Baby Cry Sound Detection: A Comparison of Hand Crafted Features and Deep Learning Approach

Rafael Torres, Daniele Battaglino, Ludovick Lepauloux

Baby cry sound detection: a comparison of hand crafted features and deep learning approach

Rafael Torres[1], Daniele Battaglino[1,2], and Ludovick Lepauloux[1]

[1] NXP Semiconductors,
Mougins, France
[2] EURECOM,
Biot, France

1 Abstract

Baby cry sound detection allows parents to be automatically alerted when their baby is crying. Current solutions in home environment ask for a client-server architecture where an end-node device streams the audio to a centralized server in charge of the detection. Even providing the best performances, these solutions raise power consumption and privacy issues. For these reasons, interest has recently grown in the community for methods which can run locally on battery-powered devices. This work presents a new set of features tailored to baby cry sound recognition, called hand crafted baby cry (HCBC) features. The proposed method is compared with a baseline using mel-frequency cepstrum coefficients (MFCCs) and a state-of-the-art convolutional neural network (CNN) system. HCBC features result to be on par with CNN, while requiring less computation effort and memory space at the cost of being application specific.

Keywords: baby cry detection, hand crafted baby cry features, support vector data description, convolutional neural networks

2 Introduction

Audio event detection (AED) has recently gained attention in the audio community [1–3]. AED is therefore pertinent to *smart-home* market where the presence of connected devices enables sounds to be detected through consumer microphones.

Thus, this work has focused on the detection of baby cry sounds specifically for home environment. This choice has been driven by the practical use case consisting of capturing the baby crying and automatically alerting his parents. AED based technologies are smarter than standard baby monitors or walkie talkies: in the former, monitoring is usually energy-based with the counterpart of being easily deceived by high energy sounds; in the latter, parents have to constantly listen to the receiver during their activities.

Existing approaches in baby cry detection literature consist of extracting meaningful features from audio signal frames. Most of them use spectral features

such as mel-frequency cepstrum coefficients (MFCCs), combined with binary classifiers such as support vector machines (SVMs) [4]. Recent researches have explored the use of convolutional neural networks (CNNs) tailored to baby cry detection [5].

Whereas showing the most promising results, CNN computation and memory requirements render it more compliant with a client-server solution, where an end-node client (i.e. low-power device equipped with a microphone) streams the audio to a central server in charge of the entire process. Some works in AED have started to question this client-server approach by focusing on battery-powered devices [6, 7] with a significant reduction of band-width and power. Moreover, moving the complexity towards the end-node has the advantage of respecting user privacy, since audio is analyzed locally in the device.

Thus, the need for an *always-active* baby cry detector calls for algorithm efficiency, essential to minimize battery consumption, and for classifier robustness, able to detect a baby cry sounds within a broad set of unknown conditions. As expressed in [8, 9], the detection in real conditions requires new types of classifiers more robust to unknown classes. In that sense the support vector data description (SVDD) is a good candidate for modeling baby cry features without being influenced by the number and the type of classes in the training set.

We herein present three methods for the baby cry detection task: the first baseline employs MFCCs and SVDD as a classifier; the second is based on CNN applied on mel-spectrogram; the third proposes a novel set of features tailored to baby cry detection.

The contributions of this work can be summarized in: i) hand crafted baby cry (HCBC) features; ii) the adoption of SVDD classifier for both MFCC and HCBC features; iii) improvements in terms of normalization and regularization of state-of-the-art CNN; iv) the comparison between hand-crafted features and deep learning approaches.

The remainder of the paper is organized as follows: Section 3 describes in details the three methods; Section 4 presents the experimental set-up, database description and results; conclusions and future works are discussed in Section 5.

3 Methods

This section describes the aforementioned methods for baby cry sound detection.

3.1 One-class classifier - SVDD

Whereas baby cry sounds can be easily collected and modeled, non-baby cry samples are more difficult to identify and categorize. In domestic environment, many sounds may resemble a baby cry sound provoking false alarms during the detection. Standard binary classifiers may fail to learn both baby cry (*target*) and non-baby cry (*non-target*) samples when these latter represent a subset of those encountered during testing. This problem has been identified as *open-set* and specific classifiers have therefore been employed [8, 9].

Instead of separating *target* from *non-target*, SVDD models only target samples with a hypersphere [10]. Once the radius R and the center of the hypersphere a have been found during training, the decision function f to determine if a new sample $\boldsymbol{z}$ belongs to the target class depends upon R and a:

$$f(\boldsymbol{z}, R, a) = sign(R^2 - ||\boldsymbol{z} - a||^2). \tag{1}$$

Drawing upon the SVM theory, SVDD finds the support vectors (SVs) using Lagrangian procedure to optimize a and R. These SVs are training samples which are selected to represent the boundary between the two classes.

Differently from prior works on SVDD, the grid-search for finding the best classifier parameters has been modified. This routine points at the presence of non-target samples in the training set to automatically select the best pair of parameters by minimizing the following function:

$$\lambda = \sqrt{(\frac{\#SVs}{T})^2 + (1 - \text{AUC})^2} \tag{2}$$

where $\#SVs$ corresponds to the number of SVs, T is the cardinality of the target class samples and AUC stands for the area under the receiver operating characteristic curve, used to evaluate the performance of SVDD on a validation set made of target and non-target samples. The function in Eq. 2 defines a trade-off between an estimation of the classifier complexity ($\frac{\#SVs}{T}$ term) and the global performance of the system expressed with AUC metric.

3.2 Baseline system - MFCC

A baby cry sound is generated by an excitation of the vocal cords producing a sequence of periodic impulses. In healthy babies the fundamental frequency ($F0$) reaches values between 250Hz - 600Hz, which has a higher range than that of adult females and males. Hence, a higher $F0$ characterizes most of the baby cry sounds, as depicted in Fig. 1.

Given these spectral properties, MFCCs have been proven in the literature [16] to be a good candidate for baby cry detection task since it represents each signal as rate of change in the frequency bands. Due to baby cry harmonics, MFCCs represent this information with higher MFCCs coefficients. This phenomenon is observable in Fig. 1. As a consequence of representing the entire spectral information, MFCCs are very dependent on overlapping sounds or additional noise which are mixed with the baby cry signal.

3.3 State-of-art system - CNN

Although primarily designed for image classification tasks, CNNs have proven to be successful in speech and music recognition [11]. Recently, these techniques have attracted interest also in AED [12] showing promising results. Inspired by

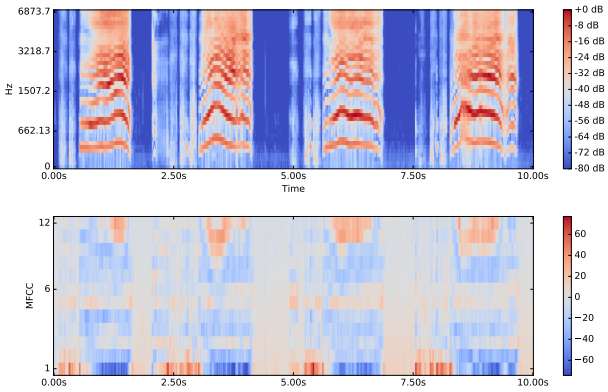

Fig. 1. The log-mel power spectrum on the top and the MFCC without the first coefficient C0 on the bottom for a sequence of baby cry sounds. The sound produced by the baby creates periodic harmonics in the log-mel power spectrogram that are captured in the MFCCs domain. In this example, each harmonic sequence is preceded by an unvoiced breath of the baby which produces noisy-like sounds in the lower frequency. This phenomenon is represented by a peak in the first MFCCs coefficients, while baby cry is captured by higher coefficients.

the architecture in [5], CNN performances are enhanced by introducing normalization and regularization layers. These modifications report a better generalization and they are even applicable on datasets of modest size.

A standard CNN is a deep architecture of successive layers, which are connected in different ways from the input data until the output layer. The global architecture is shown in Fig. 2. Differently from MFCCs which decorrelate the data with the discrete cosine transform (DCT), CNN takes as input the log mel-filtered spectrogram mimicking an image processing behavior. In the convolutional layer, each hidden unit is not connected to all the inputs from previous layer, but only to an area of the original input space, capturing local correlation. These small parts of the whole input space are connected to the hidden units through the weights. This operation is equivalent to a *convolutional filter* processing.

The pooling consists of merging close units according to some criteria (such as mean or max). This effectively performs a *downsampling* which smooths the resulting outputs of each convolutional layer, making the system more robust to small variations or translations. In the case of spectrograms as input, these local variations have to be attenuated in order to better recognize global patterns.

The main differences between the proposed CNN and the one in [5] are listed below:

1. convolutional filters of the first layer are 10×10 blocks, to capture both frequency and temporal resolution;
2. regularization techniques avoid overfitting of the network on a relatively small dataset. One of the most adopted is the so called *dropout*, which con-

sists of literally dropping out hidden units with a certain probability. At each training iteration, a random subset of hidden units is temporary disabled by multiplying the input to these units by 0. This forces the network to find robust features that do not depend on the presence of particular other neurons [13];

3. scaling inputs to zero mean and unit standard deviation is a common preprocessing step to uniform values across heterogeneous features. When they pass through a deep architecture, data progressively loose this normalization resulting in too big or too small values. Instead of computing the normalization only on the input data, the batch normalization is applied to the hidden layers so to avoid this effect [14].

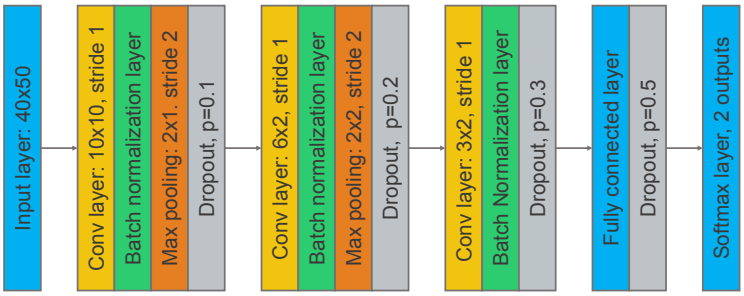

Fig. 2. The global architecture of the proposed CNN. More details of each layer are provided in Subs. 4.3.

3.4 Proposed approach - HCBC with SVDD classifier

As explained in Subs. 3.2 and Fig. 1, baby cry sounds are characterized by specific acoustic properties. Whereas the spectral content can be represented with MFCCs, there exists no standard features exploiting voiced-unvoiced recurrence in a baby cry signal. In the proposed algorithm, hand crafted features are specifically designed to characterize these temporal patterns.

HCBC features consist of frame-based descriptors which are then aggregated over longer audio-clips. For each frame, the fundamental frequency $\hat{F}0$ is estimated using an autocorrelation method. These features are composed of *voiced unvoiced counter*, *consecutive* $F0$ and *harmonic ratio accumulation* which create a 3-D feature vector used by SVDD classifier to model the target baby-cry class. The explanation of each feature is described below:

Voiced unvoiced counter (VUVC) counts all frames having a significant periodic content. This is obtained by looking at the harmonic strength, called $R0$, defined in Boersma's work [15]. For each frame, the local maximum of the frame normalized autocorrelation $R0$ must be greater than a predefined threshold t_{vuvc}.

Consecutive $F0$ (CF0) acts as an accumulator, which tracks the temporal continuity of the estimated $\hat{F0}$. Let us define $Fref$ as the most occurring $\hat{F0}$ learned from the training set (see Fig. 4). First, the distance between $\hat{F0}$ of each frame and $Fref$ is calculated. As long as this distance is smaller than a tolerance parameter ϵ, a score is computed and accumulated in $CF0$ with a weight that follows a square law. The longer a sequence of consecutive $\hat{F0}$, the greater the weight is. The corresponding method is given in Algorithm 1.

input : An array of $\hat{F0}_{i=1\cdots M}$ for a given audio-clip
Frequency sampling Fs
Tolerance parameter ϵ
Reference fundamental frequency $Fref$
output: $CF0$
begin
  $CF0 = 0$
  $counter = 1$
  for $i = 1$ **to** M **do**
    if $|\hat{F0}_i - Fref| < \epsilon$ **then**
      $score = \frac{Fs - |\hat{F0}_i - Fref|}{Fs}$
      $CF0 = CF0 + sqrt(counter) \times score$
      $counter = counter + 1$
    else
      $CF0$ not updated and $counter = 1$
    end
  end
end

Algorithm 1: Consecutive $F0$ ($CF0$) algorithm.

Harmonic ratio accumulation (HRA) is defined similarly to [16] as the ratio between the energy in harmonics and the overall frame energy. Let us define x as the microphone signal, X its discrete N points Fourier transform and n_i the closest bin of the i^{th} harmonic with i from 1 to Ny the last harmonic before Nyquist frequency. Note that the first harmonic n_1 corresponds to $2 \times F0$. For a given frame, the harmonic ratio is defined as follows:

$$HR = \frac{\sum_{i=1}^{Ny} |X[n_i]|^2}{\sum_{j=1}^{N} |X[j]|^2}. \tag{3}$$

Thus, HRA of a given audio-clip is the sum of all its frames harmonic ratios. Considering M the number of frames in an audio-clip:

$$HRA = \sum_{1}^{M} HR. \tag{4}$$

The three presented features (VUVC, CF0 and HRA) capture correlated, mono-tonal and harmonic patterns. A baby cry sound has a specific pitch, dura-

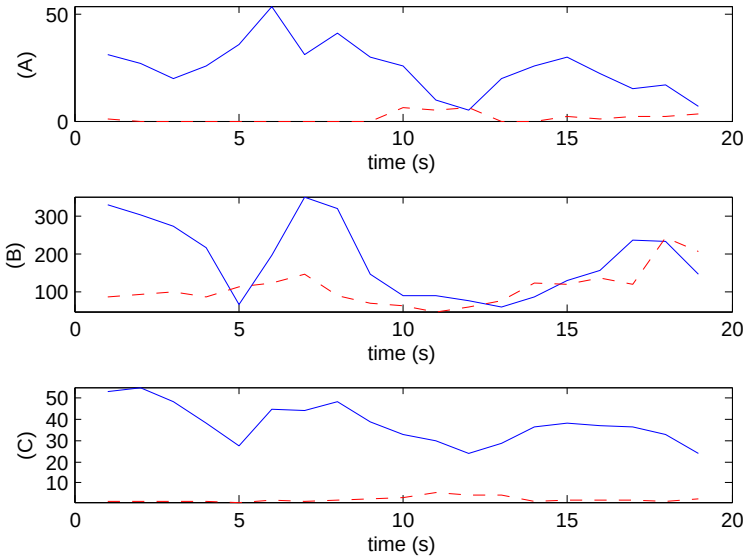

Fig. 3. Example of HCBC features behavior for a baby cry sample in solid blue line and a male cry sample in red dashed line with (A) voiced unvoiced counter, (B) consecutive F0 and (C) harmonic ratio accumulation.

tion and spectral distribution that requires fine tuning for optimal class differentiation as illustrated in Fig. 3. It must be also emphasized that HCBC features are energy independent. Differently from previous works, HCBC are self-content low-dimensional descriptors that are not concatenated with other features (such as MFCC) resulting in a better memory and computation efficiency.

4 Experimental results

This section describes the datasets used for training together with testing protocols, metrics, implementation details and results.

4.1 Dataset

Due to its recent development in AED field, no public standardized dataset has yet been released for baby cry sound detection task.

However, collaborative databases may offer a good alternative. One major advantage is the diversity of the signals in terms of device audio path and signal to noise ratio (SNR) levels. This heterogeneity covers many possible scenarios with a more robust *on-field* evaluation.

The database employed in this work comes from a set of available on-line resources[3]. For training, it includes 102 baby cry sound events (1h07m) and 93

[3] `http://www.audiomicro.com`
`https://www.freesound.org`

non-baby cry (1h24m) i.e tv, toy, adult cry, baby talk/play, music, fan, vacuum cleaner which are used for modeling target and non-target class.

The testing set is composed of 10 files that are created by mixing 26 baby cry events (0h16m) separated from each other by 30s with 10 different 5m looped background recordings at a SNR level of 18dB. These background recordings are repeated to avoid a significant noise variation along the baby cry sequence that may unfairly affect the detection.

Hence, the testing set is composed of a 4h recordings which sparsely contain target sounds and additional 2h of whole non-target sounds. This non-target noise library is made up of home environment recordings from CHiMe-Home (more details are available in [17]). All signals are 16kHz mono wave files.

4.2 Evaluation protocols & metrics

The three methods output a continuous score every second. For this type of application, it has been identified as a good trade-off between detection rate, computation cost and latency. According to that, the groundtruth of each file has been manually annotated.

Performance is evaluated by a receiver operating characteristic (ROC) curve and a prediction-recall (PR) curve. These two curves provide complementary information: the ROC curve presents how the number of correctly classified samples varies with the number of negative incorrectly classified samples. Positive and negative samples are however separately counted and normalized.

When the absolute number of positive samples (i.e. baby cry) is significantly less than the possible number of negative samples (i.e. non-baby cry), ROC curve may give a too optimistic view of the algorithm performance. The precision of the PR curve, instead, directly compares absolute number of positive and negative samples. In the case of an highly unbalanced set, the precision will be affected by the number of false positive providing a view closer to real performances [18].

4.3 Implementation details

Baseline MFCC features are extracted from a frame length of 32ms overlapped by 16ms. The filter-bank is built of 40 Mel-scaled filters up to 8kHz, resulting in 13 MFCCs for each frame. The mean and standard deviation are then computed over the 3s audio-clip overlapped by 2s, resulting in a 24 dimensional feature vector (without C0). For the remaining parameters, the default ones of *rastamat* library[4] have been selected.

The SVDD classifier in the implementation of *libsvm* [19] uses the radial basis function (RBF) kernel while the best pair of parameters C, γ are selected by minimizing the function in Eq. 2 on the validation set. In our experiments, in order to adjust classifier and features parameters, a validation set is randomly selected

`https://www.pond5.com`
`https://www.soundsnap.com`

[4] `http://www.ee.columbia.edu/ln/rosa/matlab/rastamat/`

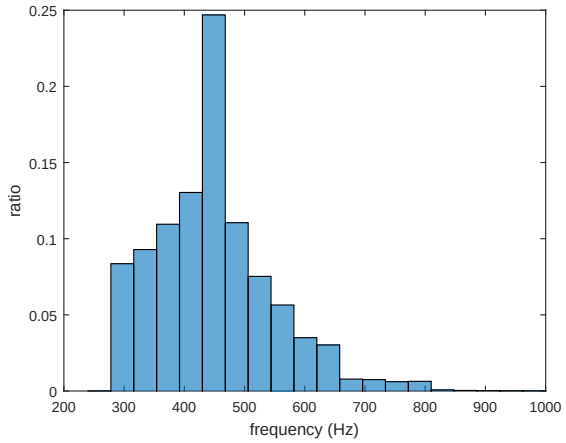

Fig. 4. Normalized histogram of the baby cry frame $\hat{F0}$ distribution of the training database with a bar width of 40Hz. The maximum is reached at 430-468Hz with a ratio of 24.7%. This is previously referenced as $Fref$.

from a 30% of the training set. Once the parameters have been estimated, the final model is trained using the entire training set.

The same strategy is then used to choose the CNN parameters and architecture, based on the lowest classification error on the validation set. Also in this case, the final model is then trained using the whole training set with a network in the order to 1 million trainable parameters. CNN structure is depicted in Fig. 2. The input layer takes a log mel-filtered spectrograms of 40 filters and 50 frames (corresponding to 1s). The activation function is the standard rectifier (ReLU), except for the last layer where the *softmax* returns the probability for the two outputs (baby cry or non-baby cry). During the training phase, a stochastic gradient descent is evaluated over a mini-batch of 256 shuffled input spectrograms. The random shuffling of the inputs is important to represent the data not in their temporal order. The momentum is set to 0.9, over 50 training iterations with a learning rate of 0.001.

Details of filters are displayed in Fig. 2. The number of filters is set to 32. The stride term refers to the amount by which each convolutional filter shifts horizontally and vertically. There are 4 dropout layers in this architecture, with an increasing probability of dropping off units (from 0.1 to 0.5). This prevents to prune too many units in the first layers where features are being built. Thus an higher dropout probability is applied to the fully connected layer, particularly prone to overfit.

The library employed to implement the CNN is the *lasagne*[5] library, a wrapper of *Theano*[6]. Experiments have been run on a Nvidia Quadro M4000 GPU.

[5] `http://lasagne.readthedocs.io/en/latest/`
[6] `http://deeplearning.net/software/theano/`

The frame length used in HCBC features is 32ms overlapped by 16ms. All frames are then aggregated over bigger audio-clips of 3s with an overlap of 2s. For each frame, $\hat{F}0$ is computed in a restricted range of 250Hz-1000Hz. VUVC threshold is set at tvuvc = 0.85. CF0 standard fundamental frequency $Fref$ is estimated at 449Hz with a tolerance $\epsilon = 20Hz$. These values are set during training phase based on the overall $\hat{F}0$ distribution (see Fig. 4) and validation results.

4.4 Results

Reported in Fig. 5 are experimental results for the three methods: MFCC in dotted black line, CNN in dashed red line and HCBC in solid blue line. The area under the curve (AUC) metric is also reported in the legend of the ROC curve.

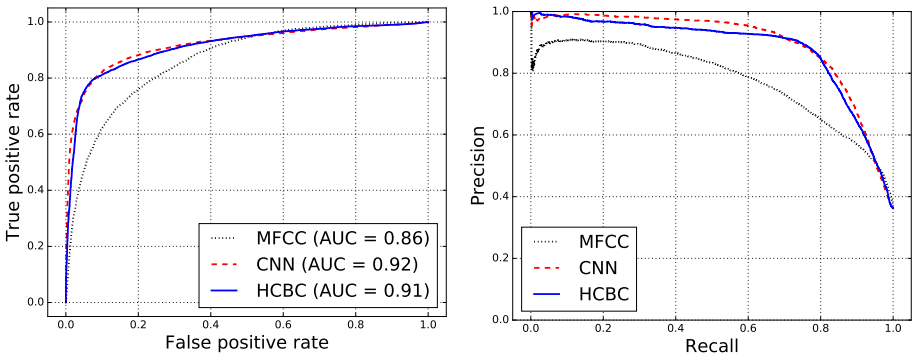

Fig. 5. The ROC on the left and the precision-recall curve on the right.

Concerning the ROC curve, CNN and HCBC outperform the baseline MFCC system, passing from an AUC of 86% to more than 90%. CNN classifier is slightly better than HCBC, with 1% improvement in AUC. The precision-recall curve is more related to the type of application, where we can choose the trade-off between an high precision (therefore less false positive) or an higher recall (with less false negative).

Let us consider that a baby statistically cries 2h per day. An acceptable metric for a baby cry detector would be a recall of 80% with a maximum of 5m per week of false positives. These numbers result in a precision of 99,3% and a recall of 80%. CNN and HCBC are the closest algorithms to this requirement, while the baseline MFCC has a drop of 20% in precision compared to them.

Albeit showing similar performance, systems must be compared also in terms of computational and memory cost. From a feature computation point of view, not using the mel spectrogram as for MFCC or CNN is a clear advantage. Referring to [20], HCBC features employ only FFT and autocorrelation as basic processing, resulting in 20 times lower computational cost than standard

MFCCs with no additional memory cost. From a classifier point of view, it has been demonstrated in [21] that CNNs may reach the highest performance at the expense of high computation complexity. This mainly prohibits their use on low-power devices. Finally, the proposed approach shows advantage over the 24 dimensional MFCC features used in the baseline system: knowing that complexity of SVDD is proportional to the number of SVs and feature vector dimensionality [22], HCBC outperforms the baseline system with only 3 dimensional feature vector.

5 Conclusions

In this work three methods were proposed for detecting baby crying in every day domestic environment: a baseline based of MFCC and SVDD classifier; a state-of-the-art CNN system and a new set of features specifically designed for this task. These 3 dimensional features capture repetition of voice-unvoiced pattern during time and therefore outperform the MFCC baseline, reaching the same level of performance of CNN. CNN is able to automatically extract meaningful patterns from log mel-filtered spectrograms, achieving the best results. Nevertheless, depending on the computational and memory resources available, the choice of CNN may not be compatible with low-power devices. The proposed method HCBC has the same level of performance and it is less computational and memory demanding. The drawback of this approach is to be suited uniquely for baby cry sound detection, with an high cost in designing specific baby cry features.

Further research should investigate ways of reducing complexity of CNN, by decreasing the number of filters and their size. Another track may encode the temporal patterns directly in the deep learning architecture.

References

1. Mesaros, A., Heittola, T., Virtanen, T.: TUT database for acoustic scene classification and sound event detection. In: 24th European Signal Processing Conference (EUSIPCO). (2016) 1128–1132
2. Barchiesi, D., Giannoulis, D., Stowell, D., Plumbley, M.: Acoustic Scene Classification: Classifying environments from the sounds they produce. IEEE Signal Processing Magazine **32**(3) (May 2015) 16–34
3. Ntalampiras, S.: Audio pattern recognition of baby crying sound events. J. Audio Eng. Soc **63**(5) (2015) 358–369
4. Saraswathy, J., Hariharan, M., Yaacob, S., Khairunizam, W.: Automatic classification of infant cry: A review. In: International Conference on Biomedical Engineering (ICoBE). (Feb 2012) 543–548
5. Lavner, Y., Cohen, R., Ruinskiy, D., Ijzerman, H.: Baby cry detection in domestic environment using deep learning. In: IEEE International Conference on the Science of Electrical Engineering (ICSEE). (Nov 2016) 1–5
6. Saha, B., Purkait, P.K., Mukherjee, J., Majumdar, A.K., Majumdar, B., Singh, A.K.: An embedded system for automatic classification of neonatal cry. In: IEEE Point-of-Care Healthcare Technologies (PHT). (Jan 2013) 248–251

7. Bnic, I.A., Cucu, H., Buzo, A., Burileanu, D., Burileanu, C.: Baby cry recognition in real-world conditions. In: 39th International Conference on Telecommunications and Signal Processing (TSP). (June 2016) 315–318
8. Battaglino, D., Lepauloux, L., Evans, N.: The open-set problem in acoustic scene classification. In: IEEE International Workshop on Acoustic Signal Enhancement (IWAENC). (Sept 2016) 1–5
9. Rabaoui, A., Davy, M., Rossignol, S., Lachiri, Z., Ellouze, N.: Improved one-class svm classifier for sounds classification. In: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS). (2007) 117–122
10. Tax, D.M.J., Duin, R.P.W.: Data domain description using support vectors. In: European Symposium on Artificial Neural Networks. (1999) 251–256
11. Deng, L., Yu, D.: Deep learning: Methods and applications. Foundations and Trends in Signal Processing **7**(34) (2014) 197–387
12. Piczak, K.J.: Environmental sound classification with convolutional neural networks. In: IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP). (Sept 2015) 1–6
13. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580 (2012)
14. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: 32nd International Conference on Machine Learning, ICML. (2015) 448–456
15. Boersma, P.: Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In: IFA Proceedings 17. (1993) 97–110
16. Cohen, R., Lavner, Y.: Infant cry analysis and detection. In: IEEE 27th Convention of Electrical and Electronics Engineers. (Nov 2012) 1–5
17. Foster, P., Sigtia, S., Krstulovic, S., Barker, J., Plumbley, M.D.: Chime-home: A dataset for sound source recognition in a domestic environment. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). (2015) 1–5
18. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS One **10**(3) (2015) e0118432
19. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27 Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
20. Wang, J.C., Wang, J.F., Weng, Y.S.: Chip design of mfcc extraction for speech recognition. Integr. VLSI J. **32**(1-3) (November 2002) 111–131
21. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2016) 4820–4828
22. Sigtia, S., Stark, A.M., Krstulovi, S., Plumbley, M.D.: Automatic environmental sound recognition: Performance versus computational cost. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**(11) (Nov 2016) 2096–2107