A Neural Network Approach for Automatic Detection of Acoustic Alarms

Alex Peiró Lilja, Ganna Raboshchuk and Climent Nadeu
TALP Research Center, Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, C. Jordi Girona, 1-3, 08034, Barcelona, Spain

Keywords: Acoustic Event Detection, Alarm Detection, Neural Network, Neonatal Intensive Care Unit.

Abstract: Acoustic alarms generated by biomedical equipment are relevant sounds in the noisy Neonatal Intensive Care Unit (NICU) environment both because of their high frequency of occurrence and their possible negative effects on the neurodevelopment of preterm newborns. This work addresses the detection of specific alarms in that difficult environment by using neural network structures. Specifically, both generic and class-specific input models are proposed. The first one does not take advantage of any specific knowledge about alarm classes, while the second one exploits the information about the alarm-specific frequency sub-bands. Two types of partially connected layers were designed to deal with the input information in frequency and in time and reduce the network complexity. The time context was also considered by performing experiments with long short-term memory networks. The database used in this work was acquired in a real-world NICU environment. The reported results show an improvement of more than 9% in absolute value for the generic input model and more than 12% for the class-specific input model, when both consider time information using the proposed partially connected layer.

1 INTRODUCTION

The acoustic environment of a Neonatal Intensive Care Unit (NICU) is very noisy and contains a high diversity of sounds that happen spontaneously and often overlap in time. The possibility of noxious effects of the noisy NICU environment on preterm newborns has been well documented and is of great concern in the medical literature (Wachman and Lahav, 2010). Acoustic alarms, which are frequently generated by various biomedical equipment, are among the most relevant types of sounds in the NICU environment and may negatively affect the neurodevelopment of the preterm infants. Besides, the excess of alarms that do not have clinical relevance can cause an alarm fatigue to the medical staff and affect the quality of healthcare provided by them (Freudenthal et al., 2013). Therefore, automatic detection of acoustic alarms in the NICU can be useful to: 1) study relations between the detected alarms and the infant health; and 2) assist the medical staff in their work.

So far, the problem of general acoustic alarm detection has not been much explored. In general, it was investigated for hearing impaired assistance or hearing support in noisy conditions (Beritelli et al., 2006; Carbonneau et al., 2013). To the best of our knowledge, the first work on automatic alarm sounds detection is reported in (Ellis, 2001), where an approach borrowed from speech recognition and an approach based on sinusoid modelling and separation were compared. Later works on acoustic alarm detection usually make use of particular properties of alarms as: detection of amplitude periodicity in a specified frequency bandwidth (Carbonneau et al., 2013); pitch detection in a predefined frequency range (Meucci et al., 2008); spectral- and time-domain properties extracted with morphological features (Xiao et al., 2009); explotation of the long-term periodicity with the autocorrelation function (Lutfi and Heo, 2012).

A system for acoustic alarm detection in a NICU environment was first presented in (Raboshchuk et al., 2014), for the task of binary alarm detection; and also an acoustic description of that environment was provided. The proposed system included a denoising pre-processinng step, employed generic features that cover the whole frequency bandwidth and pre-trained neural networks. In (Raboshchuk et al., 2015), the problem of detecting specific NICU acoustic alarm classes was addressed. The system consisted of a set of binary Gaussian mixture modelling based detectors (one per each alarm class). The knowledge about the spectro-temporal alarm characteristics was exploited. First, by using sinusoid detection around alarm-specific frequencies for feature extraction; and

PeirÃş Lilja A., Raboshchuk G. and Nadeu C.
A Neural Network Approach for Automatic Detection of Acoustic Alarms.
DOI: 10.5220/0006167200840091
In *Proceedings of the 10th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2017)*, pages 84-91
ISBN: 978-989-758-212-7

second, by including a post-processing step that takes into account the temporal structure of alarms.

An approach based on Neural Networks (NN) is proposed in this paper to detect alarm sounds in a real-world NICU hospital environment. The NNs described in (Ellis, 2001; Beritelli et al., 2006) have a conventional topology and use extracted features at the input (perceptual linear predictive cepstral coefficients in (Ellis, 2001) and mel-scale cepstral coefficients in (Beritelli et al., 2006)). Conversely, in this work, a simple magnitude spectral representation is used at the input, but a more elaborated NN structure is explored, though, due to the small size of the dataset, deep topologies could not be considered. Two types of partially connected hidden layers with limited weight sharing are implemented to reduce the number of network parameters to train. Those layers are employed in the networks to extract local information in either time or frequency, so we call them, respectively, Frequency Weighting (FW) and Temporal Weighting (TW). Two different kind of models are proposed: a Generic Input Model (GIM) and a Class-specific Input Model (CIM). GIM does not employ any knowledge about the particular alarm class, while CIM takes advantage of the knowledge about alarm-specific frequency components. Both models were studied and compared through the presented results.

The NN-based techniques which have been developed and compared in this work operate at the frame level. In order to evaluate the systems also in terms of event detection, the sequence of frame-level classification decisions is processed with a simple smoothing technique.

The rest of the paper is organized as follows. Section 2 contains the description of the database and of the acoustic alarms. In Section 3, the system development is presented, including the description of input representation, pooling techniques and partially connected hidden layers, and the description of the two input models proposed. Finally, in Section 4 the evaluation setup and the experimental results are discussed.

2 DATA DESCRIPTION

The database used in this work contains audio acquired during ten recording sessions in the NICU of Hospital Sant Joan de Déu Barcelona (108.7 minutes in total). Two electret unidirectional microphones connected to a linear PCM Recorder were used to make recordings. One microphone was placed inside the incubator, close to the infant's ear, and the other one outside the incubator, usually pointing to the center of the room. But only the recordings obtained from the microphone placed outside are used in this work. The manual annotations cover 54.3 minutes of this data, where 19.28% of time is labelled as alarms. Note that each alarm signal was annotated separately (see Figure 1).

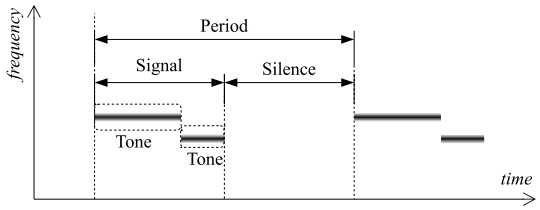


Figure 1: Typical structure of an alarm sound.

A large diversity of sounds can be found in a typical NICU environment. In the acquired database 16 different alarm classes were observed, which were generated by various types of biomedical equipment, such as cardiorespiratory, monitors, ventilators, infusion pumps, incubators, etc. Only 7 most populated alarm classes, which were also relevant from the medical point of view, are selected in this work.

Alarm sounds are periodic in time, and each period is defined by a signal and silence intervals of specific durations (see Figure 1). The signal interval contains one or several consecutive tones, which are stationary. Each tone is defined by one or several simultaneous frequency components harmonically related or not. Several alarm classes show some variations in the frequency and duration values, and such cases are referred to as different versions of the alarm class. The specific spectro-temporal properties of the selected alarm classes are presented in Table 1.

Most of the time, alarm sounds occur together with other types of sounds. However, the temporal overlaps between alarms are not rare either, and in the annotated data two or more alarms sound simultaneously almost 8% of time.

3 SYSTEM DEVELOPMENT

3.1 Input Representation

The recorded audio was downsampled from 44.1 to 24 kHz, thus, the observed frequency range is up to 12 kHz. For processing, the frame length was set to 2048 points with 50% of overlap. Then, each spectral frame length is 1024 points, which corresponds to a resolution of 11.78 Hz per spectral point. In this work, a spectrum representation rather than an extracted set of features is used directly as the input data to let the network learn the feature representation by

Table 1: Description of the chosen alarm classes. After "alarm-classes": comma-separated frequencies are simultaneous in time; information separated with brackets corresponds to a different version of the alarm; frequencies separated with slash correspond to consecutive tones.

Class	Period duration (s)	Frequency components (kHz)	Samples	Frames
a1	2.050 [2.246]	0.495, 1.465, 2.435 [0.515, 2.455, 3.445, 4.415]	238	3935
a3	15.300	0.665, 1.330, 1.990, 2.660 / 0.540, 1.60, 3.150 [0.520 / 0.420]	130	1928
a6	0.447	2.410	203	1785
a7	1.015	0.980, 2.935 [2.880]	114	2239
a8	2.245	0.490, 1.480, 2.460, 3.440, 4.420	452	2956
a10	1.000	1.140, 2.280, 3.425 [0.880]	75	1186
a16	2.053	0.495	135	969

itself. This was inspired by recent works reported in (Sainath et al., 2013; Abdel-Hamid et al., 2013).

3.2 Pooling Techniques for Input Pre-processing

Several pooling techniques are implemented to reduce the input size while preserving the relevant information. Average Pooling (AP) technique averages the small portion of frequency sub-band values. Then two non-linear pooling techniques based on maximum are proposed, where the goal is to preserve high spectral peaks with their real values: Standard Max Pooling (SMP) and Mel-Scale Max Pooling (MSMP). SMP technique has the same structure as AP, but it uses maximum rather than averaging. MSMP, on the other hand, looks for the maximum value following the mel-scale filter bank distribution and provides a more natural representation according to the human ear perception.

3.3 Model Structures

3.3.1 Generic Input Model

The aim of using this kind of model is to have a network structure and a set of features that do not depend on the alarm properties, so they have not to be changed when the model for a new alarm has to be trained. Figure 2(a) shows the general scheme of this model. The input-vector is the whole spectral frame representation (magnitude DFT of the signal frame). First, the spectral values are pre-processed to reduce the size of the representation of the spectrum, while keeping the relevant information. Then, logarithm and Mean-Variance Normalization (MVN) are used to improve the data distribution before it is fed to the classifier, which is either a feedforward NN or a long short-term memory NN.

3.3.2 Class-specific Input Model

For this model the particular spectral and temporal properties of alarms are assumed known. In fact, only the spectral information is used, since the inclusion of the temporal information (i.e. signal and silence interval duration) requires that many more network weights are trained, what is not feasible. The spectral information is exploited at the input of the network (see Figure 2(b)), where the input features are only the spectral values at the alarm-specific frequency bins (f^0, f^1, f^2, f^3) and the bins around them: Left Neighbors (LN) and Right Neighbors (RN). Logarithm and MVN are further applied to the input data. Obviously, neither pooling strategies nor partially connected layers for frequency weighting are used by this model.

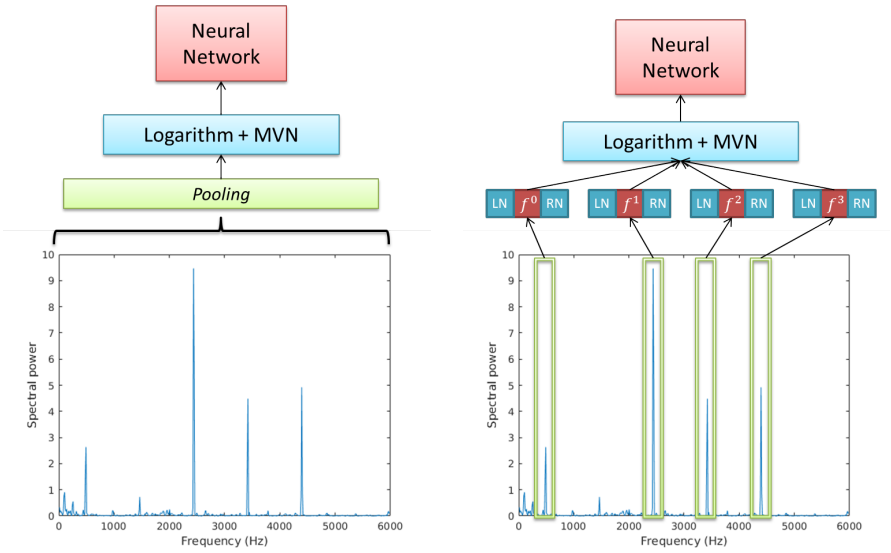
3.4 Partially Connected Layers

3.4.1 Frequency Weighting

For model size reduction we first propose to use partial connections as a weighted average of a portion of spectral bins, as shown in Figure 3(a), where F is the number of input spectral bins of frame S_t. As shown in the figure, the upper layer has a size that is the quocient between the input size and the pooling size.

3.4.2 Temporal Weighting

This layer exploits the frame temporal context as shown in Figure 3(b). The central frame S_t of the context window is the one to be classified by the NN, but the input of NN is a concatenation of all frames



(a) Generic Input Model (GIM) (b) Class-specific Input Model (CIM)

Figure 2: Proposed input models.

inside the context window. Therefore, the input vector contains information from past and future frames. The upper layer performs a time weighted average for each frequency bin, and so it has the same size F than a single input frame.

4 EXPERIMENTAL EVALUATIONS

4.1 Experimental Setup

Since the dataset is relatively small, a 10-fold cross-validation scheme was implemented to obtain more statistically relevant results. As the dataset consists in 10 recording sessions, on each fold 9 sessions were used for training and the remaining one for testing. Then, the overall metric results were obtained. For training the model for an alarm class, from each recording session, all the alarm class frames were extracted plus the same amount of non-alarm frames to create a balanced training set. In order to maximize variability, those non-alarm frames were chosen randomly.

The fully connected hidden layer employed in the neural networks of the presented experiments had 8 hidden units. The activation function of the hidden units was the sigmoid, and that of the output units was softmax. Note that there was no pre-training. Binary cross-entropy was used as objective function for networks training, which seems to perform better than mean-squared error function neither pre-training nor a good initialization are used (Golik et al., 2013). Stochastic gradient descend was used for the network optimization. The learning rate and momentum parameters of the network were set to 0.01 and 0.9, respectively. The number of epochs was 70 and the mini-batch size was 10.

With both types of models the mean and the variance required for applying MVN to both training and testing samples were obtained from the training data. According to (Lecun et al., 1998), convergence is usually faster if the average of each input variable over the training set is close to zero.

4.2 Evaluation Metrics

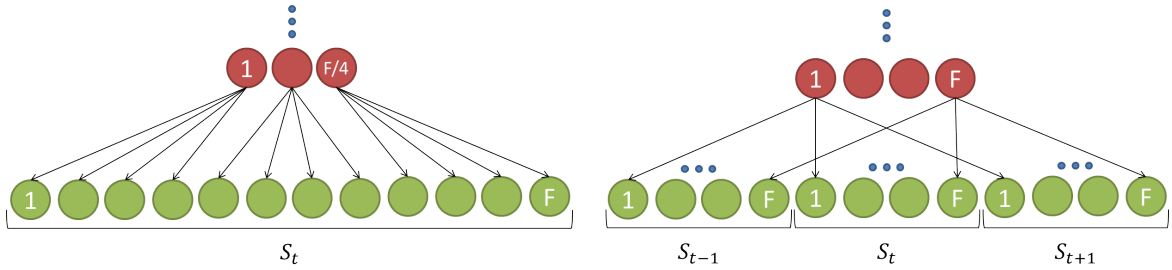
In this work, frame-level and event-level metrics are used to evaluate the detection performance. The Missing Rate (MR) and the False Alarm Rate (FAR) metrics are used for the frame-level evaluations. And these are defined as

$$MR = \frac{N_M}{N_A}, \qquad FAR = \frac{N_{FA}}{N_{NA}}, \tag{1}$$

where N_M and N_{FA} is the number of misclassified frames for alarm and non-alarm class, respectively, and N_A and N_{NA} is the total number of alarm and non-alarm frames, respectively.

The Equal Error Rate (EER) was chosen as the decision criterion at the frame level, therefore the reported MR and FAR metric scores have the same value.

All the alarms are periodic sounds. The period of the alarm was chosen for the event-level evaluation,



(a) Partially connected layer for frequency weighting

(b) Partially connected layer for temporal weighting

Figure 3: Partially connected layers implemented.

since it is a natural alarm-specific unit. The Period-Based Error Rate (PB-ERR), is defined as

$$PB-ERR = 1 - \frac{2 \cdot N_C}{2 \cdot N_C + N_{FA} + N_M}, \qquad (2)$$

where N_C is the number of correctly detected reference alarm periods, N_M and N_{FA} is the number of missed and falsely inserted periods. For calculating this metric, a smoothing post-processing, which is based on majority voting, is applied to the frame-level output labels. The length of the smoothing window was set to be the minimum of the signal and silence interval length in the alarm period. A detailed explanation of this event-level metric is presented in (Raboshchuk, 2016).

4.3 Results and Discussion

The experiments were first carried out with the most populated alarm class from the recorded database (see Table 1), which is class 'a8'. Then, the best network structures for each model were replicated for the rest of the alarm classes.

Table 2 shows the baseline results when either an amplitude (A) or log-amplitude (LogA) spectral input is used. The latter was introduced taking into account that the non-linear logarithm transformation reduces the value range, which could benefit the network's learning. Both experiments were performed using the whole spectral frame length (thus, 1024 input units) fully connected with a hidden layer of 8 units.

Table 2: Baseline results.

Input	Evaluation metrics (%) MR=FAR	PB-ERR	Network weights
A	26.42	**56.17**	8218
LogA	**23.44**	57.53	8218

It can be seen that using the log-amplitude spectrum (LogA) the EER was reduced by almost 3% in absolute value, although PB-ERR slightly increased. Since in this work the focus is on the frame-level classification, the rest of the experiments were performed using the logarithmic representation of the spectrum.

The results in Table 3 show the performance of the system using the proposed pooling techniques for the Generic Input Model (GIM): Average Pooling (AP), Standard Max Pooling (SMP) and Mel-Scale Max Pooling (MSMP). AP and SMP had a pooling size of 4 bins so the resultant input had 256 units. This size was chosen to avoid merging of spectral bins corresponding to class-specific frequencies of different alarm classes inside pooling groups, which may harm the system performance. In order to keep a minimum resolution while preserving the maximum alarm-specific spectral information, a bank of 60 filters was used in MSMP, although typically 40 filters are used in speech and acoustic recognition (e.g. (Sainath et al., 2013)).

Table 3: Pooling techniques results.

Technique	Evaluation metrics (%) MR=FAR	PB-ERR	Network weights
AP	30.24	64.15	2074
SMP	**23.55**	**55.20**	2074
MSMP	25.00	57.47	506

Non-linear pooling techniques clearly provide better results than average pooling, both in terms of frame-level and period-level metric scores. This may be explained by the fact that both max-pooling schemes preserve high spectral peaks corresponding to alarms. Notice that a strong reduction of network weights in MSMP with respect to SMP is obtained at the expenses of only a small increase in EER (6.1% relative). Similarly, at the frame-level compared to the best baseline (LogA), much smaller number of training weights is used, what is an advantage in terms of computation time and overfitting. Next experiments aimed to further decrease both the error rate and the number of training weights.

Experiments with GIM were performed including partially connected hidden layers for weighting in

frequency (FW) and in time (TW), the results from which are shown in Table 4. Spectral representations after SMP and MSMP were used with a FW layer of pooling size 4 and 5, respectively, creating a hidden layer of size 64 and 12 units. Further, a TW layer with a context window of 5 frames was implemented also on both representations. The pooling sizes were experimentally optimized.

Table 4: Generic input model results using partially connected structures.

Setup	Evaluation metrics (%) MR=FAR	PB-ERR	Network weights
SMP+FW(4)	18.27	50.14	450
SMP+TW(5)	16.67	46.03	2050
MSMP+FW(5)	19.05	45.32	98
MSMP+TW(5)	**13.87**	**41.65**	482

In terms of the frame-level evaluation, the use of FW layer with the SMP representation gives slightly better results than using it with MSMP, but the opposite is true in terms of the event-level evaluation. Using TW layer rather than FW layer with both SMP and MSMP better results were achieved, although the number of training weights was around 5 times higher. Also the size of the input representation itself affects: it seems that wider frequency representations (using SMP) work better when the input is only one frame, but shorter representations (using MSMP) work better when temporal context is included.

Recurrent neural networks were also explored for GIM, specifically, the Long-Short Term Memory (LSTM) networks. The objective was to let the network learn the temporal recurrence inherent to the sequence of alarm periods. Table 5 shows the results for different LSTM setups. Starting from a baseline setup, experiments were performed changing only one parameter value at each time. The fixed setup was the MSMP spectral input, a single hidden layer, and a hard sigmoid inner activation function. The baseline setup had 4 cells (C) and 5 timesteps (TS), i.e. the number of spectral frames to look backward. And in the rest of experiments the number of timesteps and the number of cells were modified. Also, averaging the MSMP input frames in a context window (CW) of size 5 was tried, similarly to TW layer. Note that more experiments were carried out with different parameter values, but only the best results are presented.

The various LSTM results do not differ much from their baseline performance and do not achieve those from the combination of MSMP input frames with TW layer that actually shows a smaller number of network weights.

In Table 6 the experimental results for the Class-Specific Input Model (CIM) are shown. In the first two rows, the network consists of a single fully connected (FC) layer of 8 hidden units. The network input consisted of concatenating $\pm$ 1 or 2 bins around the bins corresponding to alarm-specific frequency components (SF). TW layer was also used in CIM with the same size of the context window as in the previous experiments (i.e. 5 frames). As the number of training weights was small enough, the experiment with an extra fully connected (FC) hidden layer of 8 hidden units on the top of TW layer was also performed.

Table 5: Generic input model results employing LSTM.

Setup	Evaluation metrics (%) MR=FAR	PB-ERR	Network weights
Baseline	14.84	48.03	>1000
TS(2)	14.56	49.45	>1000
CW(5)	14.30	47.41	>1000
C(6)	14.84	47.62	>1500
C(6)-CW(5)	**14.09**	**45.66**	>1500

Table 6: Class-specific input model results using fully connected and partially connected hidden layers.

Setup	Evaluation metrics (%) MR=FAR	PB-ERR	Network weights
SF($\pm$1)+FC	16.44	44.09	146
SF($\pm$2)+FC	16.37	41.45	226
SF($\pm$2)+TW	13.23	41.02	202
SF($\pm$2)+TW+FC	**10.69**	**38.36**	376

A wider margin around specific frequency components slightly improved the evaluation results at both frame and alarm period levels. As in GIM, using a TW layer instead of fully-connected layer improved the results, reducing the frame-level metric score by almost 3% in absolute terms and also reducing the number of network parameters. Then, even better results were obtained when a second fully connected hidden layer of 8 units was added. The number of network weights still was low and further reduction of both metric scores was around 3% in absolute value.

Finally, the best structures obtained with 'a8' alarm class for the baseline, GIM and CIM, which are highlighted in bold in Tables 2, 4 and 6, respectively, were replicated to the rest of alarm classes:

1. Baseline: logarithmic representation of the magnitude spectrum and a fully connected layer of 8 hidden units (LogA+FC, see Table 2).
2. GIM: TW layer over 5 concatenated MSMP input frames (MSMP+TW(5), see Table 4).
3. CIM: concatenation of $\pm$2 spectral bins around

the alarm-specific bins at the input, TW layer over 5 concatenated MSMP input frames and a second fully connected hidden layer of 8 units (SF(±2)+TW+FC, see Table 6)

The results shown in Table 7 are the metric scores averaged over the 7 considered alarm classes. The baseline model and GIM have the same number of training weights for all the classes as they have the same input. But the CIM structure has a different number of weights depending on the number of class-specific frequency components. Namely, for each alarm class, all its frequency components (shown in Table 1) were included at the input; thus, each alarm-class has a different number of network parameters. Note that the number of network weights provided in Table 7 for CIM is an average value over all alarm-specific structures.

Table 7: Best model structure results over all alarm classes.

System	Evaluation metrics (%)		Network
	MR=FAR	PB-ERR	weights
Baseline	23.42	68.68	8218
GIM	17.76	54.80	482
CIM	**11.13**	**53.75**	330

Both kinds of input models improve significantly the baseline results, and the proposed CIM structure clearly outperforms GIM. It can be seen that the event-based evaluation was considerably improved by both input model structures with respect to the baseline, but it still has much room for improvement.

5 CONCLUSIONS

In this work several neural network based structures were presented to detect alarm sounds in a NICU environment. Two kind of models based on different input formats, that either make use or not of the knowledge about the alarm class properties, were proposed and tested: generic and class-specific. Due to the scarcity of available annotated data, the number of layers and nodes was kept small. Both linear and non-linear pooling techniques were considered in order to reduce the size of the input. Also, in order to exploit frequency and temporal information while reducing that network complexity, two types of partially connected hidden layers were implemented.

As expected, it was observed that the class-specific input model, which takes advantage of the knowledge about the alarm frequency components, yielded better results than the generic input model; however, the structure of the latter model has does not need to be adapted to each new alarm class. Also, both types of partial connections, which make a significant reduction of training weights, improved the error rate, especially the time-based one.

The detection rate is still high, but we believe that, when a much bigger dataset will be available and used, the differences in performance among the various tested neural net structures that have been observed in this work will be substantially kept.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish government (contracts TEC2012-38939-C03-02 and TEC2015-69266-P) as well as by the European Regional Development Fund (ERDF/ FEDER). The authors are grateful to Ana Riverola de Veciana and Blanca Muoz Mahamud for their work on the database collection and on the medical aspects of this study, and to Vanessa Sancho Torrents and Francisco Alarcón Sanz for their work on the database annotation.

REFERENCES

Abdel-Hamid, O., Deng, L., and Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. In *INTERSPEECH*, pages 3366–3370.

Beritelli, F., Casale, S., Russo, A., and Serrano, S. (2006). An Automatic Emergency Signal Recognition System for the Hearing Impaired. In *Proceedings of the IEEE Digital Signal Processing Workshop*, pages 179–182.

Carbonneau, M.-A., Lezzoum, N., Voix, J., and Gagnon, G. (2013). Detection of alarms and warning signals on an digital in-ear device. *International Journal of Industrial Ergonomics*, 43(6):503–511.

Ellis, D. P. (2001). Detecting alarm sounds. In *Consistent & Reliable Acoustic Cues for Sound Analysis: One-day Workshop: Aalborg, Denmark, Sunday, September 2nd, 2001*, pages 59–62. Department of Electrical Engineering, Columbia University.

Freudenthal, A., Stuijvenberg, M. v., and Goudoever, J. B. v. (2013). A quiet NICU for improved infants health, development and well-being: a systems approach to reducing noise and auditory alarms. *Cognition, Technology & Work*, 15(3):329–345.

Golik, P., Doetsch, P., and Ney, H. (2013). Cross-entropy vs. squared error training: a theoretical and experimental comparison. In *INTERSPEECH*, pages 1756–1760.

Lecun, Y., Bottou, L., Orr, G., and Müller, K. (1998). Efficient backprop.

Lutfi, R. A. and Heo, I. (2012). Automated detection of alarm sounds. *The Journal of the Acoustical Society of America*, 132(2):125–128.

Meucci, F., Pierucci, L., Del Re, E., Lastrucci, L., and Desii, P. (2008). A real-time siren detector to improve safety of guide in traffic environment. In *Signal Processing Conference, 2008 16th European*, pages 1–5. IEEE.

Raboshchuk, G. (2016). *Automatic analysis of the acoustic environment of a preterm infant in a neonatal intensive care unit*. PhD thesis, Technical University of Catalonia, Barcelona, Spain.

Raboshchuk, G., Janovi, P., Nadeu, C., Lilja, A. P., Kker, M., Mahamud, B. M., and Veciana, A. R. d. (2015). Automatic detection of equipment alarms in a neonatal intensive care unit environment: A knowledge-based approach. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Raboshchuk, G., Nadeu, C., Muñoz Mahamud, B., Riverola de Veciana, A., and Navarro Hervas, S. (2014). On the acoustic environment of a neonatal intensive care unit: initial description, and detection of equipment alarms. In *Proceedings of INTERSPEECH*.

Sainath, T. N., Kingsbury, B., Mohamed, A.-r., and Ramabhadran, B. (2013). Learning filter banks within a deep neural network framework. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 297–302. IEEE.

Wachman, E. M. and Lahav, A. (2010). The effects of noise on preterm infants in the NICU. *Archives of Disease in Childhood - Fetal and Neonatal Edition*, 96(4):305–309.

Xiao, X., Yao, H., and Guo, C. (2009). Automatic Detection of Alarm Sounds in Cockpit Voice Recordings. In *IITA International Conference on Control, Automation and Systems Engineering (CASE)*, pages 599–602.