing the chances of making mistakes and increasing the reliability of the annotated data. Other quality control measures like peer review and checking at certain intervals may also be put in place to check on the annotations and correct any anomaly or inaccuracy to improve the quality of the dataset and the classification models that are developed from it.

3.5. Model Development and Training

3.5.1. Feature Extraction Overview

Feature extraction is crucial to a wide range of speech and audio processing applications including speech recognition, speaker identification, and music organization. The main goal of this process is to convert raw audio data into features that can be efficiently used to train ML models and guarantee consistent and reliable results.

In this sense, "Mel-Frequency Cepstral Coefficients (MFCCs) represent one of the most common feature extraction techniques. MFCCs are specially known for successfully capturing the spectral characteristics of an audio signal and shaping a compact spectral signature that models human cochlear perception. The main advantage of MFCCs is the extraction of the most important characteristics, which helps to perform speech recognition and classification.

Although other feature descriptor such as GFCCs and LFCCs exist, MFCCs are preferred by many studies and applications as they are consistent with human auditory system, and are more

applicable to machine learning. In reality, many research works have proved the capability of MFCCs to effectively and expediently extract significant features from audio signals, and thus certify the extensive application in practice.

In our research, we are concentrated on the value of MFCCs attributes, due to their wide acceptability and application in the field of Machine Learning rather than using them for direct sound transformation. To accomplish this, a number of preprocessing steps are performed on the raw audio data, such as pre-emphasis, framing, windowing, Fast Fourier Transform (FFT), Mel filtering, axis scaling, and Discrete Cosine Transform (DCT). These operations are essential for emphasizing the most relevant information in the audio signal and in obtaining adapted features to be used as input of machine learning models.

The feature images ultimately represent the characteristics of the audio signal. These images that can be extracted from the audio data, can then be processed by machine learning algorithms and CNNs, in particular, to efficiently traverse and analyze the data. With the help of CNN models, then MFCC features are classified and interpreted successfully from audio signals.

In the next paragraphs, we will delve into the inner-workings of how MFCCs are extracted and discuss why this particular feature extraction was used for this research, and why it is so common in the world of machine learning applied to audio and speech analysis.

3.5.2. MFCC Technique overview

In order to extract the MFCC coefficients, we must perform a process called pre-emphasis to amplify the high-Frequency components of the signal, which are often weakened during the

recording process due to the recording environment or limitations of the recording device (microphone). This process aims to distribute the energy within the audio signal in a balanced manner, as low frequencies usually dominate the audio file. By enhancing the high-Frequency components, we improve feature extraction and ensure a more consistent and balanced spectral representation.

The energy content in most speech and audio signals is unbalanced towards lower frequencies, and typically the higher frequencies (above 4-5 kHz) are much weaker. The main reason for this discrepancy is that humans naturally will generate more energy in low frequencies and microphones or recording systems are not necessarily going to pick up the high Frequency information same as the lower Frequency information. Because these higher Frequency components (which often contain important information for telling one sound from another) become less loud and thus harder to analyze in their raw form.

The extraction of Mel-Frequency Cepstral Coefficients (MFCC) from an audio signal requires a certain amount of framing. This involves splitting the continuous audio signal into bite sized overlapping segments, called frames. In assumption, audio signal features do not change dramatically in short period of time, and the reason of framing. Many signal processing tasks make the assumption that the signal is stationary within a short interval, after which the signal is typically assumed to have taken on a different value so that each frame can be processed separately, as if it were a different signal.

Each frame's length is normally set between 20 and 40 milliseconds. This is sufficiently long to represent significant acoustic features, but sufficiently short so that the signal within that time can be assumed to be approximately stationary. A frame that was excessively long could

capture changes in the signal over time, resulting in the characteristics of being non stationary, or a frame that was too short could not capture enough information to be useful could result.

To maintain and prevent loss of important signal features, information from adjacent frames is often preserved, and frames are overlapped by certain percentage (50% to 75%)[36] or even more. The process of overlapping guarantees the temporal resolution of the signal, and also provides better transition between consecutive frames. The resulting set of frames of the audio signal is discrete, manageable chunks of the audio signal that can be analyzed independently for feature extraction. We see that the framing process is crucial to allow windowing and Fourier transformation in the following steps, since it guarantees that any portion of the signal is treated as a stable, time invariant unit.

Windowing

Windowing provides a solution to such discontinuities or abrupt changes at the beginning and end of each frame of audio signal during MFCC extraction. The audio signal is divided into small segments (frames) and each frame represents only short interval of the signal, so there are sudden transitions between adjacent frames. When further analyzing for instance with the Fourier transform, spectral leakage can introduce these transitions. In order to address this, we apply a window function to each frame before performing a Fourier Transform. A smooth, bell-shaped curve similar to this number of the Hamming window is usually employed as the most commonly acceptable window function which tapers the signal gradually towards the sides of each frame. The effect of window function is smoothing attenuation of the signal at the edges, and thus reduces the edge effects and provide more accurate Frequency domain representations.

Fast Fourier Transform (FFT)

After windowing, time domain signal is converted to the Frequency domain using Fast Fourier Transform (FFT). FFT is an efficient algorithm which computes the Discrete Fourier Transform (DFT) of a signal, this is a special case of a linear transform that decomposes a signal into constituent sinusoidal components of different frequencies. After the FFT, each frame, after windowing it, sees each frame passed through the FFT and a Frequency spectrum that shows how much energy is present there at each frequency. If our input is the FFT, then the output is the power spectrum that contains a strength from each Frequency component (represented by its amplitude). In fact, it is the power spectrum that reveals the distribution of energy over different frequencies, hence allowing us to analyze, for example, pitch, and tone and timbre as underlying characteristics of the signal.

Mel Filtering

In MFCC extraction, the next step is Mel Filtering in which the Frequency axis of the power spectrum is converted so that it is closer match to the human's perception of sound. We do have a nonlinear perception of frequencies; we are more sensitive at smaller differences of lower frequencies rather than higher frequencies. To reflect this property, the Mel scale is created for the Frequency axis to be aligned to a scale that is approximately close to human hearing. This scale compresses the higher frequencies and expands the lower frequencies. Mel filtering essentially consists of applying a set of triangular filters, in correspondence with a certain range of frequencies expressed in the Mel scale. The power spectrum is convolved with these filters to produce the Mel-

spectrogram, which emphasizes the more perceptible frequencies to humans, and de-emphasizes the less useful frequencies for human perception.

Logarithmic Scaling

The system uses an applied logarithmic Scaling after Mel filtering to simulate the loudness response of the human ear. Heck, the human auditory system perceives sound intensity in a logarithmic scale – double the sound intensity doesn't seem twice as loud, it literally seems twice as loud. In order to mimic this behavior, the Mel-spectrogram values are transformed with a logarithmic function. This transformation applies a logarithmic transformation of the spectrogram, compressing its dynamic range, and, hence, highlights the features that are important to perception of loudness, while suppressing nonessential variations. The upshot is that the obtained representation is a closer approximation to how humans perceive sound intensity.

Discrete Cosine Transform (DCT)

To reduce dimensionality of the log Mel spectrogram Discrete Cosine, transform (DCT) is applied. A log Mel spectrogram is very high dimensional, but the DCT is used to reduce the log Mel spectrogram into a set of a few Coefficients that capture most of the significant information of an audio signal. It is essential, in this process as a feature extraction step because it reduces the dimensionality of the signal while retaining its most important features. The DCT performs by taking the log Mel spectrogram and converting it into Coefficients that describe spectral feature variations across time. Usually, the first few Coefficients are the most important, evidencing the gross broad spectral shape, while the subsequent higher order Coefficients represent the finer

details. The DCT helps us make the resulting MFCCs more manageable and easier to process in machine learning tasks by its dimensionality reducing. As illustrated in the figure, we use these Coefficients as the input feature for audio classification [37].

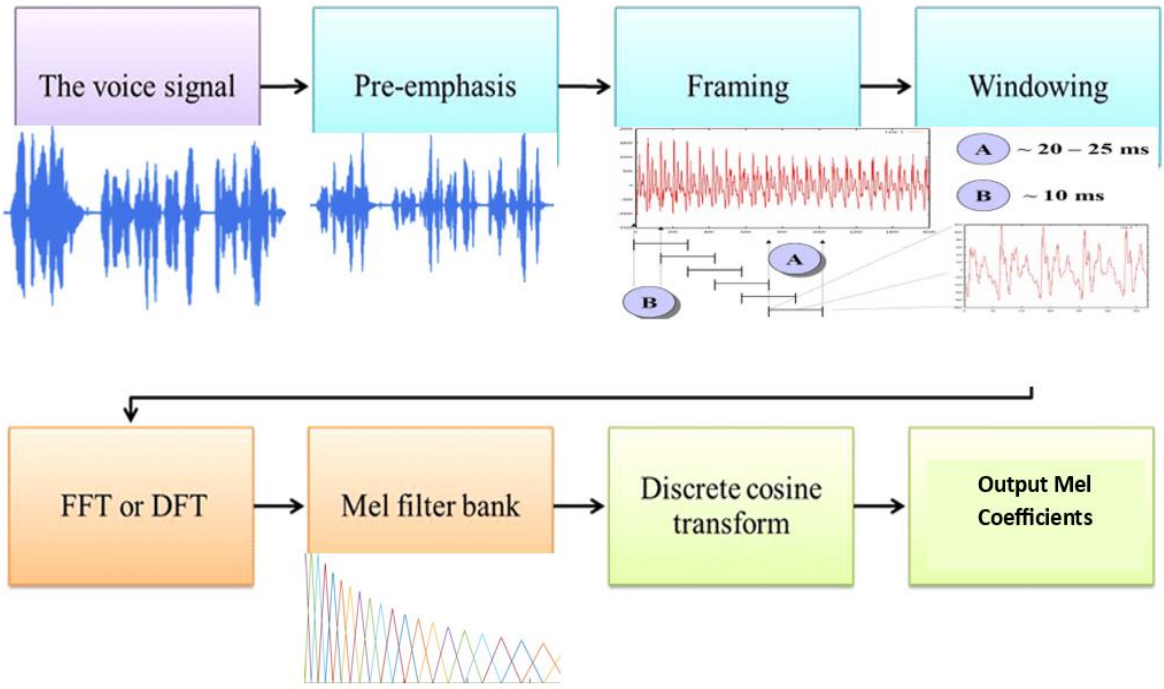


Figure 3.5.1: Steps to extract Mill Frequency Capsular (MFCC) transactions

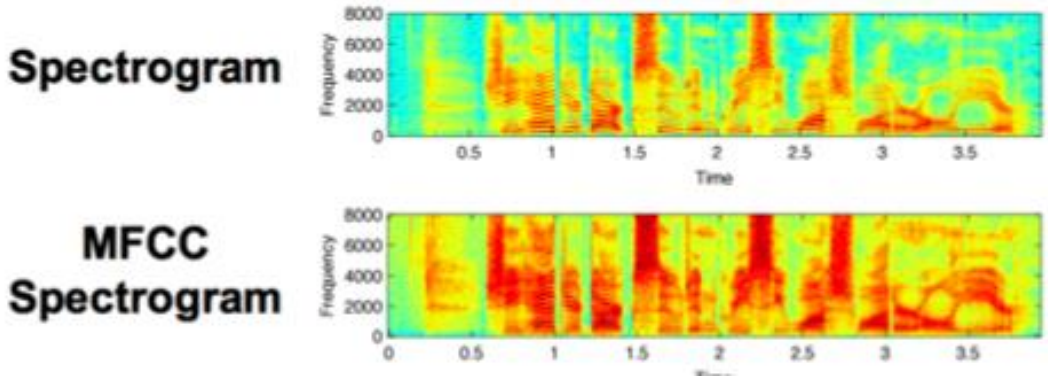


Figure 3.5.2: Mel-Frequency Cepstral Coefficients (MFCC) block diagram [38]

3.5.3. Deep learning Models:

Deep learning models have dramatically impacted the audio signal processing domain, and in particular for applications including speech recognition, sound classification and music analysis. A common way to treat audio data is to convert it from audio to visual representation (e.g., MFCC, Mel-Frequency Cepstral coefficient) that captures the spectral property of the sound. Source MFCC features are calculated from the raw audio signals and the extracted MFCC features could be considered as image data where each coefficient indicates particular Frequency components of the audio signal.

Convolutional Neural Networks (CNNs), one class of neural network that is widely used for image processing, have been successfully applied for the classification of the MFCC-based images. CNNs are natural choice for recognizing spatial hierarchies and patterns in images, and hence, are suitable for processing the MFCC images that contains highly correlated temporal and spectral patterns of sound. Using multiple convolutional layers, CNNs can identify and categorize

the pertinent features from these representations of MFCCs which contributes effectively toward classifying the audio signal.

The pipeline starts by computing MFCC features from raw audio signal and then reshaping them into image-like representations. Subsequently these MFCC images are input to deep neural network models, Convolutional Neural Networks (CNNs) in particular, for classification task. The model is exposed to various types of audio signals by classifying them from their visual representations, so that it becomes successful to differentiate each class such as speech, speaker or music genre.

Numerous state-of-the-art deep learning models have been successfully applied such as DenseNet, VGG16, ResNet, and GoogleNet . All of these models have their own characteristics which are benefiting the transformation of MFCC images:

DenseNet: DenseNet is also a kind of efficient feature reuse, which improves information flow between layers and can get a better feature extraction and classification.

VGG16: Simple and deep VGG16 architecture is well known to capture hierarchy features and known for its simplicity and depth.

ResNet: ResNet uses residual connections which mitigates the vanishing gradient problem and enables to train much deeper networks with better performance.

GoogleNet : The inception modules of GoogleNet permits multilevel feature extraction so that the model can lear from various scales of data.

With the aid of these CNN-based classifiers, made use of MFCC images, the complex audio data classification has turned into an accurate and faster process. These models have been

proved to be very effective in many audio classification tasks such as speech recognition, speaker identification and infants crying.

VGG 16

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is one of the prominent computer vision competitions that has become an annual event: teams compete in two key tasks. The first task is object localization where each image is given 200 predefined categories and object(s) present in the image are to be detected. The second task is image classification wherein each image has a label in one of 1,000 possible categories. For large scale image recognition, the VGG16 model was introduced by Andrew Zisserman and Karen Simonyan in 2014 of the Visual Geometry Group (VGG) Lab of the University of Oxford. This model was especially successful, finishing first or second in each of the categories at the 2014 ILSVRC competition. VGG16 is evaluated on the ImageNet dataset containing more than 14 million images split over 1,000 classes and reports a top-5 test accuracy of 92.7% [39].

Input images are of size 224x224 pixels and since they are represented using three RGB color channels, VGG16 takes them as input. Input to model is an image tensor of size (224, 224, 3) and output is a vector of 1,000 values (i.e., probability that image is one of the 1,000 categories). For example, let there be a model which predicts probabilities of various classes, where the probabilities predicted by SoftMax are normalized (or add up) to one. It then calculates which of the five classes has the highest probability, and then compares that to the ground truth labels, and measures the error. In order to minimize classification errors and as a result improve the accuracy

of prediction of the correct class, I define an error function, which measures the difference between the predicted and the actual classes in this problem.

VGG16 starts with a 224x224 pixels input image and 3 color channels. The first two layers involve the application of convolutional filters of size 3x3, depth 64, and a max-pooling layer which down samples the image. We repeat this process with convolutional layers of increasing depths–128, 256, 512 filters followed by max-pooling layers which progressively decrease the image size. Unlike AlexNet and ZF-Net, all the convolutional filters in VGG16 are 3x3 in size and are uniform. After each convolutional layer, we add padding so as to keep the spatial dimensions of our image.

The image is converted to a feature map of the shape (7, 7, 512) and then flatten to a feature vector of (1, 25088) as the image traverses the network. It processes the feature vector by means of three fully connected layers that lead to a layer spitting out 1,000 channels, each of which corresponds to one of the 1,000 classes of the $1{,}000^2$ classes of a larger ImageNet dataset. In the last layer, the SoftMax function classifies the input image from one of the specified classes and, in the hidden layers, the Rectified Linear Unit (ReLU) serves as the activation function for computational efficiency [40].

The following shows different configurations for VGG models (VGG16C, VGG16D) in the table below. The difference between these models is that in some layers, (3x3) convolutional filters instead of (1x1) filters are applied. The VGG16 counts about 138 million parameters, which make training computationally challenging regarding training duration and memory consumption. On a GPU such as the Nvidia Titan, the original VGG16 model took 2-3 weeks to train, and the large volume (528 MB) made storage and RAM demands more burdensome.

By substituting the output class scores for bounding box coordinates, the performance of classification with VGG16 was improved for object localization tasks. The model can easily locate the position of an object in an image using a 4-dimensional vector representing the center point, height, and width of the bounding box. As can be observed in the figure 3.5.

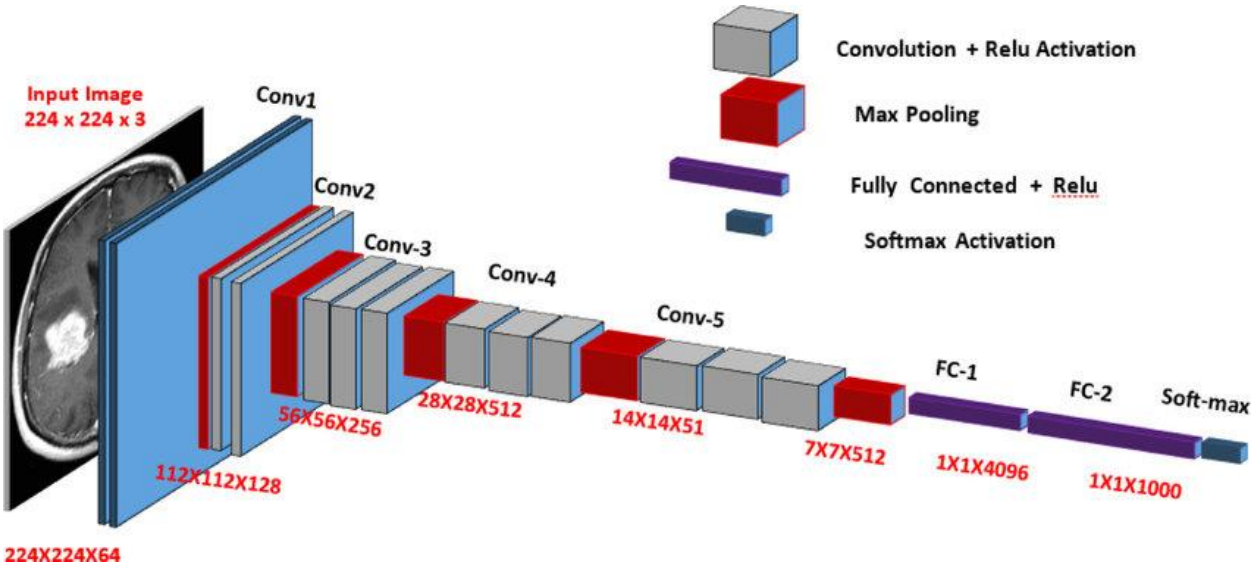


Figure 3.5.3: VGG-16 CNN model architecture layer wise [47].

The 2014 ILSVRC competed was found the strong performance of VGG16, with localization error rate 25% but VGG16 also has great limitations. It might have problems galore, like vanishing gradients, slow training speed, large models, and a whopping 138 million parameters. To address these limitations, newer models, like ResNet, were employed using skip connections that nudge over gradient issues and boost the power of deep networks. The ResNet architecture is an innovative innovation over VGG16 framework where it outperforms and reduces computational problems during large scale image recognition problem [41].

DenseNets

Densely Connected Convolutional Networks or DenseNets are a revolutionary change in deep learning as they both increase the depth and performance of convolutional networks. VGG has 19 layers, over the years CNN architectures evolved to ResNets with 100s of layers, even 1000s. But greater number of layers come at the price, as now information and gradients must travel over longer paths where the input and output layers. This can cause this extended path to vanish or dissipate before important information reaches its destination, decreasing the model's effectiveness. This is addressed in DenseNets which simply consider the connection structure and some approaches from network like Highway Networks, Residual Networks and Fractal Networks.

DenseNets are also highly parameter efficient. In contrast to conventional CNNs, DenseNets circumvent learning redundant feature maps, resulting in a more compact architecture with less parameters. Furthermore, if ResNets are researched, a lot of layers are actually useless in achieving the desired performance, which can be skipped; whereas, DenseNets use a specific concatenation mechanism. This assures that we are not reduplicating the output feature maps of one layer with the incoming feature maps of another layer, and at the same time makes the learning of weights more efficient because the information with related features gets combined together. As a result, DenseNets enable direct access to their gradients from the loss function in addition to the original input and facilitate better information flow during training allowing for deeper networks [31].

In traditional Feed forward neural networks, we have layer by layer connection where each Layer does some mapping like convolution, pooling, batch Norm, and activation functions and so on. Skip connection in ResNets aims to solve vanishing gradient problem, DenseNets further think

about concatenation instead of addition to combine feature maps. This type of structure enables DenseNets to keep consistent feature map sizes within each dense block while having changeable number of filters. However [Wang et al 2018], that DenseNets handle transitions between layers with different feature map sizes via 'transition layers' comprised of batch normalization, 1x1 convolutions, and 2x2 pooling.

They refer to the growth rate of these DenseNets with the help of the hyperparameter *k* meaning how many feature maps at each layer we generate. This is the growth rate at which knowledge in the network is absorbed as each layer has access to all of the feature maps from all preceding layers. The collective knowledge of the network is proportional to the growth rate, and therefore can be seen to be determining the overall performance of the model.

DenseNet-B is a variant where a 1x1 convolution is applied before applying the 3x3 convolution, to reduce computational efficiency by reducing the size of feature maps. DenseNet-BC takes this even further by imposing a compression factor (theta) on its growth which decreases the amount of output feature maps at each layer. This is measured by a compression factor between 0 and 1 which provides more efficient resource use without sacrificing performance. DenseNets behave just like their original form when theta is 1, but lower values of theta cause smaller feature map sizes and better efficiency [42].

Datasets like ImageNet which are high input volume are particularly suited to DenseNets. With CIFAR-10 or SVHN as small datasets the architecture can be simplified still with high accuracy. DenseNets comprise dense blocks, stacked of a number of dense layers. These networks use a set of bypass connections to add feature maps to the existing volumes, conserving and reusing

information throughout the network. The architectural design of DenseNet-121, in the simplest variant of the DenseNet family, is quite clear.

The figure 3.6 reveals that in DenseNets dense layers add new feature maps every step, effectively multiplying the size of information to add. For illustration, in the first dense block, the network can expand the number of feature maps from 64 to 256 by multiple layers of connectivity, each adding 32 feature maps. 1x1 convolutions with 2x2 pooling, termed transition blocks, serve to reduce the spatial dimensionality of feature maps by a factor of 2, in order to maintain computational efficiency of the model. However, despite this high degree of connectivity, DenseNets achieve controllability in the flow of information by virtue of the balance between dense connections and transition blocks.

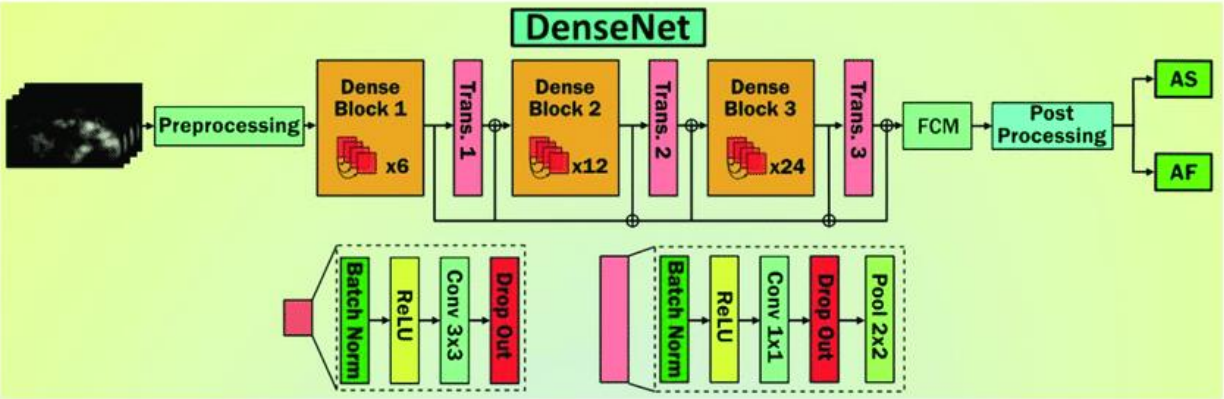


Figure 3.5.4: DenseNet architecture with three dense blocks and three transition blocks [43].

DenseNets provides an efficient and connected architecture minimizing several of the problems that plague deep networks like vanishing gradients and redundant feature maps. DenseNets draws on dense blocks, transition layers, and growth rates to increase the depth and

performance of convolutional networks. The variants DenseNetB and DenseNetBC provide incorporated adaptations that make the model much more adaptable to any given datasets and computational need. DenseNets thus provide a strong and scalable solution to tackle complex image classification problems with the power of deep learning without its drawbacks [41].

ResNet

ResNet (Residual Network) is a deep learning model for image processing introduced to solve complicated computer vision tasks. Uniquely, it depends on its architecture on a Convolutional Neural Network (CNN), but unlike CNN, this network leverages hundreds or thousands of convolutional layers. However, traditional CNN designs have been limited in performance by the capacitation problem; as more layers have been added to the model, the accuracy has been stalling. One of the issues was that it became clear when loading additional layers that the vanishing gradient problem would negatively impact the network performance. Backpropagation is a gradient descent-based technique to train neural networks by iteratively adjusting weights in order to minimize the loss function. However, gradients will suffer diminishing through repetitive multiplications and can be dropped from existence if layers are too rich. Since more layers are added, a performance saturation point, or even degradation, of the network results.

The authors of ResNet propose solving this problem of vanishing gradients using so called 'skip connections,' which is a novel idea altogether. Also, these skip connections enable the network to skip the layers and use the activation of the previous layers. ResNet initially stacks

identity mappings, convolutional layers intended to have no effect, and winds up compressing the network into fewer layers, which speeds up the initial training process. When retraining, residual parts of the network increase their size and that means being able to explore a broader range of features that compose the input images.

For example, ResNet just skips two or three layers at a time and puts non-linearity and batch normalization between. This is extended by more advanced architecture, such as Highway Networks that learn skip weights that determine how many layers to skip. Residual blocks are the key building blocks of ResNet that is useful for the network. Residual blocks follow a similar structure to the VGG16 architecture with convolutional layers interspersed with batch normalization and nonlinear activation functions generally ReLu. Early networks, such as VGG, only had 19 layers, but later research showed that adding more layers really could supercharge CNNs.

The figure at the bottom right shows an equation of $\text{output} = F(x) + x$, which is a residual block in ResNet as $F(x)$ means series of convolutional operations without any information about input x. Backpropagation for CNNs using ResNet smooths out the gradient flow, making it easy for ResNet to go anywhere from 50, 100 or even 150 layers. The integration of input from previous layers is now prevalent in several neural network architectures, including UNet and Recurrent Neural Networks (RNN); skip connections do not impose any additional computational burden [44].

From fundamental ResNet architecture, variants of this have come to be developed. ResNeXt is one such variant that allows for flexible splitting, transformation and merging of feature maps. Instead of using a single processing path, ResNeXt redirects multiple paths for

processing, thereby facilitating a variety of transformations including 1x1, 3x3, or 5x5 convolutions, or pooling. Unlike the Inception Model that uses a similar approach, it merges and adds these transformations, allowing it to deal with any tasks more efficiently. Cardinality, i.e., the number of independent paths, is introduced in the model. ResNeXt increases capacity and, similar to Inception, we benefit from increased cardinality at little cost in terms of parameter count or computation.

The other important variant is Wide ResNet introduced to overcome limitation of deep networks. Wide ResNet, introduced in 2016, permits users to transform the width or depth of the network without affecting performance. Increasing the width of the network has been found to provide better performance gains than simply adding more depth, which usually results in vanishing gradients or simply degradation in performance. Residual blocks on Wide ResNet are similar to those on original ResNet, but it allows wider network configurations, so it can train and apply to various datasets easily. There are five groups of layers with convolutional layers and residual connections. The final convolutional group is proceeded by an average pooling layer, where a classification layer is attached. As can be observed in the figure 3.7, a structure enables us to adjust network depth and width freely with corresponding intact high accuracy.

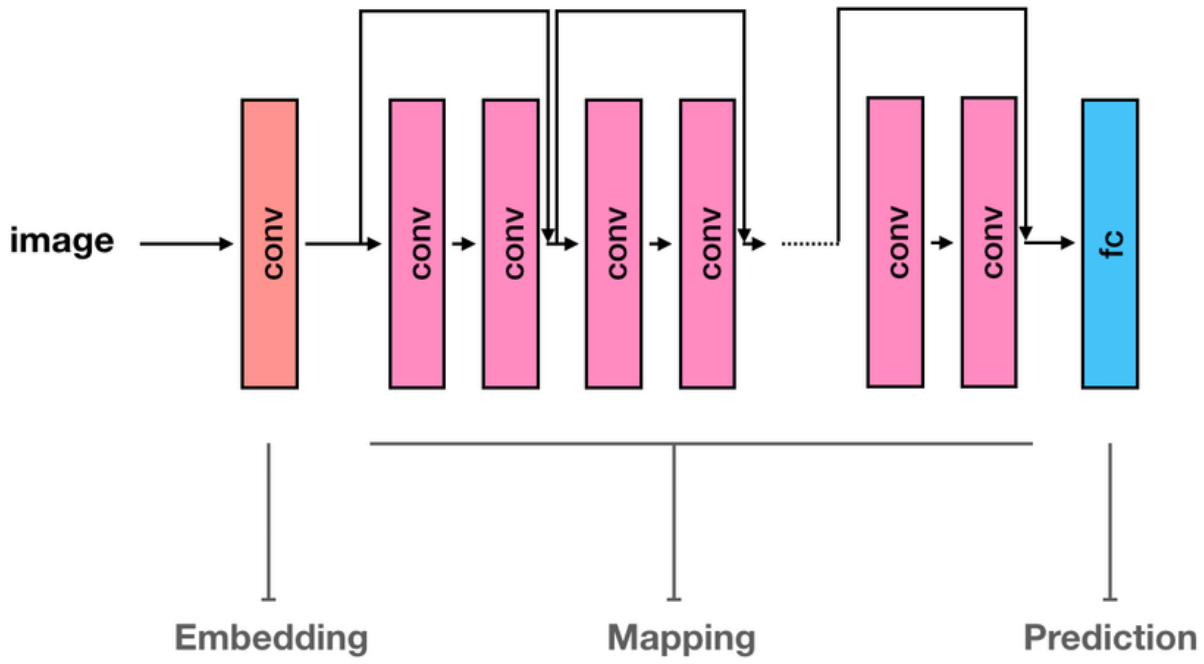


Figure 3.5.5: The ResNet architecture of a deep CNN with residual learning [45]

In IT, residual blocks of Wide ResNet are built out of 3x3 convolutions, but one can connect them with 1x1 convolutions for flexibility. The problem is how many convolutions to put per block and this is a crucial metric which governs the depth of the network and the overall performance. To gauge the network's computational drift and accuracy, it's important to frequently check both the depth and width of each residual block. Furthermore, after all convolution dropout layers are usually included to avoid getting the network to over fit and to promote generalization [44].

In short, ResNet and its variants to ResNeXt and Wide ResNet form a major leap forward in deep learning for image processing. ResNet was successful in solving the vanishing gradient problem through introducing skip connections which allow the training of very deep networks. This foundation is then extended by variants such as ResNeXt and Wide ResNet with a further

tweak from architectural perspectives, like cardinality and network width, for achieving flexibility and performance. Yet, these networks are still heavily used in computer vision tasks, thanks to their versatility to strike an equilibrium between depth, efficiency, and accuracy, and their aptness for broadort application in the real world [45].

GoogleNet

In 2014, GoogleNet (Inception V1) was introduced, and so on, in the research paper 'Going Deeper with Convolutions,' GoogleNet itself is an important milestone in the neural network architecture designed for recognition of images. This architecture, developed by researchers at Google in collaboration with several universities, made first place in the ILSVRC 2014 image classification competition. Despite challenging AlexNet (the 2012 ILSVRC winner), as well as ZF-Net (the 2013 winner), and then at times producing lower error rate then VGG (the 2014 runner up), GoogleNet however showed an impressive reduction in error rates over both their predecessors. GoogleNet uses 1×1 convolutions as its key innovation in the core design, and global average pooling, which leads to its good performance [46] .

However, the architecture of GoogleNet deviates radically from previous models, including AlexNet and ZF-Net. In its design, all 1×1 convolutions and global average pooling contribute to the increase of its complexity, even with much lower computational cost. This is a special type of convolutional operation called a 1 ×1 convolution, which is a convolution operation with a filter size 1 ×1, usually used for reducing the dimensionality of the model (to refer effectively to fewer trainable parameters while deepening the model). For instance, using 5×5

convolutions with 48 filters cannot afford to perform 112.9 million operations. However, if we use a 1×1 convolution the number of operations is decreased to 5.3 million in total.

It also contains global average pooling, see figure 3.8, a global average pooling technique replacing the fully connected layers usually used at the end of CNNs. Global average pooling averages the feature map over space and thus drastically decreases the number of trainable parameters in contrast to previous models in which most parameters were located in these fully connected layers. However, the downscaled feature map leads to 0.6% top-1 accuracy gain [31].

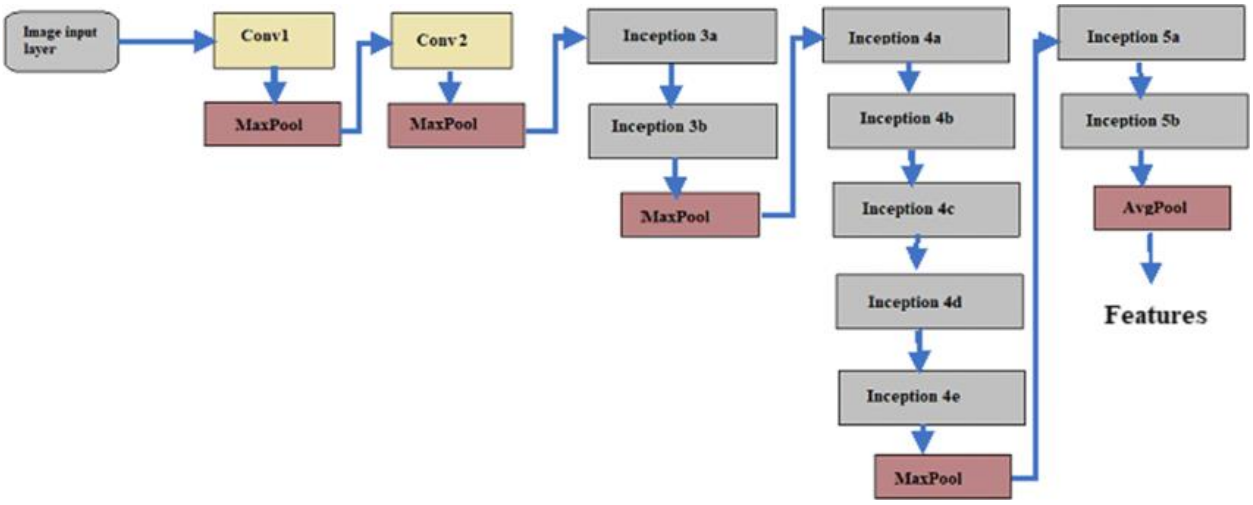


Figure 3.5.6: GoogleNet architecture for video frame feature extraction [56]

The most distinct characteristic from GoogleNet ’s architecture is Inception module, where parallel operations are carried by 1×1, 3×3, 5×5 convolutions, and 3×3 max pooling. To obtain the final result these are concatenated with the outputs of these operations. This multi scale processing enabled the model to significantly better capture objects at each scale thus improving the classification performance. In addition, GoogleNet contains auxiliary classifiers of the structure, which are utilized only in the training. The first type of these classifiers adopts a 5x5 average

pooling layer, a 1x1 convolution layer with 128 filters, two fully connected layers, and a softmax classification layer. These auxiliary classifiers provide further regularization and finally help reduce the vanishing gradient problem by contributing to overall loss function [31].

GoogleNet with a total depth of 22 layers is designed to be computationally efficient. It is built to be quickly scaled and adapted to run on devices with very low processing power. Two auxiliary classifier layers are added, connected to the outputs of Inception (4a) and Inception (4d), since the model found them helpful for training the final net. The auxiliary classifiers are characterized with a 5×5 average pooling layer with a stride of 3, a ReLU activation layer, a fully connected layer with 1024 outputs followed by a dropout regularization and a softmax classifier with 1000 classes. The ReLU activation function is used in all convolutions in the network, as it is known for introducing non-linearity to a model [31].

GoogleNet won first place in the classification and the detection tasks at the ILSVRC 2014 competition. In the classification task, it had a top 5 error rate of 6.67% which put it in the top models of its time. The collection of six GoogleNet models demonstrated its performance in object detection as well, achieving a mean average precision (mAP) of 43.9 on the ImageNet test set. GoogleNet established itself as the foundational architecture in deep learning and image classification success, and opening the door for the design of advanced neural networks in subsequent work.

3.5.4. Random Fourier Features (RFF)

Feature representation is very important in the use of machine learning — especially in applications of deep learning — because it determines the performance of the predictive models. Traditional methods typically use hand crafted features which can be expensive and might not completely encode the nuances of the data. One solution to this problem is Random Fourier Features (RFF), which allow us to transform input data into a high dimensional space within which models, such as Convolutional Neural Networks (CNNs), are able to learn more complex patterns [47].

Random Fourier Features is a means of approximating kernel functions in order to efficiently compute high dimensional feature spaces. The core idea is then based on Bochner's theorem; naMely, that any continuous kernel can be represented as an inner product in a higher dimensional space. This higher dimensional representation is defined by generating a finite number of random Fourier features of the original input data.

Mathematically, given an input data point x and a kernel function (k(x,y)), RFF approximates this kernel by mapping (x) to a random feature space defined by:

$$\emptyset(x) = \sqrt{\frac{2}{D}}\,\mathrm{Cos}(Wx + b) \qquad (1)$$

Where (W) is a randomly sampled weight matrix, (b) is a random bias term, and (D) is the number of random features. This transformation allows the model to avoid direct knowledge of complex relationships in data, while still being able to pick up on such relationships in data without the computational burden of direct kernel methods.

Random Fourier Features can help deep learning models like CNNs perform better with tasks like classification, regression, and anomaly detection on tasks involving a nonlinear data distribution.

For various reasons, Random Fourier Features (RFF) have a very important place in machine learning and deep learning. Firstly, RFF gives computationally efficient way to approximate kernel functions used, usually, in support vector machines and kernel-based methods. RFF transforms input data into a high dimensional space and is able to capture complex dependencies without suffering from the high computational cost of kernel methods.

Furthermore, use of RFF allows models including CNNs to extract the finer details of data efficiently. These high dimensional representations allow the model to better capture nonlinearities and feature interactions, resulting in more accurate and better generalization [48].

In fact, models overfit training data very easily in high dimensional spaces. This risk can be mitigated with the help of RFF, which will give the ability to expand the feature space in a controlled way, without blowing up the complexity of the model too much. So that the model is robust and well performing in new data.

Lastly, the RFF implementation is actually quite simple and can seamlessly be incorporated into pervasive machine learning pipelines. Its flexibility allows practitioners to conceptually experiment with different kernel approximations without significant modifications to the rest of their models. In summary, Random Fourier Features greatly enhance the effectiveness of machine learning algorithms at learning complex data patterns with a limited computational cost.

Random Fourier Features (RFF) for Convolutional Neural Networks (CNNs) enables the CNN model to learn complex data in tasks of high input dimensionalities. First, we transform the input data, e.g. images, using RFF, to generate a high dimensional feature representation. Before feeding the data into the CNN, this transformation happens, because many complex patterns and relationships will be missed in the original feature space and will not be captured by the model.

Once the transformation is complete, the resulting high-dimensional features are input into the CNN architecture. The CNN can then apply its convolutional layers to extract spatial hierarchies and patterns from these features, enabling it to learn both low-level and high-level representations effectively. During the training phase, the CNN is trained on the transformed features using back propagation. The model adjusts its weights based on prediction errors, refining its ability to classify or regress based on the transformed input data. Fine-tuning the model with various hyper parameters can further enhance performance [49].

After training, CNN makes predictions on new data by first applying RFF to transform the input into the high-dimensional space before passing it through the network. This allows the model to handle complex, non-linear relationships in the data, leading to improved accuracy in tasks such as classification and object detection. Finally, the model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score, and based on the results, iterative refinements can be made by adjusting parameters or experimenting with different configurations of RFF and CNN layers.

Integrating Random Fourier Features with Convolutional Neural Networks leverages the strengths of both methods, enabling the model to capture intricate patterns in the data while maintaining computational efficiency. This combination proves particularly beneficial in

applications like image recognition, audio classification, and other tasks involving complex and high-dimensional datasets.

3.6. Experimental Setup

In the experimental work phase, we started dealing with the dataset, where we made the necessary modifications to the used audio, such as noise cleaning, audio segmentation, etc., then the audio was converted via MFCC to Cepstral Coefficients and then saved as a spectral image. The image size was changed from (3.224.244) to (1.64.64) to speed up training while preserving the basic features of the image. Standard normalization techniques were applied to unify pixel values consistently, and two convolutional layers (128.64) were applied with the ReLu function activated. Then, these images were used as inputs for the CNN algorithms to be trained, which are (VGG-16, DensNet, ResNet, GoogleNet) to evaluate and compare the performance of each model separately. The same dataset was used to ensure consistency in the training and testing conditions, as the dataset was divided into two training and testing sets at a ratio of (80, 20). Labels were also encoded for the classifications of children's crying sounds (hunger Hu, pain Pa, burping Bu , sleepiness Sl, discomfort Di) , and specific performance metrics such as accuracy and loss were used to determine which of these models performed best at the classification task. We incorporated stochastic Fourier features (RFF) into the fully connected layers to approximate nonlinear maps, enhancing the model's ability to distinguish between spectral differences in infant cry types. Using RFF may allow the representation of complex patterns and relationships within the data, leading to better classification across different cry categories. This approach may improve the performance

of models in distinguishing infant needs based on crying sounds, helping caregivers respond more quickly and efficiently.

3.7. Evaluation Metrics

In machine learning, accuracy and loss refer to the same thing.

Two of the most fundamental metrics used in performing model evaluation with machine learning (especially in the context of classification tasks) are accuracy and loss.

3.7.1. Accuracy

Accuracy means the ratio of total number of correct predictions to total number of all predictions made by a model. It is a widely used metric to assess how well a classification model is performing and is calculated as follows:

A natural measure of model performance is accuracy, which is simple and easy to understand. While it is a good metric, it might not always be a very reliable one when dealing with imbalanced data. For example, in a dataset which contains 5% of the cases of one class and 95% of the other, a model can perform perfectly by predicting the majority class most of the time, gaining an accuracy score of 95% without really being good at the minority class [49].

3.7.2. Loss

A measure of how good a machine learning model is by loss which is how well its predictions approximate the actual outcomes. While training or validating, it tells you how much difference the predicted value is from the true value. In short, loss is used to figure out how poorly or how well the model is predicting the target variable [50].

Cross entropy loss, (also called log loss) is used in classification tasks. For binary classification, the cross-entropy loss is defined as:

In the case of multi class classification, the loss extends to tackle the multiple classes, but the fundamental idea is the same. In training, this loss is minimized by the model through tuning its weights via optimization methods, such as gradient descent. Better the model has the predictions closer to data, lower loss.

Accuracy tells us the percentage of correct predictions, but loss tells us how well the model is learning the patterns in the data. Although both metrics are important for evaluating a model, they tell very different things. A model can have low loss but that doesn't automatically mean high accuracy because we can predict probabilities that are very close to real labels, but make some wrong predictions. On the contrary, a model can have high accuracy together with high loss if the each of its predictions is typically right but very far off the actual probabilities (i.e., in probabilistic models).

To sum up, accuracy is a good measure for precision, especially for balanced datasets, but loss gives us a better sense of how good model is learning when performing train. So both should be considered together in deciding the overall performance of a model.

Chapter 4

Experiments and Result

4.1. Introduction

In this chapter, the practical steps used to process data and classify crying! Are outlined. After preprocessing for infant sounds, we start applying some machine learning and deep learning techniques like image processing and image and object recognition to classify our data. We used the Classification Learner tool in Jupyter Notebook (Anaconda environment), which was selected based on the nature of the problem categories to be addressed. The suitable methods for the classification task are then selected.

All experiments were carried out in Jupyter Notebook within the Anaconda environment on Apple MacBook Pro with an M2 Pro chip, 8GB of RAM, and 256GB SSD. Four classification techniques were applied: Which are DenseNet, VGG16, ResNet, and GoogleNet). The effectiveness of the proposed four algorithms is analyzed and compared with the results. Moreover, the training phase of the models was run for 5 and 20 iterations to determine their performance at different iteration cycles, expecting that, with minimal iterations, minimum error values should be achieved.

The aim of this comparison is to measure the performance of deep learning algorithms for the classification as well as prediction task in this particular experiment. The study analyzes the results and seeks to find techniques that provide the best result for this type of classification problem.

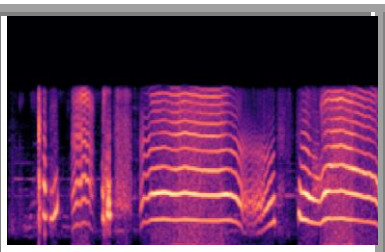
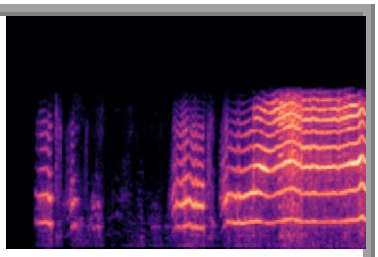
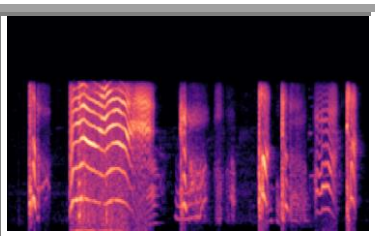
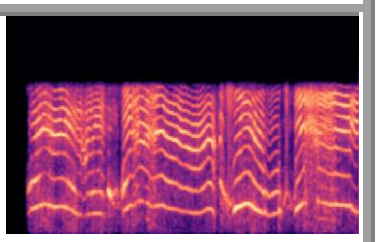
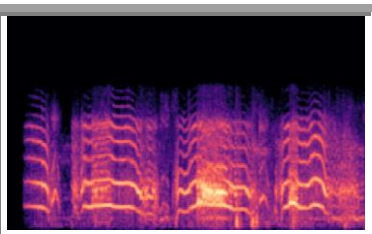
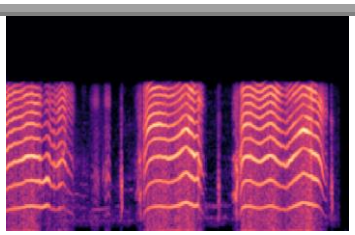
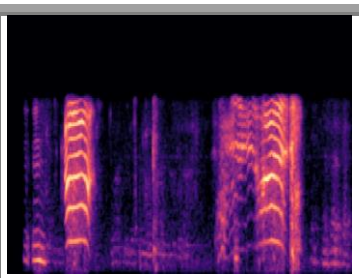
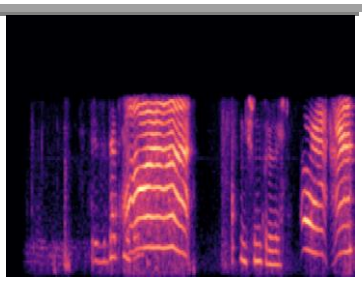
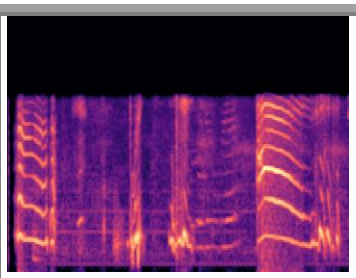
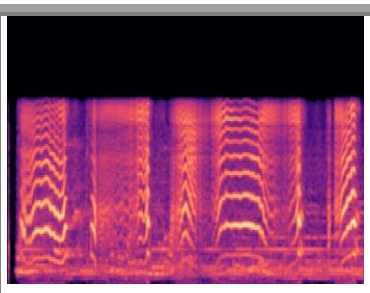
4.2. MFCC Preprocessing

First, audio data is preprocessed to obtain Mel-Frequency Cepstral Coefficients' (MFCC) from recordings of infant cries. To start, all recordings are normalized to make sure they're all the same loudness level, and a noise reduction operator is applied to reduce any background noise, Audio content noises have been effectively reduced by noise reduction software like Catchpoint and CapCut. The audio signal is then split into short segments (to be able to analyze the sound at a finer level). This is so as to avoid spectral leakage which requires windowing of each frame using a Hamming window. The edges of the signal are smoothed out.

Then a Fast Fourier Transform (FFT) is applied on each frame, converting from audio data in the time domain to the Frequency domain. Analyzing the Frequency components of the cry requires this step. The result of this leads to a series of Mel filter banks processing the result further into the Mel scale which is a scale roughly relevant to human aural perception. In this step it gives priority to lower frequencies, particularly because they are important in understanding human sounds.

The power spectrum of each Mel Frequency band is calculated, and the log of these power values is taken to compress the value range for easier data management. It ultimately applied a Discrete Cosine Transform (DCT) on the logarithmic Mel spectrum and yielded a series of spectrum coefficients, known as MFCCs, see in table 4.1.

TABLE 4.2.1 Infant cry classification based on Mfcc-extracted features

belly pain		
burping		
discomfort		
hungry		
tired		

A common library for Python which is used to extract MFCCs from audio data is `librosa` [51]. Using the librosa.feature.mfcc function easily from an audio file, and is popular in voice classification applications. Here is an example of how this can be done:

```
import librosa
import numpy as np

# Load the audio file (infant cry recording)
audio_file = 'infant_cry.wav'
signal, sample_rate = librosa.load(audio_file, sr=None)

# Extract MFCCs from the audio signal
mfccs = librosa.feature.mfcc(y=signal, sr=sample_rate, n_mfcc=13)

# The result is a matrix where each column is an MFCC feature vector
print(mfccs.shape)
```

Figure 4.2.1Using Librosa Library for Audio Signal Processing in Python

In this code snippet you can see the reading of the infant cry recording, processing with the MFCC extraction pipeline, and as an output getting the MFCC Images. Deep learning models are used with these images to classify infant cries.

4.3. Deep Learning Techniques for Global Dataset Results

In this work, we test how well deep learning algorithms can detect baby cries. This study will evaluate the classification methods using a worldwide dataset using the criteria laid out in the previous chapter as an evaluation. Scoring measures for assessment of efficacy and results of the used strategy are available. For each such experiment, the classification matrix relies on the quality

of the deep learning categories used. There were three steps to train various neural networks using the processed photos in the dataset: training roles, validation usage and testing usage.

For image analysis, four different types of CNNs, DenseNet, VGG-16, ResNet, and GoogleNet are chosen. Then the performance of the algorithms was evaluated as the number of training iterations were changed from 5 to 20 epochs. They also found that a higher number of training epochs resulted in better outcomes. This section aims at light on the real world use of these algorithms in healthcare by investigating the various results obtained using deep learning approaches for detection and classification.

4.3.1. VGG-16 Experiment Result

Given its deep structure and the known ability to extract features deep in the network, this experiment used the VGG-16 architecture to classify and analyze infant crying. Based on the figure 4.3.1, the curated photos of different infant screams made up the dataset, so the deep-learning techniques can learn patterns specific to each group. To evaluate the model, we trained and validated many different times to see what we could get in terms of the prediction results that we were indeed going to be able to predict something from the input photos. The case is shown by the findings, VGG- 16 is able to perform in the newborn field.

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 input_1 (InputLayer)         [(None, 180, 180, 3)]     0
 block1_conv1 (Conv2D)        (None, 180, 180, 64)      1792
 block1_conv2 (Conv2D)        (None, 180, 180, 64)      36928
 block1_pool (MaxPooling2D)   (None, 90, 90, 64)        0
 block2_conv1 (Conv2D)        (None, 90, 90, 128)       73856
 block2_conv2 (Conv2D)        (None, 90, 90, 128)       147584
 block2_pool (MaxPooling2D)   (None, 45, 45, 128)       0
```

```
x = GlobalAveragePooling2D()(x)
predictions = Dense(5, activation="sigmoid")(x)
model = Model(inputs=base_model.input, outputs=predictions)
```

Figure 4.3.1: Part of the VGG16 Architecture

To analyze the impact of epoch count on classification accuracy and convergence, the system was evaluated throughout many training iterations, 5 and 20 epochs, on the performance. From the figure 4.3.2 and 4.3.3 , we can see that the model was quite good as it was able to catch up with the other variations but remained consistently accurate by the 5th epoch. The results from the model showed that it had acquired certain fundamental patterns from the weeping audio data, but was not yet completely robust, so more training is needed to increase accuracy and achieve stability.

```
Epoch 5/5
12/12 [==============================] - ETA: 0s - loss: 0.2271 - accuracy: 0.8393WARNING
12/12 [==============================] - 60s 5s/step - loss: 0.2271 - accuracy: 0.8393 -
```

Figure 4.3.2: VGG16 Results after 5 Epochs of Training

```
Epoch 20/20
12/12 [==============================] - ETA: 0s - loss: 0.2210 - accuracy: 0.8393WARNI
12/12 [==============================] - 66s 5s/step - loss: 0.2210 - accuracy: 0.8393
```

Figure 4.3.3: VGG16 Results after 20 Epochs of Training

Experimental results revealed that using 20 or 5 epochs, the accuracy of the VGG16 model was 0.83 for both tests. The fact that this outcome was because the model could not increase its accuracy by increasing the number of iterations and this highlights the necessity to examine alternative models that might give better results which is illustrated in the figure 4.3.4 .

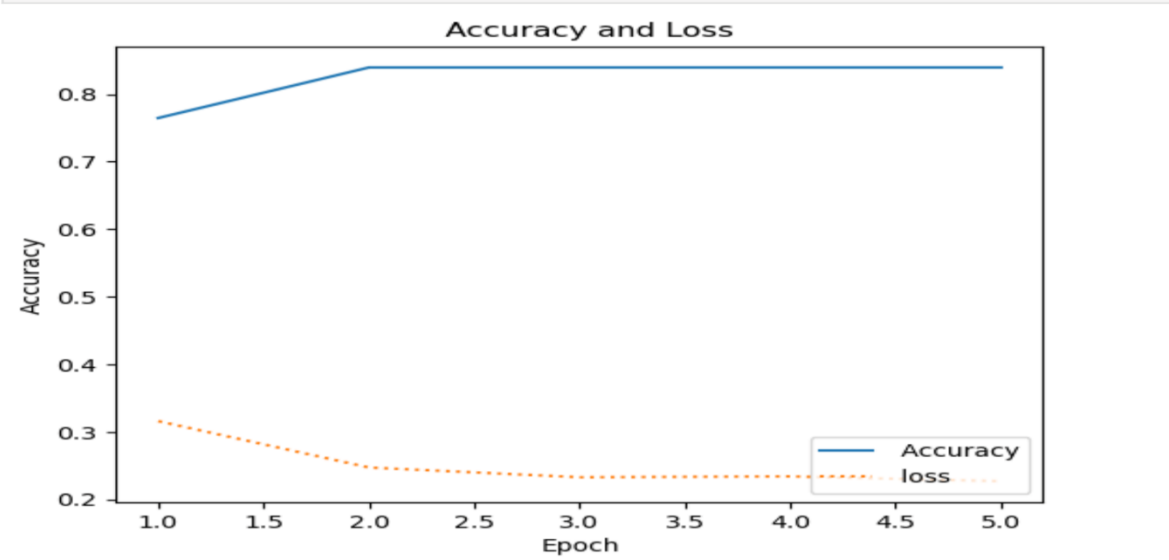


Figure 4.3.4: VGG16 Performance Graph after 5 Epochs of Training

There was a decrease in the enhancement in loss metric, from 0.27 to 0.21, presenting only a small improvement in model predictive metrics. This decrease in loss means VGG-16 is improving their data representation during training and a slight uptick in the number of patterns VGG-16 is able to identify within images containing crying infants. Specifically, this outcome

indicates the necessity of investigating new models to identify ways to improve classification accuracy and decrease loss which can be observed in the figure 4.3.5.

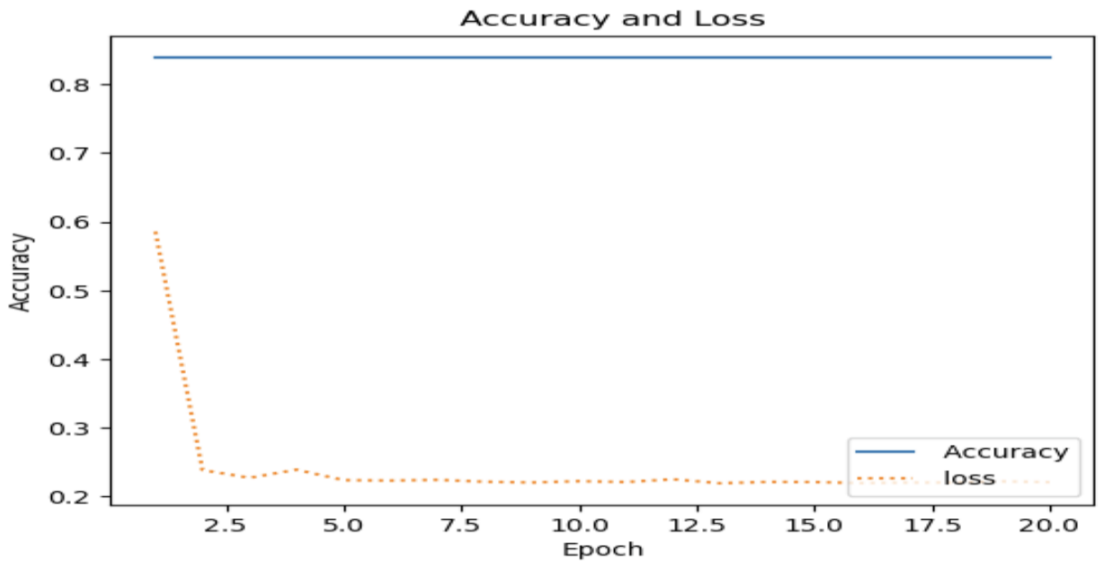


Figure 4.3.5: VGG16 Performance Graph after 20 Epochs of Training

4.3.2. DenseNet Experiment Result

For this investigation, we used DenseNet architecture to classify and analyze images of infant screams into five categories. Based on the figure 4.3.6 , dense connectivity and good feature propagation are the benchmarks of DenseNet, which enables the model to extract complicated patterns from the data with much less information. We trained the DenseNet model to build his networks and then checked the accuracy of the outcome prediction during the validation phase. The classification outcome of infant cry DenseNet emerges as powerful techniques with considerable improvement in accuracy and overall performance.

```
Model: "densenet121"
__________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #   Connected to
==================================================================================================
input_1 (InputLayer)         [(None, 180, 180, 3)]     0         []

zero_padding2d (ZeroPaddin   (None, 186, 186, 3)       0         ['input_1[0][0]']
g2D)

conv1/conv (Conv2D)          (None, 90, 90, 64)        9408      ['zero_padding2d[0][0]']

conv1/bn (BatchNormalizati   (None, 90, 90, 64)        256       ['conv1/conv[0][0]']
on)

conv1/relu (Activation)      (None, 90, 90, 64)        0         ['conv1/bn[0][0]']

zero_padding2d_1 (ZeroPadd   (None, 92, 92, 64)        0         ['conv1/relu[0][0]']
ing2D)
```

```
x = GlobalAveragePooling2D()(x)

predictions = Dense(5, activation="sigmoid")(x)

model = Model(inputs=base_model.input, outputs=predictions)
```

Figure 4.3.6: Part of the Densenet Architecture

To analyze the impact of the number of epochs on accuracy and convergence, the initial performance evaluation of the system was evaluated at different training iterations (5 and 20 epochs). From the figure 4.3.7 and 4.3.8 , we can see that After 5 epochs, the model showed acceptable accuracy and getting better over the previous iterations indicating that there were things that the model had actually picked up from the crying audio data. Results got better as more training iterations were made, and we will show more results in detail shortly.

```
Epoch 5/5
12/12 [==============================] - ETA: 0s - loss: 0.1275 - accuracy: 0.8920WARNIN
12/12 [==============================] - 28s 2s/step - loss: 0.1275 - accuracy: 0.8920 -
```

Figure 4.3.7: Densenet Results after 5 Epochs of Training

```
Epoch 20/20
12/12 [==============================] - ETA: 0s - loss: 0.0563 - accuracy: 0.9474WARNI
12/12 [==============================] - 27s 2s/step - loss: 0.0563 - accuracy: 0.9474
```

Figure 4.3.8: Densenet Results after 20 Epochs of Training

Results of the experimental indicate that there was a significant increase in the accuracy of the DenseNet model, reaching an accuracy of 0.89 during training for 5 epochs and increasing to 0.947 with 20 epochs. These results suggest that DenseNet can learn well and extract informative features from infant crying images and achieve better classification results which reflected in the figure 4.3.9 below.

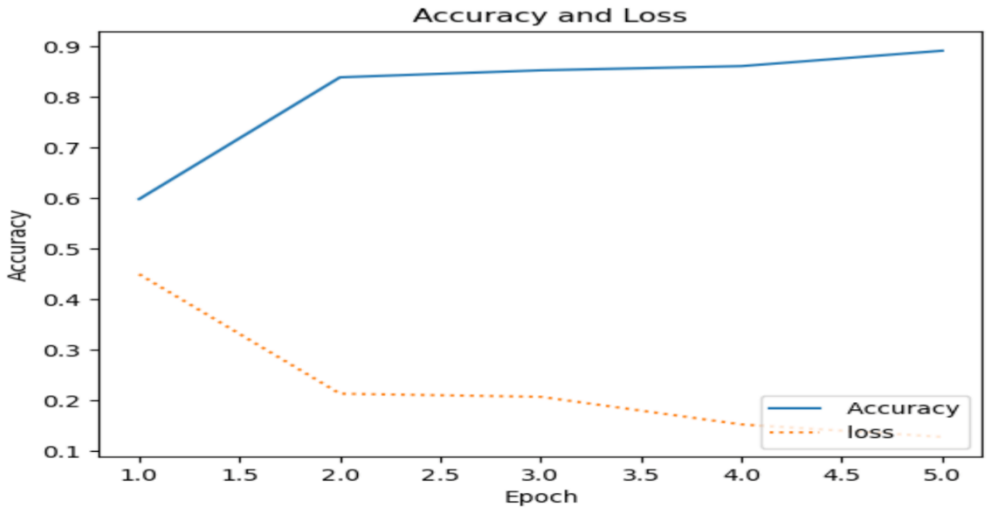


Figure 4.3.9: DenseNet Performance Graph after 5 Epochs of Training

Additionally, the loss metric was drastically improved, decreasing from 0.12 to 0.05, indicating that the model became more efficient in its predictions. From the reduction in loss it appears that DenseNet enhanced the representation of the data leading to generally better performance of the model. The findings prove that DenseNet is efficient for sophisticated classification tasks like infant cry analysis as displayed in the figure 4.3.10.

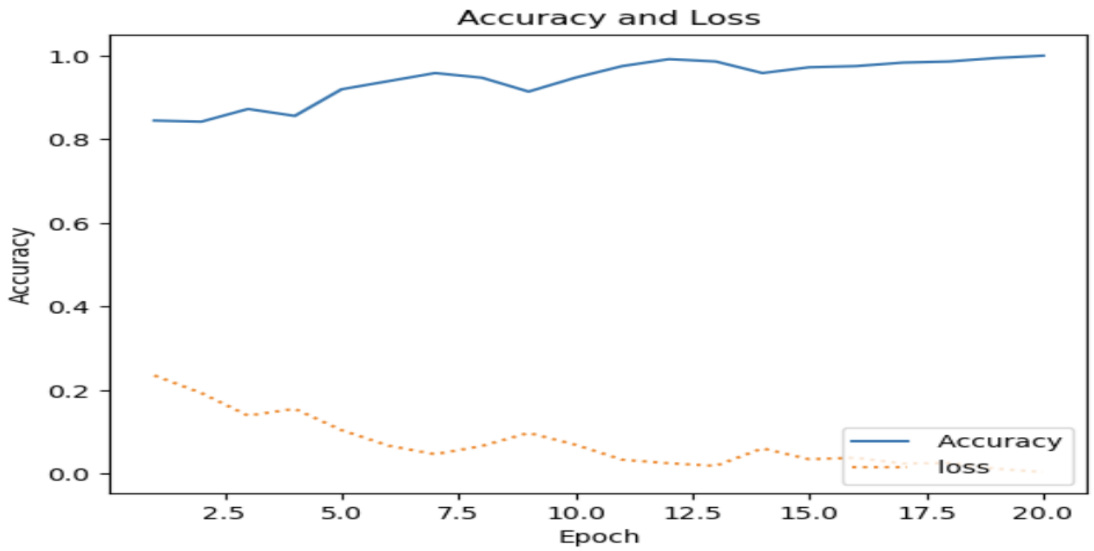


Figure 4.3.10: Densenet Performance Graph after 20 Epochs of Training

4.3.3. ResNet Experiment Result

This thesis focused on the investigation of the capabilities of ResNet architecture to understand the capabilities of various algorithms and their output for the classification of infant crying. Based on the figure 4.3.11, ResNet is renowned for its robust network, which allows it to train deeper networks effectively.

This new design enables the model to pick up subtle features that are typical of infant cries. Images in the dataset had been painstakingly curated to permit sharp distinction between the five categories. A rigorous training and validation process was performed to assess the performance of ResNet with key predictive strengths. The findings underscore ResNets potential for increasing the accuracy of infant cry classification and highlight ResNet's potential importance in this field of research.

```
Model: "resnet50"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape                 Param #   Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 180, 180, 3)]        0         []

 conv1_pad (ZeroPadding2D)      (None, 186, 186, 3)          0         ['input_3[0][0]']

 conv1_conv (Conv2D)            (None, 90, 90, 64)           9472      ['conv1_pad[0][0]']

 conv1_bn (BatchNormalizati     (None, 90, 90, 64)           256       ['conv1_conv[0][0]']
 on)

 conv1_relu (Activation)        (None, 90, 90, 64)           0         ['conv1_bn[0][0]']

 pool1_pad (ZeroPadding2D)      (None, 92, 92, 64)           0         ['conv1_relu[0][0]']

 pool1_pool (MaxPooling2D)      (None, 45, 45, 64)           0         ['pool1_pad[0][0]']

 conv2_block1_1_conv (Conv2     (None, 45, 45, 64)           4160      ['pool1_pool[0][0]']
 D)

 conv2_block1_1_bn (BatchNo     (None, 45, 45, 64)           256       ['conv2_block1_1_conv[0][0]']
 rmalization)
```

```
x = GlobalAveragePooling2D()(x)

predictions = Dense(5, activation="sigmoid")(x)

model = Model(inputs=base_model.input, outputs=predictions)
```

Figure 4.3.11: Part of the Restnet Architecture

The impact of the number of epochs on classification accuracy and convergence behavior was evaluated across different training iterations i.e. 5 and 20 epochs. Looking at the figure 4.3.12 and 4.3.13 ,once the training epochs reached 5 epochs, the model was accurate, which meant it had started to pick some of the main patterns present in the crying audio data and was performing okay. Furthermore, the losses were extreMely low, indicating that the model works! In the next section we will detail the results.

```
Epoch 5/5
12/12 [==============================] - ETA: 0s - loss: 0.2277 - accuracy: 0.8366WARNI
12/12 [==============================] - 27s 2s/step - loss: 0.2277 - accuracy: 0.8366
```

Figure 4.3.12 Restnet Results after 5 Epochs of Training

```
Epoch 20/20
12/12 [==============================] - ETA: 0s - loss: 0.0027 - accuracy: 1.0000WARNI
12/12 [==============================] - 26s 2s/step - loss: 0.0027 - accuracy: 1.0000
```

Figure 4.3.13 Restnet Results after 20 Epochs of Training

The results of the experiment show that the accuracy of the ResNet model was very much acceptable, with accuracy improving to 0.84 after the first 5 epochs and the second 20 epochs reaching one. The results demonstrate that ResNet is powerful techniques to learn well and find the main features in infant crying images and improve the classification accuracy which can be observed in the figure 4.3.14 .

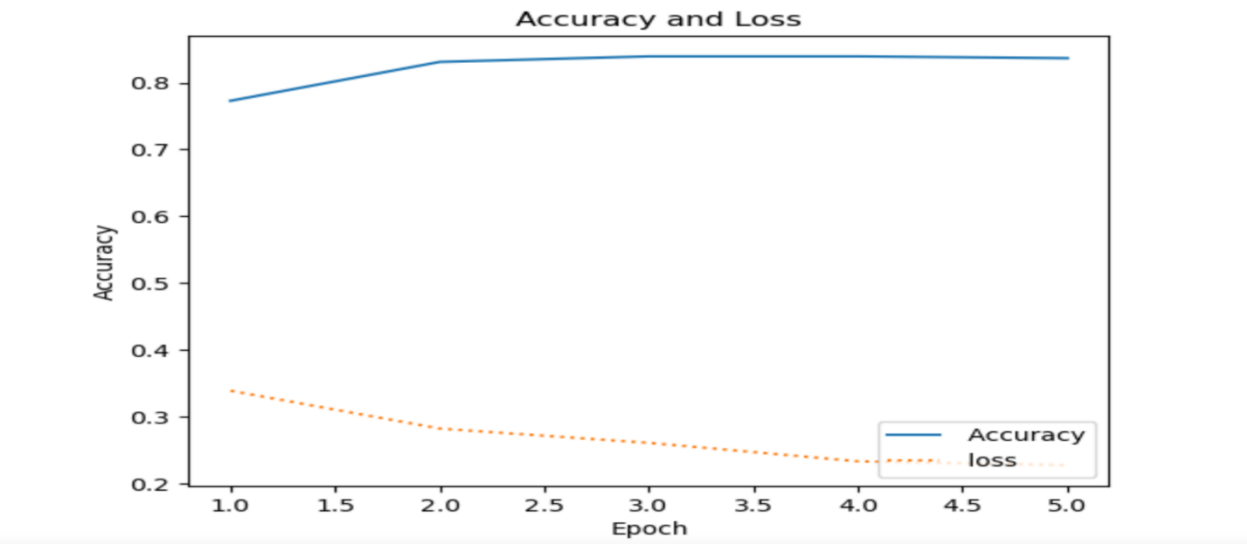


Figure 4.3.14: Restnet Performance Graph after 5 Epochs of Training

Additionally, the metric, which measures the loss of our model, improved remarkably, going from 0.227 to 0.0027, which means that our model had become more efficient in its predictions. The decrease in loss in this experiment implies that ResNet has helped the representation of data, helping the model to perform well. The ResNet results provide evidence of the network's power for advanced classification tasks, including infant cry analysis as shown in the figure 4.3.15 .

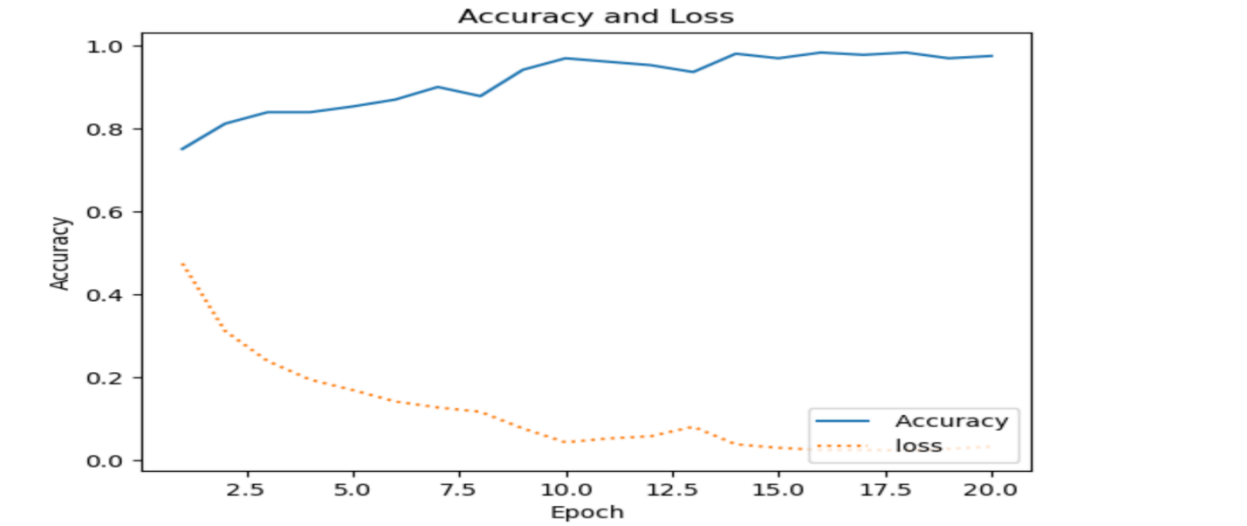


Figure 4.3.15: Restnet Performance Graph after 20 Epochs of Training

4.3.4. GoogleNet Experiment Result

In this research, based on the figure 4.3.16, we evaluate the performance of the GoogleNet architecture for classifying different data types with respect to infant cries. In this sense, GoogleNet stands out due to its inception modules which allow us to obtain progressive information capturing a wide variety of features, in an efficient and computationally effective way. The images that constituted the dataset were curated carefully and thus facilitated the accurate classification of infant cries into those five specific categories. The predictive performance of GoogleNet was evaluated entirely through a series of structured training and validation steps. As can be seen in the findings, GoogleNet is found to be a very powerful tool for classifying infant cry, and it is shown there has been good in accuracy and overall performance of infant cry classification using GoogleNet.

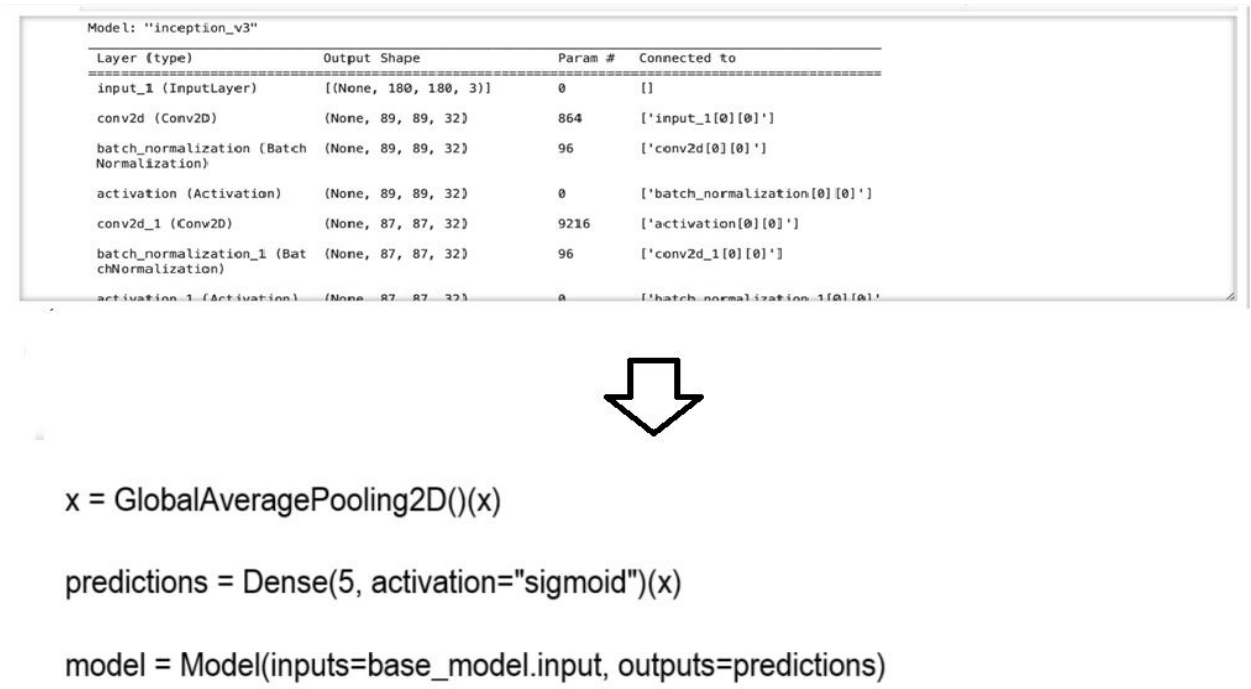


Figure 4.3.16: Part of GoogleNet Architecture

It has evaluated the accuracy of the system across different training iterations (at 5 and 20 epochs) to study the effect of epochs on classification accuracy and convergence. From the figure 4.3.17 and 4.3.18, we can see that after going through 5 epochs, we were able to get very good accuracy, which was better compared to the one obtained from ResNet architecture. This result illustrates that the model has been picking up some of the features within the crying audio data meaning they had a strong initial learning ability. But the outcome was not as very good on 20 epoch as expected.

```
Epoch 5/5
12/12 [==============================] - ETA: 0s - loss: 0.0600 - accuracy: 0.9418WARNIN(
12/12 [==============================] - 16s 1s/step - loss: 0.0600 - accuracy: 0.9418 -
```

Figure 4.3.17: GoogleNet Results after 5 Epochs of Training

```
Epoch 20/20
12/12 [==============================] - ETA: 0s - loss: 0.0454 - accuracy: 0.9723WARNI
12/12 [==============================] - 14s 1s/step - loss: 0.0454 - accuracy: 0.9723
```

Figure 4.3.18: GoogleNet Results after 5 Epochs of Training

The training results indicate that the GoogleNet model is able to make a good prediction which improves by 0.94 after 5 epochs till a maximum improvement of 0.97 after 20 epochs. The results show that GoogleNet is capable of learning and improving the accuracy of classification by learning important features from types of crying babies as seen in the figure 4.3.19.

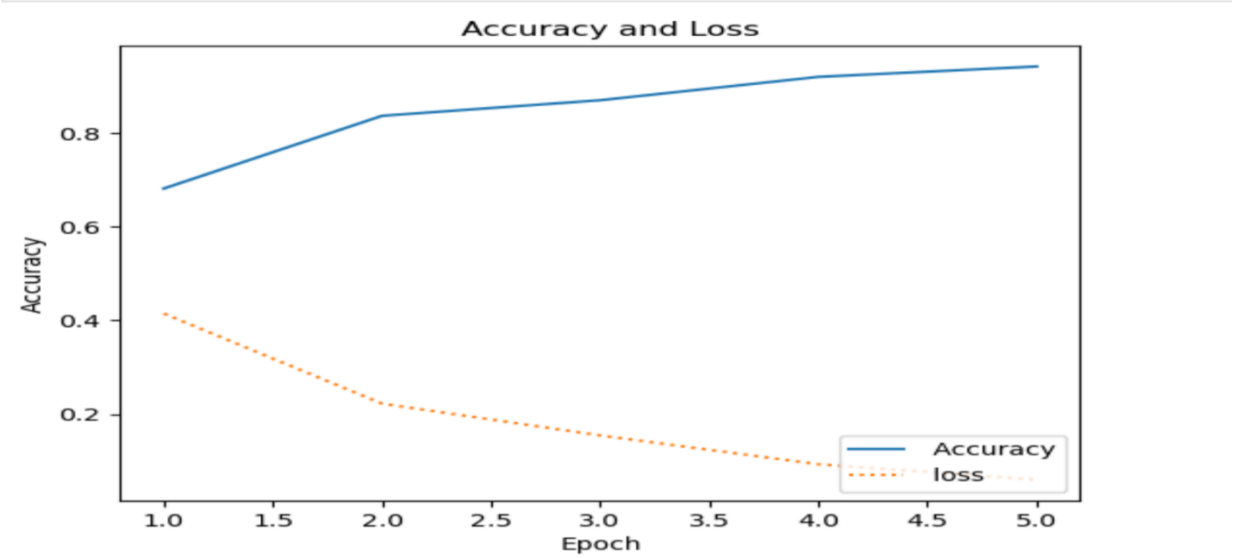


Figure 4.3.19: GoogleNet Performance Graph after 5 Epochs of Training

Also, the loss metric has improved slightly climbing from 0.06 to 0.054. The improvement realized in this model suggests that it became more efficient in its predictions. The less loss on this dataset means that GoogleNet is able to train the data representation, which contributed to the model's performance. This proves GoogleNet is adept in advanced classification applications e.g. infant cry analysis as displayed in the figure 4.3.20.

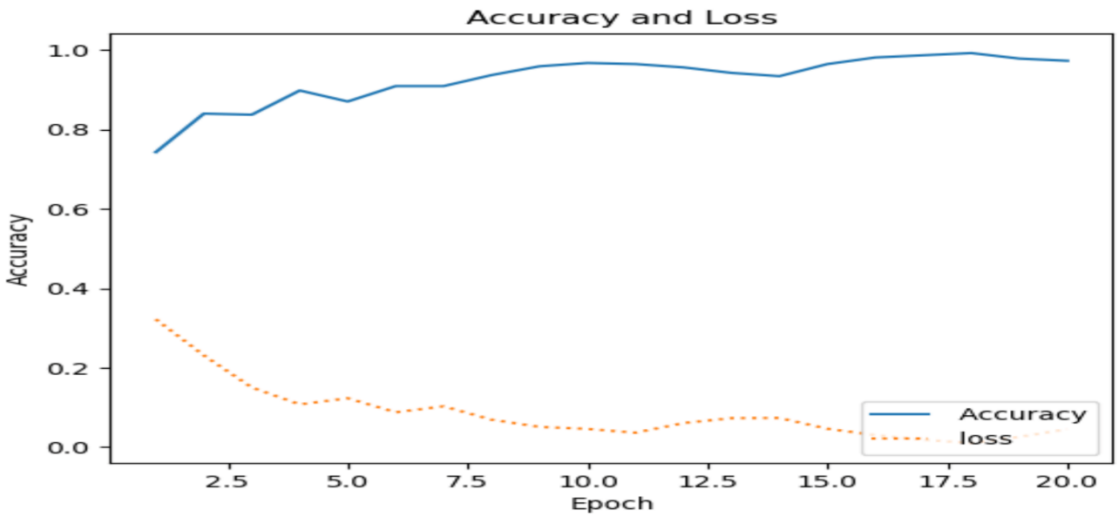


Figure 4.3.20: GoogleNet Performance Graph after 20 Epochs of Training

4.4. Local dataset results using deep learning techniques

This work evaluates the results obtained from classifying infant cry using the deep learning techniques on the locally collected data. We took 15 samples per category and put 75 samples total into creating these datasets. The first part of this section examines the result of classification, using these approaches, where the assessment metrics discussed in the previous chapter are subsequently used for the assessment. We show through analysis of these scoring metrics how to evaluate an implementation of the techniques we present and determine the level of effectiveness and quality.

The accuracy of each deep learning model across scenarios is the basis of which we extract the classification matrices and the quality of each algorithm's prediction.

After processing all the images in the dataset, they are utilized across three stages to train the neural networks: You have the training, the validation, and the test. The selected models for image analysis are as follows: The choice of different techniques based on Convolutional Neural

Networks (CNN) naMely DenseNet, VGG16, ResNet, and GoogleNet . The resulting algorithmic results are compared against adjustments made to the number of training epochs (5 and 20 epochs, respectively). Also shown is how classification performance is affected as a function of the growth in the number of training epochs.

We give details of the different detection and classification results achieved using the specialized deep learning technique used in this study.

4.4.1. VGG-16 Experiment Result

In our work, we exploited primarily the deep structure and efficient feature extraction ability in the VGG-16 architecture to classify and analyze infant cries. The chosen images belong to types of infant cries and the model could learn patterns in each category. With the help of training and validation stages, we evaluated the model's accuracy, including its capability to predict outcomes correctly from the given input images. These results show the ineffectiveness of VGG -16 in the infant cry classification domain.

The classification accuracy and convergence of the system were tested over different training iterations: 5 epochs and 20 epochs in order to ascertain what impact the number of epochs has on the accuracy of the classification. From the figure 4.4.1 and 4.4.2 , we can see that after running for 5 epochs, the model was not able to predict correctly and failed completely to classify.

Figure 4.4.1 VGG16 Results after 5 Epochs of Training

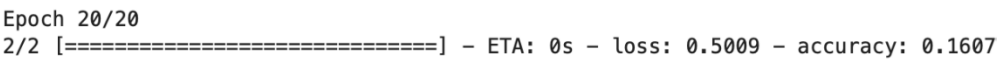


Figure 4.4.2 VGG16 Results after 20 Epochs of Training

In the test with 20 epochs, the VGG-16 model had an accuracy of 0.16, and in the test with 5 epochs, it was 0.19. This result indicates that the model is not good at learning from the data that is provided, which is likely why it fails to accurately categorize MFCC images of babies crying which is illustrated in the figure 4.4.3 .

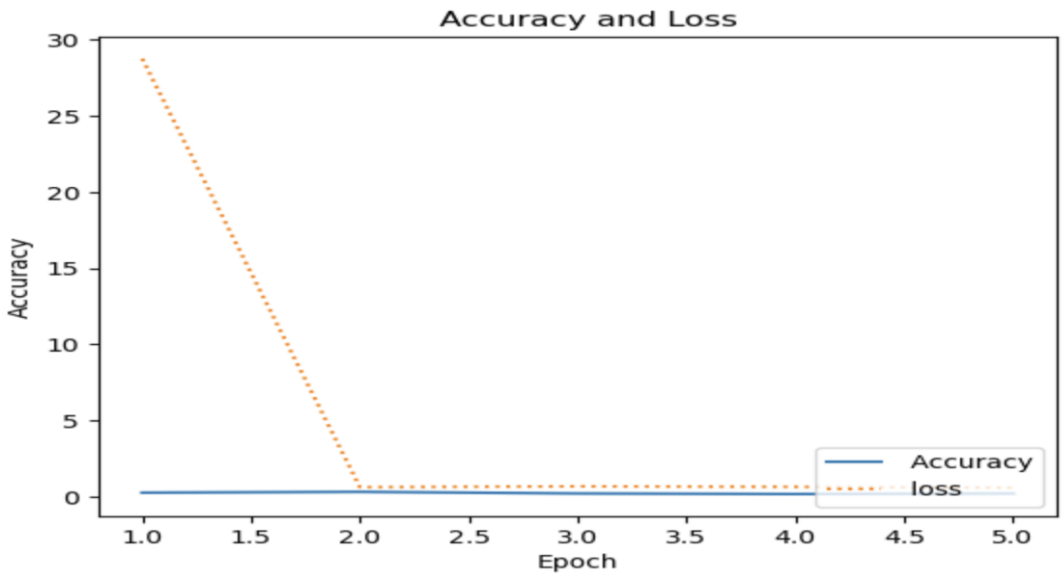


Figure 4.4.3: VGG16 Performance Graph after 5 Epochs of Training

The low accuracy stayed even after adding more epoch, which is really disappointing. We have a result related to this: this result shows us that the model can't learn the data adequately even with more training iterations which can be observed in the figure 4.4.4 .

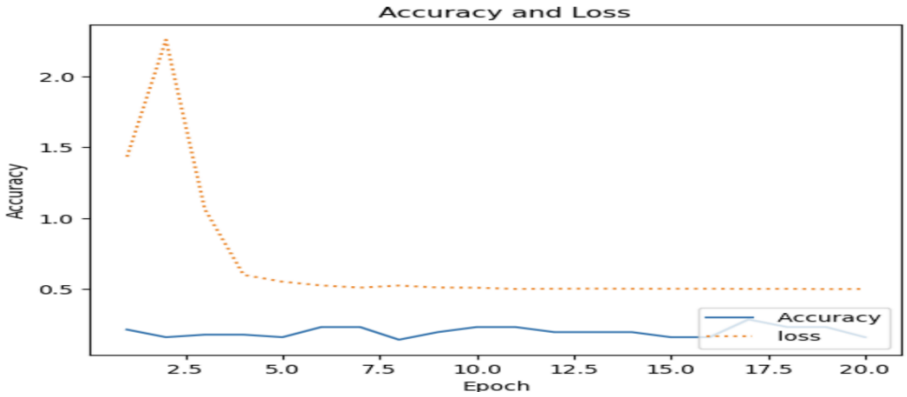


FIGURE 4.4.4 VGG16 PERFORMANCE GRAPH AFTER 20 EPOCHS OF TRAINING

Additionally, graphical representations show a very high loss model with a maximum of 2.5. This means that there are some big errors in the predictions that the model makes showing that after training for so long, it cannot correctly classify the data.

4.4.2. DenseNet Experiment Result

The DenseNet architecture was used to analyze and classify infant screams into five distinctive categories within this work. Due to its dense connection and effective feature propagation, DenseNet can learn fine patterns from data yet reduce information loss. The accuracy of DenseNet was methodically trained and validated, demonstrating it effective for outcome prediction. We find that DenseNet acts as a very robust instrument in infant cry categorization, significantly improving overall metrics including accuracy.

The results showed how the system performance changes with epoch count, while assessing the performance of the system for different training iterations — 5 and 20 epochs — to determine its impact on classification accuracy and convergence. From the figure 4.4.5 and 4.4.6 , we can see that after five epochs the model was sufficiently accurate indicating that it began improving over prior work and was now learning critical patterns from the cry data. As the training iteration increased, the accuracy scores of the model remained stable, which showed that the model has approached a state of learning stability.

```
Epoch 5/5
2/2 [==============================] - ETA: 0s - loss: 0.0279 - accuracy: 0.9821
```

Figure 4.4.5: Densenet Results After 5 Epochs of Training

```
Epoch 20/20
2/2 [==============================] - ETA: 0s - loss: 0.0126 - accuracy: 0.9821
```

Figure 4.4.6: Densenet Results after 20 Epochs of Training

It is observed that DenseNet model performs significantly better than the previous VGG 16 with DenseNet reaches an accuracy of 0.98 after 5 epochs and 0.9821 after 20 epochs. Our results show that DenseNet novel learning, and extracting discriminant features, from Infant cry images improves classification accuracy as seen in the figure 4.4.7.

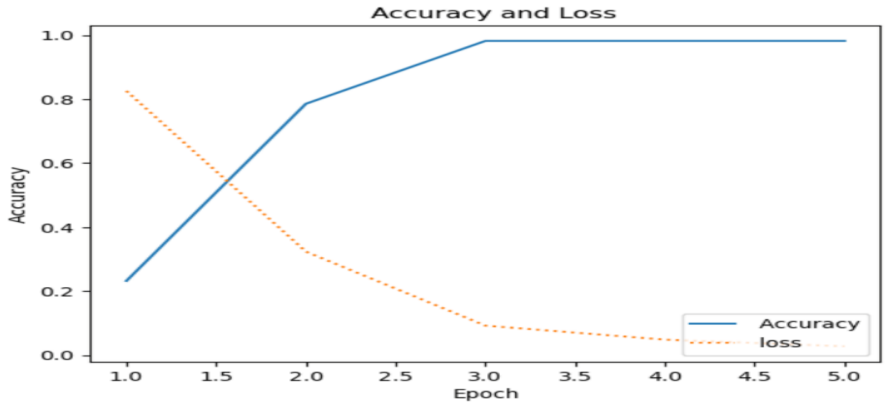


FIGURE 4.4.7 DENSENET PERFORMANCE GRAPH AFTER 5 EPOCHS OF TRAINING

Moreover, the loss metric was greatly improved, from 0.27 to 0.12, meaning the model improved at classification. This lowering of loss means that DenseNet actually taught the model to better represent the data as shown in the figure 4.4.8 .

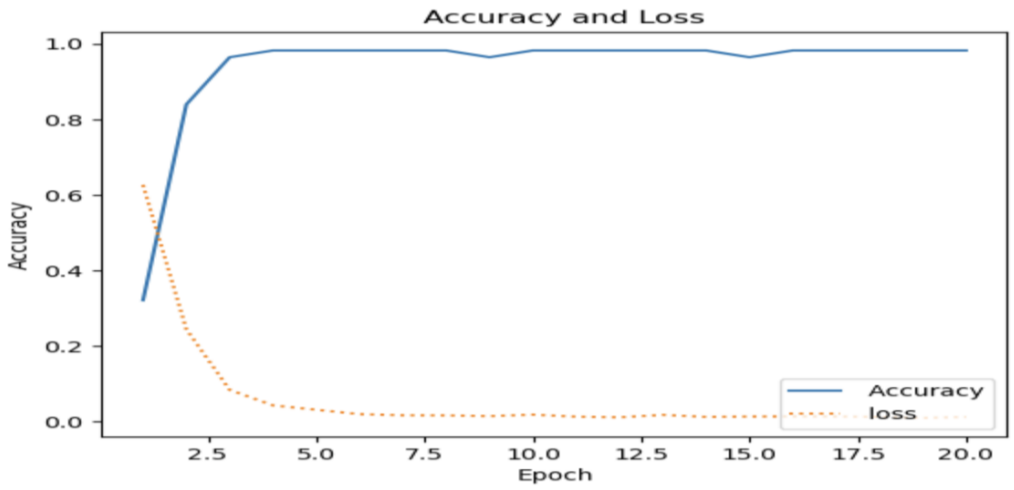


Figure 4.4.8: Densenet Performance Graph after 20 Epochs of Training

On this data, DenseNet classification was shown to significantly outperform VGG-16. Results show that DenseNet was able to optimize this classification assignment by being able to collect and utilize more underlying patterns in the data.

4.4.3. ResNet Experiment Result

In this study, we explore the ability of the ResNet architecture to inform the analysis of the algorithms and their outputs to classify infant cries. The resilience of the vanishing gradient problem during the training of deeper networks makes ResNet interesting because of its innovative implementation of residual connections, which enables the training of deeper networks. The beauty of this design is that the model can capture so much detail in the infant's cries. The MFCC images in the dataset were carefully curated to be easily separable into the categories. We evaluated

ResNet's performance via rigorous training and validation wherein predictive capabilities were emphasized. The results show that ResNet is capable of boosting infant cry classification.

Classification accuracy and convergence behavior of the system were examined across different training iterations, at 5 and 20 epochs in particular to see the effect of epoch count on classification accuracy. From the figure 4.4.9 and 4.4.10 , we can see that after 5 epochs, the model had started picking up some strongest patterns related to cry data and taken decent steps in terms of its robustness. With very low losses, it meant the model was working pretty well. The detailed results are discussed in the next section.

Figure 4.4.9: Restnet Results after 5 Epochs of Training

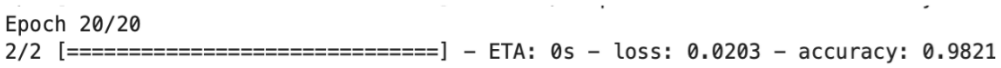


Figure 4.4.10: Restnet Results after 20 Epochs of Training

Results of experiments show that the ResNet model has an accuracy value of 0.94 reached after 5 epochs of training, but after 20 epochs this value was 0.98. Results show that ResNet is able to learn and learn features from cry infant image, increasing iteration helps to increase the accuracy of the algorithm as displayed in the figure 4.4.11.

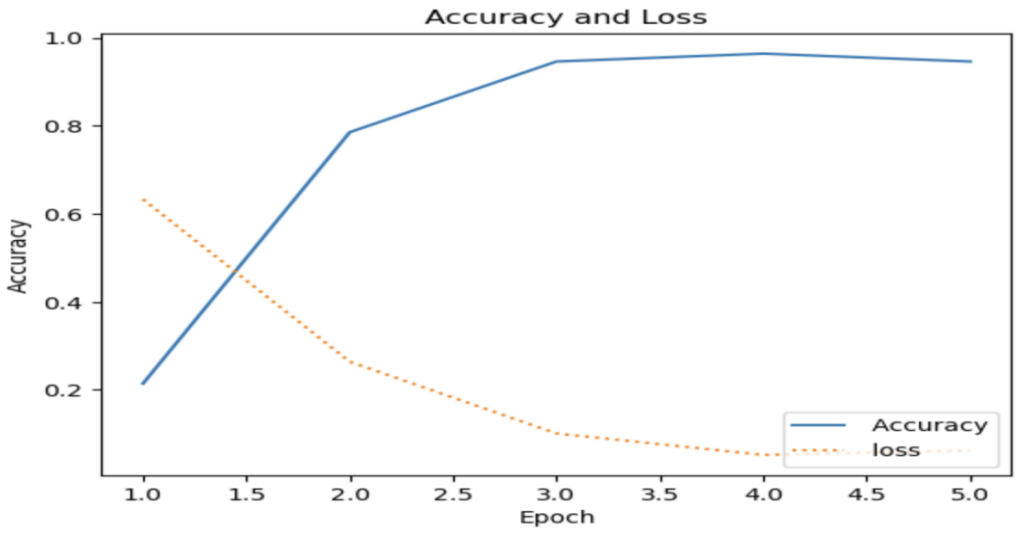


Figure 4.4.11: Restnet Performance Graph after 5 Epochs of Training

The efficiency in predicting the model using the loss metric presented proved to be improved, with the loss metric from 0.0626 to 0.0203. This reduction in loss suggest that ResNet found an efficient way to optimise their representation for that data across the training process, resulting in the model improving its overall performance which reflected in the figure 4.4.12 below.

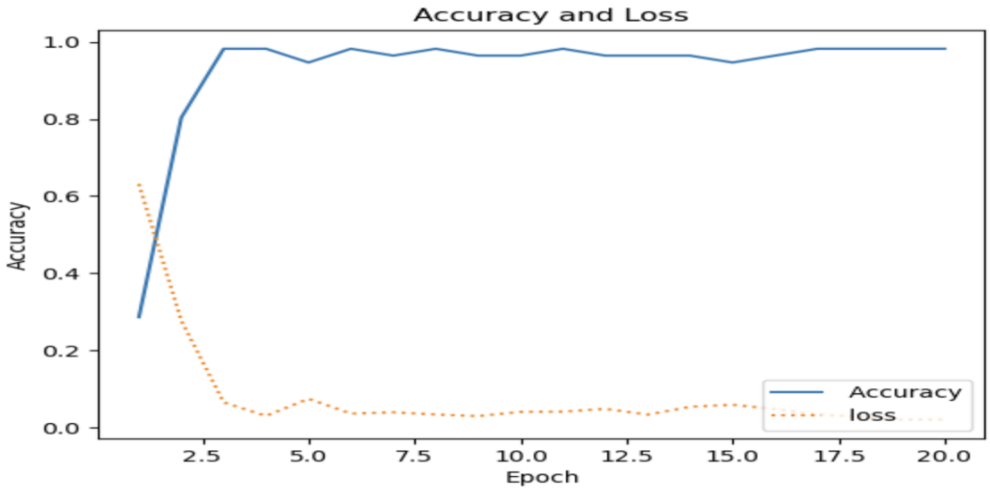


Figure 4.4.12: Restnet Performance Graph after 20 Epochs of Training

After 20 iterations, the model's accuracy was similar to DenseNet while surpassing VGG-16. Thus, feature extraction in this context as well as the effectiveness of ResNet is evidenced.

4.4.4. GoogleNet Experiment Result

In this study we provide an assessment of the efficacy of the GoogleNet architecture used to categorize different types of data associated with infant cries. The inception modules of GoogleNet characterize GoogleNet and enable efficient capture of a variety of features while improving computational performance. The systematically arranged images of different types of infant cry images proved suitable for accurate classification in the dataset. We systematically evaluated Google Net's performance including training and validation phases and specific examination of its predictive capabilities.

To assess the system performance over different training iterations, several experiments were performed with epoch count equal to 5 and 20, in order to examine the impact of epoch count on the classification accuracy and convergence behaviors. As revealed in figure 4.4.13 and 4.4.14 , the model showed similar accuracy (as ResNet does) after five epochs as after twenty epochs. After 20 epochs we saw a decline of accuracy.

At the beginning of training, the model showed low loss values during the first 5 epoch. The lesser loss means that the model is able to parameterize successfully towards data that we trained on.

```
Epoch 5/5
2/2 [==============================] - ETA: 0s - loss: 0.0330 - accuracy: 0.98
```

Figure 4.4.13: GoogleNet Results after 5 Epochs of Training

```
Epoch 20/20
2/2 [==============================] - ETA: 0s - loss: 0.0300 - accuracy: 0.96
```

Figure 4.4.14: GoogleNet Results after 20 Epochs of Training

Experimental results show that the GoogleNet model accuracy was substantially good since 0.98 is reached in the first 5 epochs of training but it falls down to 0.96 in the first 20 epochs. This demonstrates the model's huge predictive power; yet with a rise in iterations there was an increase in error, and a decrease in accuracy as shown in the figure 4.4.15.

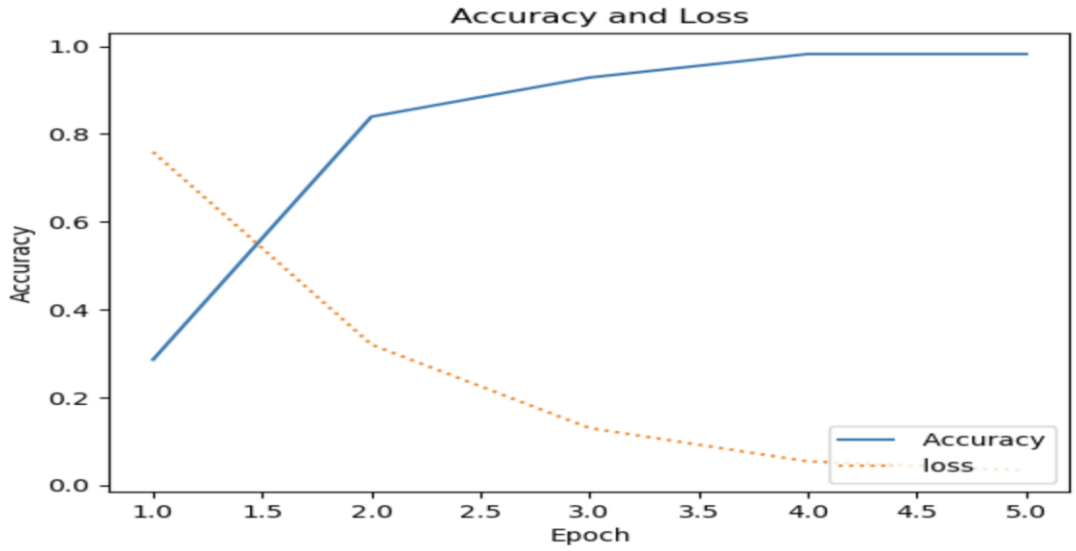


Figure 4.4.15: GoogleNet Performance Graph after 5 Epochs of Training

The loss metric was constant at 0.3, meaning that there was no growth of prediction loss when we got a drop in accuracy. However, even if the accuracy decreases, the stability in loss implies that the loss did not increase while keeping a learning phenomenon such as overfitting or learning plateau happening this means that the model still maintained a constant error level in its prediction, the error is simply little bit high than Restnet which reflected in the figure 4.4.16 below.

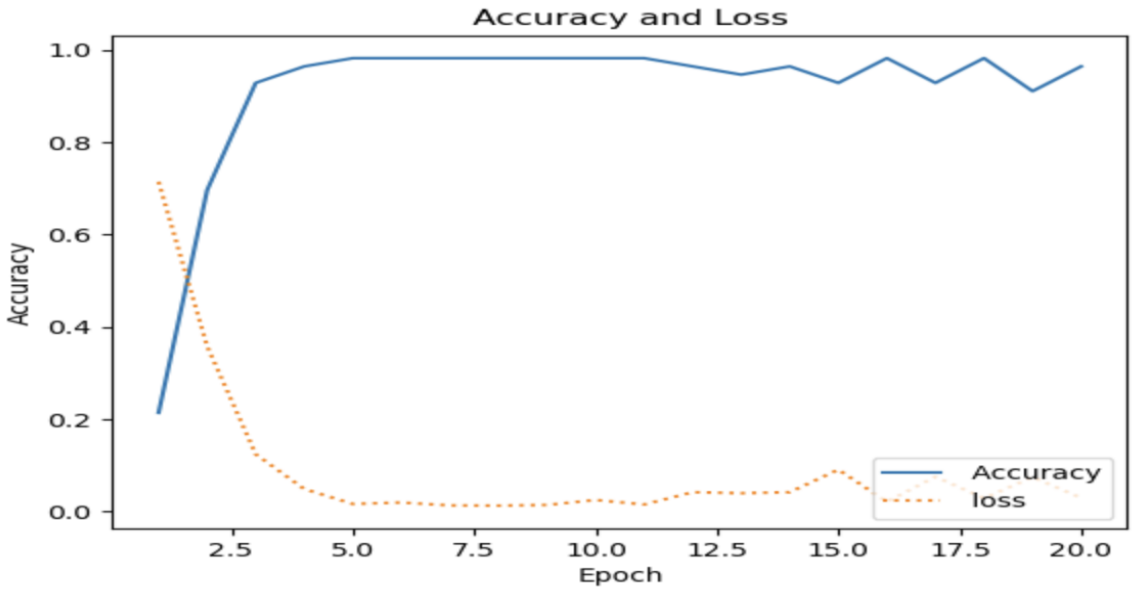


Figure 4.4.16: GoogleNet Performance Graph after 20 Epochs of Training

No critical improvement in results was shown for GoogleNet compared to the previous two methods, which its accuracy was even lower. This decrease implies that for the task which the classification should perform, GoogleNet could be less fitted compared to the other tested models.

4.5. Integrating Random Fourier Features in Deep Learning Models

4.4.5. Random Fourier Features (RFF)

Random Fourier Features (RFF) is a way to efficiently approximate kernel functions to improve the classification of complicated data such as infant crying. The integration of Random Fourier Features (RFF) with Convolutional Neural Networks (CNNs) allows data input to be transformed into a higher dimensional feature space such that complex patterns and patterns in the audio representation of the infant cry may be captured by the model. It allows us to distinguish clearly different kinds of cries which correspond to different infants' needs (such as hunger or discomfort.

We implemented the algorithm described in the paper [61], and we got variations in the results that the outcomes change each time the code is run. This discrepancy can for example be caused by the variation in input data, the model initialization, or the methodology used for splitting the dataset into training and testing sets. The results reconfirm the importance of repetition and intensive examination of the use of methodology in the study.

Also, the code that they used shows how to use Random Fourier Features (RFF) in the final layer predictions to generate five classifications of infant cries. Typically, when implementing this, the output from the previous layers will also be passed to the prediction layer, see figure 4.5.1 , however, it will first convert this output to a higher dimensional space than the features are in before transmitting this to the prediction layer. Using the RFF in this way may allow the model to represent complex patterns and relations within the data resulting in better classification

performance over the five distinct cry categories. This method may help to improve the model's performance in discriminating between different infant needs so that careers can respond to them more efficiently.

```
model = Sequential()
model.add(InputLayer(input_shape=(224, 224, 3)))
model.add(Resizing(64,64))
model.add(Normalization())
model.add(Conv2D(64, 3, strides=1, activation='relu'))
model.add(Conv2D(128, 3, strides=1, activation='relu'))
model.add(MaxPooling2D(2, 2))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(RandomFourierFeatures(5))
model.compile(optimizer=AdamW(lr=0.01), loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Figure 4.5.1: Model Architecture Implemented in the Paper

We found that the accuracy, with a corresponding loss of 8.5, after 50 epochs of training was 0.58. Basically, the results show that there is poor performance in classification which may mean that the model did not manage to retrieve the underlying patterns in the data. The noted performance is shown in the graph whose performance is based on the absence of an accuracy increase and high loss values as shown in the figure 4.5.2 .

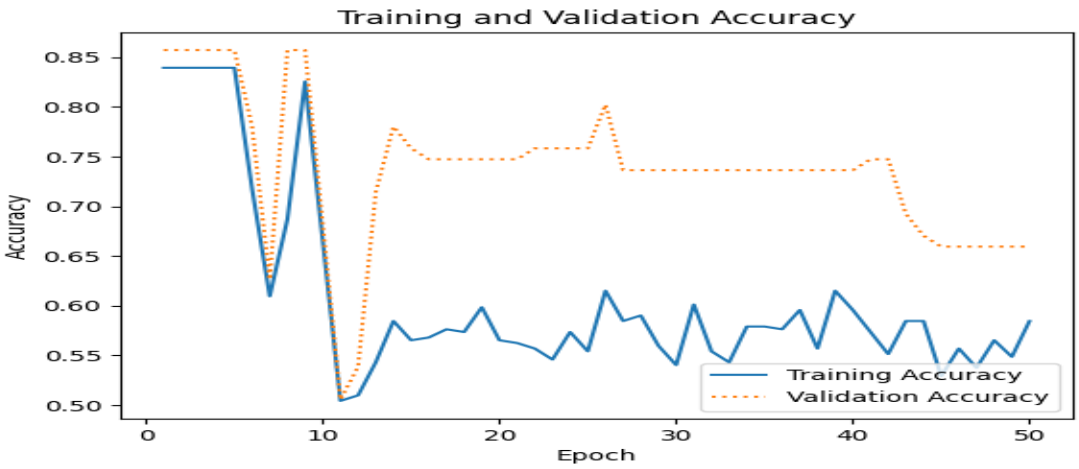


Figure 4.5.2: Results of the Model Implemented in the Paper

With this model, we ran multiple experiments which produced different outputs. The result is 0.25 accuracy with 12.5 loss and 0.35 accuracy with 3.85 loss. The best outcome we achieved was an accuracy of 0.7 with a loss of 0.8 which is illustrated in the figure 4.5.3 . These results show the inconsistency in the model's performance. This shows that the model requires further optimization and the possibility of changing training parameters or architecture to improve classification.

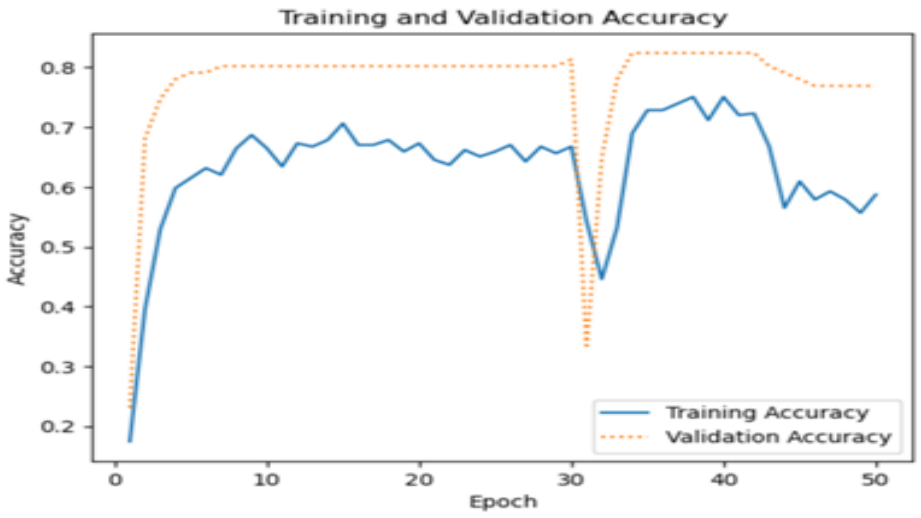


Figure 4.5.3: Best Performance Results of the Model Implemented in the Paper

A lot of investigations and experiments were performed, it is aimed at integrating RFF with the deep learning techniques previously explored, and the integration was motivated by the prior discrepancies. We applied this approach to the DenseNet algorithm and measured the possible improvement or stability in the model performance. The goal of this integration is to refine the classification results as well as to minimize the discrepancies found in earlier experiments. Also, we will then discuss this integration in the application context, as well as how its inclusion within DenseNet can positively impact overall algorithm performance and accuracy as related to the classifying of infant cries.

4.4.6. 4Integrating Random Fourier Features with DenseNet for Enhanced Classification 1

With this layer we will carry out several tests using DenseNet to assess its impact. Random Fourier Features are integrated with the DenseNet architecture to investigate potential improvement in model accuracy and consistency in classifying infant cries. Through systematic experimentation, subsequently we will analyze performance metrics (e.g., accuracy and loss) to investigate the effect of the integration of these two approaches. The results will yield insights into the effectiveness of RFF on DenseNet and its implication for the improvement of classification.

The code shows that the final layer was changed, and the dense layer was replaced with the RFF layer and configured as the output layer, see figure 4.5.4. The motivation for this adjustment is to improve the quality of the prediction using the transformed features generated by Random Fourier Features. To improve the model's classification performance for different types of infant cries, this new layer structure is proposed. We anticipate that integrating this layer will allow learning from the input data and ultimately the predictive accuracy of the model to be improved overall.

```
#model = Sequential()
base_model = DenseNet121(include_top=False, weights='imagenet')
x = base_model.output

x = GlobalAveragePooling2D()(x)

predictions = RandomFourierFeatures(5)(x)

model = Model(inputs=base_model.input, outputs=predictions)


#model.add(base_model)
#model.add(GlobalAveragePooling2D())
#model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Figure 4.5.4: Proposed Initial Architecture of the Model

The code has been executed and the results are as shown in the figure below. Here we show the performance metrics our model achieved (accuracy and loss values) along with what trends were significant during training. The figure 4.5.5 below gives an idea of how well or not our model works at classifying the infant cries.

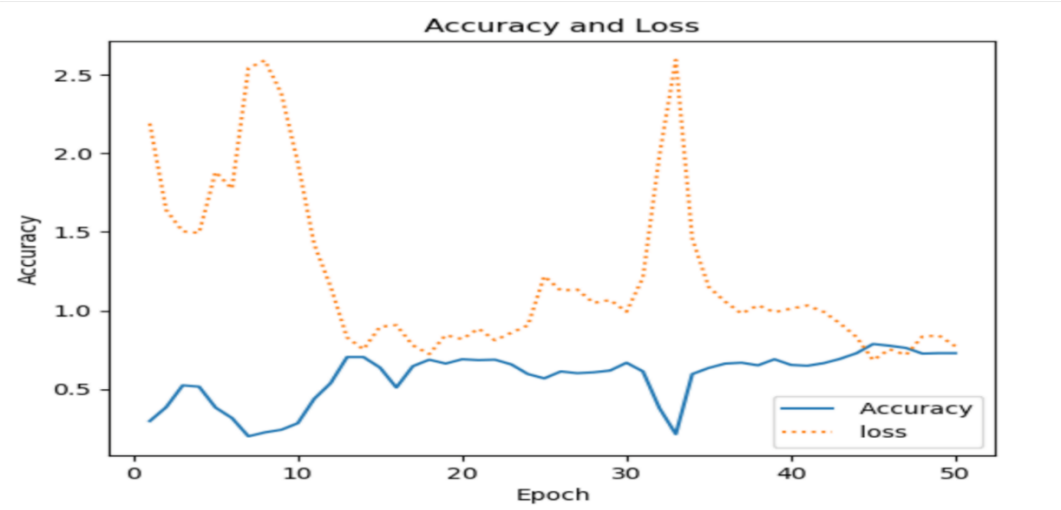


Figure 4.5.5: Results from Applying the Proposed Initial Architecture to the Model

The performance is relatively stable during training in all attempts because the loss did not exceed 2.5 in all. There wasn't also improvement in accuracy, it did not attain more than 0.7. This also indicates that despite keeping the loss at a constant level, the model was not able to improve its ability to classify infant cries. The results obtained show that there is a need for additional optimization, e.g., fine-tuning the model architecture, in order to get higher accuracy estimates while keeping the stability given by the loss values.

Reconstructing the model has been made in the next section. And it has been evaluated, to check if the Random Fourier Features layer can help capture useful features and improve the classification accuracy. We then analyzed the results from this configuration and discussed them

in subsequent analysis, and how they affect the performance of the model as a whole for classifying infant cries.

4.4.7. Integrating Random Fourier Features with DenseNet for Enhanced Classification 2

In this test, the dense layer has been put after the Random Fourier Features (RFF) layer to see if doing so could improve the model's performance. Reconstructing the model has been done so that we can analyze the effects and advantages of the dense layer along with the RFF transformation. The basic module configuration used in this case is shown in the code shown in figure 4.5.6 below, where the combination of these layers may help the model acquire knowledge about data distributions and improve the learning of the classification. With this implementation, we can test and measure the model outcome and test how accurate it is.

```
#model = Sequential()
base_model = DenseNet121(include_top=False, weights='imagenet')
x = base_model.output

x = GlobalAveragePooling2D()(x)

x= RandomFourierFeatures(5)(x)

predictions = Dense(5, activation="sigmoid")(x)

model = Model(inputs=base_model.input, outputs=predictions)

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Figure 4.5.6: the Second Proposed Architecture to the Model

The results were satisfactory when reconstructing the model in this arrangement, the system has an accuracy of around 0.92 with a loss of 0.23. During all training attempts, the loss did not exceed 0.24, this means the model is stable. Interestingly, the loss was still decreasing while accuracy did not change from the tenth epoch. These results show that the model is getting better indicating that Random Fourier Features (RFF) can be a powerful tool to be used to enhance model performance in future studies. Also, it has been found that it seems that integrating RFF may improve model accuracy and robustness and recommend RFF as a possible exploratory tool for future research in this domain which reflected in the figure 4.5.7 below.

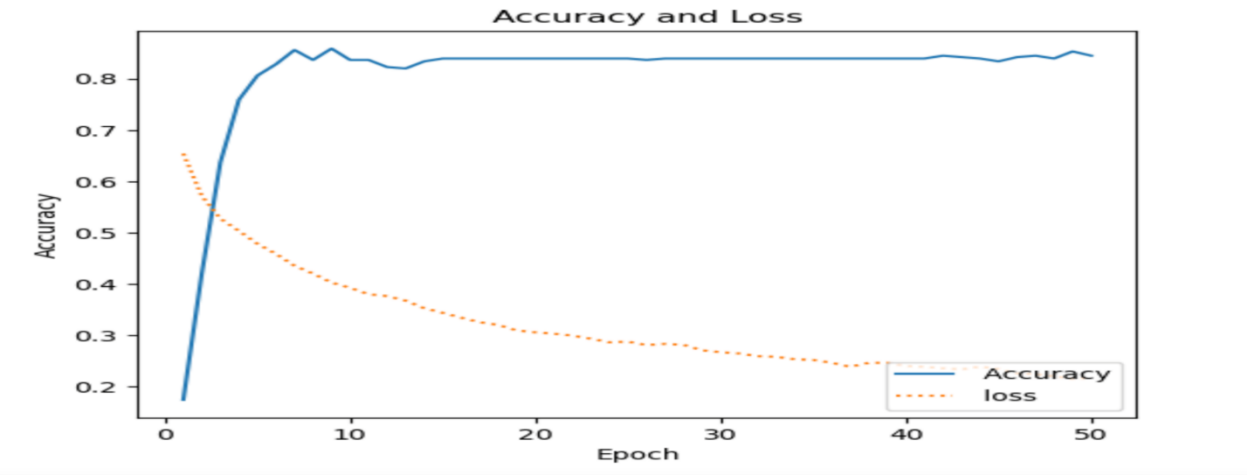


Figure 4.5.7: Results from Applying the Second Proposed Architecture to the Model

In short, when the Random Fourier Features (RFF) layer is put after the dense layer in a model, the model's performance is significantly boosted due to RFF's examples with an enriched feature representation. The RFF layer converts input data into a high-dimensional space in which complex patterns and relations within the data are better captured under the assumption of a Gaussian random process. This transformation allows the dense layer to receive better features, so

it learns and understands the features better! Thus, the model can provide higher classification, accuracy, and generalization in the cases of complex and nonlinear relationships.

4.6. Result discussions

After significant experimental and testing work producing image prediction using advanced machine learning techniques together with evaluation of local and global datasets and integration between this research and the current theories, we move on to discuss the results from these evaluations.

On global datasets, ResNet and DenseNet algorithms have observed that the performance is strong and efficient for this type of data. While GoogleNet performed satisfactorily but not as expected. Moreover, the worst results were obtained with VGG16 as displayed in the figure 4.6.1 below.

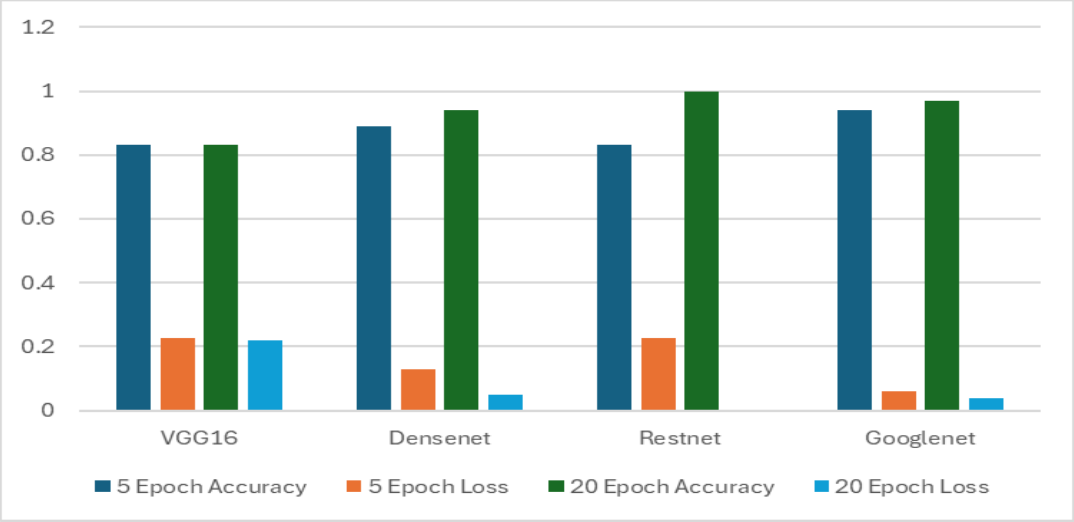


Figure 4.6.1: DL Techniques Performance on Global Datasets

It has been found that ResNet and DenseNet algorithms perform very well and efficiently on Local datasets. While GoogleNet did a decent job, it wasn't quite what was anticipated. On top of that, VGG16 produced the most subpar outcomes. See figure 4.6.2.

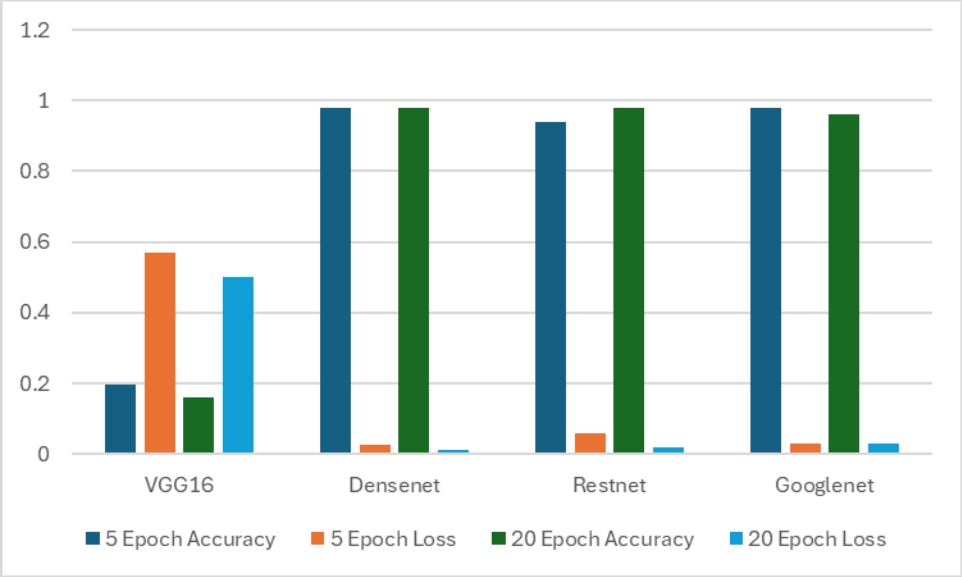


Figure 4.6.2: DL Techniques Performance on Local Datasets

Random Fourier Features (RFF) layer has been integrated with the Dense layer by putting the Dense layer at the end of the architecture, the model's performance and stability have been largely improved which can be observed in the figure 4.6.3. A combination seems to enhance the model to learn complex patterns makes its predictions more accurate.

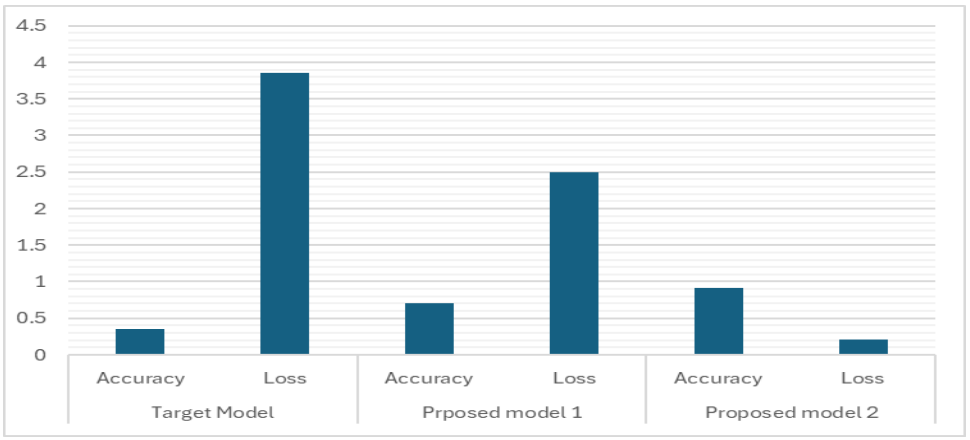


Figure 4.6.3: Proposed Models Performance on Global Datasets

Due to its enhanced feature representation, Random Fourier Features (RFF) samples improve model performance when placed after the dense layer. The RFF layer turns input data into a high-dimensional space where a Gaussian random process better captures complicated patterns and relations. This change improves the dense layer's feature learning and understanding! Thus, the model can classify accurately.

This study successfully addresses the primary research question of identifying the most effective method for classifying infant crying signals by transforming them into input formats for Deep Learn models and evaluating the ability of different Deep Learn models to accurately classify infant crying based on acoustic features. The results of the study demonstrate a significant improvement in the accuracy and stability of the models after incorporating the Fourier randomization properties of the hybrid architecture, compared to traditional convolutional neural network models such as (VGG-16, DenesNet, ResNet, GoogleNet). This confirms that hybrid architectures not only enhance feature learning but also generalize better across diverse datasets.

Also, the addition of Random Fourier Features (RFF) to CNN-based models answers our first research question: What is the most appropriate approach to classify infant crying signals using deep learning? The results unambiguously show that the use of RFF inside DenseNet buildings does increase the accuracy of the model (reaching up to 0.92) and its stability (when conformed with the loss level less than 0.24), especially in the second approach (Dense followed RFF transformation). This fusion enabled the model to more effectively model complex nonlinear acoustic structures in infant cry sounds that traditional CNN architectures could not robustly learn.

Compared with related works, [22] which used the MFCC features with SVM or Random Forest and had good accuracy, our approach shows better generalization capacity over datasets and is more robust in learning deeper representations. Unlike [31] which tries to improve the temporal learning using CNN-RNN hybrids, our approach sought to improve internal feature representation while avoiding the increase in temporal complexity. The results also differ from the baseline implementation in [61]: when using RFF only, the performance was inconsistent and low (accuracy was as low as 0.25), showing that RFF alone is not enough. Our optimized fusion with DenseNet effectively addresses these problems, suggesting a joint consideration of the structural configuration and the feature transformation is necessary to ensure high-quality classification performance.

As we can see, the results presented in our paper demonstrate that RFF, if well integrated into a deep network (as, for example, DenseNet is), are able to improve both accuracy and generalization, thus proving to be an interesting approach for infant cry classification and to provide an advance to the state-of-the-art in such field. This shows a significant improvement over

prior work, which either did not generalize well, were limited to a single dataset or were only able to learn shallow patterns from complex acoustic data.

Based on these results, it seems like future studies should focus more on hybrid architecture and how to set them up to improve accuracy in classification.

Chapter 5

Conclusion and Future Work

5.1. Challenges and Difficulties

Despite the hard work and ultimately satisfying results, I faced several challenges, most notably during the data collection phase. The dataset for classified, high-quality infant cries was very limited. Furthermore, the dataset collected locally was limited due to the lack of response from a large number of healthcare institutions or mothers, who refused to record infant cries. Analyzing and interpreting crying sounds was also difficult due to the similarity in tone, such as hunger cries and sleep cries, such as sleepy cries (Owh) and discomfort cries (Heh). It was also difficult to find a specialized expert to help classify the collected cries. Furthermore, environmental noise during the recording of sounds significantly affected the quality of some sounds, limiting the number of sounds in the final dataset to be processed and applied to deep learning techniques. Furthermore, environmental noise during the recording of sounds significantly affected the quality of some sounds. These challenges highlight the need for continued research and development in this area, including expanding datasets and improving model structures.

5.2. Conclusion

Infant cry classification is an important area in healthcare and behavior analysis since knowing the state the child is in and what their needs are is essential in the healthcare area and in

determining the correct method of behavior analysis. In this work, we evaluate the performance of multiple deep learning models on classifying infant cries to minimize the error in correctly anticipating what type of cry an infant is emitting, e.g., for hunger or pain.

In this study, we answered two main questions. The first question was: What are the most appropriate methods for converting infant cries into images that can be used with deep learning models? After researching, examining, and comparing acoustic feature extraction techniques, we found that the Mill-Frequency Coefficients (MFCCs) are the most popular and widely used in scientific articles on sound classification. In our study, we used MFCCs to extract meaningful acoustic features and convert them into spectrograms. This technique provided us with an informative visual input for training deep learning models. For the second question, we compared four well-known deep learning frameworks—VGG-16, DenseNet, ResNet, and GoogleNet —to see which model performed best at extracting the most important features from the dataset and performing accurate classifications. After training deep learning models, specifically convolutional neural network algorithms, on the collected, classified, and converted data into spectrograms, the models were able to analyze the acoustic patterns and distinguish each type with high efficiency. The ResNet and DenseNet algorithms, in particular, demonstrated the best performance, outperforming both the global and local datasets used with high accuracy. Therefore, we can assert that AI algorithms are capable of recognizing infant cry types at different levels and are a promising tool for improving and diagnosing infants' conditions based on their voice, facilitating responses in childcare settings, and meeting their needs.

The study results provided that the ResNet algorithm achieved a classification accuracy of 100% when analyzing global data, and both DenseNet and GoogleNet achieved 98% accuracy. On

the contrary, VGG-16 performed the worst testifying that every algorithm should be tested in a way that suits the data. Analyzing ResNet on local data also proved to be equally effective in classifying infant cries thus validating the overall success of ResNet.

The findings from the state-of-the-art model with Random Fourier Features (RFF) combined by DenseNet infer a significant progress of baby crying signals classification. In particular, the performance of the reformed model reached the accuracy of about 0.92 and the low consistent loss of 0.23, revealing the high predictive capacity and the robustness of the training. This is a significant improvement, in comparison to the previous configurations (such as the original model or other baseline architectures) where the accuracy was consistently underwhelming with 70%. The smooth decline of loss in every training epoch without notable fluctuation shows the reliability of the improved architecture.

The results of both of these experiments verify that the addition of RFF allows the model to map the input data into a high-dimensional space, in which more complex and non-linear acoustic patterns can be modeled. This capability is especially desirable for medical use cases, as correct classification of an infant's cry can lead to promptly addressing the child's needs. Thus, the enhanced model not only achieves the central goal of the study — improving cry classification using deep learning — but also is valuable to the field by proposing a practical architectural improvement. The success of this integration shows promise of combining advanced feature transformation techniques such as RFF with deep CNN model architecture to achieve high level of accuracy and generalization in difficult tasks involving audio classification.

The results show how important it is to keep experimenting with the models by trying different setups and improvements as you build them. You can't have reliable outcomes in

healthcare applications without first knowing how accurate the model is and how well it handles different kinds of data. Thus, it is crucial to do additional study in the subject to back the development of diagnostic tools for healthcare purposes.

5.3. Future works

While the present study illustrates that MFCC features combined with CNN are successful in infant cry classification, there are various paths for future study to refine and expand these results. One way to go is to examine alternative feature extraction strategies, including spectrograms or wavelet transforms, to seize extra details contained in cry audio signals. Over time, the model's capability for pat-tern recognition can advance thanks to the addition of temporal features, along with Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, especially for longer audio sequences.

There’s a need for more expansive research involving the dataset to include a wider variety of cry types from babies of assorted ages, health conditions, and environments. More comprehensive data incorporating greater diversity would advance the development of uniform models that perform effectively in a range of circumstances.

Also, researching the immediate realization and deployment of the model on mo-bile or edge devices could yield practical benefits for both caregivers and healthcare professionals. Improving the model for fast, on-device inference would make the system more usable and available in everyday life settings. In addition, employing techniques related to transfer learning

derived from alternative audio tasks might speed up the model while utilizing few computational re-sources.

In short, part of the system's design includes integrating it into a multimodal framework, which unites audio data with either video or physiological signals, in order to boost overall accuracy and give more complete understandings of an infant’s situation. It is conceivable that it would involve robust and dependable systems for the recognition of a greater diversity of infant requirements and health challenges.

References

M. Hammoud, M. N. Getahun, A. Baldycheva, and A. Somov, “Machine learning-based infant crying interpretation,” Front. Artif. Intell., vol. 7, p. 1337356, Feb. 2024, doi: 10.3389/frai.2024.1337356.

A. Laguna et al., “Multi-modal analysis of infant cry types characterization: Acoustics, body language and brain signals,” Comput. Biol. Med., vol. 167, p. 107626, Dec. 2023, doi: 10.1016/j.compbiomed.2023.107626.

K. Rezaee, H. G. Zadeh, L. Qi, H. Rabiee, and M. R. Khosravi, “Can You Understand Why I Am Crying? A Decision-making System for Classifying Infants’ Cry Languages Based on DeepSVM Model,” ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 23, no. 1, pp. 1–17, Jan. 2024, doi: 10.1145/3579032.

R. Saharoy, A. Potdukhe, M. Wanjari, and A. B. Taksande, “Postpartum Depression and Maternal Care: Exploring the Complex Effects on Mothers and Infants,” Cureus, Jul. 2023, doi: 10.7759/cureus.41381.

P. Kumari and K. Mahto, “A Narrative Review on Different Novel Machine Learning Techniques for Detecting Pathologies in Infants From Born Baby Cries,” J. Voice, p. S0892199724000778, May 2024, doi: 10.1016/j.jvoice.2024.03.009.

A. EkiNci and E. Küçükkülahli, “Classification of Baby Cries Using Machine Learning Algorithms”.

Y. Kristian, N. Simogiarto, M. T. A. Sampurna, E. Hanindito, and V. Visuddho, “Ensemble of multimodal deep learning autoencoder for infant cry and pain detection,” F1000Research, vol. 11, p. 359, Jan. 2023, doi: 10.12688/f1000research.73108.2.

R. Radhu Krishna, G. A. D. C, C. N. Reddy, and V. V. G, “Beyond Tears: An AI Framework for Automated Interpretation of Baby Cries,” in 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India: IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/ICCCNT61001.2024.10725318.

T. Ozseven, “Infant cry classification by using different deep neural network models and hand-crafted features,” Biomed. Signal Process. Control, vol. 83, p. 104648, May 2023, doi: 10.1016/j.bspc.2023.104648.

N. F. Muhammad, R. Dewan, J. Pusppanathan, and F. A. Suryanata, “Baby Crying Sound Classification using Convolutional Neural Network,” J. Hum. Centered Technol., vol. 3, no. 1, pp. 67–74, Feb. 2024, doi: 10.11113/humentech.v3n1.66.

B. M. Mala and S. S. Darandale, “Effective infant cry signal analysis and reasoning using IARO based leaky Bi-LSTM model,” Comput. Speech Lang., vol. 86, p. 101621, Jun. 2024, doi: 10.1016/j.csl.2024.101621.

A. Abbaskhah, H. Sedighi, and H. Marvi, “Infant cry classification by MFCC feature extraction with MLP and CNN structures,” Biomed. Signal Process. Control, vol. 86, p. 105261, Sep. 2023, doi: 10.1016/j.bspc.2023.105261.

Y. Zayed, A. Hasasneh, and C. Tadj, “Infant Cry Signal Diagnostic System Using Deep Learning and Fused Features,” Diagnostics, vol. 13, no. 12, p. 2107, Jun. 2023, doi: 10.3390/diagnostics13122107.

Y.-C. Liang, I. Wijaya, M.-T. Yang, J. R. Cuevas Juarez, and H.-T. Chang, “Deep Learning for Infant Cry Recognition,” Int. J. Environ. Res. Public. Health, vol. 19, no. 10, p. 6311, May 2022, doi: 10.3390/ijerph19106311.

C. Ji, T. B. Mudiyanselage, Y. Gao, and Y. Pan, “A review of infant cry analysis and classification,” EURASIP J. Audio Speech Music Process., vol. 2021, no. 1, p. 8, Dec. 2021, doi: 10.1186/s13636-021-00197-5.

M. Hammoud, M. N. Getahun, A. Baldycheva, and A. Somov, “Machine learning-based infant crying interpretation,” Front. Artif. Intell., vol. 7, p. 1337356, Feb. 2024, doi: 10.3389/frai.2024.1337356.

D. Widhyanti and D. Juniati, “Classification of Baby Cry Sound Using Higuchi’s Fractal Dimension with K-Nearest Neighbor and Support Vector Machine,” J. Phys. Conf. Ser., vol. 1747, no. 1, p. 012014, Feb. 2021, doi: 10.1088/1742-6596/1747/1/012014.

A. W. Manigault et al., “Acoustic Cry Characteristics in Preterm Infants and Developmental and Behavioral Outcomes at 2 Years of Age,” JAMA Netw. Open, vol. 6, no. 2, p. e2254151, Feb. 2023, doi: 10.1001/jamanetworkopen.2022.54151.

P. Kumari and K. Mahto, “A Narrative Review on Different Novel Machine Learning Techniques for Detecting Pathologies in Infants From Born Baby Cries,” J. Voice, p. S0892199724000778, May 2024, doi: 10.1016/j.jvoice.2024.03.009.

Y. Zayed, A. Hasasneh, and C. Tadj, “Infant Cry Signal Diagnostic System Using Deep Learning and Fused Features,” Diagnostics, vol. 13, no. 12, p. 2107, Jun. 2023, doi: 10.3390/diagnostics13122107.

K. Sujatha et al., “Internet of medical things for abnormality detection in infants using mobile phone app with cry signal analysis,” in Implementation of Smart Healthcare Systems using AI, IoT, and Blockchain, Elsevier, 2023, pp. 169–191. doi: 10.1016/B978-0-323-91916-6.00012-6.

J. Basavaiah and A. A. Anthony, “A Pragmatic Approach for Infant Cry Analysis Using Support Vector Machine and Random Forest Classifiers,” Wirel. Pers. Commun., vol. 137, no. 4, pp. 2269–2280, Aug. 2024, doi: 10.1007/s11277-024-11491-8.

M. Hammoud, M. N. Getahun, A. Baldycheva, and A. Somov, “Machine learning-based infant crying interpretation,” Front. Artif. Intell., vol. 7, p. 1337356, Feb. 2024, doi: 10.3389/frai.2024.1337356.

K. Świtała, A. Bojarczuk, J. Hajto, M. Piechota, M. Buryta, and A. Leońska-Duniec, “Impact of the DRD2 Polymorphisms on the Effectiveness of the Training Program,” Int. J. Environ. Res. Public. Health, vol. 19, no. 9, p. 4942, Apr. 2022, doi: 10.3390/ijerph19094942.

T. Nadia Maghfira, T. Basaruddin, and A. Krisnadhi, “Infant cry classification using CNN – RNN,” J. Phys. Conf. Ser., vol. 1528, no. 1, p. 012019, Apr. 2020, doi: 10.1088/1742-6596/1528/1/012019.

K. Rezaee, H. G. Zadeh, L. Qi, H. Rabiee, and M. R. Khosravi, “Can You Understand Why I Am Crying? A Decision-making System for Classifying Infants’ Cry Languages Based on DeepSVM Model,” ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 23, no. 1, Art. no. 1, Jan. 2024, doi: 10.1145/3579032.

P. A. Riadi, M. R. Faisal, D. Kartini, R. A. Nugroho, D. T. Nugrahadi, and D. B. Magfira, “A Comparative Study of Machine Learning Methods for Baby Cry Detection Using MFCC Features,” J. Electron. Electromed. Eng. Med. Inform., vol. 6, no. 1, Art. no. 1, Jan. 2024, doi: 10.35882/jeeemi.v6i1.350.

T. Nadia Maghfira, T. Basaruddin, and A. Krisnadhi, “Infant cry classification using CNN – RNN,” J. Phys. Conf. Ser., vol. 1528, no. 1, Art. no. 1, Apr. 2020, doi: 10.1088/1742-6596/1528/1/012019.

S. Lahmiri, C. Tadj, C. Gargour, and S. Bekiros, “Deep learning systems for automatic diagnosis of infant cry signals,” Chaos Solitons Fractals, vol. 154, p. 111700, Jan. 2022, doi: 10.1016/j.chaos.2021.111700.

K. De Barbaro, M. Micheletti, X. Yao, P. Khante, M. Johnson, and S. Goodman, “Infant crying predicts real-time fluctuations in maternal mental health in ecologically valid home settings.,” Dev. Psychol., vol. 59, no. 4, pp. 733–744, Apr. 2023, doi: 10.1037/dev0001530.

L. Alzubaidi et al., “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” J. Big Data, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique, and J. Henkel, “A Comprehensive Survey of Convolutions in Deep Learning: Applications, Challenges, and Future Trends,” Feb. 28, 2024, arXiv: arXiv:2402.15490. doi: 10.48550/arXiv.2402.15490.

S. Jeyaraman et al., “A review: survey on automatic infant cry analysis and classification,” Health Technol., vol. 8, no. 5, pp. 391–404, Nov. 2018, doi: 10.1007/s12553-018-0243-5.

B. Akash Kumar, V. Vasugi, and S. Elavenil, “Strength predictions of GGBS based cement mortar with different M-Sands using Neural Networks,” J. Phys. Conf. Ser., vol. 1716, no. 1, p. 012015, Dec. 2020, doi: 10.1088/1742-6596/1716/1/012015.

F. Li, C. Cui, and Y. Hu, “Classification of Infant Crying Sounds Using SE-ResNet-Transformer,” Sensors, vol. 24, no. 20, p. 6575, Oct. 2024, doi: 10.3390/s24206575.

L. R. Rabiner and R. W. Schafer, Theory and applications of digital speech processing, 1. ed. Upper Saddle River, NJ: Pearson Higher Education, 2011.

M. S. Sidhu, N. A. A. Latib, and K. K. Sidhu, “MFCC in audio signal processing for voice disorder: a review,” Multimed. Tools Appl., vol. 84, no. 10, pp. 8015–8035, Apr. 2024, doi: 10.1007/s11042-024-19253-1.

M. Hoq, M. N. Uddin, and S.-B. Park, “Vocal Feature Extraction-Based Artificial Intelligent Model for Parkinson’s Disease Detection,” Diagnostics, vol. 11, no. 6, p. 1076, Jun. 2021, doi: 10.3390/diagnostics11061076.

H. Qassim, A. Verma, and D. Feinzimer, “Compressed residual-VGG16 CNN model for big data places image recognition,” in 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV: IEEE, Jan. 2018, pp. 169–175. doi: 10.1109/CCWC.2018.8301729.

S. Mascarenhas and M. Agarwal, “A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification,” in 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru, India: IEEE, Nov. 2021, pp. 96–99. doi: 10.1109/CENTCON52345.2021.9687944.

H. M. Rai and K. Chatterjee, “Detection of brain abnormality by a novel Lu-Net deep neural CNN model from MR images,” Mach. Learn. Appl., vol. 2, p. 100004, Dec. 2020, doi: 10.1016/j.mlwa.2020.100004.

S. Cong and Y. Zhou, “A review of convolutional neural network architectures and their optimizations,” Artif. Intell. Rev., vol. 56, no. 3, pp. 1905–1969, Mar. 2023, doi: 10.1007/s10462-022-10213-5.

S. S. Sanagala et al., “Ten Fast Transfer Learning Models for Carotid Ultrasound Plaque Tissue Characterization in Augmentation Framework Embedded with Heatmaps for Stroke Risk Stratification,” Diagnostics, vol. 11, no. 11, p. 2109, Nov. 2021, doi: 10.3390/diagnostics11112109.

S. A. Hasanah, A. A. Pravitasari, A. S. Abdullah, I. N. Yulita, and M. H. Asnawi, “A Deep Learning Review of ResNet Architecture for Lung Disease Identification in CXR Image,” Appl. Sci., vol. 13, no. 24, p. 13111, Dec. 2023, doi: 10.3390/app132413111.

G. E. Manucharyan, L. Siegelman, and P. Klein, “A Deep Learning Approach to Spatiotemporal Sea Surface Height Interpolation and Estimation of Deep Currents in Geostrophic Ocean Turbulence,” J. Adv. Model. Earth Syst., vol. 13, no. 1, p. e2019MS001965, Jan. 2021, doi: 10.1029/2019MS001965.

A. S. Modi, “Review Article on Deep Learning Approaches,” in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India: IEEE, Jun. 2018, pp. 1635–1639. doi: 10.1109/ICCONS.2018.8663057.

T. Ozseven, “Infant cry classification by using different deep neural network models and hand-crafted features,” Biomed. Signal Process. Control, vol. 83, p. 104648, May 2023, doi: 10.1016/j.bspc.2023.104648.

C. Fournier et al., “Position-controlled quantum emitters with reproducible emission wavelength in hBN,” in 2021 Conference on Lasers and Electro-Optics Europe & European Quantum

Electronics Conference (CLEO/Europe-EQEC), Munich, Germany: IEEE, Jun. 2021, pp. 1–1. doi: 10.1109/CLEO/Europe-EQEC52157.2021.9542642.

A. E. Maxwell, T. A. Warner, and L. A. Guillén, "Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review," Remote Sens., vol. 13, no. 13, p. 2450, Jun. 2021, doi: 10.3390/rs13132450.

[50] B. J. Erickson and F. Kitamura, "Magician's Corner: 9. Performance Metrics for Machine Learning Models," Radiol. Artif. Intell., vol. 3, no. 3, p. e200126, May 2021, doi: 10.1148/ryai.2021200126.

"librosa.feature.mfcc[1]."

https://www.kaggle.com/datasets/bhoomikavalani/donateacrycorpusfeaturesdataset

https://github.com/gveres/donateacry-corpus

Appendices

Appendix **A:**

Arab American University
Faculty of Graduate Studies
الجامعة العربية الأمريكية
كلية الدراسات العليا

2024/5/27

إلى من يهمه الأمر

تسهيل مهمة بحثية

تحية طيبة وبعد،

تُهديكم كلية الدراسات العُليا في الجامعة العربية الأمريكية أطيب التحيات، وبالإشارة الى الموضوع أعلاه، تشهد كلية الدراسات العُليا في الجامعة أن الطالبة **أسماء جودت محمد شويكة** والتي تحمل الرقم الجامعي 201720327هي طالبة ماجستير في برنامج علم الحاسوب وتعمل على رسالة الماجستير الخاصة بها بعنوان:

"Utilizing Deep Learning Models to Identify and Classify Various Types of Infant Cries"

تحت إشراف الدكتور معاذ صبحة، نأمل من حضرتكم الإيعاز لمن يلزم لمساعدتها للحصول على المعلومات اللازمة للدراسة، علماً أن المعلومات ستستخدم لغاية البحث فقط وسيتم التعامل معها بغاية السرية، وقد أُعطيت هذه الرسالة بناءً على طلبها.

وتفضلوا بقبول فائق الاحترام

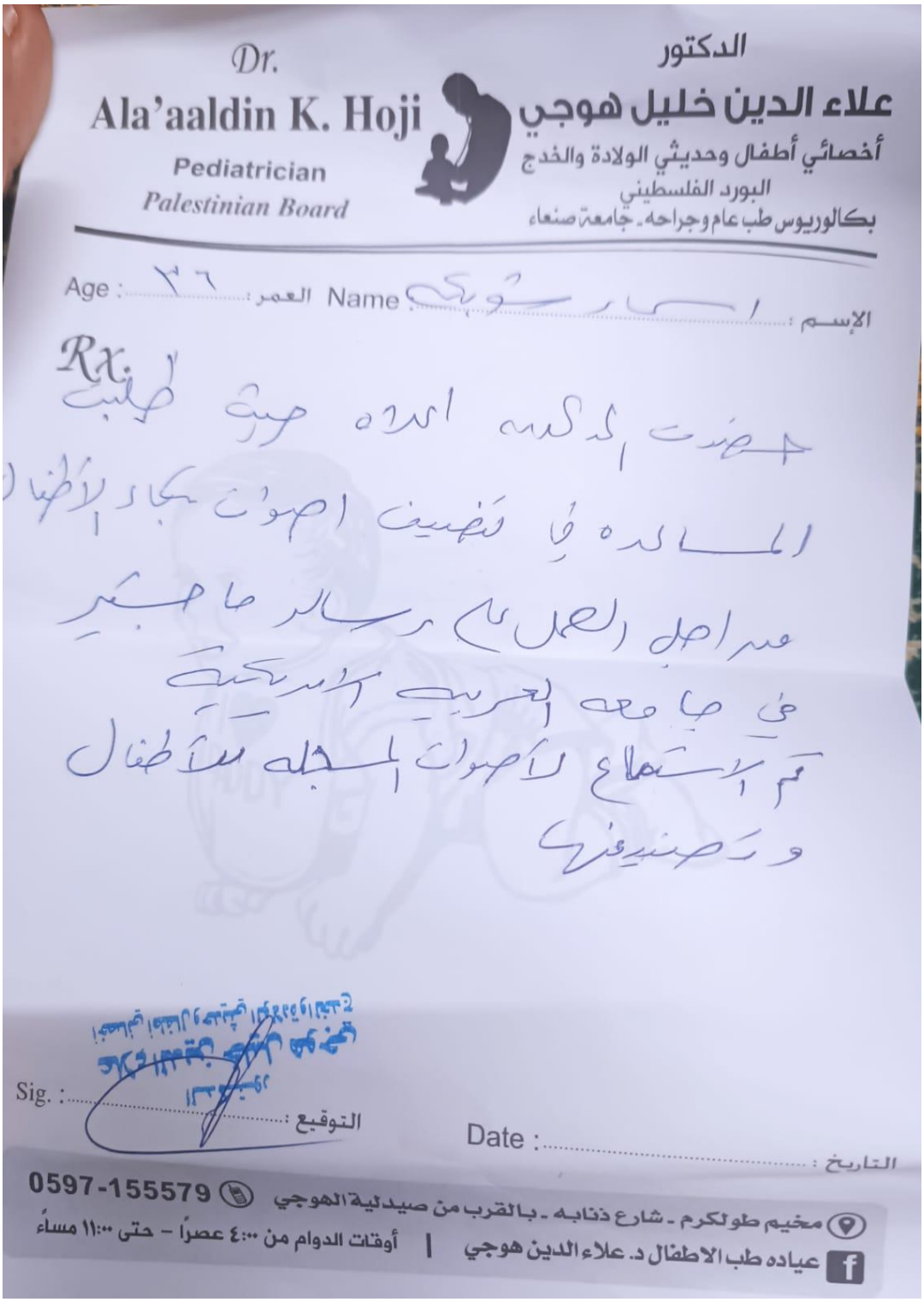
عميد كلية الدراسات العليا

د. نوار قطب

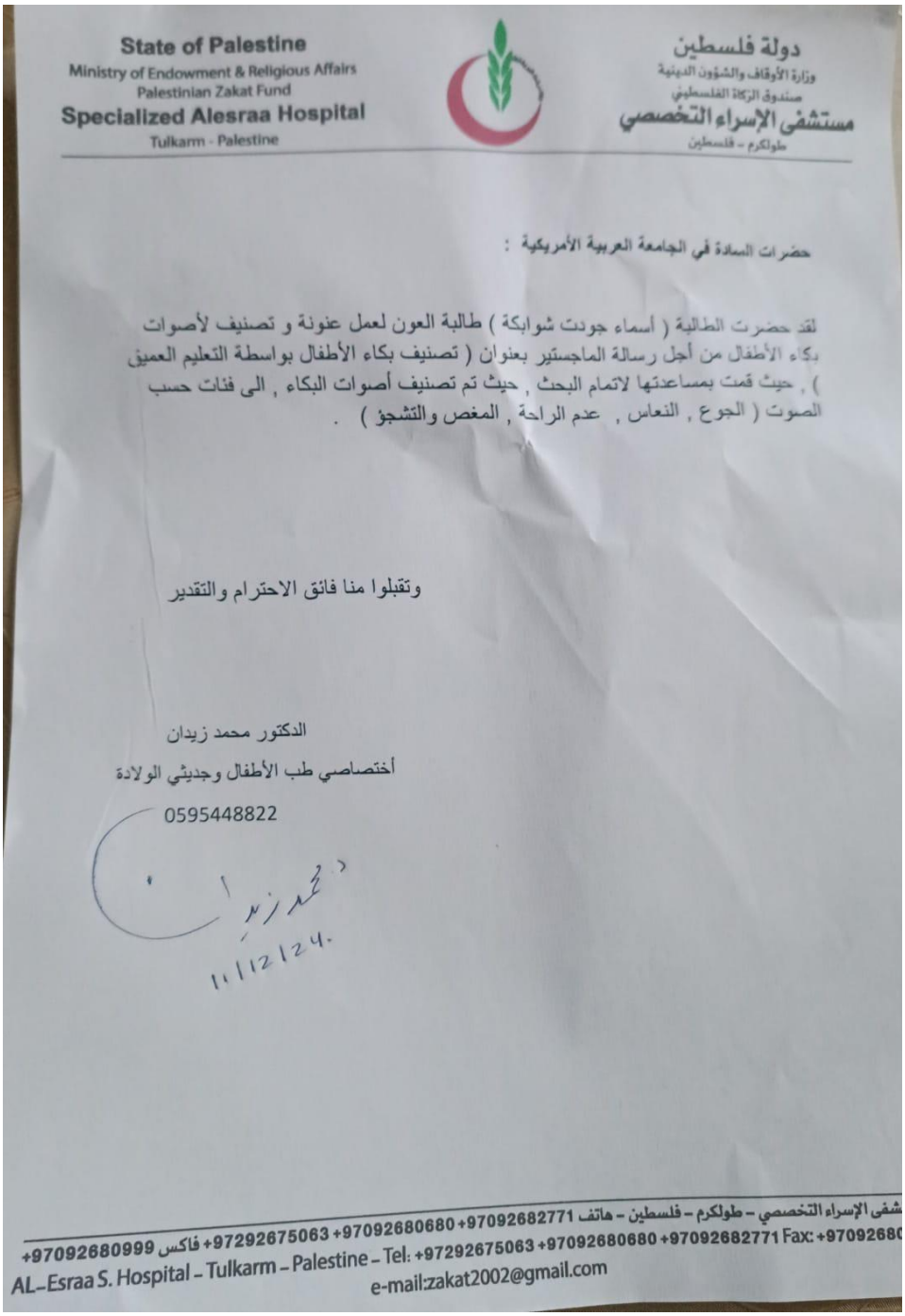
كلية الدراسات العليا
FACULTY OF GRADUATE STUDIES

Jenin Tel: +970-4-2418888 Ext.:1471,1472 Fax: +970-4-2510810 P.O. Box:240
Ramallah Tel: +970-2-2941999 Fax: +970-2-2941979 Abu Qash - Near Alrehan
E-mail: FGS@aaup.edu ; PGS@aaup.edu Website: www.aaup.edu

Appendix B:


Dr.
Ala'aaldin K. Hoji
Pediatrician
Palestinian Board
الدكتور
علاء الدين خليل هوجي
أخصائي أطفال وحديثي الولادة والخدج
البورد الفلسطيني
بكالوريوس طب عام وجراحه - جامعة صنعاء
Age : ٢٦ العمر
Name
الإسم
Rx
Sig. :
التوقيع
Date :
التاريخ
0597-155579
مخيم طولكرم - شارع ذنابه - بالقرب من صيدلية الهوجي
عياده طب الاطفال د. علاء الدين هوجي | أوقات الدوام من ٤:٠٠ عصراً - حتى ١١:٠٠ مساءً

Appendix C:



State of Palestine
Ministry of Endowment & Religious Affairs
Palestinian Zakat Fund
Specialized Alesraa Hospital
Tulkarm - Palestine

دولة فلسطين
وزارة الأوقاف والشؤون الدينية
صندوق الزكاة الفلسطيني
مستشفى الإسراء التخصصي
طولكرم – فلسطين

حضرات السادة في الجامعة العربية الأمريكية :

لقد حضرت الطالبة (أسماء جودت شوابكة) طالبة العون لعمل عنونة و تصنيف لأصوات بكاء الأطفال من أجل رسالة الماجستير بعنوان (تصنيف بكاء الأطفال بواسطة التعليم العميق) , حيث قمت بمساعدتها لاتمام البحث , حيث تم تصنيف أصوات البكاء , الى فئات حسب الصوت (الجوع , النعاس , عدم الراحة , المغص والتشجؤ) .

وتقبلوا منا فائق الاحترام والتقدير

الدكتور محمد زيدان
أختصاصي طب الأطفال وجديثي الولادة
0595448822

10/12/24

شفى الإسراء التخصصي – طولكرم – فلسطين – هاتف 97092682771+ 97092680680+ 97292675063+ فاكس 97092680999+
AL-Esraa S. Hospital – Tulkarm – Palestine – Tel: +97292675063 +97092680680 +97092682771 Fax: +9709268
e-mail:zakat2002@gmail.com

استخدام نماذج التعلم العميق لتحديد وتصنيف أنواع مختلفة من بكاء الرضع.

أسماء جودت محمد شوابكة

د. معاذ صبحة

د. أحمد حساسنة

د. أنس سمارة

ملخص

تشير الأبحاث الحالية إلى أهمية تطوير نماذج فعالة لمساعدة المتخصصين في تصنيف الحالات السلوكية بدقة وكفاءة، ومن المجالات البحثية البارزة في هذا السياق تصنيف بكاء الرضع. يُعد بكاء الرضيع مؤشرًا حيويًا لصحة الطفل، حيث يعمل كوسيلة اتصال غير لفظية للتعبير عن احتياجاته. ومع ذلك، يمكن أن يكون فهم سبب بكاء الرضيع مهمة معقدة، خصوصًا للأهل ومقدمي الرعاية. ولهذا، أصبحت هذه المشكلة محورًا للبحث العلمي.

ازدادت استخدامات التعلم العميق في مجال الرعاية الصحية، بما في ذلك في العيادات والمستشفيات. تتيح منهجيات التعلم العميق تحسين القدرة على التشخيص المبكر، مما يعزز من إمكانية تحديد أسباب بكاء الرضيع دون الحاجة إلى تعريضه لمعاناة إضافية. تقدم هذه الدراسة رؤى مهمة حول عوامل ميزات البكاء التي ترتبط بشكل ضعيف بوظائف وحالات البكاء، وهو ما يمكن أن يفيد في تطوير أدوات تشخيصية دقيقة لتصنيف بكاء الرضع. ويُجرى تقييم قدرة تقنيات التعلم العميق المستخدمة في هذا العمل على تحسين دقة التشخيص ونتائج الرعاية الصحية للأطفال.

تم استخدام تقنيات تحويل الصوت إلى صور باستخدام معاملات الترددات الميلية (MFCC)، تلتها تصنيف الصور الناتجة باستخدام تقنيات تعلم عميق مختلفة مثل VGG16، DenseNet، ResNet، وGoogleNet. واعتمدت الدراسة على تقنيات قياسية للمصفوفات لتقييم أداء هذه النماذج ومقارنته.

تم تطبيق تقنيات التعلم العميق في تصنيف بكاء الرضع باستخدام بيانات مجمعة من مراكز رعاية محلية ومنصات بيانات عالمية مثل Kaggle. وتم إجراء تحليل شامل لأداء الخوارزميات المختلفة لتحديد أكثر النماذج ملاءمة لتصنيف بكاء الرضع بدقة في بيئات بيانات متنوعة.

تشير النتائج إلى تفوق خوارزميتي ResNet وDenseNet على غيرهما، حيث حققت ResNet أعلى دقة على مجموعة البيانات العالمية، تليها DenseNet. بينما كانت فاعلية VGG-16 الأقل، مما يعكس ضرورة تقييم كل خوارزمية وفقًا لمدى ملاءمتها للبيانات المستخدمة. أما على مستوى البيانات المحلية، فقد أثبتت ResNet دقة عالية، مما يعزز من موثوقيتها وقوتها على هذا النوع من البيانات.

إضافة إلى ذلك، تسلط الدراسة الضوء على أهمية التحسينات المعمارية في تعزيز أداء النماذج. حيث أدى دمج طبقة عشوائية للميزات (Random Forest Features) مع طبقات كثيفة (Dense Layers) إلى تحسينات كبيرة في استقرار ودقة النموذج. تشير هذه التعديلات المعمارية إلى أن النهج الهجينة يمكن أن تحسن بشكل كبير من قدرة النموذج على تعلم الأنماط المعقدة، وهو أمر ضروري لتطبيقات التصوير الطبي والتشخيص.

بناءً على ما سبق، يمكننا تقديم مساعدة فعالة لمقدمي الخدمات الصحية في تحديد السبب الفعلي لبكاء الرضع بسرعة ودقة باستخدام أحدث تقنيات التعلم العميق. كما يمكن إنتاج نموذج وتوزيعه على هذه المراكز مستقبلاً للاستفادة منه.

الكلمات المفتاحية: تصنيف، بكاء الرضع، التعلم العميق، خوارزميات، الذكاء الاصطناعي.