Article

A study of deep learning models for audio classification of infant crying in a baby monitoring system

Denisa Maria Herlea 1, Bogdan Iancu 2 and Eugen-Richard Ardelean 3,\*

|  |
| --- |
| Academic Editor: Firstname Lastname  Received: date  Revised: date  Accepted: date  Published: date  **Citation:** To be added by editorial staff during production.  **Copyright:** © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/). |

1 Computer Science Department, Technical University of Cluj-Napoca; denisa.herlea@student.utcluj.ro

2 Computer Science Department, Technical University of Cluj-Napoca; bogdan.iancu@cs.utcluj.ro

3 Computer Science Department, Technical University of Cluj-Napoca; ardeleaneugenrichard@gmail.com

**\*** Correspondence: ardeleaneugenrichard@gmail.com;

**Abstract:** This study investigates the ability of audio classification of well-known deep learning models, such as ResNet and EfficientNet. By comparing the performance of different machine learning algorithms, this study seeks to determine the most effective approach for the detection of infant crying, enhancing the functionality of baby monitoring systems and contributing to a more advanced understanding of audio-based deep learning applications. Understanding and accurately detecting a baby's cries is crucial for ensuring their safety and well-being, a concern shared by new and expecting parents worldwide. Despite advancements in child health, as noted by UNICEF's 2022 report of the lowest ever recorded child mortality rate, there is still room for technological improvement. This paper also presents a baby monitoring system that utilizes Deep Learning and Internet-of-Things technologies to enhance parental peace of mind and infant safety. The system employs a Raspberry Pi board equipped with a camera, microphone, and speaker to monitor and communicate a baby’s status. Through this research, various machine learning algorithms are evaluated to identify the most effective approach for detecting infant cries and thus audio classification as well.

**Keywords:** deep learning; convolutional neural network; classification; infant crying; resnet; efficientnet;

1. Introduction

Ensuring the wellbeing of babies is the top priority for any new or expecting parent, which often leads to physical or emotional exhaustion. Monitoring children's safety and health both in the first months of life and beyond is not only a crucial necessity, but also a moral obligation that lays the foundations of the broad term 'parenting'. At the beginning of their lives, babies are sensitive and vulnerable to various external factors that can jeopardize their life and health, which is why constant monitoring of their developmental parameters. According to UNICEF, the year 2022 saw the fewest number of deaths among children under five, but nevertheless, the reported 4.9 million number can and should be improved every year.

Against this backdrop, the use of technology has increased significantly in recent years and has played a key role in streamlining the aforementioned tasks. It is not news that mobile applications are part of everyone's life nowadays. Also, the use of IoT (Internet of Things) devices has gained momentum in various fields, with various monitoring devices such as smart cameras, remote communication devices, devices for monitoring breathing or heart rate and many such examples now on the market.

Thus, through the integration of an artificial intelligence model, trained to recognize baby crying sounds, the stress of parents could be lessened such that even when not in the vicinity of their baby, they would know whether their baby is well or not. Such a model, integrated with hardware components would be capable of alerting parents in very close to real time to any discomfort their child is experiencing, thus being able to reach their child and solve problems as quickly as possible.

2. Materials and Methods

Data preprocessing

Data preprocessing methods make up one of the most important steps in training an artificial intelligence model because the quality of the preprocessing determines the quality of the predictions made by it.

There are many methods by which data can be modified to match the architecture of a model, among which the most common ones are the subject of this paper:

* Cleaning: by replacing missing or null values with other suggestive values close to the real data, such as mean, median or other derivatives, or removing duplicates and irrelevant data for the training process.
* Normalize the data by fitting them into a more precise and easier to analyze range.
* Encode data by transforming labeled categories into numbers, vectors or other data structures.
* Data augmentation, by modifying the dataset to force the model to generalize better on unknown data and to broaden the range of cases recognized by the model.
* Data resizing, by adding new data to match a desired length or deleting existing data to bring all the data in the dataset to the same size.

Data representations

In the context of sound classification, we are interested in the appropriate ways of data representation for classifying and recognizing infant crying. This task becomes more complicated as various other sounds occur in the background or the environment is acoustically changing [1]. In Figure 1, we can see the different options for representing five seconds of infant crying, each of which is important and different, which is why choosing the right option is challenging.

Among the possibilities mentioned graphically, we list the spectrogram, which is a visual representation of a frequency in relation to another unit such as time. It is obtained by applying the Fourier transform on the segment containing the audio data.

Due to the result as a linear representation, overlapping noises of similar frequencies may not be correctly determined. If this is possible with the input data, the use of Mel-filter log representation is more beneficial and correct, as this method is able to distinguish more clearly between similar frequencies coming from different sources.

In practice it is more common to use the discrete Fourier transform [2], a reason resulting from the fact that the time we are working with is discrete. The calculation of the discrete Fourier transform is performed according to the formula:

The spectrogram and Short-Time Fourier Transform (STFT) are two very closely related concepts, the difference in their applicability being in the way the frequency of the data changes (stationary or non-stationary). Spectrogram is more suitable for stationary data, it displays only the magnitude of the power spectrum , whereas STFT displays both magnitude and phase. The calculation formula for the Short-Term Fourier Transform is as follows:

Waveforms are another way of representing audio data for the purpose of preprocessing and extracting relevant features for machine learning models. These, however, are not usually used in this form for training models due to their complexity. This graphical representation depends on the signal amplitude, as described in [3], which can change if the pressure around the microphone changes.

The waveform is generally used for extracting other key features needed by the machine learning domain. For example, one can infer from it the spectrogram mentioned above, but also other data such as Mel Frequency cepstral Coefficients (MFCC). The differences between the two, displayed graphically in Figure 1 – A sample of a crying sample with its signal, its MFCC and its Fourier spectrogram., are in the time-frequency representation, as the spectrogram contains details of the entire audio spectrum, whereas the MFCC uses a logarithmic scale based on human perception of sound frequencies. Therefore, their dimensions differ, with MFCC being more reduced by synthesizing spectral information into cepstral coefficients.

A screenshot of a computer screen

Description automatically generated

*Figure 1 – A sample of a crying sample with its signal, its MFCC and its Fourier spectrogram.*

Data augmentation

Dataset augmentations play a role in the robustness and generalization abilities of deep learning models, espectially when a high number of samples is lacking. For audio data, augmentations such as time stretching, pitch shifting, addition of noise, and changes in volume introduce variations that can occur when testing such a model in real environments. This aids in not overfitting to the training data and therefore assures the model's performance over unseen data segments. Augmentation can also help to balance datasets, especially when classes are imbalanced, by artificially increasing the diversity and quantity of training samples. These techniques thereby help improve pattern recognition and feature extraction within a model that has invariance to some distortions, which has shown better performance in tasks such as speech recognition, music genre classification, and environmental sound detection.

* Pitch shifting seeks to alter the pitch of the audio without changing its duration. This is used in applications such as music genre classification or speech recognition, in which many pitches would help refer to various speakers or musical instruments. By training on pitch-shifted audio, a model has a greater likelihood of becoming invariant to these variations.
* Adding noise involves overlaying background noise onto the original audio, usually done through the addition between the original signal and a noise signal that is multiplied with a noise factor. This simulates real-world environments where background noise can be present and helps in making models more robust. Different types of noise can be used, such as white noise, pink noise, brown noise, or realistic environmental sounds like traffic or crowd noise.
* Time shifting moves the audio signal forward or backward in time by a certain amount. This augmentation can help a model become invariant to slight timing differences, which can be especially useful in tasks like speech recognition or event detection where exact timing can vary. This is especially useful for longer audio signals that only have certain segments with sounds, such that the model does not correlate the position in time with the classification; thus, making it more robust.
* Volume change, also known as gain adjustment, involves altering the amplitude of the audio signal to make it louder or quieter. This is achieved by multiplying the audio waveform by a constant factor greater than 1 for amplification or between 0 and 1 for attenuation. Volume change is a simple yet effective augmentation that helps in simulating real-world scenarios where the audio might be recorded at different levels of loudness due to varying distances from the microphone or different recording equipment.

Machine learning models

Support Vector Machines (SVM) are a simple yet widely used classification technique that has shown high accuracy in various pattern recognition tasks. The SVM operates by finding the optimal hyperplane that separates data points of different classes in a high-dimensional space. This hyperplane is determined such that the margin, which is the distance between the hyperplane and the nearest data points of each class, is maximized. By leveraging the kernel trick, SVMs can efficiently perform non-linear classification by implicitly mapping input features into higher-dimensional spaces. This ability to handle complex boundaries and avoid overfitting makes SVMs particularly suitable for tasks involving intricate data structures, such as audio classification.

Random Forest (RF) is an ensemble learning method that uses multiple decision trees during training and outputs the majority vote of the classes for classification. Random Forest reduce the overfitting that individual trees are prone to, through the use of this array of decision trees and the majority vote to decide. Each tree in the forest is trained on a random subset of the data and features, which enhances the model's generalizability and resilience to noise.

Deep learning models

Neural networks are a subset of machine learning that have seen extensive application and development in various domains, including speech and sound recognition, which are pertinent to the baby monitoring system proposed in this study. This section identifies relevant contributions toward progress in neural networks, especially in domains such as audio recognition and health monitoring systems. Neural networks have revolutionized the field of audio recognition, making significant developments in tasks such as speech and sound classification.

DeepSVM [4] is an ensemble learning approach based on SVM to create a decision-making system for the classification of infant crying using windowing and the Fourier transform for feature extraction and selection. In this paper, they managed to classify the reason for which the infant was crying with high accuracy.

In [5], convolutional and recurrent neural networks were integrated for the classification of infant crying using various features extraction methods such as mel frequency, bark frequency and linear prediction cepstral coefficients. They have shown that their model is capable of achieving higher performance than previous models.

In another study [6], a pretrained convolutional neural network, specifically VGG16, was used in conjunction with Gammatone frequency cepstral coefficients and spectrograms in order to diagnose infant crying. They have showed that the best performance was obtained through the fusion of various audio features and the applicability of transfer learning.

Performance evaluation

Accuracy is the most common metric used in classification. However, it can be misleading for imbalanced datasets. Nevertheless, this effect can be mitigated through augmentation for the balancing of class samples. In simple words, accuracy represents the probability with which the classifier predicts correctly and it can be expressed mathematically as:

F1-score is used as an alternative to accuracy, which better reflects the classifier quality. It is often used when optimising recall or precision results in decreased model performance. Mathematically, F1 is the harmonic mean of precision and recall, but it can also be defined using the confusion matrix.

Proposed Method

Overall system architecture

In this paper, we propose a baby monitoring system that uses deep learning for the detection of infant crying but for it to work properly the integration of hardware and software is necessary. The core elements of its architecture and how they interact are shown in Figure 2. These include:

* The user, who is the main actor using the system.
* The Raspberry Pi board, the main component and brain of the entire hardware device.
* Camera, for recording video frames and sending them to the mobile application.
* The microphone and speaker component that plays a dual role in the hardware: sending the data for further processing to the baby crying detection model and communicating with the mobile application both by recording the sound and playing it back through the hardware system speaker, and in the reverse direction, by recording it through the microphone and playing it back to the application.
* Baby cry detection model, which sends notifications to the app via the Flask server.
* The application itself, which communicates with the server via the HTTP communication protocol for audio and video transmission, and via a websocket connection for receiving notifications sent by the server to it.
* SQLite database, for storing user data and using it for further analysis.

A diagram of a computer network

Description automatically generated

*Figure 2 – The general system architecture*

The main advantage of such a system is its high potential for extensibility. Existing functionalities can be enhanced to include additional sensors, integration with other smart devices in the home, or the development of advanced analytics and predictive algorithms. The data collected and analyzed can provide useful information about babies' sleep, feeding and growth patterns, thus helping to identify potential health problems early and improve care strategies.

The central component for the functionality of the baby monitoring system is the deep learning-based baby cry detection model. The accurate detection of infant crying is critical for the system's efficacy. The deep learning model processes audio data in real-time, distinguishing between crying and other sounds. Below, we describe the components and processes involved in the machine learning detection of infant crying.

Data manipulation

To obtain a robust model for such a system, a copious amount of data is required and as such, data was extracted from several sources:

* CryCeleb, segmented cry sounds from 786 infants [7]
* Donateacry, a corpus of various instances of infant crying that have been classified by the reason for crying [8]
* LibriSpeech, a dataset of read english speech from audiobooks [9]
* Infant’s Cry Sound, a dataset containing 3 classes of crying [10]
* SilentBabyMonitor, containing various classes of sounds, such as noise, silence and crying [11]

These datasets have been converted to the “.wav” format from their original formats and have either been segmented or padded to a 5 second sample of audio.

Such a system should be capable of detecting crying with a high accuracy. But more than that, it should be able to precisely detect the crying with as few false positives as possible and at the same it should not miss the actual crying of the infant. Thus, such a model requires both high precision and high recall. To increase the robustness of any model trained using this curated dataset, it contains samples of noise, silence, animal sounds, ambiental sounds and adult speech.

Furthermore, these samples have then been augmented to increase the ability of the model to generalize. The data augmentations applied were: pitch shifting, noise addition (white, pink, brown), time shifting and volume change. Through this augmentation of the data, balanced classes can be obtained. The augmented dataset was set to a size of 10000 samples where a random audio sample was chosen from the crying or non-crying samples to obtain a balanced dataset. Each augmentation had a probability of 20% to be applied to the sample. Finally, the dataset was partitioned into training, validation and testing of 70%, 15% and 15% percentages, respectively. The spectrograms and MFCCs of each sample were extracted to be used as the features for training.

Machine learning algorithms

We utilized SVM with a linear kernel for the classification of flattened spectrograms from audio signals. The choice of a linear kernel is motivated by its simplicity and effectiveness in high-dimensional feature spaces. We set the regularization parameter *C* to 1.0 to balance the trade-off between a low training error and a high margin. Features were standardized before training to ensure equal contributions. The dataset was divided into training and testing sets with an 80-20.

We utilized the RF classifier to perform the classification of flattened spectrograms from audio signals. The model is configured to use 100 decision trees. The dataset was divided into training and testing sets with an 80-20 and they were standardized before training.

Deep learning models

Several architectures of neural networks were developed for the evaluation of their ability to classify infant crying. All of these models were trained using the Adam optimizer, with sparse crossentropy loss and they had ReLU activation functions for the hidden layers and softmax for the output layer. The architectures are as follows:

* *CNN1*: Convolutional neural network (1x convolution + 1x max pooling) with 3 dense layers
* *CNN3*: Convolutional neural network (3x convolution + 3x max pooling) with 5 dense layers
* *VGG16* [12]: 16 layers, including 13 convolutional layers and 3 fully connected layers. It is widely used due to its balance of depth and computational efficiency.
* *ResNet50* [13]: introduces residual blocks, which help train deeper networks by avoiding the vanishing gradient problem. This makes it a good choice for large datasets with complex patterns.
* *InceptionV3* [14]: uses parallel convolutions with different kernel sizes, allowing the network to capture a variety of features at different scales.
* *Inception-ResNetV2* [15]: combines the inception architecture with residual connections for enhanced learning capabilities.
* *Xception* (Extreme Inception) [16]: deep learning model that improves upon Inception by replacing the standard Inception modules with depthwise separable convolutions.
* *MobileNet* [17]: lightweight and efficient, making them ideal for mobile and edge devices. They use depthwise separable convolutions to reduce computational cost.
* *DenseNet* [18]*:* connects each layer to every other layer in a feed-forward fashion. This enhances the flow of information and gradients through the network, making it highly efficient.
* *EfficientNet* [19]*:* scales up the network width, depth, and resolution in a compound manner, providing a more balanced approach to scaling compared to other models.

The last 8 architectures are used as encoders and they continue with a flattening and a series of 5 dense layers to obtain the classification output. These have been pretrained on the ImageNet dataset. The models are presented in the Results section in two variants, frozen and non-frozen. This choice is applied to the architectures, not the last layers of the model.

Although not presented here, simple neural networks containing only dense layers have been evaluated as well, but they were unable to properly learn as their accuracy remained close to that of random. The same conclusions was reached for using only the signal.

3. Results

For the analysis of this array of neural network architectures, we present the loss (Table 1) and accuracy values (Table 2) for the training, validation and testing of these models. Moreover, the F1-score (Table 4) is presented for the testing data. Furthermore, we have trained each neural network model using the spectrogram or the MFCCs in order to determine which of the two types of features are more appropriate for training of such a model in the detection of infant cries. This same procedure was applied to the machine learning models (SVM and RF), however no loss values are present for these models in Table I and the validation subset of the dataset was moved to the testing subset.

Another analysis on the pretrained models was the option of freezing the encoding layers of the pretrained architectures used. The effect of this choice on accuracy can be observed through the comparison of Table 2 and Table 3. Furthermore, in Table 4, the effect on the F1-score for the testing data is presented.

**Table 1.** Loss values across architectures.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Model** | **Train** | **Validation** | **Test** |
| Spectrogram | SVM | - | - | - |
| Spectrogram | RF | - | - | - |
| Spectrogram | CNN1 | 0.926 | 0.928 | 0.958 |
| Spectrogram | CNN3 | **0.222** | **0.153** | **0.230** |
| Spectrogram | VGG16 | 0.467 | 0.401 | 0.420 |
| Spectrogram | ResNet50 | 0.476 | 0.358 | 0.352 |
| Spectrogram | InceptionV3 | 0.569 | 0.469 | 0.467 |
| Spectrogram | Inception-ResNetV2 | 0.562 | 0.441 | 0.441 |
| Spectrogram | Xception | 0.530 | 0.409 | 0.410 |
| Spectrogram | DenseNet | 0.492 | 0.402 | 0.399 |
| Spectrogram | MobileNet | 0.588 | 0.518 | 0.522 |
| Spectrogram | EfficientNet | 0.554 | 0.467 | 0.470 |
| MFCC | SVM | - | - | - |
| MFCC | RF | - | - | - |
| MFCC | CNN1 | 0.541 | 0.512 | 0.507 |
| MFCC | CNN3 | 0.542 | 0.442 | **0.443** |
| MFCC | VGG16 | **0.488** | 0.679 | 3.582 |
| MFCC | ResNet50 | **0.489** | 0.594 | 1.146 |
| MFCC | InceptionV3 | 0.641 | 0.741 | 1.077 |
| MFCC | Inception-ResNetV2 | 0.617 | 0.676 | 0.994 |
| MFCC | Xception | 0.542 | 0.537 | 0.555 |
| MFCC | DenseNet | 0.501 | **0.407** | 0.614 |
| MFCC | MobileNet | 0.578 | 0.527 | **0.783** |
| MFCC | EfficientNet | 0.607 | 0.992 | 0.986 |

**Table 2.** Accuracy values across architectures.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Model** | **Train** | **Validation** | **Test** |
| Spectrogram | SVM | **100** | - | 91.73 |
| Spectrogram | RF | **100** | - | 96.83 |
| Spectrogram | CNN1 | 96.34 | 96.53 | 96.33 |
| Spectrogram | CNN3 | 95.13 | 98.27 | 97.80 |
| Spectrogram | VGG16 | 93.09 | 98.53 | 97.33 |
| Spectrogram | ResNet50 | 95.73 | **99.27** | **99.60** |
| Spectrogram | InceptionV3 | 92.67 | 98.67 | 99.00 |
| Spectrogram | Inception-ResNetV2 | 92.77 | 99.07 | 99.06 |
| Spectrogram | Xception | 93.41 | **99.27** | 99.20 |
| Spectrogram | DenseNet | 93.64 | 99.13 | 99.26 |
| Spectrogram | MobileNet | 93.27 | 99.00 | 98.46 |
| Spectrogram | EfficientNet | 92.47 | 96.73 | 96.40 |
| MFCC | SVM | **100** | - | 92.39 |
| MFCC | RF | **100** | - | 96.26 |
| MFCC | CNN1 | 97.46 | **99.07** | **98.73** |
| MFCC | CNN3 | 97.11 | 98.47 | 98.20 |
| MFCC | VGG16 | 91.13 | 73.07 | 50.00 |
| MFCC | ResNet50 | 95.11 | 86.60 | 50.00 |
| MFCC | InceptionV3 | 91.29 | 85.00 | 51.06 |
| MFCC | Inception-ResNetV2 | 92.16 | 92.07 | 72.80 |
| MFCC | Xception | 93.14 | 93.20 | 55.53 |
| MFCC | DenseNet | 92.37 | 97.73 | 91.42 |
| MFCC | MobileNet | 92.91 | 97.20 | 85.13 |
| MFCC | EfficientNet | 90.71 | 51.00 | 51.26 |

**Table 3.** Accuracy values across architectures with a frozen pretrained encoder.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Model** | **Train** | **Validation** | **Test** |
| Spectrogram | VGG16 | 88.30 | 96.67 | 97.66 |
| Spectrogram | ResNet50 | 83.37 | 94.40 | 95.26 |
| Spectrogram | InceptionV3 | 85.13 | 97.07 | 97.40 |
| Spectrogram | Inception-ResNetV2 | 86.69 | 94.53 | 94.40 |
| Spectrogram | Xception | 91.21 | 97.20 | 96.26 |
| Spectrogram | DenseNet | 90.89 | 97.00 | 97.46 |
| Spectrogram | MobileNet | **91.59** | **98.07** | **98.66** |
| Spectrogram | EfficientNet | 50.74 | 50.00 | 50.00 |
| MFCC | VGG16 | 81.81 | 88.80 | 87.60 |
| MFCC | ResNet50 | 81.41 | 85.67 | 86.13 |
| MFCC | InceptionV3 | 80.79 | 88/13 | 89.27 |
| MFCC | Inception-ResNetV2 | 86.69 | 94.53 | 94.40 |
| MFCC | Xception | 87.73 | 93.27 | 92.40 |
| MFCC | DenseNet | 85.43 | 90.00 | 89.53 |
| MFCC | MobileNet | **83.69** | **95.73** | **94.47** |
| MFCC | EfficientNet | 50.41 | 50.00 | 50.00 |

**Table 4.** F1-Score values for the testing data across architectures.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Spectrogram** | **Spectrogram** | **MFCC** | **MFCC** |
| **Models** | **Non-frozen** | **Frozen** | **Non-frozen** | **Frozen** |
| SVM | 91.96 | - | 92.39 | - |
| RF | 96.80 | - | 96.19 | - |
| CNN1 | 96.30 | - | 98.72 | - |
| CNN3 | 96.30 | - | **98.18** | **-** |
| VGG16 | 97.40 | 97.64 | 66.66 | 85.93 |
| ResNet50 | **99.59** | 95.26 | 66.66 | 85.10 |
| InceptionV3 | 99.00 | 97.30 | 66.66 | 90.22 |
| Inception-ResNetV2 | 99.06 | 94.64 | 71.90 | 94.43 |
| Xception | 99.20 | 96.16 | 68.10 | 91.85 |
| DenseNet | 99.26 | 97.50 | 91.89 | 88.41 |
| MobileNet | 98.48 | 98.67 | 86.03 | 94.57 |
| EfficientNet | 96.44 | 00.00 | 66.81 | 00.00 |

4. Discussion

The development of an effective baby monitoring system such as is described in this paper depends on the integration of both hardware and software components. The key feature of the system is the baby cry detection model, which must achieve a high accuracy with minimal false positives for a robust performance in real-world conditions. In this work, we present an analysis of various machine learning models (including deep learning models) to determine the best option for such as system. In this analysis, we also include two feature spaces created from the recorded signals based on signal processing and we analyze the impact of training the encoding part of deep learning models.

The creation of a baby cry detection model necessitates a comprehensive dataset, which was compiled from various sources, ensuring a diverse range of cry sounds and other audio inputs for robustness. The extended dataset created through augmentation, enhances the model's ability to generalize in real-world environments.

The evaluation of different machine learning models, including neural network architectures, has provided insights into their performance in classifying infant cries and have demonstrated the potential to achieve high precision and recall rates as indicated by the F1-scores obtained on the previously unseen test data. This dual emphasis on precision and recall is essential to minimize false positives while ensuring that actual cries are not missed.

A critical observation from the different options for data features indicate that the pretrained more complex models are more capable of learning from spectrograms than MFCC features, while previously untrained (and simpler) convolutional neural networks do not have this preference. Normally, it is expected that spectrogram-based feature extraction better preserves the **time-frequency characteristics of the data**, allowing deep learning architectures to distinguish between crying and background noises more effectively, while MFCCs, commonly used in speech processing, may lose essential frequency-related information when applied to this specific task. However, our results suggest that more complex models require more complex data, this is especially visible in the test accuracy obtained by the more complex models when applied on the MFCC data.

For the non-frozen models, we observed that their performance is satisfactory only for the spectrogram features, while for the MFCC features, they are often unable to learn with the validation and test accuracy being close to that of random labelling. This is not the case for the frozen architectures, where the models are capable of learning from both the spectrograms and the MFCC features; however, the performance is slightly lower than that of non-frozen models combined with spectrogram features. The exception is the EfficientNet architecture which is able to learn only on the non-frozen setting combined with spectrogram features. Two models escape this pattern, specifically DenseNet and MobileNet, which are capable of learning from both spectrogram and MFCC features regardless whether their layers are frozen or not.

Overall, from the results obtained, the best option is the non-frozen setting with spectrogram features, where the best performance is obtained by the ResNet50 architecture. However, a simpler architecture such as CNN3 with no pretraining is capable of learning from both sets of features with satisfactory accuracy with a considerable lower amount of training time. For a more performant and robust architecture, our results indicate that DenseNet and MobileNet are the best options as they are capable of high accuracy on any combinations tested.

The best training accuracy was obtained by the machine learning models (SVM and RF), with RF having a comparable performance on the testing data with the neural network models, yet still lower than the more performant models.

In summary, the baby monitoring system proposed in this study is based on and necessitates an automatic cry detection model. We have analysed several architectures with various feature types on a comprehensive dataset to determine the best approach for the detection of infant cries in various real-world conditions and environments. Future work could explore further enhancements to the model or other architectures and expanding the dataset with additional diverse audio samples.

**Supplementary Materials:** The following supporting information can be downloaded at: www.mdpi.com/xxx/s1, Figure S1: title; Table S1: title; Video S1: title.

**Author Contributions:** For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used “Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.” Please turn to the [CRediT taxonomy](https://img.mdpi.org/data/contributor-role-instruction.pdf) for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

**Funding:** Please add: “This research received no external funding” or “This research was funded by NAME OF FUNDER, grant number XXX” and “The APC was funded by XXX”. Check carefully that the details given are accurate and use the standard spelling of funding agency names at https://search.crossref.org/funding. Any errors may affect your future funding.

**Institutional Review Board Statement:** In this section, you should add the Institutional Review Board Statement and approval number, if relevant to your study. You might choose to exclude this statement if the study did not require ethical approval. Please note that the Editorial Office might ask you for further information. Please add “The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval).” for studies involving humans. OR “The animal study protocol was approved by the Institutional Review Board (or Ethics Committee) of NAME OF INSTITUTE (protocol code XXX and date of approval).” for studies involving animals. OR “Ethical review and approval were waived for this study due to REASON (please provide a detailed justification).” OR “Not applicable” for studies not involving humans or animals.

**Informed Consent Statement:** Any research article describing a study involving humans should contain this statement. Please add “Informed consent was obtained from all subjects involved in the study.” OR “Patient consent was waived due to REASON (please provide a detailed justification).” OR “Not applicable.” for studies not involving humans. You might also choose to exclude this statement if the study did not involve humans.

Written informed consent for publication must be obtained from participating patients who can be identified (including by the patients themselves). Please state “Written informed consent has been obtained from the patient(s) to publish this paper” if applicable.

**Data Availability Statement:** We encourage all authors of articles published in MDPI journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Where no new data were created, or where data is unavailable due to privacy or ethical restrictions, a statement is still required. Suggested Data Availability Statements are available in section “MDPI Research Data Policies” at https://www.mdpi.com/ethics.

**Acknowledgments:** In this section, you can acknowledge any support given which is not covered by the author contribution or funding sections. This may include administrative and technical support, or donations in kind (e.g., materials used for experiments).

**Conflicts of Interest:** Declare conflicts of interest or state “The authors declare no conflicts of interest.” Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results. Any role of the funders in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results must be declared in this section. If there is no role, please state “The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results”.

Abbreviations

The following abbreviations are used in this manuscript:

|  |  |
| --- | --- |
| MDPI | Multidisciplinary Digital Publishing Institute |
| DOAJ | Directory of open access journals |
| TLA | Three letter acronym |
| LD | Linear dichroism |

References

[1] R. Cohen, D. Ruinskiy, J. Zickfeld, H. IJzerman, and Y. Lavner, “Baby Cry Detection: Deep Learning and Classical Approaches,” 2020, pp. 171–196. doi: 10.1007/978-3-030-31764-5\_7.

[2] J. Benesty, M. M. Sondhi, and Y. (Arden) Huang, “Introduction to Speech Processing,” in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. A. Huang, Eds., Berlin, Heidelberg: Springer, 2008, pp. 1–4. doi: 10.1007/978-3-540-49127-9\_1.

[3] J. K. Das, A. Ghosh, A. K. Pal, S. Dutta, and A. Chakrabarty, “Urban Sound Classification Using Convolutional Neural Network and Long Short Term Memory Based on Multiple Features,” in *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, Oct. 2020, pp. 1–9. doi: 10.1109/ICDS50568.2020.9268723.

[4] K. Rezaee, H. Ghayoumi zadeh, L. Qi, H. Rabiee, and M. 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, Jan. 2024, doi: 10.1145/3579032.

[5] S. Rajagopal, P. Poonkodi, M. Kavitha, and K. Subburathinam, “Premature Infant Cry Classification via Deep Convolutional Recurrent Neural Network Based on Multi-class Features,” *Circuits Syst. Signal Process.*, vol. 42, pp. 1–20, Aug. 2023, doi: 10.1007/s00034-023-02457-5.

[6] Y. Zayed, A. Hasasneh, and C. Tadj, “Infant Cry Signal Diagnostic System Using Deep Learning and Fused Features,” *Diagnostics*, vol. 13, p. 2107, Jun. 2023, doi: 10.3390/diagnostics13122107.

[7] D. Budaghyan, C. C. Onu, A. Gorin, C. Subakan, and D. Precup, “CryCeleb: A Speaker Verification Dataset Based on Infant Cry Sounds,” Mar. 21, 2024, *arXiv*: arXiv:2305.00969. doi: 10.48550/arXiv.2305.00969.

[8] G. Veres, *gveres/donateacry-corpus*. (Jul. 02, 2024). Accessed: Jul. 11, 2024. [Online]. Available: https://github.com/gveres/donateacry-corpus

[9] “Librispeech: An ASR corpus based on public domain audio books | IEEE Conference Publication | IEEE Xplore.” Accessed: Jul. 11, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/7178964

[10] Y. D. Rosita and H. Junaedi, “Infant’s cry sound classification using Mel-Frequency Cepstrum Coefficients feature extraction and Backpropagation Neural Network,” in *2016 2nd International Conference on Science and Technology-Computer (ICST)*, Oct. 2016, pp. 160–166. doi: 10.1109/ICSTC.2016.7877367.

[11] A. E. Jang, *eunbeejang/SilentBabyMonitor*. (Jul. 10, 2024). Python. Accessed: Jul. 11, 2024. [Online]. Available: https://github.com/eunbeejang/SilentBabyMonitor

[12] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Apr. 10, 2015, *arXiv*: arXiv:1409.1556. doi: 10.48550/arXiv.1409.1556.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *ArXiv151203385 Cs*, Dec. 2015, Accessed: Nov. 15, 2021. [Online]. Available: http://arxiv.org/abs/1512.03385

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Dec. 11, 2015, *arXiv*: arXiv:1512.00567. doi: 10.48550/arXiv.1512.00567.

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” Aug. 23, 2016, *arXiv*: arXiv:1602.07261. doi: 10.48550/arXiv.1602.07261.

[16] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” Apr. 04, 2017, *arXiv*: arXiv:1610.02357. doi: 10.48550/arXiv.1610.02357.

[17] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” Apr. 16, 2017, *arXiv*: arXiv:1704.04861. doi: 10.48550/arXiv.1704.04861.

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” Jan. 28, 2018, *arXiv*: arXiv:1608.06993. doi: 10.48550/arXiv.1608.06993.

[19] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *ArXiv190511946 Cs Stat*, Sep. 2020, Accessed: Nov. 15, 2021. [Online]. Available: http://arxiv.org/abs/1905.11946