Baby Cry Detection: Deep Learning and Classical Approaches

Rami Cohen[1], Dima Ruinskiy[2, 3], Janis Zickfeld[4], Hans IJzerman[5], and Yizhar Lavner[2]

[1]Viterbi Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel
rc@technion.ac.il
[2]Department of Computer Science, Tel-Hai College, Upper Galilee, Israel
dima.ruinskiy@gmail.com, yizhar.lavner@gmail.com
[3]Network.IO Innovation Lab, Haifa, Israel
[4]Department of Psychology, University of Oslo, Norway
jhzickfeld@gmail.com
[5]LIP/PC2S, Université Grenoble Alpes, France
h.ijzerman@gmail.com

Abstract. In this chapter, we compare deep learning and classical approaches for detection of baby cry sounds in various domestic environments under challenging signal-to-noise ratio conditions. Automatic cry detection has applications in commercial products (such as baby remote monitors) as well as in medical and psycho-social research. We design and evaluate several convolutional neural network (CNN) architectures for baby cry detection, and compare their performance to that of classical machine-learning approaches, such as logistic regression and support vector machines. In addition to feed-forward CNNs, we analyze the performance of recurrent neural network (RNN) architectures, which are able to capture temporal behavior of acoustic events. We show that by carefully designing CNN architectures with specialized non-symmetric kernels, better results are obtained compared to common CNN architectures.

Keywords: Baby cry detection, Deep learning, Convolutional neural networks, Audio detection

1 Introduction

In recent years, *deep neural networks* have been used with great success in a variety of practical real-world problems. As opposed to the traditional feature-extraction stage in classical machine-learning algorithms, deep neural networks

automate the formation of useful features from the data. Such networks consist of multiple layers, resembling the way computations are performed in the brain. In these layers, a hierarchy of non-linear features is formed, growing in complexity with the depth of the network. A combination of these features is used in the last layer of the network to generate a prediction.

The most impressive results are perhaps in the area of computer vision, starting with the seminal work of [1], where deep learning networks exhibit state-of-the-art performance in various tasks, such as object detection and classification. With the advances in deep-learning techniques and the availability of large databases for training, deep learning is also becoming an important tool for automatic audio event detection [2, 3].

In this chapter, we compare deep-learning and classical approaches for the detection of baby cry events in acoustic signals. Accurate and reliable detection of infant cry events in a stream of audio is a prerequisite for classification algorithms and screening tasks, which rely on the acoustic properties of the cry. One of the main difficulties in detecting baby cry in a domestic environment or in other natural environments, such as neonatal clinic units or nurseries, is the presence of noise and background sounds - speech, music, electronic toys, door opening, phone ringing, and many others. This poses a considerable challenge for classical machine-learning approaches, which typically start by extracting a set of distinguishing features from the acoustic signal. Background noise may have fundamental frequency or vocal qualities similar to those of infant cry, hindering the detection algorithm. In addition, the signal-to-noise ratio (SNR) often varies. Speech in particular poses a considerable challenge for the detection, due to frequency content similar to baby cry, which may introduce false-positive events.

In the research described here we devise and evaluate deep-learning approaches for baby cry detection. We design specialized convolutional neural networks (CNNs) for this task and study appropriate image representations of audio signals for serving as inputs to the CNNs. We use the non-linear log Mel-filter bank (log MFB) representation, where each pixel represents a frequency range according to the logarithmic Mel-scale; this representation is known to capture well the relevant frequencies that distinguish different types of the acoustic signals. We compare the performance of our CNN architectures to traditional machine-learning algorithms, such as logistic regression and support vector machine (SVM) classifiers.

The performance evaluation is carried out using an annotated database containing several hours of recordings of babies in domestic environments. In addition to baby cry, these recordings contain various types of domestic sounds, such as phone ringing, door opening and parent speech. We discuss the trade-off between the false-positive rate and the detection rate, based on a receiver operating characteristic (ROC) curve, and provide performance analysis of CNN detection results with a varying number of layers and units. We show considerable performance gain compared to classical machine-learning approaches, especially at the low false-positive rate regime.

1.1 Approaches in audio event detection

Audio event detection is the task of spotting specific acoustic events within long clips or streams of audio data. Examples of acoustic events are human voices, specific utterances, different types of domestic or urban noises, sounds produced by various musical instruments, and many others. The detection task can have varying requirements: from simple spotting (presence or absence of the event in question, such as in a voice activity detector [4]) to accurate demarcation of the event boundaries (onset and offset), e.g., [5]; in some cases detection of multiple event types may be desired, which introduces a related task of sound event *classification*: assigning each audio portion to one of several predefined classes (sometimes referred to as *annotation*). The classification may be applied only to the detected events, for example distinguishing different types of fricative consonants [6, 7], or to entire segments, for instance, when discriminating between speech and music [8, 9], or between various musical genres [10].

Automatic detection and classification of acoustic events in audio signals is a major research field in machine learning, due to its many applications. In the broadest sense, it is a key part in *auditory machine perception* (also known as *machine listening*, [11]) - where a computer learns to interpret and analyze audio information similarly to a human. Some examples of specific applications are speech recognition [12], voice-controlled appliances [13], audio surveillance [14], health monitoring [15] and audio signal enhancement [5, 16].

One of the early implementations of deep learning for acoustic event detection [17], used a convolutional neural network (CNN) for a robust sound event recognizer in different kinds of noisy environments. In [18], a deep model consisting of 2 convolutional layers with max-pooling and 2 fully connected layers was used for detecting various urban and environmental sounds. A CNN architecture was also employed for acoustic scene classification task in [19]. [20] showed that a convolutional recurrent neural network (RNN) approach yielded better results in polyphonic sound detection (where multiple types of events can be detected simultaneously) than separate CNN/RNN or traditional approaches. A similar approach was used in [21] for a low-latency monaural sound source separation system. Applicability of deep learning approaches for automatic music tagging (assigning properties such as genre, instrumentation, rhythm, etc.) was also demonstrated [22, 23].

Other types of deep learning architectures have also been investigated. For example, in [24], the authors used a deep network consisting of convolutional layers, followed by fully connected layers and a single-element output layer with a sigmoid activation function; [25] used a topology known as Capsule Network (CapsNet, [26]) and found that it could outperform a traditional CNN. Another study compared deep learning methods to a hand-crafted support vector data description (SVDD) classifier with a few carefully selected features, and found that the latter can achieve comparable performance to a CNN at lower computational cost, but with the drawback of having to design features specific to the task [27].

Various methods have also been investigated for the specific task of cry detection; a good survey of traditional approaches is available in [28]. In re-

cent years, advances in deep learning made it a popular technique as well. In [29], a specially-designed CNN was shown to outperform a traditional logistic regression-based classifier in very low false-positive rate regimes. In [30], a CNN running on audio captured from a microphone array installed next to a baby carriage could detect cry with 86% accuracy. [31] compared a CNN followed by a Hidden Markov Model (HMM) to a Linear Discriminant Analysis (LDA) classifier.

An automatic baby cry detector has many applications: it is commonly employed in safety-related devices, such as baby monitors [32], and has been proposed as part of a system to detect children forgotten in vehicles [33]; some commercial products featuring cry detection technology include [34–36]. Identification, followed by classification of the cry signals, can be useful for medical purposes, such as detection of pathologies based on the auditory properties of the cry signal (e.g., [37, 38]), or assessment of the neurological state of infants based on differences in the crying between full-term and preterm babies [39].

1.2 The origin and role of infant cry

The human cry has several roles, depending on developmental stage. The initial function, and thus developmental origin, is relatively clear. For some of the other aspects, there are generally-accepted theories and hypotheses, but not solid conclusions, as we are not sufficiently confident of psychological theory as a means for prediction. The vocal cry is one of the first forms of communication to interact with the caregiver [40], and is common to more species than humans. [41] proposed that feedback-sensitive attachment behavior is vital to retain the caregivers proximity, and that crying is one of the most important channels for establishing it [42–44]. In humans, the infant depends on the caregiver for food, safety and warmth [45]. As a result, much of our functioning focuses on a kind of co-regulation that caregivers provide to infants early in life, and adult partners to each other later in life [46].

Crying in human infants is elicited from rhythmical transitions between inhalation and exhalation, due to a vibration of the vocal cords that produces periodic air pulses. The period of these pulses is called the fundamental frequency (pitch), and its typical values in healthy babies are $250-600$ Hz. The cry signal is shaped by the vocal tract, leading to resonant frequencies termed as formants. The first two formants occur typically around 1100 Hz and 3300 Hz, respectively [47]. Some studies [44, 48] point out that humans early in life already display various forms of crying: protest crying (in which the infant faces loss, like being left in the crib, and wants to undo the loss), sad crying of despair (a low wail signifying acceptance of loss), and detached inhibited crying (typically an absence of outward crying, associated with a life-threatening separation from the caregiver). Sometimes additional types, such as hunger and pain crying are considered.

For a long time, researchers assumed that different types of cries cannot be reliably distinguished, as even mothers are not always accurate at discriminating pain versus hunger crying (e.g., [43]). Nowadays, with drastically improved accuracy of measurement equipment and analysis software, and with much larger sample sets available, there are reasons to doubt this assumption. Re-

cent research suggests that one can reliably distinguish between pain, hunger, sadness, fear, and anger cries on the basis of facial expressions and cry characteristics [49, 50], and it seems plausible that deep learning methods can be applied for successful automatic classification of different types of cry.

The type, frequency, and duration of crying are highly variable, especially after early infancy (e.g., [43, 51]). Crying peaks at about 6 weeks and then declines until 4 months after which it remains rather stable [52]. The development of crying depends in part on the caregiver's response to the cry (or the lack of it). In young infants, if the caregiver responds (by holding, touching, and/or feeding) crying typically ceases [53]. A positive response from the caregiver signals that the infant can rely on others to help meet environmental demands, while a lack of it signals that the infant needs to cope with environmental demands itself. Later in life, the type, frequency, and duration may depend on an individuals temperament [54, 55], or attachment [56]. For example, infants whose mothers did not respond consistently to their cry later started oscillating between clinging to their mother and resisting contact (see e.g., [43]). In many cases, infants adapt to whatever is required for survival (for example, self-reliance in case of a non-responsive caregiver); however, differences in crying related to such adaptations have not yet been fully investigated.

There is no complete knowledge of the characteristics of pathological cry, in part due to a lack of sufficiently large samples and accurate methodologies. At the very least, a cry signal with fundamental frequency (pitch) above 600Hz has been generally regarded as indicating health issues [57]. Another function of crying may thus well be to convey the infant's fitness, preventing caregivers from investing [47]. Pain crying usually starts abruptly, is usually longer, and often involves hyperphonated cries (characterized by high pitch; [43]). A commonly accepted pathology, known as 'colic', is indicated by excessive crying; researchers typically distinguish between intrinsic causes for colic, such as food allergies, or extrinsic ones, for example inappropriate physical contact.

Classification of normal and pathological infant crying (especially in newborns) has been the subject of extensive research. Some studies focused on a specific pathology, such as deafness, or respiratory distress syndrome (RSD), while in others multiple pathologies were investigated [58]. Differences of cry between normal babies and those with high-risk for autism have been studied as well [59, 60]. A recent survey of the research status is available [61].

While pathological cry in newborns has been extensively studied, little is known about how crying early in life relates to potential maladaptive behaviors later in life. Large samples of crying recordings, combined with novel machine learning and deep learning techniques can help identify possible connections (for examples in psychology, see [62, 63]). One of the difficulties with research in the early stages of life is that it is often too intrusive. To make it less intrusive, we have integrated automatic cry detection in an Android-based smartphone app[1]. This smartphone app can be used to detect, in real-time, baby cry events, in the infant's earliest days. Usage of the app allows the infant cry to be correlated to other variables (like caregiver peripheral temperature).

[1]Code available at https://github.com/co-relab/bioapp

2 Deep Learning Approach

Convolutional neural networks (CNNs) [64, 65] have wide applications in the fields of computer vision, natural language processing and many others, especially where huge amounts of data have to be processed and classified. Like ordinary neural networks, CNNs consist of multiple layers connected by neurons that have learnable weights. Commonly used layer types are: convolutional layers (applying convolution / dot product operation), pooling layers (combining outputs of several neurons into a single neuron in the next layer), rectified linear unit (ReLU) layers, fully-connected (dense) layers and more. The exact number and configuration of layers is application-dependent.

CNNs learn the parameters (usually termed as *weights*) of each layer in a training process. This process is carried out using a gradient descent approach and the backpropagation technique [66]. The complexity of the training process scales with the number of *trainable* parameters of the network. In typical CNNs, there might be thousands to millions of parameters, whose values are learned in the training stage.

In contrast to traditional classification algorithms (e.g., support vector machines), no features are typically extracted from the data prior to CNN-based classification and detection. Instead, the input to a CNN usually consists of the raw data, e.g. images, which is one of their primary advantages. However, the performance of CNN-based classifiers depends heavily on the architectural structure of the CNN in use. Designing a CNN architecture requires choosing the number of layers, the kernel size, connectivity between layers and more.

In this section, we design and evaluate multiple CNN architectures for the detection of baby cry, operating on log Mel-filter bank representation of the audio data.

2.1 Data representation

In image classification or detection tasks based on CNNs, the input to the network is typically composed of the raw images. The CNN role is then to efficiently extract spatial features from the input images and to propagate them to deeper layers, such that correct prediction is obtained at the CNN output. For example, the detection of an object such as a cat can be viewed as the detection of its eyes, mouth and tail, by dedicated filters. However, using raw audio signals as an input to a CNN is typically undesired, as the convolution filters will be applied to temporally-adjacent samples. When dealing with sampling rates such as $44,100$Hz, the output of such filters is of limited benefit, in particular when typical small (one-dimensional) kernels are used.

To better exploit the power of CNNs for audio classification tasks, it is often beneficial to convert audio signals to an image representation with meaningful spatial information. A natural approach for this aim is the use of time-frequency representations. The most common type of such representation is spectrograms, generated by applying the short-time Fourier transform to the data. However, the linear scale of the spectrogram (in time and frequency domains both) makes it difficult to separate simultaneous sounds with similar frequency content based. Thus, the efficiency of spectrogram representation for

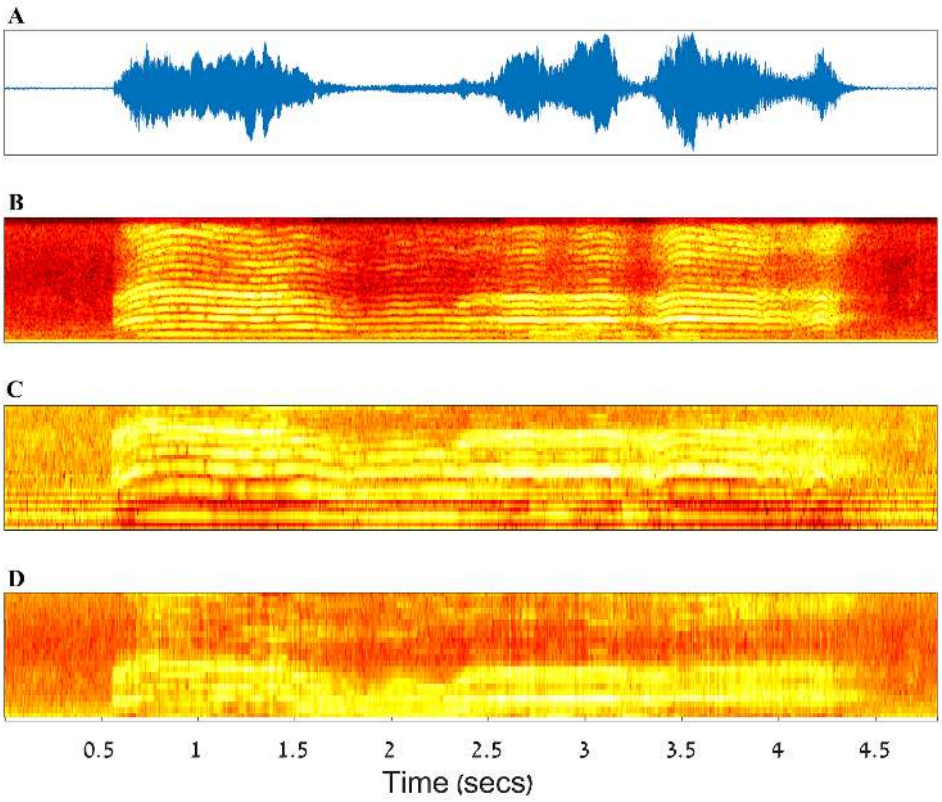


Figure 1. Different representations of a short segment (5 seconds) of a baby cry signal. A: Waveform of the signal; B: Spectrogram; C: Mel Filter Bank (MFB) representation; D: Linear Filter Bank (LFB) representation. In B, C, and D the vertical axis represents the frequency axis in a range of 0-5kHz. Note that in B (spectrogram) and D (LFB) the frequency axis is linear, while in C it is logarithmic (mel-frequency).

our task is likely to be of limited benefit, in particular in presence of noise with characteristics similar to those of cry signals.

To improve the robustness of the CNNs to noise, we use a *log Mel-filter bank* (log MFB, LMFB) representation [67] of the audio signals. The Mel-scale aims to mimic the non-linear human ear perception of sound, by being more discriminative at lower frequencies and less discriminative at higher frequencies. This logarithmic representation is often beneficial for sound classification, as it better separates different types of signals with similar frequency content. The main difference between MFB and Mel-Frequency Cepstrum coefficients (MFCC) [67] is that the discrete cosine transform (DCT) of the log-power spectrum is skipped in MFB. This is because DCT decorrelates the data, whereas spatial correlation of the input is actually advantageous for a CNN. The reason is that the 2D convolution operation used in convolutional layers is motivated by the correlation between neighbouring pixels in natural images.

To produce a log MFB representation of the data, the input audio signal is divided into consecutive segments of 4096 samples each. These segments are further divided into frames of 512 samples each, with a step size of 128 samples. As the contribution of high frequency bands to the detection of cry signals is limited, a low-pass filter at 11025 Hz is applied to the signal. A log MFB representation is then produced for each frame, using 50 triangular filters distributed according to the Mel scale in the frequency range $[0, 11025]$ Hz. Given segments of 4096 samples and a step size of 128 samples, this leads to a 50×29 "image" representation of each segment. Another representation which has been used is the log Linear Filter Bank (LFB), which is produced using the same procedure as the MFB but with linearly-distributed filters. An example is shown in Figure 1. In this figure, a short segment of a baby cry signal waveform is shown, with the corresponding spectrogram, MFB and LFB representations.

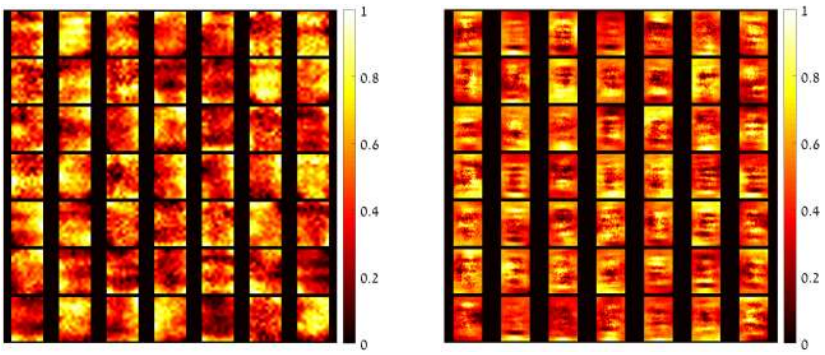
2.2 Feed-forward architectures

In natural images, both image dimensions carry the same content (pixel color values). Therefore, CNNs for images typically use two-dimensional filters that share weights across both dimensions and symmetric (e.g., 3×3) kernels. However, in the audio domain, a crucial observation is that for time-frequency representations, the x and y axes represent fundamentally different units, i.e., time in seconds and frequency in Hz. In addition, the scale of each axis might be different. Taking the LMFB representation as an example, the frequency axis is in logarithmic scale whereas the time axis is in linear scale. This calls for a careful design of the filter kernels, preferably concentrating on frequency rather than time content.

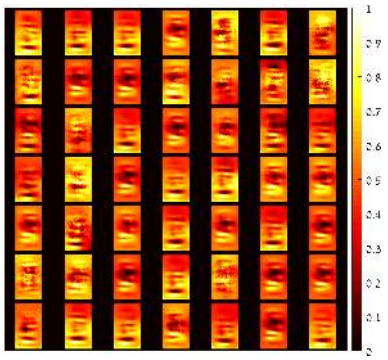
In [29], we developed a specialized CNN architecture for cry detection. Most notably, we used convolution layers with "tall" filters, i.e., non-symmetric kernels with height (frequency content) larger than width (time content). This choice of kernels is motivated by the logarithmic scale of the frequency in the LMFB representation. The use of "tall" filters makes the network "focus" on the frequency behaviour, better capturing subtle changes in signals with similar frequency content.

In this study, we improve the architecture proposed in [29] and compare the results to an architecture based on the Inception module [68]. In our CNN architecture, we have five convolutional layers followed by a fully-connected layer for final classification. We start with a 14×10 kernel for the first convolutional layer, reducing each dimension of the kernel by 2 for each subsequent layer. This gradual decrease of the kernel dimensions can be seen as multi-scale processing of the input data, which captures the frequency behaviour in an efficient manner. The architecture is presented in Table 1. Note that each convolutional layer is followed by ReLU (omitted in Table 1).

In our experiments, we were looking to test whether our specialized "tall" kernels perform better than the more common 3×3 kernels. In addition, we were interested in architectures with a smaller number of trainable parameters. For this purpose, we considered two additional architectures similar to those presented in Table 1: the first architecture has kernels replaced by 3×3 kernels



(a) Conv1 **(b)** Conv3



(c) Conv5

Figure 2. A visualization of the feature maps (scaled) produced by our CNN architecture with "tall" kernels. Note that only the first 49 maps are shown for each layer.

for all convolutional layers; the second has the same "tall" kernels, but with a reduced number of filters, such that the total number of trainable parameters is $270,000$. A visualization of the feature maps obtained for the first, third and fifth convolutional layers after the training phase is provided in Figure 2 and in Figure 3. Note that the differences in scale of the feature maps between the figures are due to the different kernel sizes. This visualization demonstrates that the learned features in each architecture are significantly different. As expected, the features in the first layer are less structured, whereas the last features exhibit more organized patterns.

For comparison, we used an architecture based on modules similar to the Inception-ResNet-A module used in the *Inception-Resnet-v2* network [68]. In the Inception-ResNet-A module, the input is processed through two parallel convolution paths, composed of convolutional layers with different kernel sizes. For improved performance, a residual connection [69] is used, so that output of the convolution operation of the inception module is added to the input. In addition, batch normalization [70] is applied to improve training convergence.

Layer	Filter size, #filters	Activations	#Parameters
Conv1	14×10, 300	$37 \times 20 \times 300$	$42,300$
Conv2	12×8, 250	$26 \times 13 \times 250$	$7,200,250$
Conv3	8×6, 250	$19 \times 8 \times 150$	$1,800,150$
Conv4	6×4, 150	$14 \times 5 \times 150$	$540,150$
Conv5	4×2, 50	$11 \times 4 \times 50$	$60,050$
Fully-connected	-	2	$4,402$
Total	-	-	$9,647,302$

Table 1. Our CNN 9.6M architecture.

To match the input and output depth size, 1×1 kernels are applied to both the input and its processed version. The inception architecture is relatively fast to train, as it is mostly based on convolutional layers.

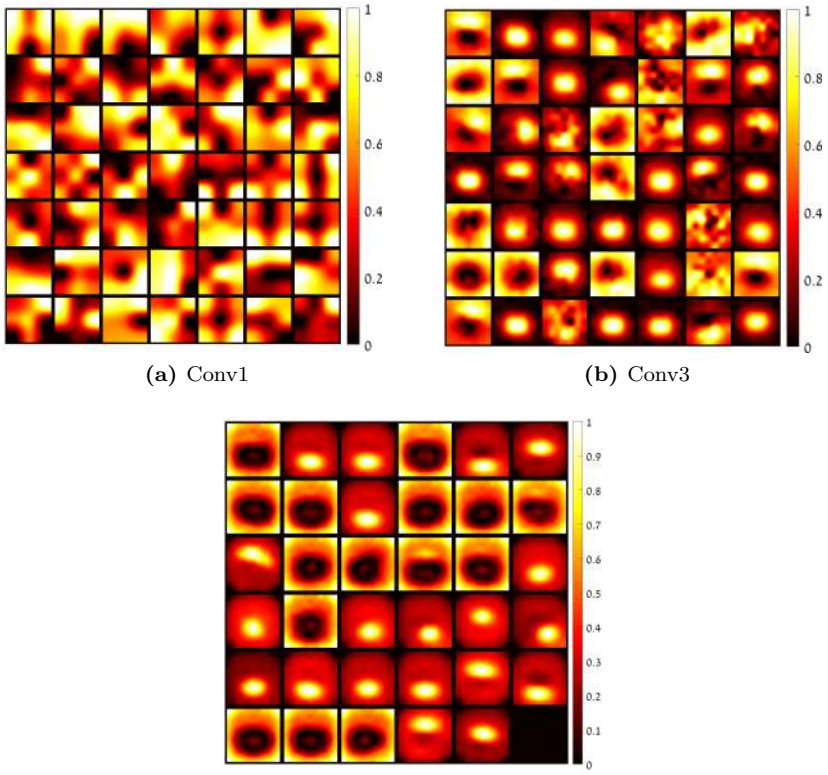
In our variation of Inception-ResNet-A module we omit the last 384 filters of 1×1 kernels for reduced complexity and due to the relatively low dimensionality of the input. In our experiments, we consider up to three Inception-ResNet-A modules. To obtain a proper two-class (cry/not cry) distribution, we reduce the output depth of the last module to 2 using two fully-connected layers (with 10 and 2 units, respectively). The final output is obtained by applying softmax to the output of the last fully-connected layer.

2.3 Recurrent neural networks (RNNs)

In traditional feed-forward architectures, the inputs and the outputs are independent of each other. As a result, such networks are not capable of modeling sequences; for example, feed-forward networks might not be an optimal choice for tasks such as predicting words in a sentence (e.g., auto-complete), as we need to know what the previous words were. On the other hand, recurrent neural networks (RNNs) [65] are designed to capture temporal information, by introducing a memory component. In recent years, RNNs had great success in a variety of problems such as language modeling, translation, image captioning and more. They have also been found useful for tasks of audio detection and classification [71], in particular for speech recognition [72].

In RNNs, the output at each time instant depends on previous computations. This is obtained by learning a *state* for each time instant, which depends on the current input and the previous state. The initial state is typically initialized to all zeroes. Compared to a possible approach of using 3D convolutions (i.e., operating on the temporal axis as well), RNNs with 2D convolutions (and states) offer a more efficient and less complex approach for learning spatiotemporal features. For our application of cry detection, the use of memory is expected to be beneficial, as cry sequences are likely to be correlated.

In our experiments, we studied the performance of *bidirectional* recurrent neural network (BiRNN) architectures [73]. In BiRNNs, the output at each time instant depends on future inputs as well as on past ones: the network



(a) Conv1 **(b)** Conv3

(c) Conv5

Figure 3. A visualization of the feature maps produced by our CNN architecture with 3×3 kernels.

has both *forward* and *backward* states, which are used at each time instant to compute an output. BiRNNs typically exhibit improved performance and convergence behaviour over standard one-sided RNNs [72, 74, 75]. We considered two kinds of BiRNN architectures, each with two layers. In the first architecture, we used standard BiRNN layers. In our second architecture, we replaced the BiRNN layers with bidirectional long short-term memory (BiLSTM) layers [76–78], in which the computation of the state is more complex compared to BiRNN, resulting in better capturing of long-term dependencies. In addition, BiLSTM layers are less susceptible to the problem of vanishing or exploding gradients [79].

As a preliminary step, the input data is processed by an all-convolutional network, with the same structure (apart from the fully-connected layer) as in Table 1, but with only $270,000$ parameters. This number of parameters is obtained by reducing the number of filters in each layer of the architecture presented in Table 1 by an appropriate factor. In both our BiRNN and BiLSTM architectures, we used states with 128 units.

3 Classical Approaches

Classical approaches for baby cry detection typically involve extraction of distinguishing features from segments of the audio signal, and using them to train a classifier. Common features include pitch, formants, and various temporal and spectral properties, such as short-time energy, Mel-frequency cepstrum coefficients (MFCC) and others [28, 37, 80]. In our study we compared the deep learning architectures to two traditional techniques - one using a logistic regression classifier, and the other using a Support Vector Machine (SVM). The duration of the audio segments and the size of the feature vectors, were similar to those used in the deep learning algorithms.

3.1 Preprocessing and feature extraction

The audio recordings are divided into consecutive overlapping segments of 4096 samples (about 93ms) with an overlap of 50%. For pitch detection purposes, the segments are further divided into frames of 16ms.

The following features are computed for each audio segment:

1. **Pitch-related features**. The pitch detection algorithm uses peaks in the cepstrum domain $c(n) = \texttt{IDFT}(\log(\texttt{DFT}|x(n)|))$ to obtain a rough estimation, and cross-correlation in the time-domain for refinement of the initial pitch value [81]. Once the prominent peak N_p is found in the section of a cepstrum that corresponds to the expected periodicity in baby cry signals (200-600Hz), a more accurate value is obtained by finding the maximum cross-correlation between adjacent signal vectors of length K, where K is a value in a neighborhood of size δ around N_p. This approach is based on the assumption that maximal similarity between the vectors is obtained when their length is equal to the pitch period N_0:

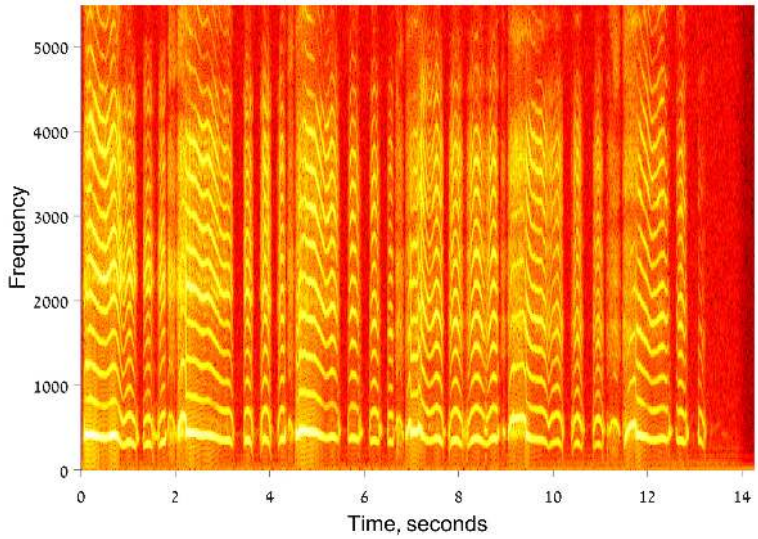


Figure 4. A spectrogram of an infant cry signal, demonstrating the harmonic structure of voiced bursts.

$$N_0 = \underset{N_p-\delta \le K \le N_p+\delta}{\operatorname{argmax}} \frac{\sum_{j=1}^{K} y_1(j) \cdot y_2(j)}{\sqrt{\sum_{j=1}^{K} y_1^2(j) \cdot \sum_{j=1}^{K} y_2^2(j)}} \tag{1}$$

If the maximum cross-correlation is over 0.85, the pitch is considered valid and the frame is considered voiced. Since the pitch is computed on a frame level, the following segment level features are extracted from it: the *median value* across the segment, and *run-length* (number of consecutive voiced frames).

2. **Harmonics analysis**. Cry bursts are predominantly voiced, and therefore their signal is characterized by a harmonic structure, as demonstrated in Figure 4. Therefore, we expect high spectral energy content around the harmonic frequencies, and compute two parameters to capture that. The first, known as *Harmonicity Factor* (H_f) measures the harmonic content of a given frame by finding the frequencies of the L most prominent peaks f_i in the spectrum, and calculating the amount of their deviation from a predicted harmonic frequency according to the corresponding f_0 (pitch) estimation, as follows:

$$H_f = \frac{1}{L} \sum_{i=1}^{L} \min(f_i \bmod f_0, f_0 - f_i \bmod f_0) \tag{2}$$

For harmonic peaks the distance between the frequency and an integer

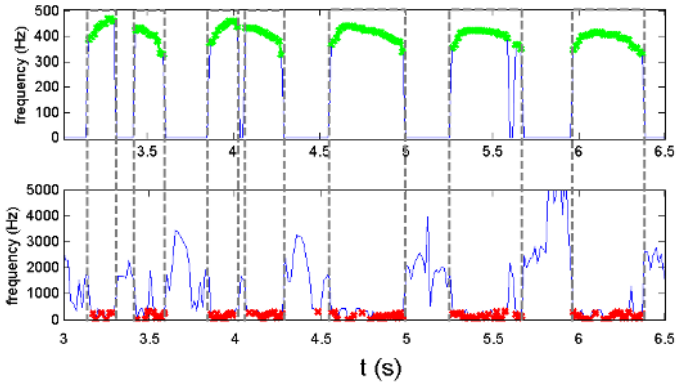


Figure 5. Fundamental frequency F_0, and Harmonicity Factor (H_f) aligned for several cry bursts.

multiple of f_0 should be small, and therefore for frames with harmonic structure H_f should be small (see Figure 5).

The second parameter is the *Harmonic-to-Average Power Ratio* (HAPR) which measures the ratio between the power of the first M harmonic components to that of the average spectral power of the frame. If $X(k)$ is the DFT of the audio frame, and f_m is the frequency of peak in the vicinity of the m^{th} harmonic, *HAPR* is computed as follows ([82,83]):

$$\text{HAPR} = \frac{1}{M}\sum_{m=2}^{M} 10\log_{10}\frac{|X(f_m)|^2}{(\frac{1}{N}\sum_{k=0}^{N-1}|X(2\pi k/N)|^2)} \tag{3}$$

3. **Filter banks and cepstrum coefficients**. A filter-bank representation is obtained by multiplying the power spectrum $X_m(K)$ by triangularly-shaped filters $V_i(K)$, where U_i and L_i are the lower and upper bounds of each filter, respectively, and $S_i = \sum_{K=U_i}^{L_i}|V_i(K)|^2$ is a normalization coefficient to compensate for the variable bandwidth of the filters:

$$E_i = \frac{1}{S_i}\sum_{K=U_i}^{L_i}|X_m(K)V_i(K)|^2 \tag{4}$$

A commonly used filter-bank representation uses the Mel scale, which arranges the filters and their widths in a way that attempts to mimic the auditory perception of the human ear [84,85]. A comparison between the Mel filter-bank (MFB) and Linear filter-bank (LFB) is shown in Figure 6, and their spectral representations can also be seen in Figure 1.

When MFB representation is used, the Mel-Filter Cepstrum Coefficients (MFCC) vector is obtained by applying the Discrete Cosine Transform

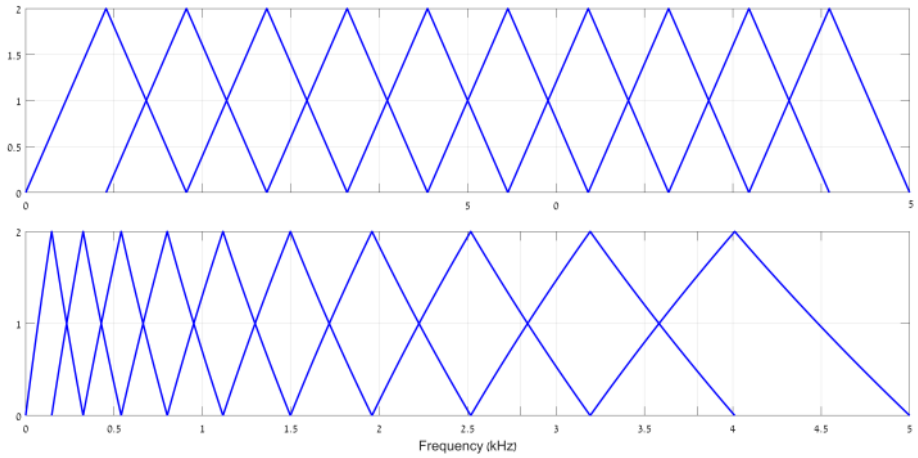


Figure 6. A schematic description of Mel-filter bank (bottom) vs Linear filter bank (top), both with 10 filters in the range of 0-5kHz.

(DCT) to the logarithm of the energy vector E_i. In our algorithm, we used the first 38 MFCC.

Figure 7 shows an example of the distribution of the 5th MFC coefficient among baby cry sections (red) vs. all other sound events (blue) in the training set (about 320 seconds). The discriminating potential of this feature is evident, although there is a wide overlapping area.

4. **Spectral energy parameters.** Different types of audio signals often exhibit different patterns in the spectral energy, as represented by the power spectrum $|X_m(K)|^2$. We computed the following two parameters: the *Spectrum rolloff point* f_R - the frequency below which 75% of the spectral energy is concentrated, and the *Band energy ratio* between the total spectral energies of two frequency bands $[0, \texttt{2500Hz}]$ and $[\texttt{2500Hz}, F_s/2]$, where F_s is the sampling frequency:

$$f_R = f : \sum_{K=0}^{f} |X_m(K)|^2 = 0.75 \cdot \sum_{K=0}^{F_s/2} |X_m(K)|^2 \tag{5}$$

$$\text{BER} = \texttt{10} \log_{10} \frac{\sum_{K(f_{\texttt{2500Hz}})}^{F_s/2} |X_m(K)|^2}{\sum_{K=0}^{f_{\texttt{2500Hz}}} |X_m(K)|^2} \tag{6}$$

5. **Time-domain features.** Simple time-domain features, such as the zero-crossing rate (ZCR) and short-time energy were also used:

$$\text{ZCR}(m) = \frac{1}{2N} \sum_{n=1}^{N-1} |\text{sign}(x_m(n)) - \text{sign}(x_m(n-1))|. \tag{7}$$

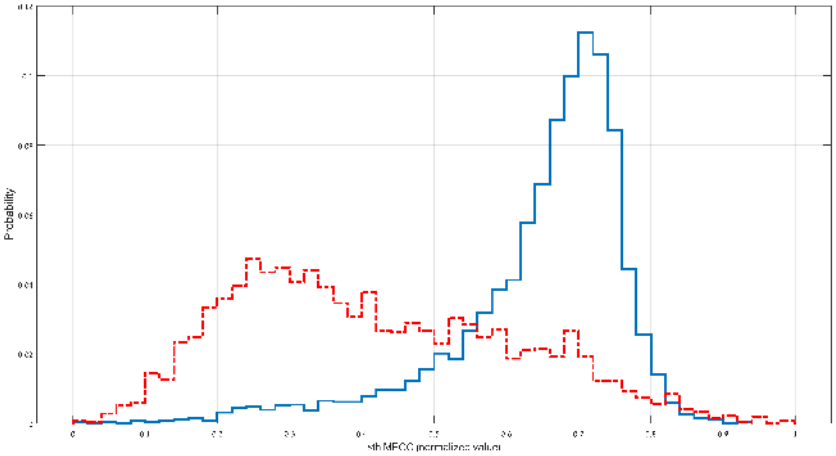


Figure 7. A histogram of the 5th MFC coefficient. Dashed line: cry events, solid line: other events.

$$E(m) = \frac{1}{N} \sum_{n=0}^{N-1} x_m^2(n). \tag{8}$$

A detailed description of the features and their computation is available in [9, 67, 80].

3.2 Logistic regression

The *logistic regression* classifier [64] is a simple supervised discriminative algorithm, with low computational complexity. The logistic regression is a non-linear hypothesis function of the form:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + \exp\left(-\boldsymbol{\theta}^T \boldsymbol{x}\right)}, \tag{9}$$

where $\boldsymbol{x}$ is a d-dimensional feature vector and $\boldsymbol{\theta}$ is a weight vector. In our case, $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \in (0, 1)$ predicts the likelihood of a segment to be a cry sound (values close to 1), or a different sound (values close to 0). The final binary classification $y \in \{0, 1\}$ (where 1 denotes a cry event) is obtained by comparing $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \in (0, 1)$ to a threshold value. In the training phase of the classifier, a gradient descent algorithm is used to find $\boldsymbol{\theta}$ that minimizes the (L_2) regularized

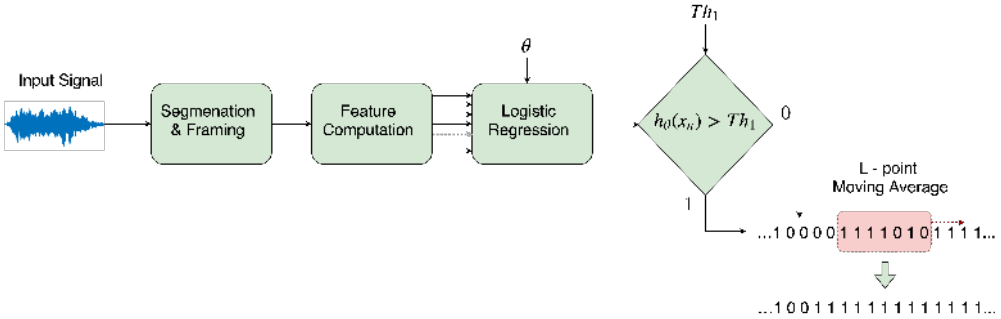


Figure 8. A schematic block diagram of the logistic regression algorithm.

cross-entropy cost function

$$\begin{aligned}\mathbb{E}(\boldsymbol{\theta}) = &-\frac{1}{n}\sum_{j=1}^{n} y^{(j)} \log\left(\frac{1}{1+\exp(-\boldsymbol{\theta}^T\boldsymbol{x}^{(j)})}\right) \\ &-\frac{1}{n}\sum_{j=1}^{n}\left(1-y^{(j)}\right)\log\left(\frac{\exp(-\boldsymbol{\theta}^T\boldsymbol{x}^{(j)})}{1+\exp(-\boldsymbol{\theta}^T\boldsymbol{x}^{(j)})}\right) \\ &+\frac{\lambda}{2n}\sum_{k=1}^{d}{\theta_k}^2,\end{aligned} \tag{10}$$

given a dataset of n labeled samples $\left\{\boldsymbol{x}^{(j)}, y^{(j)}\right\}_{j=1}^{n}$, where λ is a regularization parameter. The $\boldsymbol{\theta}$-minimizer found by the stochastic gradient descent algorithm is then assigned to (9) to classify new unlabeled samples.

A schematic block diagram of the logistic-regression-based algorithm is shown in Figure 8. The input data is divided into consecutive segments of 4096 samples. For each segment a 50-dimensional feature vector is computed. The trained regularized logistic regression is then applied to each feature vector, and the hypothesis function $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ is obtained, representing an estimation of the posterior probability $p(y|\boldsymbol{x})$, where $y \in \{0, 1\}$ is the sound event to be classified as cry or non-cry and $\boldsymbol{x}$ is the feature vector. Using a threshold value $\mathtt{Th}_1$, an initial decision value for each segment is set according to the following rule:

$$d(n) = \begin{cases} 1, & \text{if } h_{\boldsymbol{\theta}}(\boldsymbol{x}) > \mathtt{Th}_1 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The duration of a single segment is about 93ms, while most cry events are at least several hundred of milliseconds long. In order to avoid erroneous detection of sections that are too short to be a likely cry event, a smoothing operation is performed as follows: a sliding window of length L is applied to the initial sequence of decisions and the smoothed decision $d_s(n)$ for the central segment is updated according to the following rule:

$$d_s(n) = \begin{cases} 1, & \text{if } \sum_{k=-M}^{M} d(n-k) > \mathtt{Th}_2 \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

where L is odd, $M = (L-1)/2$ and $\texttt{Th}_2 \in [1, L]$ is a predefined threshold value.

3.3 Support Vector Machine

The Support Vector Machine (SVM) is a supervised large-margin classifier, which attempts to find a separating hyperplane between two classes of data, such that the margin between the hyperplane and the closest samples in either set is maximized. To train the SVM, we applied the Sequential Minimal Optimization (SMO) algorithm [86], which solves the following minimization problem:

$$\min_{\theta_0,\theta,\xi} \frac{1}{2} \|\theta\| + C \sum_{m=1}^{M} \xi_m \;\; s.t. \;\; x_m \geq 0, y_m(\theta^T x_m + \theta_0) \geq 1 - \xi_m, \;\; m = 1, 2, ..., M \tag{13}$$

where θ and θ_0 are parameters of the maximal margin hyperplane to be learned, x_m are the input data points (feature vectors), $y_m \in \{-1, 1\}$ are the output points (data labels - "cry" or "not cry"), ξ are slack variables that permit margin failure and C trades off a small number of margin failures and wide margin [86].

The trained SVM can be used in place of the logistic regression classifier in the block diagram of Figure 8. Performance comparison of the two classifiers is given in Table 2.

4 Performance Evaluation

4.1 Database

The database for this study consists of three hours of audio recordings (sampled at 44,100 Hz) of 0-6 month old babies in the Netherlands using off-the-shelf smartphones [87]. The babies were recorded continuously for several days in a domestic environment. The recordings were fully annotated, with about 50 different event types, such as crying, parents talking, door opening/closing, etc.

4.2 Training and test process

Our training corpus contained 14% of the labeled data, whereas the test corpus contained the remaining 86%. We trained our feed-forward CNN architectures using MATLAB and our RNN architectures (BiRNN and BiLSTM) with TensorFlow. We used Adam [88], which is an adaptive learning rate optimization algorithm, with an initial learning step of 0.00001. The gradient in each iteration was evaluated using mini-batches of 32 segments. Our loss function was cross-entropy loss. To avoid over-fitting, we applied L2 regularization to the weights, with a scale of 0.0001. The networks were trained over 20 epochs of the training data. The hardware used includes Intel Core i7-7700K 4.2GHz CPU and NVIDIA GTX 1080Ti 11GB GDDR5 GPU.

During testing (i.e., inference), we used the majority vote process depicted in Section 3.2 with $L = 17$. That is, a segment was classified as 'cry' if at least

8 other segments in a neighbourhood of 17 segments were classified as 'cry'. The measured inference latency (in software) for CNN 9.6M was smaller than 2.5ms per segment. In fact, this latency can be considerably reduced by using dedicated tools for deploying trained networks in hardware, such as TensorRT by NVIDIA or Deep Learning Deployment Kit by Intel.

4.3 Results

As performance metrics, we used two important measures known as the *detection rate* and the *false-positive rate*. The detection rate (also known as *sensitivity* or *recall*) is defined as the ratio between the number of true-positive events (TP), i.e. the number of cry events correctly identified, and the total number of cry events in the recording set (true positives TP and false negatives FN). The false-positive (or *false-alarm*) rate is defined as the ratio between the number of false positives (non-cry events identified erroneously as cry events (FP)) and the total number of non-cry events in the recording set (false positives (FP) and true negatives (TN)). The detection rate is therefore $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, and the false-positive rate is $\mathrm{FP}/(\mathrm{FP}+\mathrm{TN})$.

In the analysis of the cry-detection performance of different classifiers we focus on the trade-off between the false-positive rate and the detection rate. In particular, we are interested in the performance in the low false-positive rate regime, which is required in practice. The performance evaluation was carried out using receiver operating characteristic (ROC) curves.

First, we were interested in comparing the MFB and LFB representations (see Section 2.1), by comparing the ROC curves of our CNN 9.6M architecture for both representations. As shown in Figure 9, the MFB representation is superior to LFB. As discussed earlier, this is expected as the Mel-scale in MFB is better suited for detecting signals in the presence of noise with similar characteristics.

In our next experiment, we compared the performance of the feed-forward architectures described in Section 2.2. The results are presented in Figure 10. First, for all false-positive rates, our CNN 9.6M "tall" architecture outperforms all other architectures. This is most noticeable for false-positive rates below 2%. It is interesting to note that the CNN 270K "tall" network has very good performance, despite using far fewer parameters compared to the CNN 9.6M "tall" architecture. In addition, it has similar results to CNN 9.6M 3×3. That is, the use of "tall" filters is shown to be highly beneficial, even for networks with a small number of parameters.

We later compared our CNN 9.6M to the BiRNN and BiLSTM architectures described in Section 2.3. The results are shown in Figure 11. As evident by the result, the introduction of memory through the use of RNN has limited impact. The reason might be the short duration of cry events, which limits the benefit of architectures with memory. This perhaps hints as well that a more powerful CNN for feature extraction in addition to a longer training process are required for such architectures.

Finally, we compared our CNN 9.6M "tall" architecture to logistic regression and SVM. This is shown in Figure 12. We see that the deep learning approach outperforms the traditional classifiers, though the performance of the latter is

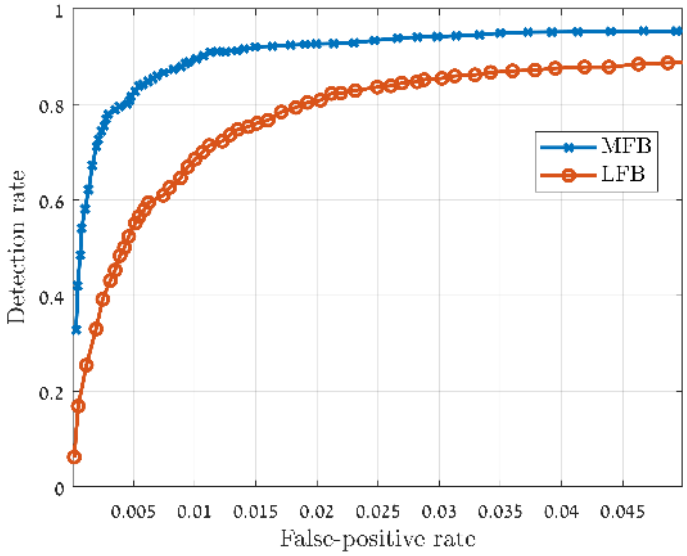


Figure 9. A comparison between MFB and LFB ROC curves for the CNN with 9.6M parameters.

quite good, and at very low false-positive regimes it even outperforms CNN architectures "weaker" than CNN 9.6M "tall".

The evaluation results are summarized in Table 2. For detection rates of 75%, 80%, 85% and 90% and 95%, the false-positive rates of the CNN 9.6M "tall" classifier are the lowest, compared to other deep-learning approaches as well to conventional machine learning algorithms such as the SVM or logistic regression.

5 Conclusion

In this study, we evaluated the performance of both deep learning and traditional approaches for baby cry detection. We investigated several CNN architectures, as well as recurrent neural networks (including LSTM) for better capturing temporal behaviour. We studied image representations of the input audio signals and found the log Mel-filter Bank (MFB) to be an appropriate representation. We demonstrated that by carefully choosing the kernel sizes and shapes in accordance to the MFB representation, better performance is achieved compared to common deep learning architectures. Our CNN classifier was shown to yield considerably better results compared to a traditional machine learning classifiers such as SVM or logistic regression, especially for low false-positive rates (which is highly required in practice). Our study demonstrates the power and advantages of deep learning when applied to audio event detection.

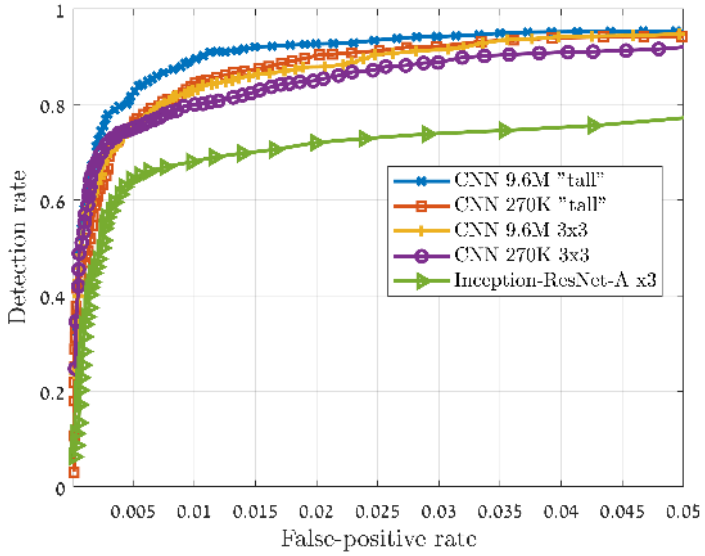


Figure 10. ROC curves of our feed-forward CNN architectures.

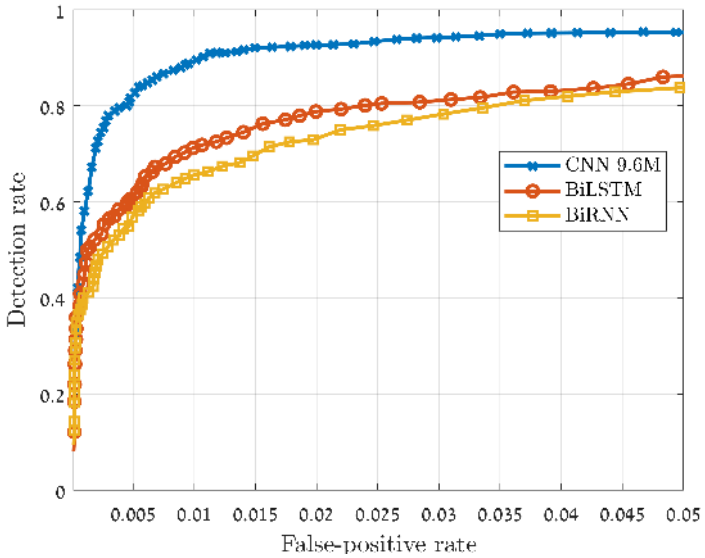


Figure 11. A comparison between CNN with 9.6M parameters, BiLSTM and BiRNN.

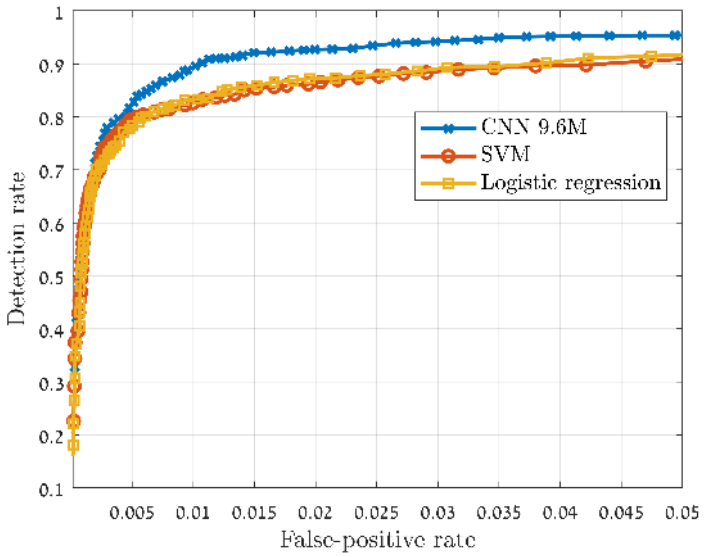


Figure 12. ROC curves for the CNN, SVM and logistic regression classifiers.

Classifier/Detection rate	75%	80%	85%	90%	95%
CNN 9.6M "tall"	0.25%	0.44%	0.66%	1.07%	3.50%
CNN 270K "tall"	0.47%	0.72%	0.95%	1.95%	6.30%
CNN 9.6M 3×3	0.51%	0.73%	1.30%	2.39%	5.41%
CNN 270K 3×3	0.48%	0.98%	2.01%	3.22%	7.8%
SVM	0.31%	0.52%	1.39%	4.13%	9.75%
Logistic Regression	0.38%	0.6%	1.23%	3.81%	9.80%

Table 2. A summary of the false-positive rates for a given detection rate.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[3] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 international joint conference on neural networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[4] J. Ramírez, J. M. Górriz, and J. C. Segura, "1 voice activity detection . fundamentals and speech recognition system robustness," 2007.

[5] D. Ruinskiy and Y. Lavner, "An effective algorithm for automatic detection and exact demarcation of breath sounds in speech and song signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 838–850, 2007.

[6] Y.-Y. Kong, A. Mullangi, and K. Kokkinakis, "Classification of fricative consonants for speech enhancement in hearing devices," in *PloS one*, 2014.

[7] A. Frid and Y. Lavner, "Spectral and textural features for automatic classification of fricatives," *XXII Annual Pacific Voice Conference (PVC)*, pp. 1–4, 2014.

[8] C. Panagiotakis and G. Tziritas, "A speech/music discriminator based on rms and zero-crossings," *IEEE Transactions on Multimedia*, vol. 7, pp. 155–166, 2005.

[9] Y. Lavner and D. Ruinskiy, "A decision-tree-based algorithm for speech/music classification and segmentation," *EURASIP J. Audio, Speech and Music Processing*, 2009.

[10] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002.

[11] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, May 2015.

[12] N. Morgan and H. Bourlard, "Continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 12, no. 3, pp. 24–42, May 1995.

[13] C. Aruna, A. D. Parameswari, M. Malini, and G. Gopu, "Voice recognition and touch screen control based wheel chair for paraplegic persons," in *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, March 2014, pp. 1–5.

[14] V. Carletti, P. Foggia, G. Percannella, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance using a bag of aural words classifier," in *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, Aug 2013, pp. 81–86.

[15] J. Ye, T. Kobayashi, and T. Higuchi, "Audio-based indoor health monitoring system using flac features," in *2010 International Conference on Emerging Security Technologies*, Sep. 2010, pp. 90–95.

[16] D. Kawano, T. Ogawa, and H. Matsumoto, "A proposal of the method to suppress a click noise only from an observed audio signal," in *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Nov 2017, pp. 93–96.

[17] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 559–563.

[18] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2015, pp. 1–6.

[19] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "Dcase 2016 acoustic scene classification using convolutional neural networks," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*. Tampere University of Technology. Department of Signal Processing, 9 2016.

[20] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, June 2017.

[21] G. Naithani, T. Barker, G. Parascandolo, L. Bramslw, N. H. Pontoppidan, and T. Virtanen, "Low latency sound source separation using convolutional recurrent neural networks," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2017, pp. 71–75.

[22] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6964–6968, 2014.

[23] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *ISMIR*, 2018.

[24] D. Ferretti, M. Severini, E. Principi, A. Cenci, and S. Squartini, "Infant cry detection in adverse acoustic environments by using deep neural networks," in *EUSIPCO*, September 2018.

[25] M. A. T. Turan and E. Erzin, "Monitoring infant's emotional cry in domestic environments using the capsule network architecture," in *Interspeech*, 2018.

[26] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *NIPS*, 2017.

[27] R. Torres, D. Battaglino, and L. Lepauloux, "Baby cry sound detection: A comparison of hand crafted features and deep learning approach," in *EANN*, 2017.

[28] J. Saraswathy, M. Hariharan, S. Yaacob, and W. Khairunizam, "Automatic classification of infant cry: A review," in *2012 International Conference on Biomedical Engineering (ICoBE)*, Feb 2012, pp. 543–548.

[29] Y. Lavner, R. Cohen, D. Ruinskiy, and H. IJzerman, "Baby cry detection in domestic environment using deep learning," *2016 International Conference on the Sceience of Electrical Engineering (ICSEE 2016)*, 11 2016.

[30] X. Zhang, Y. Zou, and Y. Liu, *AICDS: An Infant Crying Detection System Based on Lightweight Convolutional Neural Network*. Springer, June 2018, pp. 185–196.

[31] Y. Xu, M. Hasegawa-Johnson, and N. McElwain, "Infant emotional outbursts detection in infant-parent spoken interactions," in *Interspeech*, 2018.

[32] G. Silva and D. Wickramasinghe, "Infant cry detection system with automatic soothing and video monitoring functions," *Journal of Engineering and Technology of the Open University of Sri Lanka (JET-OUSL)*, vol. 5, no. 1, 2017. [Online]. Available: http://digital.lib.ou.ac.lk/docs/handle/701300122/1476

[33] J. Gao and L. Pabon, "Hot car baby detector," Illinois College of Engineering, Tech. Rep., December 2014.

[34] "Lollipop smart baby monitor," 2018. [Online]. Available: https://www.lollipop.camera/

[35] "Cocoon cam baby monitor," 2019. [Online]. Available: https://cocooncam.com/

[36] "Evoz wifi baby vision monitor," 2019. [Online]. Available: https://myevoz.com/

[37] G. Varallyay, "The melody of crying," *International Journal of Pediatric Otorhinolaryngology*, vol. 71, no. 11, pp. 1699–1708, Nov. 2007.

[38] A. Zabidi, L. Y. Khuan, W. Mansor, I. M. Yassin, and R. Sahak, "Classification of infant cries with asphyxia using multilayer perceptron neural network," in *Proceedings of the 2010 Second International Conference on Computer Engineering and Applications - Volume 01*, ser. ICCEA '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 204–208.

[39] S. Orlandi, C. A. Reyes-Garcia, A. Bandini, G. Donzelli, and C. Manfredi, "Application of pattern recognition techniques to the classification of full-term and preterm infant cry," *Journal of voice : official journal of the Voice Foundation*, vol. 30, 10 2015.

[40] K. Michelsson and O. Michelsson, "Phonation in the newborn, infant cry," *International Journal of Pediatric Otorhinolaryngology*, vol. 49, pp. S297 – S301, 1999.

[41] J. Bowlby, *Attachment and loss*. Basic Books, 1969, vol. 1.

[42] P. Ostwald, "The sounds of infancy," *Developmental Medicine & Child Neurology*, vol. 14, no. 3, pp. 350–361, 1972.

[43] D. Owings and D. Zeifman, "Human infant crying as an animal communication system: Insights from an assessment/management approach," *Evolution of Communication Systems: A Comparative Approach*, pp. 151–170, 01 2004.

[44] J. Nelson, *Seeing through tears: Crying and attachment*. Routledge, February 2005.

[45] H. IJzerman et al., "A theory of social thermoregulation in human primates," *Frontiers in Psychology*, vol. 6, no. 464, 2015.

[46] E. A. Butler and A. K. Randall, "Emotional coregulation in close relationships," *Emotion Review*, vol. 5, no. 2, pp. 202–210, 2013.

[47] L. L. LaGasse, A. R. Neal, and B. M. Lester, "Assessment of infant cry: Acoustic cry analysis and parental perception," *Mental Retardation and Developmental Disabilities Research Reviews*, vol. 11, no. 1, pp. 83–93, 2005.

[48] M. Hendriks, J. K. Nelson, R. Cornelius, and A. Vingerhoets, "Why crying improves our well-being: An attachment-theory perspective on the functions of adult crying," *Emotion Regulation: Conceptual and Clinical Issues*, pp. 87–96, 01 2008.

[49] P. Pal, A. N. Iyer, and R. E. Yantorno, "Emotion detection from infant facial expressions and cries," *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, pp. II–II, 2006.

[50] S. Barajas-Montiel and C. A. Reyes-Garcia, "Identifying pain and hunger in infant cry with classifiers ensembles," 12 2005, pp. 770 – 775.

[51] O. Wasz-Höckert, *The Infant cry: a spectrographic and auditory analysis*, ser. Clinics in developmental medicine. Spastics International Medical Publications in association with W. Heinemann Medical Books, 1968.

[52] A. Vingerhoets, *Why only humans weep. Unravelling the mysteries of tears*. Oxford University Press, 2013.

[53] S. M. Bell and M. D. Salter Ainsworth, "Infant crying and maternal responsiveness," *Child development*, vol. 43, pp. 1171–90, 01 1973.

[54] M. L. Lounsbury and J. E. Bates, "The cries of infants of differing levels of perceived temperamental difficultness: Acoustic properties and effects on listeners," *Child Development*, vol. 53, no. 3, pp. 677–686, 1982.

[55] P. Zeskind and R. Barr, "Acoustic characteristics of naturally occurring cries of infants with colic," *Child Development*, vol. 68, pp. 394 – 403, 06 1997.

[56] A. Laan, M. V. Assen, and A. Vingerhoets, "Individual differences in adult crying: the role of attachment styles," *Social Behavior and Personality An International Journal*, 2012.

[57] F. Bryant Furlow, "Human neonatal cry quality as an honest signal of fitness," *Evolution and Human Behavior*, vol. 18, pp. 175–193, 05 1997.

[58] Y. Kheddache and C. Tadj, "Acoustic measures of the cry characteristics of healthy newborns and newborns with pathologies," *Journal of Biomedical Science and Engineering*, vol. 06, no. 08, pp. 796–804, 2013.

[59] S. Orlandi, C. Manfredi, L. Bocchi, and M. L. Scattoni, "Automatic newborn cry analysis: A non-invasive tool to help autism early diagnosis," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2012, pp. 2953–2956.

[60] S. J. Sheinkopf, J. M. Iverson, M. L. Rinaldi, and B. M. Lester, "Atypical Cry Acoustics in 6-Month-Old Infants at Risk for Autism Spectrum Disorder," *Autism Research*, vol. 5, no. 5, pp. 331–339, 2012.

[61] S. Jeyaraman, H. M, K. Wan, S. Jeyaraman, T. Nadarajaw, S. Yaacob, and S. Nisha, "A review: survey on automatic infant cry analysis and classification," *Health and Technology*, vol. 8, 07 2018.

[62] H. IJzerman, M. Čolić, M. Hennecke, Y. Hong, C.-P. Hu, J. Joy-Gaba, D. Lazarevic, L. Lazarevic, M. Parzuchowski, K. G. Ratner, T. Schubert, A. Schuetz, D. Stojilovi, S. Weissgerber, J. Zickfeld, and S. Lindenberg, "Does distance from the equator predict self-control? : Lessons from the human penguin project," *Behavioral and Brain Sciences*, vol. 40, 01 2017.

[63] H. IJzerman, S. Lindenberg, I. Dalgar, S. Weissgerber, R. Clemente Vergara, A. Cairo, M. oli, P. Dursun, N. Frankowska, R. Hadi, C. Hall, Y. Hong, C.-P. Hu, J. Joy-Gaba, D. Lazarevic, L. Lazarevic, M. Parzuchowski, K. G. Ratner, D. Rothman, and J. Zickfeld, "The human penguin project: Climate, social integration, and core body temperature," *Collabra: Psychology*, vol. 4, no. 1, 2018.

[64] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science+Business Media, 2006.

[65] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[66] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back propagating errors," *Nature*, vol. 323, pp. 533–536, 10 1986.

[67] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, 2001.

[68] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016. [Online]. Available: http://arxiv.org/abs/1602.07261

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[70] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[71] H. Phan, P. Koch, F. Katzberg, M. Maaß, R. Mazur, and A. Mertins, "Audio scene classification with deep recurrent neural networks," in *INTERSPEECH*, 2017.

[72] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *CoRR*, vol. abs/1303.5778, 2013. [Online]. Available: http://arxiv.org/abs/1303.5778

[73] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov 1997.

[74] A. Graves, N. Jaitly, and A. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Dec 2013, pp. 273–278.

[75] T. Ben-Yehuda, I. Abramovich, and R. Cohen, "Low-complexity video classification using recurrent neural networks," *2018 International Conference on the Sceience of Electrical Engineering (ICSEE 2018)*, December 2018.

[76] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, November 1997.

[77] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 385.

[78] H. Fei and F. Tan, "Bidirectional grid long short-term memory (bigridlstm): A method to address context-sensitivity and vanishing gradient," *Algorithms*, vol. 11, 2018.

[79] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. PMLR, May 2010, pp. 249–256.

[80] R. Cohen and Y. Lavner, "Infant cry analysis and detection," *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel (IEEEI 2012)*, pp. 2–6, 2012.

[81] A. M. Noll, "Cepstrum pitch determination," *The Journal of the Acoustical Society of America*, vol. 41, no. 2, pp. 293–309, 1967.

[82] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: state of the art and future challenges," *Proc. IEEE*, vol. 99, no. 2, pp. 288–327, Feb. 2011.

[83] ——, "Comparative evaluation of howling detection criteria in notch-filter-based howling suppression," *J. Audio Eng. Soc.*, vol. 58, no. 11, pp. 923–940, Nov. 2010.

[84] L. R. Rabiner and R. W. Schafer, *Theory and applications of digital speech processing*. Pearson Upper Saddle River, NJ, 2011, vol. 64.

[85] T. Quatieri, *Discrete-time speech signal processing: principles and practice*. Prentice Hall, 2002.

[86] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.

[87] K. Frederiks, P. Sterkenburg, Y. Lavner, R. Cohen, D. Ruinskiy, W. Verbeke, and H. IJzerman, "Mobile social physiology as the future of relationship research and therapy: Presentation of the bio-app for bonding (bab)," *PsyArXiv*, 2018.

[88] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, December 2014.