Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Elettronica, Elettrotecnica e delle Telecomunicazioni

---

Signal Processing algorithms and Learning Systems for Infant Cry Detection

Ph.D. Dissertation of:
Daniele Ferretti

Advisor:
Prof. Stefano Squartini

Ph.D. School Supervisor:
Prof. Francesco Piazza

XVII edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Elettronica, Elettrotecnica e delle Telecomunicazioni

---

Signal Processing algorithms and Learning Systems for Infant Cry Detection

Ph.D. Dissertation of:
Daniele Ferretti

Advisor:
Prof. Stefano Squartini

Ph.D. School Supervisor:
Prof. Francesco Piazza

XVII edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

a Valentina

Abstract

Newborns' cry signals contain valuable information related to the state of the infant. Extracting this information requires a cry detection algorithm able to operate in environments with challenging acoustic conditions, since multiple noise sources, such as interferent cries, medical equipments, and persons may be present. Cry detection is an important facility in both residential and public environments, which can answer to different needs of both private and professional users. In the current dissertation the issue of cry detection in professional and acoustic noisy environments such as Neonatal Intensive care units (NICUs) will be investigate. The research, presented in this thesis, describes the developed approaches for the infant cry detection suitable for NICUs as well as an effective training methodology that does not require labeled data collected in the specific domains of use. In the described approaches the acoustic noise reduction is performed processing multiple audio channels using digital signal processing techniques as well as neural strategies. These approaches use Deep Neural Networks, whose training is conducted on a synthetic dataset created by means of a suitable Acoustic Scene Simulation procedure. The Acoustic Scene Simulation allows the creation of a synthetic dataset that, differently from a real-life dataset, can be acquired without access a NICU. The obtained detection results confirm the goodness of the developed approaches overcoming the performance achieved by the algorithms of the state of art taken as reference and proving that a synthetic dataset can be a useful replacement with respect to a real-life dataset, at least in the early design process. The proposed training methodology permits to lower the interaction with a sensitive environment such as a NICU, to the bare minimum and can be exploited to include changes to the environment as needed, without requiring additional acquisition sessions.

Sommario

I segnali associati al pianto dei neonati contengono preziose informazioni relative allo stato del bambino. L'estrazione di queste informazioni richiede un algoritmo di rilevazione del pianto in grado di operare in ambienti con condizioni acustiche difficili caratterizzati dalla presenza di fonti di rumore come pianti interferenti, apparecchiature mediche e persone. Il rilevamento del pianto infantile è una funzione importante sia negli ambienti residenziali che in quelli pubblici, in grado di rispondere alle differenti esigenze dei professionisti e degli utenti privati. Nella presente dissertazione viene presentata una indagine riguardo alla problematica questione della rilevazione del pianto infantile in ambienti professionali ed acusticamente rumorosi come le unità di terapia intensiva neonatale (UTIN). La ricerca descritta in questa tesi è volta allo sviluppo di approcci per la rilevazione del pianto adatti alle UTIN, nonché alla definizione di una efficace metodologia di allenamento degli algoritmi che non necessiti di dati raccolti negli specifici domini di utilizzo. Negli approcci descritti, la riduzione del rumore acustico viene eseguita su canali audio multipli con tecniche di elaborazione del segnale digitale e strategie neurali. Questi approcci utilizzano delle reti neurali profonde addestrate su un set di dati sintetico creato mediante un'adeguata procedura di simulazione di scene acustiche, senza la necessità di accedere ad una UTIN. I risultati ottenuti confermano la bontà degli approcci sviluppati superando le prestazioni ottenute dagli algoritmi dello stato dell'arte presi come riferimento, dimostrando che un set di dati sintetico può essere un utile rimpiazzo rispetto ad un set di dati della vita reale. La metodologia proposta per l'allenamento delle reti neurali consente di ridurre al minimo l'interazione con ambienti sensibili come le UTIN e permette di elaborare modifiche dei domini di utilizzo senza richiedere sessioni di acquisizione aggiuntive.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Engineering collaboration in the biomedical field has accelerated the development of new diagnostic technologies that simplify the management and treatment of diseases and, at the same time, improve our understanding of medicine. Nowadays, the need for electronic assistance systems in the global health-care sector is increasingly recognized. Existing solutions can be quickly integrated into health-care processes by providing smarter and more accessible services [4]. Pervasive and ubiquitous computing solutions, are already at the basis of the most modern medical monitoring and diagnostic systems. These technologies can be used to take a big step towards reducing morbidity and mortality patterns.

A field of use is that of neonatology. In Italy, it is estimated that 6.5% of the total birth corresponds to premature babies, i.e. children born under 37 weeks of gestation. The premature birth, in particular the critical one (newborn's weight under 0.9 kg), negatively influences the anatomical and functional development of the infant, causing permanent health issues. The repercussions on society are of enormous value, both for the impact on families and in economic terms. The Health problems associated to premature birth can be avoided or reduced with a prompt medical treatment with a high technological content.

Infant cry detection systems offer support to early diagnosis systems that can be used in the clinical setting resulting in significant benefits both in terms of managing the neonatal care process and disease prevention.

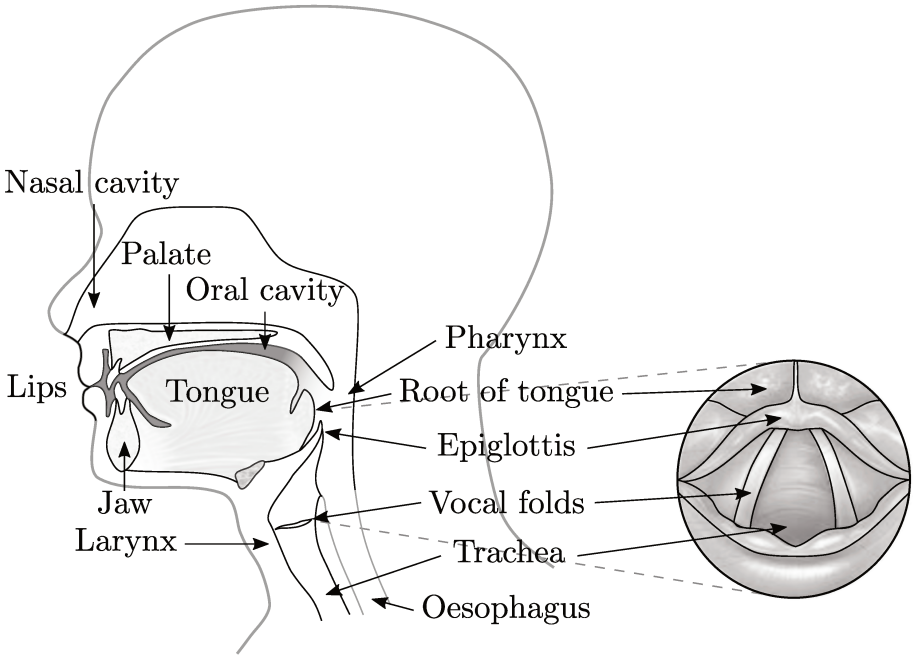


Figure 1.1: Anatomy of the human vocal apparatus.

1.1 Infant Cry

Crying is a basic and innate form of communication of infants, through which they voice their needs deriving from internal or external stimuli. In nature, crying is an important act for survival. It has acoustic properties that elicit maximal attention and strong emotions in parents. Physiologically, crying involves the central and autonomic control of arousal/inhibitory mechanisms, the coordination of cardiorespiratory activity and the laryngeal musculature [5]. The multiplicity of physiological and psychological factors involved in the formation of crying make it a valuable source of information regarding the health of the infants [6–11].

A cry is a series of four movements called phases. Figure 1.2 shows the spectrogram computed from an infant cry with the phases in evidence. They are inspiration (strain phase), exhalation or expiratory (sigh phase), pause (nonphonatory in nature), and then a quick inspiratory gasp that precedes the next cry [12]. The anatomy of the human vocal apparatus is depicted in Figure 1.1. The cry sounds is generated in

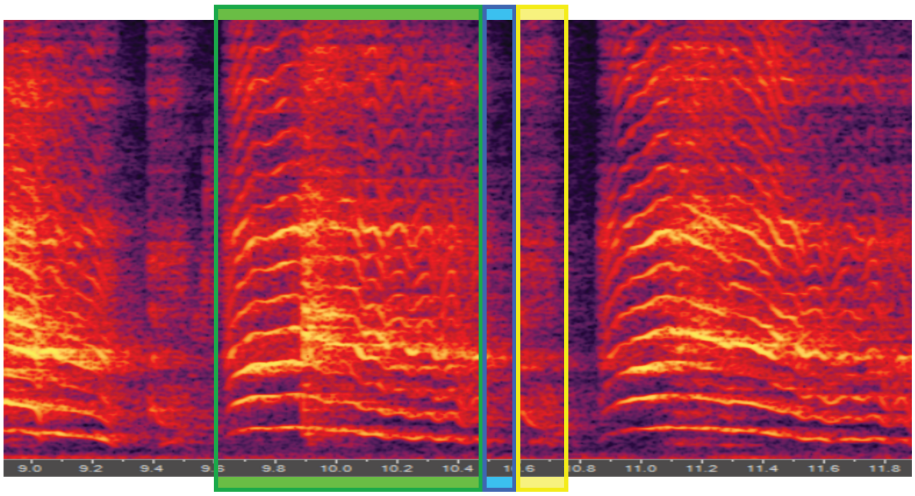


Figure 1.2: The phases of a cry unit: expiration (green), pause (blue), inspiration (yellow).

the larynx, which contains the vocal folds, through a mechanism named phonation, that is the same of speech production in an adult. The gap between the vocal folds is called the glottis. The larynx also protects the trachea against food aspiration during swallowing. The glottis is fully open in the normal respiratory phases but closes during a phonation. The air, pumped by the lungs through a constricted tube, causing the vocal cords to open and close rapidly producing a vibration. This vibration corresponds to the fundamental frequency ($F0$) of cry and in normal, healthy newborns typically ranging from 300 to 600 Hz. The part of the human vocal apparatus located above the glottis (supraglottal system) shapes the sound to produce the formant frequencies. They are referred to as first formant ($F1$), second formant ($F2$), and so forth. Formant frequencies are usually independent of the fundamental frequency and its harmonics. The part of the human vocal apparatus located under the glottis (subglottal system) is instead associated with the rhythms of expiratory and inspiratory sounds and with the loudness of cry. During crying, the volume, the pitch and the tone color are changeable. The changing of the pitch is called melody and describe a curve that, except for sudden pitch shifts, has continuous nature.

In the study [1], György Várallyay tried to find out if there were any differences between the crying sound of normal hearing and hard-

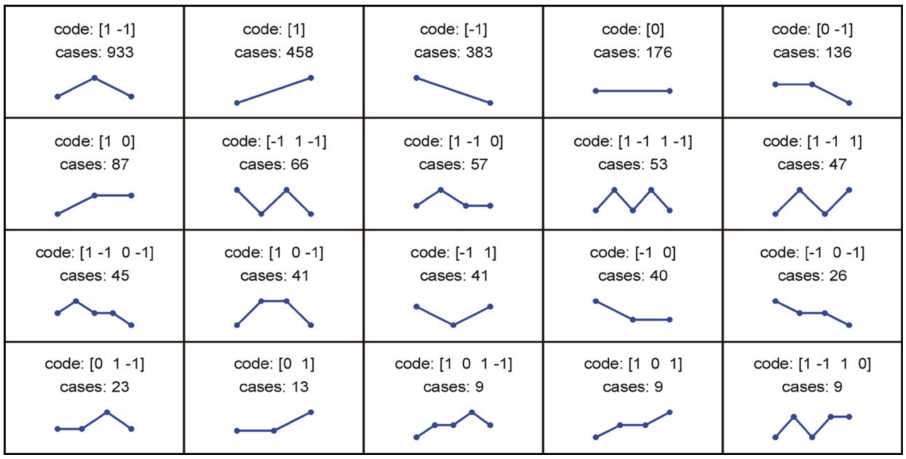


Figure 1.3: The 20 most common melody shapes [1]

of-hearing infants. He collected individual recordings from 316 infants that started to cry during examination of the eardrum, a painful procedure. All the recordings were made in quiet places in the hospitals or at homes, but not in special silence rooms. Leveraging on simple attributes of a crying (energy, duration longer than a few tenths of seconds, regular spectral structure) the audio fragments of 2762 cry sounds were automatically cleaned from disturbing parts such as noises, pauses, coughs, etc. Each segment has been partitioned into 40 ms long windows in order to compute the spectrum and then detect the the fundamental frequency F0. The detected consecutive F0 values formed the melody of the segment. All the melodies has been analyzed as a sequence of three fundamental units, i.e. falling (-1), flat (0) and rising (1). In this way Várallyay determined 77 different categories of melody shapes and discover that the first 20 categories covered the 95% of the all melodies computed from a cry (Figure 1.3).

As written above, newborns' cry signals contain valuable information related to the state of the infant, thus their acoustic analysis can provide a cost-effective and non-intrusive monitoring approach in different environments, spanning from simple households to infant wards or even Neonatal Intensive Care Units (NICUs) [13]. From in-depth analysis of the audio signals can be detected a specific clinical situation, such as the presence of a pathology [6–8], or even the cause of a cry (e.g., hunger, pain) [9–11].

Figure 1.4: A Neonatal intensive care unit

1.2 Neonatal intensive care unit

The NICUs are characterized by very chaotic environments. Many cribs and incubators are "packed" together in a room, each with its own medical instrumentation which, monitoring the infants, emits routine acoustic signals and alarms. In addition to the medical staff working in the room, often young patients are assisted by their parents. An example of NICU is shown in Figure 1.4 [1]

Despite an high level of sound pressure inside NICUs represent a risk factor for newborns and care-givers [14, 15], in the literature there are many contributions [2,14,16] that describe how recommended maximum levels are often exceeded in these environments.

Excessive noise level in NICU disturbs sleep, causes stress, and interferes with the development of the brain auditory neurons of infants. Since the first year of a newborn's life is fundamental in the development

[1] photo source: `http://worldpediatrics.alliedacademies.com/events-list/neonatal-intensive-care-unit-nicu`

of hearing and language, failure to maintain noise levels under the maximum levels recommended may result in numerous adverse noise-induced health effects, particularly to the premature infants

The high noise level of most NICU environments has two essential sources:

- background noises from incubator motors, ventilator equipment and medical equipment processes.
- impulsive events, such as the noise from alarms equipment, conversational sounds and various activities of the attending staff.

Given the high number of patients who typically stay in a NICU, even the crying of newborns is a significant and frequent source of environmental noise. Tipically, during the first three days of life of a newborn, 6.7 hours (402 minutes) per day of crying is common. The mean duration of crying per day in full-term infants peaks at a range of 42.7 to 120 minutes at six weeks, and at 29.3 minutes at 13 weeks [12].

An interesting findings is shown in [2] that describe a sound spectral analysis conducted in two NICUs. Figure 1.5 shows one of the most intense detections obtained near an extracorporeal membrane oxygenation (ECMO) equipment. In particular, the sound pressure levels surpasses security levels at higher frequencies increasing the risk factor for the health of newborns. This detection is a great example of extreme acoustic situations that can occur in a NICU.

1.3 Aims and outline

Cry detection, which consists on the identification of a cry within the audio stream, is the fundamental part of any cry analysis system. It can alert the medical staff regarding the onset of a state of unease of the newborn, and can be used to evaluate the overall health status of the infant based on the analysis of the cries occurrence over time. Moreover, without algorithms for infant cry detection, analysis procedures can not be automated, making it impossible to continuously monitor newborns or perform large-scale studies.

The ability to infer useful medical information from infant cry depends on the goodness of the process used to identify and extract the cry units

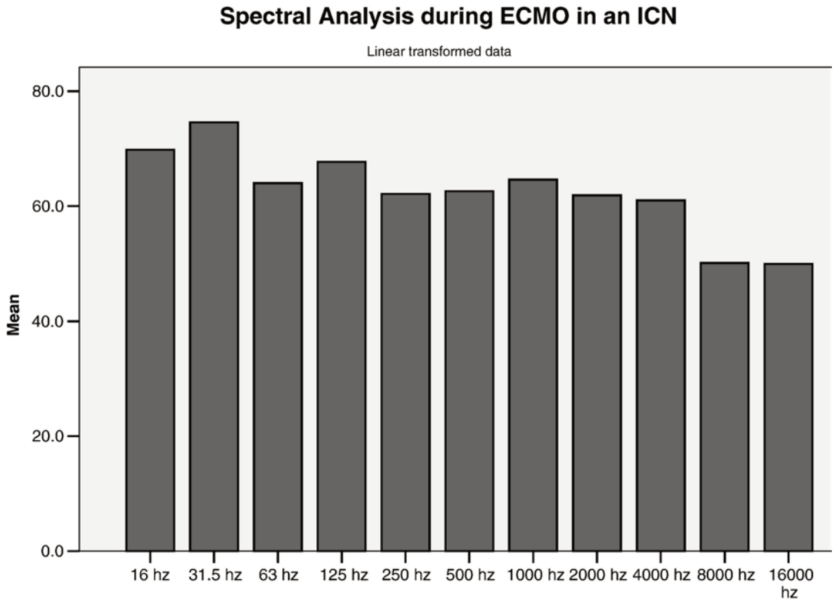


Figure 1.5: Bar chart indicating sound frequency spectrum during ECMO. [2]

from the audio stream. This aspect can present major challenges in those environments characterized by high level of acoustic noise. Moreover, in professional environments, such as infant wards or NICUs, is necessary to take into account sanitary concerns resulting from researchers accessing a NICU environment, and bureaucratic concerns to obtain the authorization to access in a NICU and make audio recording.

The research, described in the current dissertation, is the groundwork that has been carried out in order to develop approaches for the infant cry detection suitable for noisy and professional environments as well as an effective training methodology that does not require labeled data collected in the specific domains of use

The outline of the thesis is the following:

Chapter 2 presents the Machine Learning (ML) and the digital signal processing (DSP) techniques used for the development of the Infant Cry Detection (ICD) approaches studied on this research work. Herein, the concepts underlying the neural approaches are introduced and the functioning of the Feedforward Networks (FFNN), Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN) are explained.

Subsequently, two well-known DSP techniques, i.e. the Beamforming and the Optimally-Modified Log-spectral Amplitude (OMLSA) are detailed. For the development and test phases of the main contribution of this thesis, two different datasets of infant crying have been used, one real and the other synthetic. The synthetic one has been developed with an Acoustic Scene Simulation procedure, starting from raw records discovered in the network. The real one has been collected into the NICU of the Salesi Hospital in Ancona. In Chapter 3 the recording device used for the audio acquisitions is shown, the operations performed on data are explained and the compositions of the two datasets are described. In Chapter 4 a neural approach for estimating the Direction Of Arrival (DOA) of the sound emitted by a target source is presented and is showed how this technique when used in synergy with a beamformer improve its performance. The neural approach for the estimation of DOA is a preliminary study thought in anticipation of development of the audio enhancement techniques used for the ICD algorithm presented in Chapter 5. After reviewing the state of the art of the ICD algorithms, different approaches to the problem are proposed and motivated. Subsequently, the experiments conducted in order to determinate the performances of the proposed approaches are presented. The results obtained has been compared with a reference algorithm presented in the same chapter. Chapter 6 present three other contributions developed by the author during the PhD research. They concern the more general topic of the ambient assisted living. One concerns the automatic recognition of activity of daily living and other two the automatic detection of human falls. Chapter 7 draws the conclusion of this work.

Chapter 2

Background

2.1 Neural Network

The biological neural networks of human brain are made up of a large amount of specialized cells, named neurons, connected among them, which continually receives information, perceives it, and makes appropriate decisions. Figure 2.1 illustrates the shape of a multipolar neuron, which is one of the most common types of cortical neurons. It is composed of:

- Dendrite: input terminal
- Nucleus: processing core
- Axon: output way-out
- Synapses: output terminal (with weight)

Typically, neurons are five to six orders of magnitude slower than silicon logic gates; events in a silicon chip happen in the nanosecond range, whereas neural events happen in the millisecond range. However, a conventional digital computer can not compete with a biological neural network in terms of speed and accuracy when dealing with tasks of recognition (people, objects, sounds, etc.), comprehension of natural language, ability to learn, categorize, generalize and memorise. The main reason behind this gap has to be found in the different information processing strategies: actual computers typically works according to sequential paradigms, while the biological neural networks of the brain perform calculations by means of a huge number of simple operations in parallel. A biological neural networks has also the following remarkable properties:

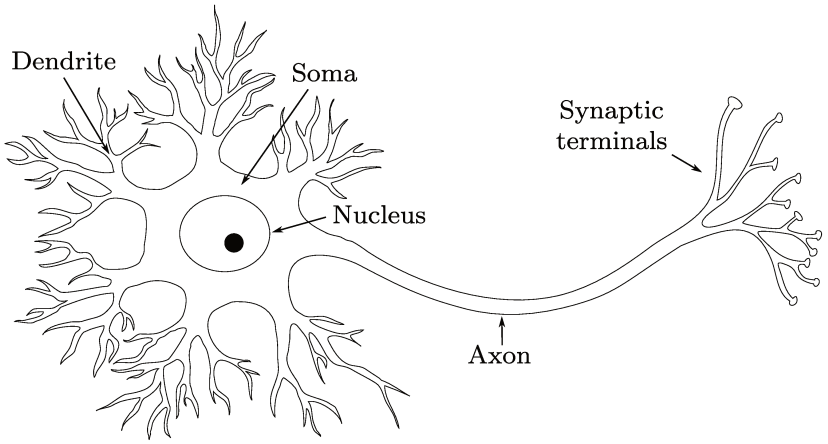


Figure 2.1: The neuron model.

- Local simplicity: the neuron receives stimuli (excitation or inhibition) from dendrites and produces an impulse to the axon which is proportional to the weighted sum of the inputs;
- Global Complexity: the human brain possess approximately 10 billion neurons, with $\mathcal{O}(10^{18})$ synapses or connections.
- Learning: even though the topology of the network is relatively fixed, the strength of connections (synaptic weights) can change when the network is exposed to external stimuli;
- Distributed Control: no centralized control, each neuron reacts only to its own stimuli;
- Tolerance to failures: performance slowly decrease with the increase of failures.

In its most general form, an artificial neural network (ANN), commonly referred to as "neural networks", is a machine designed to model the way in which the brain performs a particular task or function of interest. It is possible to formulate the following definition of a neural network seen as an adaptive machine [17]:

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

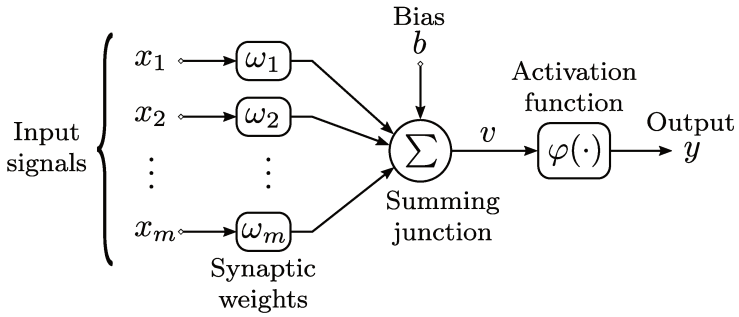


Figure 2.2: The artificial neuron model.

1. *Knowledge is acquired by the network from its environment through a learning process.*

2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.*

Through some kind of learning algorithm, the synaptic weights of the network are modified in order to reach a desired design goal. However, it is also possible for a neural network to modify its own topology, which is motivated by the fact that neurons in the human brain can die and new synaptic connections can grow.

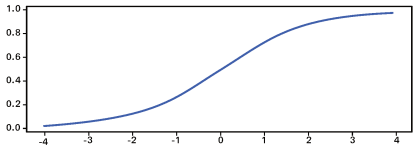
As in biological neural networks, the fundamental process unit of an ANN is the neuron. As shown in Figure 2.2, the neuron model is composed of:

- Synapses: a set of connecting links, each of which is characterized by a weight or synaptic strengths. Specifically, a signal x_i at the input of synapse i connected to neuron is multiplied by the synaptic weight w_i.

- Bias: external parameter of the neuron.

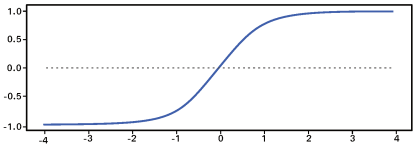
- Adder: it sums the bias and the input signals, weighted by the respective synaptic strengths.

- Activation function: limits the amplitude of the output and decides whether the neuron should "fired" or not.



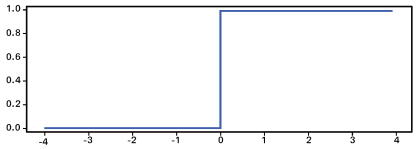
Sigmoid

$$\varphi(v) = \frac{1}{1+e^{-v}} \tag{2.2}$$



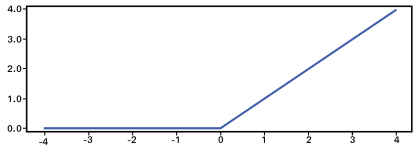
Hyperbolic Tangent (*tanh*)

$$\varphi(v) = \frac{e^{v}-e^{-v}}{e^{v}+e^{-v}} \tag{2.3}$$



Rectified Linear (ReLU)

$$\varphi(v) = \begin{cases} 0 & \text{if } v < 0 \\ 1 & \text{if } v \geq 0 \end{cases} \tag{2.4}$$



Rectified Linear (ReLU)

$$\varphi(v) = \begin{cases} 0 & \text{if } v < 0 \\ v & \text{if } v \geq 0 \end{cases} \tag{2.5}$$

Figure 2.3: Common types of activation function

The corresponding mathematical formulation is:

$$y = \varphi\left(\sum_{i=1}^{m} w_i x_i + b\right) \tag{2.1}$$

where $x_1, x_2, \ldots, x_m$ are the input signals, $w_1, w_2, \ldots, w_m$ are the respective synaptic weights, b is the bias, $\varphi(\cdot)$ is the activation function and y is the output signal of the neuron.

Regarding the activation function, its choice highly affects the performance of the model and plays an important role for the learning algorithms. In Figure 2.3 four common types of activation function are shown.

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. We may therefore speak of learning algorithms (rules) used in the design of neural networks as being structured.

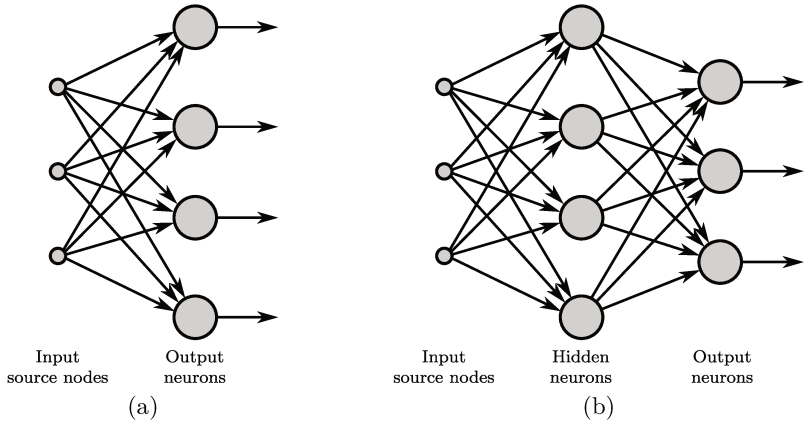


Figure 2.4: Layered perceptron: (a) Single-layer (b) Multi-layer

2.1.1 Feedforward Networks

A neural network is defined as feedforward if the connections between its nodes do not form cycles and, therefore, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. The feedforward networks have no memory of input occurred at previous times, so the output is determined only by the current input. The simplest feedforward network is the Single-layer Perceptron (SLP). The term layered indicates that the neurons are organized in the form of layers. An SLP consists of an input layer, followed directly by the output. The input layer of source nodes is not included in the count of layers since no computation is performed there. Each input unit is connected to each output unit (Figure 2.4a). In practice, this type of neural network has only one layer that performs data processing. A SLP can be trained by a simple learning algorithm that is usually called the delta rule. It calculates the errors between calculated output and sample output data, and uses this to create an adjustment to the weights, thus implementing a form of gradient descent.

SLP can only solve linearly separable problems. This limitations has led to the development of multi-layer feedforward networks with one or more hidden layers, called Multi-layer perceptron (MLP). A MLP is characterized by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden

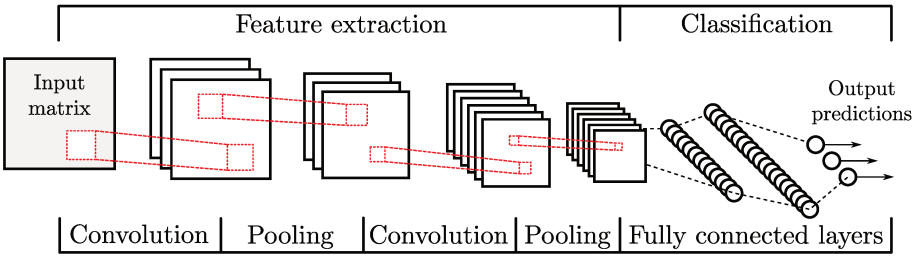


Figure 2.5: The Convolutional Neural Network

units (Figure 2.4b). The term hidden refers to the fact that this part of the neural network is not seen directly from either the input or output of the network. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding one or more hidden layers, the network is enabled to extract higher-order statistics from its input. In a rather loose sense, the network acquires a global perspective despite its local connectivity, due to the extra set of synaptic connections and the extra dimension of neural interactions. The neurons in each layer of the network have as their inputs the output signals of the preceding layer only. If every node in each layer of the network is connected to every other node in the adjacent forward layer, the NN is said to be fully connected. If, however, some of the links are missing from the network, we say that the network is partially connected. The set of output signals of the neurons in the output layer of the network constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input layer. A MLP is often trained with the back propagation, a form of supervised learning. Error data at the output layer is back propagated to earlier ones, allowing incoming weights to these layers to be updated.

2.1.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a particular type of feedforward networks that are particularly effective in recognizing two-dimensional forms with a high degree of translational invariance, scaling, tilt and other forms of distortion.

The CNNs became popular during the 2012 ImageNet Computer Vision competition when Alex Krizhevsky introduced the AlexNet [18].

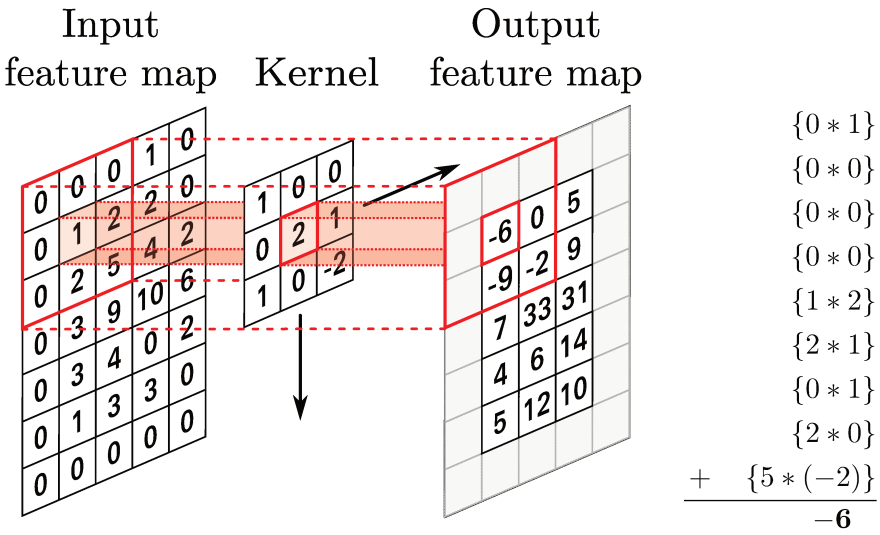


Figure 2.6: The convolution operation

This NN, based on a 14 years earlier work of Yann LeCun [3], it was capable to achieved an error of only 15.8% in classifying of millions of images from thousands of categories. Nowadays, state-of-the-art Convolutional Neural Networks surpass the human-level performance in the imaging recognition task [19]. Although primarily designed for image classification tasks, CNNs have proven to be successful in speech and music recognition [20].

The CNN architecture is essentially made up of an MLP preceded by several special layers (Figure 2.5):

- *Convolutional layers:* in these layers the convolution operation is performed between a filter, named kernel, and the input matrix. An example of convolution is shown in Figure 2.6. The inputs of each neuron are then restricted to a local receptive field in the previous layer, thereby forcing it to extract local features. Once a feature has been extracted, its exact location becomes less important, so long as its position relative to other features is approximately preserved. When the feature is present in part of an image, the convolution operation between the filter and that part of the image results in a real number with a high value. If the feature is not present, the resulting value is low. For each com-

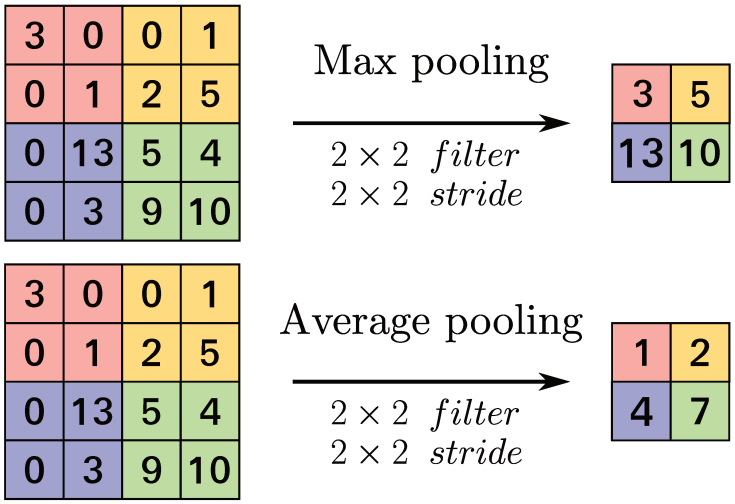


Figure 2.7: The convolution operation

putational layer the network extract multiple feature maps. The neurons are constrained to share the same set of synaptic weights within a single feature map. This constrain reduce the capacity of the learning machine but at the, same time, improves the machine's generalization ability. Moreover, the use of kernel of small size in convolution, followed by a sigmoid function, gives the shift invariance property to the network. Weight sharing, as well as reducing the number of free parameters, also makes it possible to implement the convolutional network in parallel form.

- *Subsampling layers:* Each convolutional layer is followed by a computational layer that performs local pooling and subsampling. As shown in Figure 2.7, the pooling operation consist in a window passes over a feature map according to a set stride (how many units to move on each pass). At each step, the maximum value (Max Pooling) or the average value (Average Pooling) within the window is pooled into an output feature map. This operation has the effect of reducing the sensitivity of the feature map's output to shifts and other forms of distortion.

The idea of convolution followed by subsampling is inspired by the studies of Hubel and Wiesel (1962, 1977) on locally sensitive and orientation-selective neurons of the visual cortex of a cat. Appending more compu-

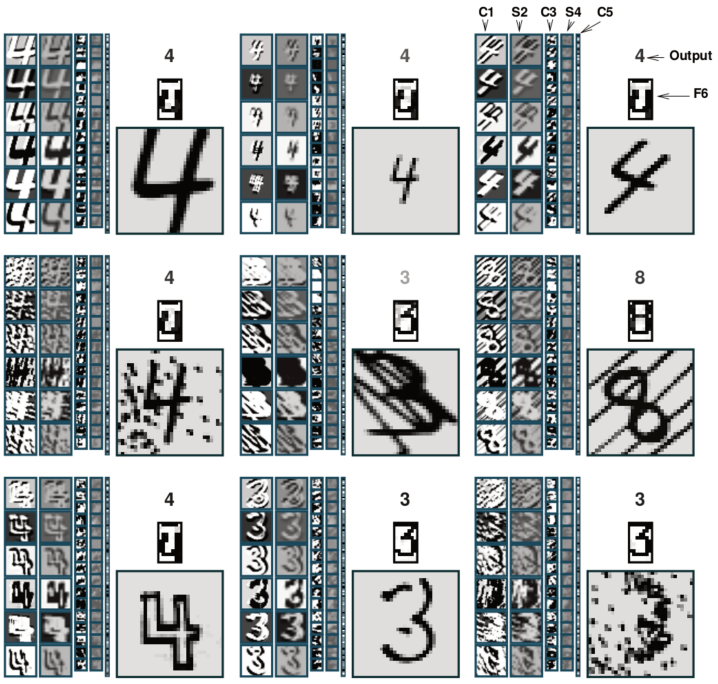


Figure 2.8: Feature Maps and output of LeNet-5 [3].

tational layers alternating between convolution and subsampling, we get a "bipyramidal" effect. That is, at each convolutional or subsampling layer, the number of feature maps is increased while the spatial resolution is reduced, compared with the corresponding previous layer. An example of feature maps produced by LeCun's CNN described in [3] is shown in Figure 2.8. For each set, the first, second and fifth images column are the outputs of convolutional layers, the second and fourth the outputs of subsampling layers. What is even more remarkable is the fact that the adjustments to the free parameters of the network are made by using the stochastic mode of backpropagation learning and all weights in all layers of a convolutional network are learned through training. Moreover, the network learns to extract its own features automatically.

2.1.3 Deep Neural Network

The Deep Learning methods are those formed by the composition of multiple nonlinear transformations with the goal of yielding more abstract—and ultimately more useful-representations [21]. The Deep Neu-

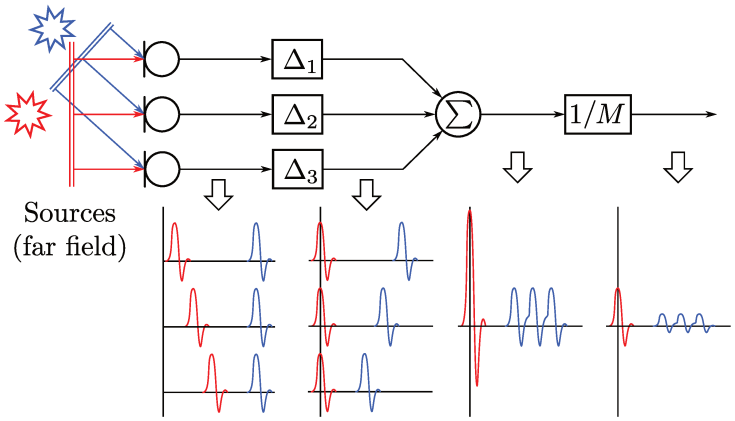


Figure 2.9: The Dalay-and-Sum beamformer.

ral Networks (DNN) are a subclass of the NN that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification. Deep learning removes the manual identification of features in data and, instead, relies on whatever training process it has in order to discover the useful patterns in the input examples. This makes training the neural network easier and faster, and it can yield a better result that advances the field of artificial intelligence. Neural Networks are often referred as deep when they have more than 1 or 2 hidden layers. Both CNN and MLP can be referred as deep network architectures.

2.2 Beamforming

Beamforming or spatial filtering is a signal processing technique used in sensor arrays for directional reception. This filtering is achieved by combining the signals acquired by the array elements in an appropriate manner. Using beamformer techniques, it is possible to isolate the audio signals coming from the desired acoustic sources, from the other coherent sources present in a room, or it is possible to locate a specific source spatially.

The simplest beamformer is the Delay And Sum, depicted by the block diagram in Figure 2.9. When a sound wave reaches the microphones of the array, this capture very similar waveforms, but differ in

delay and phase. These differences are related to the specific paths traveled by the sound waves from sources to each microphones of the array. When the microphone channels are added together, there are phenomena signal interference. In particular, the signal components that are in phase, experience constructive interference while the others experience destructive interference, i.e. they are filtered out. Its possible to focus the array to a specific direction of arrival of signal by choosing an appropriate delay for each input channel.

More generally, the Beamforming operation consists in filtering the signal acquired by each microphone and summing the outputs. Denoting with $s(t)$ the desired source, with $a_m(t)$ the room impulse response between the m-th microphone and $s(t)$, and with $n_m(t)$ the noise term related to microphone m, the signal acquired by the m-th microphone is given by:

$$z_m(t) = a_m(t) * s(t) + n_m(t). \tag{2.6}$$

Analyzing the signals with the short-time Fourier transform (STFT), Eq. (2.6) can be expressed in vector form as:

$$\mathbf{Z}(k,l) = \mathbf{A}(k)S(k,l) + \mathbf{N}(k,l), \tag{2.7}$$

where l is the frame index and k is the frequency bin index. Given the filter $W_m^*(k,l)$, $m = 1, \ldots, M$, the vector formulation of the beamforming operation is:

$$Y(k,l) = \mathbf{W}^H(k,l)\mathbf{Z}(k,l). \tag{2.8}$$

The filter coefficients can be fixed or can vary according to an adaptive strategy. Different approaches can be adopted for the calculation of the coefficients, depending on the characteristics desired for the beamformer. With the linearly constrained minimum-variance (LCMV) algorithm [22], the filters coefficients $\mathbf{W}^H(k,l)$ are obtained by minimizing the output power $E\{Y(k,l)Y^*(k,l)\}$, and constraining the signal component of $Y(k,l)$ to be equal to $S(k,l)$. It can be proved [22] that the steepest descent formulation of the adaptive solution is given by the following expression:

$$\mathbf{W}(k,l+1) = P(k)[\mathbf{W}(k,l) - \mu\mathbf{Z}(k,l)Y^*(k,l)] + \mathbf{F}(k), \tag{2.9}$$

where

$$P(k) = \mathbf{I} - \mathbf{A}(k)\mathbf{A}^H(k)/||\mathbf{A}(k)||^2$$

and

$$\mathbf{F}(k) = \mathbf{A}(k)/||\mathbf{A}(k)||^2.$$

Neural Beamforming

The number of approaches to speech enhancement that exploits completely neural or hybrid techniques are growing. For example, actual DNN-based beamformers for automatic speech recognition (ASR) have been proposed in [23, 24]. In [24], the algorithm operates on multiple complex-valued short-time Fourier transforms (STFTs) to estimate a single enhanced signal. The network exploit generalized cross correlation (GCC) coefficients to estimate the weights of a filter and sum beamformer that then processes the STFTs related to the microphone signals. The algorithm is employed as a preprocessing stage of an ASR and it can be trained jointly with the acoustic model in order to further optimize the process. A similar approach has been proposed in [23, 24], where the spatial filter coefficients are estimated by a long short-term memory (LSTM) network [25] that takes raw waveform signals as input, instead of STFT coefficients. In [26, 27], the authors employ a DNN and a Bidirectional-LSTM for estimating the time-frequency mask in a minimum variance distortionless response (MVDR) beamformer.

2.3 Post filter

In this thesis an audio enhancing technique is used to post-filtering the residual diffuse noise after a beamforming operation. As many audio enhancement systems for the automatic speech recognition or similar activities, it include two major components: the noise power spectral estimator and the speech power spectral estimator. The optimally-modified log-spectral amplitude (OMLSA) algorithm [22] is chosen as speech power spectral estimator, while, as noise power spectral estimator, the Improved Minimal Controlled Recursive Averaging (lMCRA) approach [28] is used .

Given the output of the beamformer $Y(k, l)$, the OMLSA algorithm

applies an adaptive gain function $G(k,l)$:

$$|\hat{Y}(k,l)|^2 = G(k,l)|Y(k,l)|^2, \tag{2.10}$$

where

$$G(k,l) = \frac{\xi(k,l)}{1+\xi(k,l)} \exp\left(\frac{1}{2}\int_{\nu(k,l)}^{\infty} \frac{e^{-t}}{t} dt\right), \tag{2.11}$$

$$\xi(k,l) = \frac{\sigma_x^2(k,l)}{\sigma_n^2(k,l)}, \quad \gamma(k,l) = \frac{|Y(k,l)|^2}{\sigma_n^2(k,l)}, \tag{2.12}$$

and $\nu(k,l) = \gamma(k,l)\xi(k,l)/(1+\xi(k,l))$. The noise variance $\sigma_n^2(k,l)$ is estimated using the IMCRA algorithm by recursively averaging past spectral power values of noisy speech using a smoothing parameter that is adjusted by the speech presence probability in subbands. The recursive averaging continuously updating the noise estimate even during weak speech activity. This allows to follow rapid changes in the noise spectral power. The IMCRA method is particularly suitable in non stationary noise environment, in presence of weak speech component and low input SNR.

In OMLSA, the optimal spectral gain function is obtained as a weighted geometric mean of the hypothetical gains associated with the speech presence uncertainty. The modified gain function takes the following form:

$$G(k,l) = [G_{H_1}(k,l)]^{p(k,l)} G_{\min}^{1-p(k,l)}, \tag{2.13}$$

where $G_{H_1}(k,l)$ is the same as Eq. (2.11), $p(k,l)$ is the *speech presence probability* (SPP) and $G_{\min}$ is a lower threshold [22]. The speech presence probability is computed as

$$p(k,l) = \left\{1 + \frac{q(k,l)}{1-q(k,l)}(1+\xi(k,l))e^{-v(k,l)}\right\}^{-1}, \tag{2.14}$$

where $q(k,l)$ is the *a priori* speech absence probability estimated using a soft-decision approach [22].

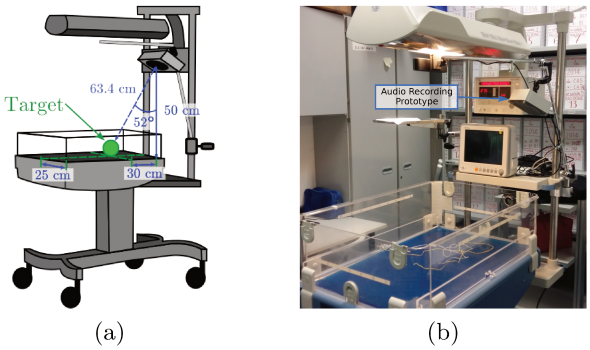
Chapter 3

Data acquisition

3.1 Case study

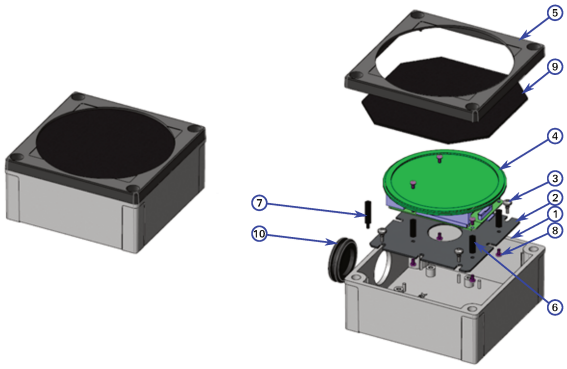
Cry detection technologies can support the NICUs medical staff by providing additional monitoring abilities integrated in the crib. However, NICUs typically present noisy acoustic environments [14, 16] that involves many challenges for an infant cry detection system. Concerning the acquisition of audio samples, a microphone array allows the use of multi-channel techniques (such as beamforming), to remove coherent noise sources, thus enhancing the audio signal by removing interfering noise sources. Concerning such a technology, the prototype at the basis of the present work is shown in Figure 3.1. Figure 3.1a present a scheme where the microphone array targeting the head of the infant is evidenced, while, in Figure 3.1b the actual prototype is presented. Moreover, in Figure 3.2, the exploded view of the prototype is shown.

The crib is a Draeger Babytherm 8004/8010 that has been equipped



(a) (b)

Figure 3.1: Cry detection crib prototype.



1	Fibox MNX123HB
2	MD284PPS0-01
3	Screw KN3.9x9.5
4	Matrix Creator and Raspberry
5	Fibox MNX12310C
6	Spacer Essentra HNSM2.5-15-5-1
7	Spacer Essentra HNSM2.5-20-5-2
8	Screw Essentra 0082110000VR Pi 3
9	Replaceable Speaker Cloth
10	Fairlead Essentra_19431

Figure 3.2: Prototype exploded

with the MATRIX Creator development board, which hosts a circular microphone array, featuring 8 digital MEMS microphones (model MP34DB02 by ST Microelectronics), distributed uniformly on a circumference with radius of 5.25 cm[1]. The microphone array is suspended above the crib by means of a supporting arm. The clamp that binds the array to the arm allows to tilt the array towards the head of the infant. The arm, on the other hand, allows for a partial rotation, to move the array whenever it hinders the caregiving activities. The crib is located in the NICU of the Salesi hospital in Ancona. Figure 3.3 shows the planimetry of the room in which are located 16 medical workstations that can accommodate both cribs and infant incubators. Each workstations is equipped with its specific set of medical instruments. Typically, 8 caregivers are present in the room during the normal daily work routine and several times a day (at least once in the morning and one in the evening), or when it's needed, 3 doctors and 2 interns visit the young patients. One parent for each baby can be present in the room during the day.

The complexity of the scenario requires a robust and effective cry detection method, able to overcome the interfering noises such as other infants' cries, the voices of the medical staff, the noise originating from the multiple devices in the NICU. The analysis of recorded data also revealed that among the issues there is the distortion due to the microphone array misplacement.

[1] www.matrix.one

3.2 Real dataset

The prototype described in the previous section was used to create a real dataset of infant cries. More than 900 hours of raw audio has been acquired and stored into 10-minute WAV files during the audio recording campaign, which lasted about 2 months. Over an half of recordings have been manually labeled in order to distinguish between cry and non-cry portions. It was not possible to identify the portions of the crying units corresponding to the inspiration phases, mainly due to environmental noise, but also due to the distance of the microphones with respect to the mouth of the newborns that was excessive for this purpose. The files that did not contain any cry sound have been discarded. The Real Dataset is composed of 3 hours of 16 kHz audio data, of which 45 minutes and 55 seconds are cry signals. As shown in Table 3.1, a total of 2 female and 3 male infants have been monitored. All the infants were born premature with gestational age between 28 weeks and 34 weeks and 2 days, while their age span from 2 days up to 208 days. All of them suffer, or have suffered, from some illness, including respiratory ones that are very common in preterms children. The dataset has been divided into 535 audio fragments with durations comprised between 2 and 150 seconds; the average duration is about 20 seconds.

3.3 Synthetic dataset

The synthetic dataset has been designed to simulate the acoustic environment of the target NICU. Based on the observations made in the target NICU hosting the crib, a room model has been created using the

Figure 3.3: The planimetry of NICU of Salesi hospital

Table 3.1: Real Dataset composition by subjects

Subject	Sex	Gestational Age $[weeks^{+days}]$	Age $[days]$	Audio Fragments	Recording Time $[s]$	Cry Time $[s]$
Baby 1	M	28	24	113	2280	930
Baby 2	M	34	2	259	4892	897
Baby 3	M	28^{+1}	208	108	2014	497
Baby 4	F	31^{+1}	34	24	606	62
Baby 5	F	34^{+2}	5	31	833	369

Python framework Pyroomacoustics [29], which allowed to emulate the impulse responses between the simulated audio source and the simulated microphone, by including the room geometry and reverberation in the calculation.

As shown in Figure 3.4, a simulated microphone array and a simulated crib meant to monitor a baby (show in blue) were placed in the model along with different types of noise sources. The microphone array has been based on the geometry of its real counterpart, with 8-channels placed on a circular pattern with a radius of 5.25 cm. Its position and orientation towards the audio source within the crib (target source) also matches the prototype geometry (i.e. towards the head of the monitored baby). In the model, the noise sources are placed within a radius of about 5.5 m from the microphone array. Three coherent noise source types and two incoherent noise source types have been considered. Among the coherent noise source types there are:

- human speech: it emulates the presence of the medical staff
- infant cry: it emulates other infants within other cribs nearby the target
- "beep" sound: it emulates the typical noises of a medical equipment.

Concerning the incoherent noise source types, the sounds of a fan and of an oxygen concentrator have been used. The sampling frequency is 16 kHz for all audio data. A total of 64 infant cry recordings belonging to 29 different subjects have been combined with 12 background realization, 23 beep sounds and 26 human speech recordings in order to create 64 audio sequences of 30 s that simulate realistic acoustic scenarios. Half of

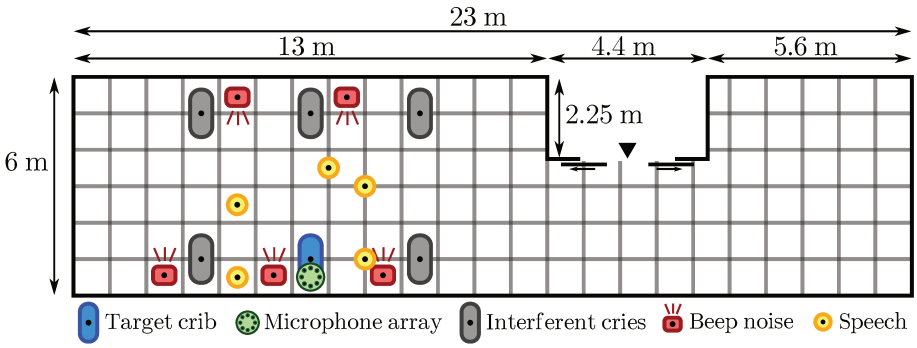


Figure 3.4: Plan of the NICU used to create the Synthetic Dataset.

simulated scenarios presents a SNR of 0 dB, while the other half presents a SNR of 5 dB. The total amount of cry signal is 15 minutes and 1 second, while the cry/silence ratio in each recording is about 50%.

The speech signals are extracted from a widely used mono clean speech dataset with American English sentences (WSJ0) [30]. All the other audio signals are collected from different web sources[2,3].

[2] `http://www.freesound.org`
[3] `http://www.youtube.com`

Chapter 4

Neural Beamforming for Speech Enhancement

In smart environments scenarios [31], automatic speech recognition (ASR) is widely employed and a cause of high word error rate is represented by the acoustic distortion due to additive noise and reverberation [32]. The automatic recognition of infant cry can be considered as a special kind of ASR designed specifically for infant's phonations, therefore, it suffering of the same issues. In the field of communication technologies, a number of algorithms has been already developed that can benefit from the use of multiple signal sources to enhance the perceived speech quality. Beamforming algorithms play a key role, being able to reduce noise and reverberation by spatial filtering. To that extent, an accurate knowledge of the Direction of Arrival (DOA) is crucial for the beamforming to be effective.

In this contribution a well known algorithm for DOA estimation, Multiple Signal Classification (MUSIC) [33], is first compared with a recently introduced neural approach [34] with the intent of verifying the effectiveness of the experiments conducted in [34] and providing additional experimental information. The results are very promising and motivated for application to a multi-channel speech enhancement scenario where the estimated DOA angle feeds a FS beamforming algorithm to improve the intelligibility of speech signals affected by noise and reverberation. The results are evaluated in terms of speech quality and compared with a well known speech enhancement method [35], showing a satisfying improvement employing objective evaluation methods. In other words, by using the neural DOA estimation in conjunction with beamforming, speech signals affected by reverberation and noise improve their quality.

These first contribution is reported to be taken as a reference for future works which aim to refine the approaches to infant cry detection proposed in the Chapter 5.

4.1 State of the art

Beamforming [36], as explained in the Section 2.2, is one of the most popular multi-channel approaches and it is ideally equivalent to steering the microphone polar pattern in the direction of the source. In delay-and-sum beamformer (DS), signals are aligned by taking into account the phase shifts among the microphones signals, while the filter-and-sum (FS) beamformer includes additional processing with linear filters. More advanced beamformers such as the Minimum Variance Distortionless Response (MVDR) [37] and the Generalized Sidelobe Canceller (GSC) [38, 39] adapt to the acoustic environment.

Nonlinear speech enhancement [40] based on deep neural networks (DNN) has also been recently proposed for enhancing speech from multiple microphone signals, in particular with the objective of improving the performance of ASRs. The advantage of this approach is that the parameters of the enhancement algorithm, i.e., the network weights, can be trained jointly with the ASR acoustic model, thus they are optimized under the same objective function. In [41], a DNN is employed as a speaker separation stage and the ASR employs bottle neck features as well as filter-bank coefficients extracted from multi-channel signals. In [42, 43], multi-channel mel filterbank coefficients are employed as input to an ASR based on convolutional neural networks (CNNs). Hoshen *et al.* [44] developed a similar approach but using raw waveforms as input to the network. Actual DNN-based beamformers for ASR have been proposed in [23, 24]. In [23], the algorithm operates on multiple complex-valued short-time Fourier transforms (STFTs) to estimate a single enhanced signal. The network operates on generalized cross correlation (GCC) coefficients to estimate the weights of a FS beamformer that then processes the STFTs related to the microphone signals. The algorithm is employed as a preprocessing stage of an ASR and it can be trained jointly with the acoustic model in order to further optimize the process. A similar approach has been proposed in [24],

where the spatial filter coefficients are estimated by a long short-term memory (LSTM) network [25] that takes raw waveform signals as input, instead of STFT coefficients. In [26, 27], the authors employ a DNN and a Bidirectional-LSTM for estimating the time-frequency mask in the MVDR beamformer.

Up to the author's knowledge, few works propose DNN-based algorithms targeted at enhancing the quality and the intelligibility of the perceived speech. In an early work [45], a single-layer perceptron filter is introduced in the GSC beamformer framework to suppress noise. In [46], the work is extended by introducing alternative structures of the noise reduction algorithm. In a more recent work [47], the authors employ a denoising autoencoder with multi-channel features. In particular, they augment single-channel log-mel filter-bank features with information extracted from multiple channels. The authors evaluated the noise suppression performance with segmental signal-to-noise ratio (SSNR) and the cepstral distortion (CD) measures, and showed that employing pre-enhanced speech features their approach improves with respect to the single-channel autoencoder.

These works suggest that investigating on the capabilities of neural networks to improve current state-of-art speech enhancement algorithm may be fruitful.

4.2 Proposed approach

In this contribution, the MUSIC method is taken as a baseline and it is compared to a more recent and promising technique for neural DOA estimation based on machine learning reported in [34], from now on referred to as NDOA. NDOA follows a data-driven approach, where a corpus of data is provided during the training phase to estimate the algorithm parameters. A feature-set is first extracted and treated as input to a multi-layer perceptron (MLP). The output of the MLP is the DOA estimate required by subsequent stages (e.g. beamforming). Figure 4.1 resumes the complete algorithm, including beamforming and speech quality evaluation.

Among possible feature-sets, the generalized cross-correlation coefficients (GCC) are employed for their wide acceptance in the field of DOA

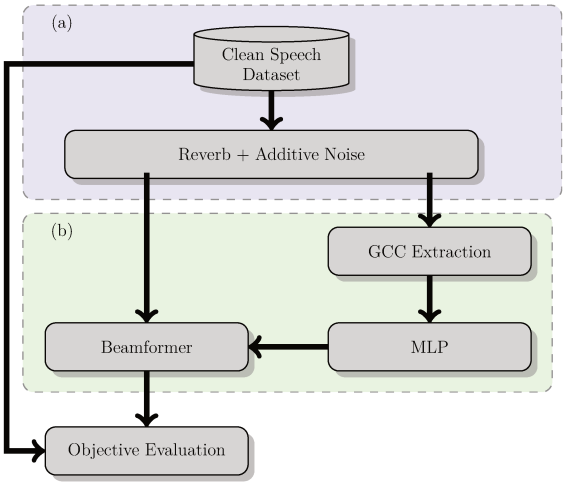


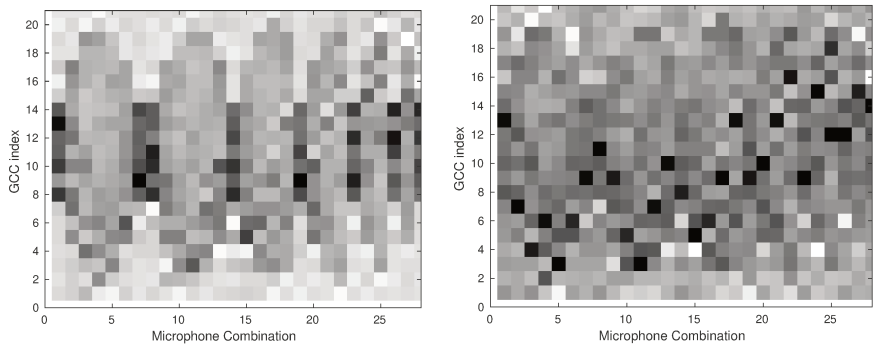
Figure 4.1: Flow diagram of the dataset generation (a) and neural DOA estimation (b). The clean speech is finally compared to the processed speech for the objective evaluation.

Estimation and their ability in capturing phase related information [34]. Specifically, the GCC are more reliable compared to time difference of arrival (TDOA). In this contribution, the GCC-PHAT algorithm [48] is used to extract GCC vectors, based on the cross-correlation of spectral coefficients between all microphone pairs. For each microphone pair combination C, only a part of the GCC values are taken, depending on the microphones distance. Let D be the maximum distance between microphones in the array, the time delay is $\tau = D/c$ seconds, or $N = F_s \cdot \tau$ samples. Under such conditions only the center $2N + 1$ GCC values contain useful information, i.e. those values corresponding to delays in the range $\pm N$ samples. The rest of the GCC values can be discarded. In more rigorous terms, the cross correlation between the power spectra of any two microphone signals in the array, i.e. $S_{12}(f)$, is defined as:

$$S_{12}(f) = X_1(f) * \mathring{X}_2(f)^*, \tag{4.1}$$

where $\mathring{X}_2(f)$ is a circular-shifted version of the FFT of $x_2(t)$ needed to operate with coherent GCC vectors. The circular shift is of N samples, thus the only GCC containing information are $S_{12}(0 \div 2N + 1)$.

To improve the robustness of the GCC, a further processing stage,



(a) (b)

Figure 4.2: GCC matrix extracted from a speech frame in the dataset before applying HEQ (a) and after (b).

histogram equalization (HEQ) is undertaken. This improves reliability of the GCC by increasing the spread between noisy and useful coefficients. Figure 4.2 shows a GCC feature set before and after histogram equalization.

Finally, under stationary conditions for the sound source position, averaging of the GCC can be done between successive frames. Under testing and training this is implemented by averaging all coefficients from the same file in the dataset, which is known to be stationary. In real-world conditions, under the reasonable hypothesis of a slowly time-varying position, a moving average can be employed, with a suitable averaging window.

Once feature extraction is completed, the features are given as input to a multilayer perceptron (MLP). The MLP employed for DOA classification has an input layer with nodes equal to the input feature dimensions $C(2N + 1)$. One hidden layer is employed, with a sigmoid activation function. Differently from [34] the output layer is fed to a nonlinear combination neuron which outputs a continuum estimate of the DOA. In the original paper classes of 1 degree or more were used, making difficult to compare different experiments with varying noise and reverberation. The activations of the hidden nodes are converted to DOA class posterior probabilities by using a linear transformation and a softmax activation function. The stochastic gradient descent (SGD)

algorithm is used to train the MLP iteratively.

A voice activity detection (VAD) algorithm [49] is employed to discard GCC from audio frames not containing speech, that would harm learning.

4.3 Dataset

The generation of the dataset for training and testing follows previous works in the field. Specifically, the multichannel noisy speech is generated from a widely used mono clean speech dataset with American English sentences (WSJ0) [30]. All speech signals are sampled at 16 kHz. The speech signals are transformed into multichannel signals with noise and reverb, in order to test the robustness of the proposed approach.

The generation of the dataset assumes a microphone displacement following a uniform circular array (UCA) of 8 microphones, with a radius of 0.1 m. Different configurations of reverberation and additive noise are introduced to simulate different use cases, and they have been applied according to the following rules:

1. Reverberation was added employing the RIR Generator[1] tool for Matlab, based on [50].

2. Three reverberation schemes were used, simulating small, medium and large rooms, each one with the speaker in the far field or near field. One shortcoming of the work in [34] was that three different datasets were created depending on the room size, and each one was evaluated on its own.

3. The direction of arrival in the UCA were randomly selected.

4. Other randomly selected parameters were: speaker distance, room dimension, T60 reverberation time, SNR, noise type.

5. The added noise came from noise samples provided by the REVERB CHALLENGE dataset and its SNR was selected randomly from 0 to 20 dB. 40 different types of noise were used, divided in simulated rooms, according to the Reverb Challenge [51].

[1] `https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator`

	TRAINING	TESTING
content	7768 sentences from the WSJ0 training set	507 sentences from the WSJ0 test set
Room size (m)	small (7x5), medium (12x10), large (17x15)	small (6x4), medium (10x8), large (14x12)
Distance (m)	near (1) and far (2, 4, 6.5 for small, medium, large)	near (1) and far (1.5, 3, 5 for small, medium, large)
T60 (s)	0.1s to 1.0s with 0.1s step	three steps: 0.3, 0.6, 0.9 s
SNR (dB)	randomly selected from 0 to 20dB	randomly selected from 0 to 20dB

Table 4.1: Training and Testing datasets details.

The microphone array displacement affects some of the algorithm parameters. Specifically, the maximum distance $D = 0.1\ m$ implies a delay of $N = 10$ samples, and the selection of 21 GCC coefficients. The array of 8 microphones allows $C = 28$ combinations of signals to compute the GCC. A total of 588 features are, thus, computed for each frame and are fed as input to the MLP. The signals sampling frequency also affects the frame size employed for FFT of the input signals. In our implementation the frame size chosen was 0.2s with 50% overlap.

Training and Testing sets were organized as shown in Table 4.1.

4.4 Experimental set-up

Traning and testing has been performed using Keras [2] running on Theano as a backend [3], while all the audio preprocessing was done in Matlab. For each MLP parameter set, the training was done over 5000 epochs, interleaved with periodical validation every 1000 epochs. MLP weights were taken from the validation obtaining the lowest RMS error.

Experimenting all possible parameter sets is not feasible as it would require a large number of very time-consuming experiments. To reduce the number of trials, discrete steps have been used for all numerical parameters. Furthermore, a heuristic procedure has been employed to look for a sub-optimum. Its first step consists in conducting several experiments by varying a single parameter, with all other parameters fixed to a initial value. The value of the parameter under test yielding best results is taken and the procedure is repeated for another parameter until all parameters have been experimented with. Finally, a number of

[2] `https://keras.io/`

[3] `http://deeplearning.net/software/theano/`

random experiments are conducted to gather more confidence that there are no other RMS error minima below the one previously found. Details regarding the experiments follow:

- MLP network size: from 80 to 512 in discrete steps;
- MLP update rule: stochastic gradient descent (SGD), Adam, AdaMax [52];
- activation functions: tanh, rectified linear unit (ReLU);
- mini-batch: 1 to 3000 in discrete steps;
- learning rate: 1e-9 to 1e-5 in discrete steps;
- momentum: 0.8, 0.9.

During preliminary tests, the latter three parameters were shown to yield improved performance with values, respectively, 3000, 1e-8, 0.9, notwithstanding the choice of the former three parameters. Furthermore, preliminary tests show that the first three combinations of the activation functions leading to good results are the following:

1. A: (tanh, tanh, tanh);
2. B: (tanh, ReLU, tanh);
3. C: (ReLU, ReLU, tanh).

Choice of MLP network size, weight optimization algorithm and activation functions has been done according to the heuristic procedure described above. Some results are reported in Table 4.2. The activation functions combination yielding the best results in first place is B. AdaMax, is found to be the best optimization algorithm and a network size of (588, 160, 80) largely improves performance.

4.5 Results and remarks

After training and selection of the best parameter set, a first evaluation has been conducted on the DOA estimation algorithm with respect to the MUSIC algorithm. The results are extremely convincing as the error decrease is larger than one order of magnitude:

Net Size	Activations	Optimizer	RMS Error
(588,100,80)	C	SGD	11.95
(588,100,80)	A	SGD	11.37
(588,100,80)	B	SGD	11.09
(588,100,80)	B	Adam	14.92
(588,100,80)	B	AdaMax	10.02
(588,250,200)	B	AdaMax	16.79
(588,160,80)	B	AdaMax	**4.48**

Table 4.2: Some of the MLP parameter sets employed during training and the RMS error obtained during testing of the related parameter set. The RMS error is expressed as the difference in angle with respect to the correct DOA. The last layer has dimension 1 and outputs a floating point value.

1. MUSIC RMS Error: 122.8.

2. NDOA RMS Error: 4.48.

The excellent capability of the NDOA algorithm to track the DOA is shown in Figure 4.3, where DOA estimation errors are reported from an excerpt of 75 randomly selected sentences. Error values of NDOA are up two orders of magnitude below MUSIC. This motivates for application of NDOA to many scenarios, such as speech enhancement. To verify the effect of the improved accuracy of DOA estimation, both NDOA and MUSIC are first applied to a FS beamforming algorithm and the resulting speech quality is evaluated. Processed audio evaluation is carried on by employing two speech quality measures [53]: Perceptual Evaluation of Speech Quality (PESQ) and Itakura-Saito distance (IS). The former is defined as a standard for speech quality assessment for communication technologies, standardized as ITU-T recommendation P.862(02/01)[4]. It provides off-line evaluation of speech signals quality by amplitude and time alignment, in order to provide a meaningful sample-by-sample comparison of the original signal and the processed signal. Furthermore it makes use of auditory transforms and cognitive models to predict a human Mean Opinion Score (MOS) speech quality assessment. The Itakura-Saito distance, on the other hand, is not a perceptual measure and it provides a measure of the difference between two spectra, in this

[4] `http://www.itu.int/rec/T-REC-P.862/en`

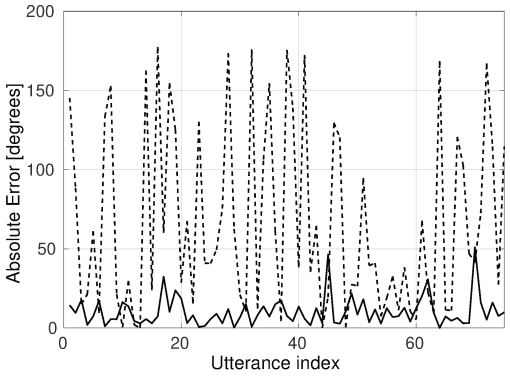


Figure 4.3: DOA estimation RMS Error for MUSIC (dotted line) and NDOA (solid line).

	NONE	MUSIC	NDOA	SE	NDOA + SE
PESQ	1.74	1.8	1.89	1.88	**1.95**
IS	3.4	3.55	3.49	3.28	**3.11**

Table 4.3: Speech Quality comparison between unprocessed speech (NONE), beamformed speech with NDOA (NDOA), beamformed speech with MUSIC, speech enhanced with [35] (SE), speech enhanced with NDOA beamforming and [35] (NDOA+SE). Please note that with IS, lower values correspond to better performance.

case the original signal and its processed version.

The results, reported in Table 4.3 and compared to the original speech source (with noise and reverb applied), show that the higher DOA estimation accuracy achieved by the NDOA improves also the speech signal quality. Motivated by these findings, we tested whether the NDOA can further improve speech quality when applied in conjunction to an established speech enhancement technique by Ephraim et al. [35] (in short SE).

These results are summarized in Table 4.3. The speech quality improvement obtained by the combination of both SE and NDOA, compared to SE only, is of 50% when evaluated with PESQ and of 70% when evaluated with IS. This confirms the validity of the approach.

The experiments reported in this contribution show that neural DOA estimation exhibits excellent performance with respect to a reference technique such as MUSIC. When used in conjunction with a classic

beamforming algorithm, its higher precision also improves its capability in enhancing the quality of speech affected by noise and reverberation with respect to a MUSIC DOA estimator in conjunction with the same beamforming algorithm. The performance, evaluated in terms of both PESQ and Itakura-Saito distance, is further increased in conjunction with a well-known speech algorithm by Ephraim et al.

These preliminary results in the field of speech enhancement motivate for further research in neural beamforming in order to specialize this strategy for the infant cry detection task. More recent machine learning algorithms can be applied to DOA estimation. For instance, while the computational cost of a MLP network is lower compared to most RNN techniques, these may potentially yield improved results in accuracy that are worth investigating. End-to-end learning could be applied, resulting in a whole beamforming architecture based solely on machine learning. Following such approach, a deep neural network could be trained to cover both DOA estimation and beamforming.

Chapter 5

Infant Cry Detection with Deep Neural Network

5.1 State of the art

Cry Detection has been already addressed in the literature. Few automated methods, are based on classical or parametric approaches [54–57]. More recently, machine learning methods have been proposed on purpose [58–62].

Orlandi *et al.* tackle the problem at the roots proposing an automatic method for extracting "voiced candidate" from audio recordings of long duration for both clinical and home applications. The method is reliable also for detect infant cries [63]. First of all, the system perform a filtering of inputs with a Butterworth filter of order 5 and a cut-off frequency of 50–1000 Hz since most common human laryngeal sounds have main components in theis range of frequencies. Then, the signal are down-sampled at 11.025 kHz To deal with the non stationarity of human vocalizations, the pre-processed signal is divided into windows of 20 ms overlapped for 50%. At second stage the detection process are carry out with the computation of Short-Term Energy (STE) measure. The threshold that discriminate "voiced" portions from "un-voiced" portions of signals, is selected with a nonparametric and unsupervised approach named Otsu's method. To avoid incorrect splitting of a single event into several intervals a double thresholding scheme is used. In the paper, the method has been tested on a sinthetyc dataset composed of adult voices and newborn cry. Four different kinds of noise were added to the signal, i.e. white noise, brown noise, background voices and air conditioner noise.

Also in [56], the infant cry detection is performed exploiting the short-

time energy measure which is compute over rectangle windows of 50 ms. The threshold is empirically chosen as a quarter of the average short-time energy measured in whole recording. In this work, the detection task is subordinated to the identification of qualitative features for cry analysis. For this purpose, either cries with a length less than 200 ms and inspiratory cries are not needed and then have to be removed. On a successive paper of the same authors [64], the simple classical approach just described has been tested on two small sets of samples, from the Chillanto data base from the Instituto Nacional de Astrofísica óptica y Electrónica. The records where obtained from healthy infants with different devices and in different ambiences. The results shows an acurracy of 96,15% on average.

The approach proposed by Cohen and Lavner [58] is an algorithm for cry detection which is aimed at alerting parents about potentially dangerous situations, such as when infants are being left alone (either in apartments or vehicles). For this reason, a synthetic corpus including car horns, car engines, passers-by and others street noises, has been used to evaluate the algorithm. Moreover, since the algorithm is designed to be performed on mobile devices, in order to limit their energy consumption, the first stage of the proposed approach is a voice activity detector, which disabled the operation of the algorithm when voice activity was not present. The Mel-frequency cepstral coefficients (MFCCs) are extracted from those fragments that contain more than 30% of audio activity. A K-nearest neighbors (k-NN) classifier is used to have a first classification of 16 msec long fragments as cry or not cry. If a fragment is classified as no cry, a post-processing stage is conducted, where some harmonic feature of the signals (Pitch frequency, Harmonicity Factor and Harmonic-to-Average Power Ratio) are checked in order to revalue the initial decision. The algorithm also implement a smoothing strategy for keeping a low rate of false positives. The results showed good performance of the proposed algorithm, even at low SNR.

Several works address the issue by means of methods based on Hidden Markov Models (HMMs), such as the ones proposed in [60, 61]. In [61], Abou-Abbas *et al.* present an ICD system based on Hidden Markov Models for the discrimination of expiratory and inspiratory parts of cry. The system is trained and tested on a real database collected in the

neonatology departments of several hospitals. The data acquired comprend noises of many types, such as human speech (nurses, doctors, parents), the sounds of the recording device and medical equipment in the neonatal Intensive Care Unit (the beeping of machines), and occasional noises like doors slams and running water. In the group of infants who have been recorded there are both preterm and full term subjects of both gender. Some of them suffered of some pathological conditions (central nervous system diseases, blood disorders, congenital cardiac anomaly, respiratory system diseases, chromosomal abnormality). All the recordings were acquired with a digital 2-channel recorder hand-helded at 10 to 30 cm from the newborn's mouth. The recorded audio signals are registered at a sampling frequency of 44.1 kHz and sample resolution of 16 bit. The total duration of the recordings is about of 6 hours and 5 minutes. A vector composed of 12 MFCCs along with the relative deltas and delta-deltas coefficients is extracted from pre-emphasized data. A different HMM is trained for six classes: Expiration, Inspiration, BIP sounds, Silence and Noise. The experimental results indicate that the system yields accuracies of 83.79% for the detection of expiratory phase and for inspiratory phase 77.93%.

In a successive work of the same authors [59], the feature extraction is performed starting from three different signal decomposition approaches: FFT, wavelet packet decomposition (WPD), and empirical mode decomposition (EMD). Subsequently, the MFCCs are computed for the FFT and EMD signals, while for the wavelets, the discrete cosine transform has been applied. The EMD+MFCC features results the most performing with HMM classifier.

In [60] Naithani *et al.* also present a multiclass approach based on HMM able to distinguish between expiratory and inspiratory phases of cries and residual acoustic regions (i.e. non-cry segments). The datasets used were recorded in real clinical environments (corridor, normal pediatric ward, NICU, waiting room, and nurse's office) and include acoustic material from two cohorts of infants, one in Tampere, Finland, and the other in Cape Town, South Africa. A total of 3 hours and 10 minutes of audio data was stored in WAV audio files with 48 kHz sampling rate and two audio channels of 24 bit depth. The authors examine different HMM topologies fed with different combinations of features: MFCCs

with deltas and delta-deltas, fundamental frequency (F0) and aperiodicity. The results show that the segmentation system works sufficiently well for expiratory phases but performs rather poorly for inspiratory phases. In terms of F-score, for the expiratory and inspiratory phases, they obtain respectively 83.3% and 48.9% for the Tampere dataset and 78.0% and 39.2% for the Cape Town dataset. Moreover, in the paper is shown a method to adapting system trained on acoustic material captured in a particular acoustic environment to a different acoustic environment by using feature normalization and semi-supervised learning (SSL). This adaptation method yield F-score up to 73.2% when the system is trained with the Tempere dataset and tested on Cape Town dataset.

In [62] three different approaches for infant cry detection designed specifically for home environment are presented. The authors make a comparison between a revised version of the deep neural network (DNN) presented in [65], modified with the introduction of dropout and batch normalization layers, a baseline approach employs MFCCs and a support vector data description (SVDD) as classifier, and a novel set of hand crafted baby cry (HCBC) features. The effort is driven to the development of a computationally light algorithm suitable for a low-power device for audio event detection, able to capture the baby crying and automatically alert his parents. The database employed for the experiments includes 102 baby cry sound events (1 h and 07 m) and 93 non-baby cry (1 h and 24 m) i.e. tv, toy, adult cry, baby talk/play, music, fan, vacuum cleaner which are used for modeling target and non-target class. The comparison shows for the method that exploits HCBC features a performance level equivalent to that obtained for the DNN but with less computational and memory demanding. The drawback of HCBC approach seem to be the high cost in designing specific domain features.

Up to the author's knowledge, no works explicitly asses the infant cry detection in adverse environments like neonatal intensive care units (NICUs).

5.2 Proposed approaches

Considering the system proposed in Section 3.1, a block-scheme of the proposed approach is shown in Figure 5.1. Acoustic signals are acquired with an eight-channel circular microphone array and processed by a filter-and-sum LCMV beamformer for reducing coherent noise source, followed by the OMLSA post-filter that reduces residual diffuse noise. The feature extraction stage calculates the Log-Mel spectrogram, whereas the DNN takes in multiple frames and classifies the central one as a cry frame or not, exploiting the information from temporally adjacent frames.

The investigation of the collected audio signals, however, revealed a few concerns relating the microphone array position and orientation. As an example, during the medical staff activity, the arm supporting the microphone has been moved to the side and has not been restored to in its original position for a prolonged period of time. This issue is representative of the unpredictable nature of a NICU environment. In fact, during the first steps in the development of this research, although a part of the collected data has been used to evaluate the cry detection method, the data did not present this issue, thus resulting in a good performance. Further investigation on the collected data showed a performance drop, revealing the issue described above.

Some alternatives to the original approach have been investigated, and a representative scheme is depicted in Figure 5.2. One of the revised approach operates on a single-channel, without additional preprocessing. The second approach under evaluation exploits 3 channels, among the 8 provided by the array, as input of a DNN, thus incorporating the multi-channel processing directly into the DNN.

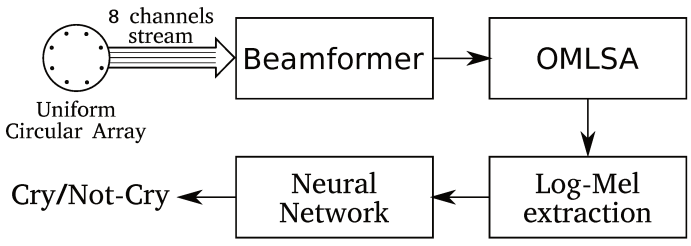


Figure 5.1: Block-scheme of the proposed approach

Furthermore, as shown in Figure 5.3, the observation over the spectrum of the audio stream, presented in Figure 5.3a, reveals that the cry signals (Figure 5.3b) occupies all the frequency components up to 8 kHz [11,66], whereas most of the noise types, such as the "beep" noises (Figure 5.3c) produced by medical equipment and the interfering voices from the medical staff (Figure 5.3d), affect mostly the signal frequency components below 4 kHz.

To investigate this spectral feature, we also included an evaluation restricted to the 4-8 kHz frequency band, by discarding the lower frequency features of the Log-Mels, as described in Section 5.2.1.

5.2.1 Feature extraction

All the methods presented above share the same feature extraction stage. For each input channel of the neural network, the signal is divided in frames 20 ms long and overlapped by 10 ms. The Fast-Fourier Transform of the frame is then filtered with a filter-bank composed of 40 triangular filters equally spaced in the mel-space. The mel scale is a perceptual scale of pitches judged by listeners to be equal in distance from one another. The correlation of the mel scale to the physical frequency is nonlinear, like the human auditory system. Figure 5.4 show the filter-bank in the frequency domain. The scale is roughly linear below 1000 Hz, and then decays logarithmically. An empirically formula to convert f hertz into mels is:

$$Mel(f) = 2595log\left(1 + \frac{f}{700}\right) \tag{5.1}$$

Log-Mel coefficients are widely used acoustic features in audio analysis with Convolutional Neural Networks, since they allow a compact representation of the audio signals while retaining discriminative information [67,68]. By calculating the energy in each band, and applying the

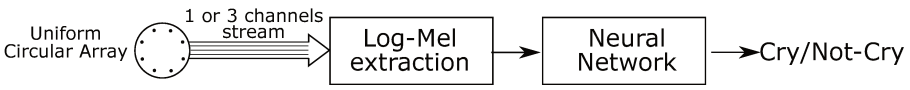


Figure 5.2: Block-scheme of the single-channel and multi-channel approaches.

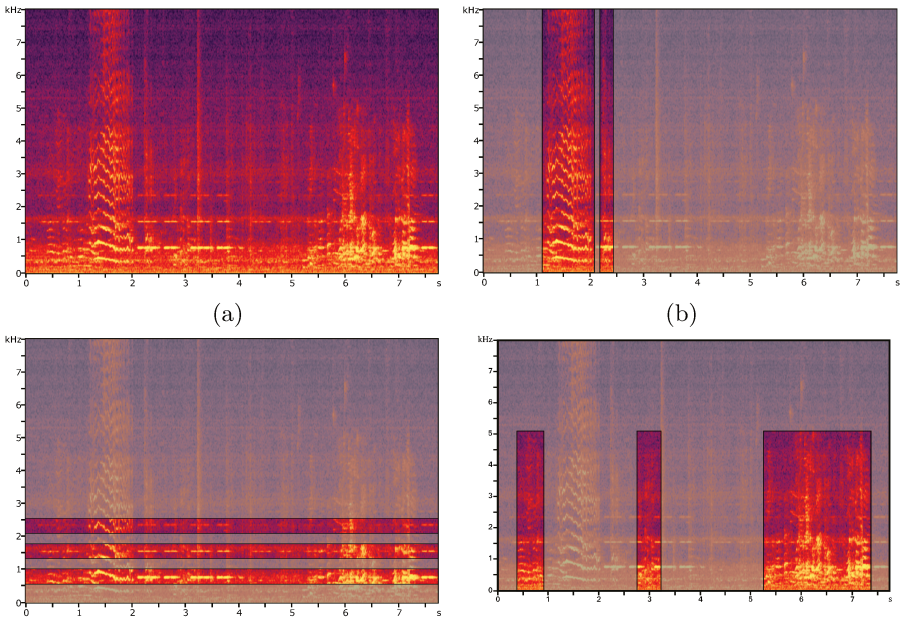


Figure 5.3: Audio sample spectrogram: (a) full spectrogram, (b) cry target detail, (c) "beep" noise detail, (d) interfering voice detail

logarithm operator, the Log-Mels coefficients are obtained, in the form of a feature vector composed of 40 elements. This approach (Full-band) is used to investigate both the raw and enhanced signal. To remove the signal frequency components below 4 kHz (Half-band), on the other hand, the corresponding coefficients are omitted, thus halving the vector length.

The classifier does not operate on individual feature vectors, but it exploits the temporal information contained in adjacent frames. The input of the neural network is thus a $(2N + 1) \times C$ matrix, where N is the size of the temporal context, i.e., the number of frames preceding and following the frame being classified, whereas F is the number of features being used. In this work, N has been set to 99 frames, corresponding to about 1 s.

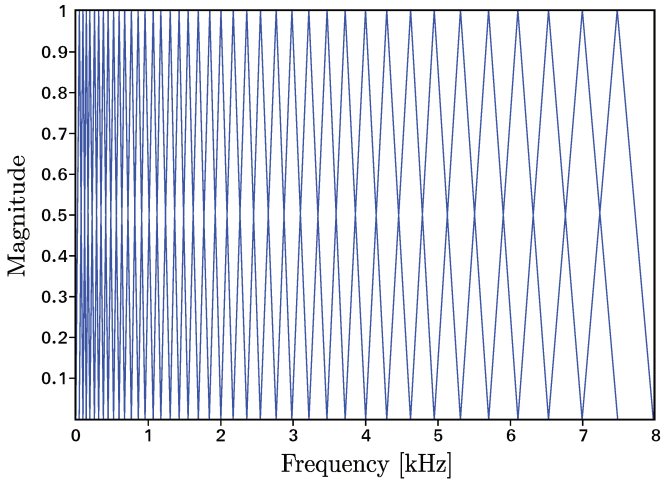


Figure 5.4: Mel-frequency filterbank.

5.2.2 Single-channel DNN approach

The single-channel neural network architecture (1Ch-DNN) used for cry detection is shown in Figure 5.5. The exact topology of the network is defined through the search of the hyperparameters by using a validation set (Section 5.4). Its general structure is defined as follows: the first part of the network consists in one or more convolutional layers, each followed by batch normalization [69], rectifier linear unit (ReLU) activation function, dropout and max-pooling operator. The output of convolutional layers is processed by one or more fully connected layers, each followed by batch normalization, ReLU activation function, and dropout. The output layer is composed of a single neuron, with a sigmoid activation function, that outputs the probability of the central frame of being a cry. The network training is performed by minimizing the binary cross-entropy loss with the Adam algorithm [69].

The hyperparameters related to the network topology that are determined in the experimental phase are the number of convolutional and fully connected-layers, the size and the number of the kernels of convolutional layers, the size of the max-pooling operator, the dropout rate, the number of units in the fully-connected layers, as well as the learning rate, and the batch size used to train and validate the network.

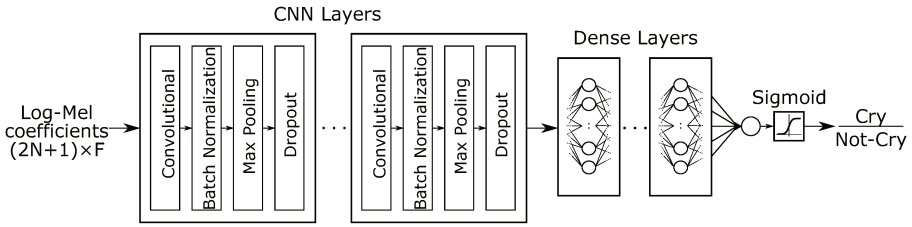


Figure 5.5: Single-channel DNN architecture used for cry detection.

5.2.3 Multi-channel DNN approach

The multi-channel neural network architecture uses 3 input channels (3Ch-DNN) and is shown in Figure 5.6. In this case, as well, the exact topology of the network is determined in the experimental phase by using a validation set (Section 5.4). The general structure, however, is defined as follows: the first part of the network consists of three identical blocks, each with one or more convolutional layers, each followed by batch normalization [69], rectifier linear unit (ReLU) activation function, dropout and max-pooling operator. These three blocks, share the same exact topology and operate in parallel. Each channel input corresponds to a specific microphone of the array, that is, the first, the fourth and the seventh.

The outputs of the three blocks are then placed side by side. At one time each block produces one frame, the three frames are then merged in a single frame with the same row number of the original frames and three times the number of columns of the original frames. The resulting frame is then processed by an additional convolutional layer also followed by batch normalization [69], ReLU activation function, dropout and max-pooling operator. The output of this convolutional layers is then processed by one or more fully connected layers, each followed by batch normalization, ReLU activation function, and dropout. The output layer is composed of a single neuron with a sigmoid activation function. that outputs the probability of the central frame of being a cry. Training of the network is performed by minimizing the binary cross-entropy loss with the Adam algorithm [69].

The hyperparameters related to the network topology that are determined in the experimental phase are the number of convolutional and fully connected-layers, the size and the number of the kernels of convo-

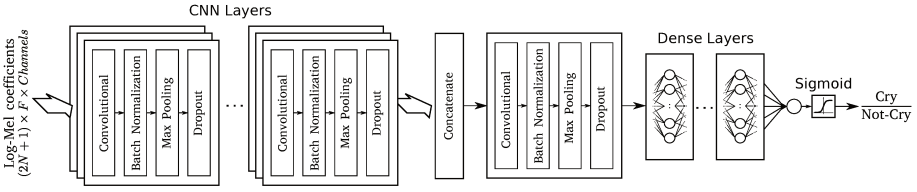


Figure 5.6: Multi-channel DNN architecture used for cry detection.

lutional layers, the size of the max-pooling operator, the dropout rate, the number of units in the fully-connected layers, as well as the learning rate, and the batch size used to train and validate the network.

5.2.4 Signal enhancement approach

In the Signal Enhancement approach (SE-DNN) the audio signals are processed according to the block-scheme shown in Figure 5.1. A 8-channels circular microphone array is used to acquire the audio samples, which are then processed by a filter-and-sum adaptive beamformer and a OMLSA post-filter to enhance the signal quality and reduce the noise. Following the enhancement a feature extraction stage computes the Log-Mel spectrogram whereas the DNN classifies the frames as cry or non-cry frame. In this case, also, the general structure of the neural network matches the one that has been already presented in Section 5.2.2. The exact topology, in this case as well, has been determined in the experimental phase by using a validation set (Section 5.4).

5.3 Comparative method

The proposed approach has been compared to the work by Raboshchuk et al. in [70], that describes a complete pipeline that handles the acoustic data recorded in NICUs and performs the vocalizations detection task. A "vocalization" is a sound produced through a vocal tract. The first stage of the comparative method performs the audio enhancement of the input signals by applying, in sequence, the Non-negative Matrix Factorization (NMF) and the spectral Subtraction (SS) methods. The vocalization detection is then performed by a Gaussian Mixture Model (GMM) based detector.

The NMF algorithm is used in order to reduce the non-stationary noises. Denoting with $V_{F\times N}$ the matrix representing the spectrograms of the input signals, where F are the frequency bins and N the number of frames, it is possible to approximate it with two non-negative matrices:

$$V_{F\times N} \approx W_{F\times R} \cdot H_{R\times N}, \tag{5.2}$$

The columns of W should be intended as bases, while the rows of H as their corresponding activations in each frame. It fallows that $R \leq F$. NMF attempts to find the matrices W and H through the solution of the minimization problem:

$$\text{argmin}_{W,H} D(V||WH) + \lambda |H|_1 \quad W, H \geq 0 \tag{5.3}$$

where D is Kullback-Leibler divergence and $\lambda \leq 0$ is used to promote a sparsity constraint on the activations. For each source, the matrix of bases is estimated on a training dataset and then are used at source separation step of the whole dataset. In the cry detection problem the ensemble matrix of bases $W_t = [W_{Cry}; W_{No-Cry}]$ is kept fixed in equation 5.3 in order to estimate the matrix $H = [H_{Cry}; H_{No-Cry}]$ for the test dataset. The spectrum of each source can be obtained as

$$\widehat{V}_i = \frac{W_i H_i}{\sum_i W_i H_i} \otimes V, \quad i \in [\text{Cry, No-Cry}] \tag{5.4}$$

where multiplication $\otimes$ and division operations are element-wise. The output enhanced signal is obtained joining the spectrogram $\widehat{V}_{Cry}$ and the phase of the original input audio.

The SS algorithm is applied in order to attenuate the stationary noise contributions. The clean signal spectrum $\widehat{X}(n,k)$ can be estimated from the noisy input spectrum $Y(n,k)$ by subtracting an estimate of the noise spectrum $D(n,k)$:

$$|\widehat{X}(n,k)|^\gamma = \begin{cases} |\widehat{Y}(n,k)|^\gamma - \alpha|\widehat{D}(n,k)|^\gamma, \\ \quad \text{if } |\widehat{Y}(n,k)|^\gamma > (\alpha+\beta)|\widehat{D}(n,k)|^\gamma \\ \beta|\widehat{D}(n,k)|^\gamma, \quad \text{otherwise} \end{cases} \tag{5.5}$$

where n is the frame index, k the frequency bin, $\gamma = 2$ corresponds to perform a power spectrum subtraction, α is the subtraction factor

and $0 < \beta \ll 1$ is the spectral floor parameter. The noise estimate is obtained by employing the Minima-Controlled Recursive-Averaging (MCRA) algorithm [28]:

$$\begin{aligned} |\widehat{D}(n,k)|^{\gamma} = \alpha_d(n,k)|\widehat{D}(n-1,k)|^{\gamma} \\ +(1-\alpha_d(n,k))|\widehat{Y}(n,k)|^{\gamma}, \end{aligned} \tag{5.6}$$

with $\alpha_d(n,k) = \alpha + (1-\alpha)p(n,k)$, where $p(n,k)$ is the speech-presence probability calculated exploiting the ratio between the noisy signal spectrum and its local minimum. The ratio is first smoothed by a factor α_s and then compared to a certain threshold value, where a higher ratio indicates presence of speech. Subsequently, a recursive temporal averaging is carried out, to reduce fluctuations between speech and non-speech segments.

A feature vector composed by 16 Frequency-Filtered Logarithmic FilterBank Energy (FF-LFBE) coefficients, along with their 16 first temporal derivatives, is extracted from the enhanced audio signals divided into frames using 30 ms long Hamming windows, overlapped by 10 ms.

Vocalization detection is performed by a single Gaussian probability density function with a diagonal covariance matrix used to model each class, i.e., Cry and Non-Cry.

5.4 Experimental set-up

The methods described in Section 5.2 have been implemented by means of the Python programming language. The DNNs have been implemented by means of the Keras framework, using Tensorflow as the back-end, whereas the feature extraction has been carried out by using *librosa* [71]. On the other hand the reference method, described in Section 5.3, has been provided by the authors, and is implemented by means of Matlab computing environment and the HTK toolkit [72] which has been used to build the GMM classifier.

By combining the three methods described in Section 5.2 and the two feature extraction procedures, we investigated five cry detection strategies, namely:

- Full-band - 1Ch DNN: single-channel DNN with full feature vector

as input

- Full-band - 3Ch DNN: multi-channel DNN with full feature vector as input
- Half-band - 1Ch DNN: single-channel DNN with half feature vector as input
- Half-band - 3Ch DNN: multi-channel DNN with half feature vector as input
- SE-DNN: single-channel DNN with signal enhancement and full feature vector as input.

To evaluate the strategies described above, and to define the topology for each of the DNN networks used in the evaluation, a random search approach [73] has been used, defining a pool of 300 configurations, based on the hyperparameter distributions and ranges reported respectively in the first and second column of Table 5.1. To this purpose, the synthetic dataset has been divided in 4 folds with the same number of audio sequences, corresponding to the 25% of the dataset each. On the other hand, the real dataset has been divided in three parts. One third has been used as test set, whereas the remaining part has been further divided in training set (75%) and validation sets (25%), corresponding to 49.5% and 16.5% of the whole real dataset respectively.

The experimentation consists of three main phases:

- cross validation conducted on the synthetic dataset: 3 folds have been used as a training set and 1 fold has been used as a validation set; the process has been carried out for each of the strategies and each of the configuration; the best performing topology for each of the strategies, has been reported in Table 5.1 and discussed in Section 5.5;
- training conducted on the real dataset, followed by a test on the real dataset: each of the network configurations reported in Table 5.1 have been trained and validated by means of the training and validation set of the real dataset; the test has been conducted on test set of the real dataset and the results have been discussed in Section 5.5;

Table 5.1: Hyperparameters explored in the random search and network architectures for the proposed configurations. "U": uniform distribution; $\log U$ uniform distribution in the log-domain.

Parameter (Distribution)	Range	Full-band 1Ch-DNN	Full-band 3Ch-DNN	Half-band 1Ch-DNN	Half-band 3Ch-DNN	SE-DNN
Batch size (U)	{512, 1024, 2048}	512	2048	512	1024	1024
Learning Rate ($\log U$)	$[4.88 \cdot 10^{-4}, 5.52 \cdot 10^{-3}]$	$1.02 \cdot 10^{-3}$	$8.54 \cdot 10^{-4}$	$2.18 \cdot 10^{-3}$	$5.55 \cdot 10^{-4}$	$2.89 \cdot 10^{-3}$
CNN layers						
Nr. of CNN layers (U)	$[1, 3]$	1	1	3	2	3
Kernel shape (U)	$[1, 10] \times [1, 10]$	2×1	1×1	1×1, 1×1, 1×1	2×2, 1×1	5×3, 2×2, 2×1
Kernel number ($\log U$)	$[16, 64]$	18	36	63, 18, 19	4×1, 2×1	29, 54, 61
Strides ($\log U$)	$[1, 6] \times [1, 6]$	3×1	2×1	2×4, 5×1, 4×1	27, 32	2×2, 2×5, 3×1
Pooling Shape (U)	$\{1, 2\} \times \{1, 2\}$	1×2	1×2	2×1, 2×2, 2×1	1×1, 1×1	1×1, 2×1, 1×2
Pooling Strides (U)	$\{1, 2\} \times \{1, 2\}$	1×1	1×2	1×2, 1×2, 1×2	1×1, 2×2	1×1, 1×1, 1×2
Dropout Rate (U)	$\{0, 0.1\}$	0.1	0	0.1, 0.2, 0.3	0.0, 0.1	0.1, 0.2, 0.3
Last CNN Layer (Multi-Channel DNN Only)						
Nr. of CNN layers (U)	1	-	1	-	1	-
Kernel shape (U)	$[1, 4] \times [1, 4]$	-	1×1	-	2×2	-
Kernel number ($\log U$)	$[16, 64]$	-	35	-	53	-
Strides (U)	$[1, 7] \times [1, 7]$	-	2×2	-	2×1	-
Fully-connected layers						
Nr. of fully-connected layers (U)	$[1, 3]$	1	3	3	1	2
Units $\log U$	$[100, 1024]$	181	251, 153, 127	154, 113, 107	796	148, 140
Dropout Rate (U)	$\{0, 0.5\}$	0	0.5, 0.5, 0.5	0.5, 0.5, 0.5	0	0.5, 0.5
Overall network parameter count	-	4.193.535	4.455.596	49.570	4.117.515	305.996

- training conducted on the synthetic dataset, followed by a test on the real dataset: the whole synthetic dataset (4 folds) has been used to train each of the network configurations reported in Table 5.1; the test has been conducted on the overall real dataset and on the test set of the real dataset; the results have been discussed in Section 5.5.

Concerning the reference method, the hyperparameters have been obtained from the original work of its authors [70], and then, it has been trained over both the training set of the real dataset and the synthetic dataset. Each time a test over the real dataset has been carried out.

To evaluate the performance of the aforementioned strategies in the proposed experiments, the Area under Curve of the Precision-Recall (PR-AUC) [74] has been used to measure the detection abilities of each of them.

5.5 Results and remarks

The experiments described above have been carried out and the PR-AUC values have been collected and summarized in Tables 5.2, 5.3 and 5.4. For instance, Table 5.2 summarizes the results obtained during the cross validation process by using the synthetic dataset. Table 5.3 summarizes the results that the DNNs trained over the real dataset achieve during the validation and the test conducted on the real dataset. In the case of Table 5.4 the training has been carried out on the whole synthetic dataset, whereas the testing has been carried out, respectively, on the overall real dataset, and on the test set of the real dataset.

From a general standpoint, based on the collected results, we observe that the use of a synthetic dataset can produce good cry-detection results, up to 83.25% over the overall real dataset and up to 80.48% over the test set of the real dataset (Table 5.4). This result proves that the use of a synthetic dataset can represent a viable alternative to real life counterparts. This performance is even more notable, if we consider that the synthetic dataset does not model the issue encountered with the real dataset, which lies in the deviation of the microphone array from its reference position due to accidental rotation and shift of the supporting arm. In other words, with an improved modelling of the environment

Table 5.2: PR-AUC on synthetic validation dataset (training on synthetic dataset)

Detection Strategy	Validation
Full-band 1Ch-DNN	85.31%
Full-band 3Ch-DNN	90.54%
Half-band 1Ch-DNN	82.97%
Half-band 3Ch-DNN	89.12%
SE-DNN	90.55%
Raboshchuk et al. [70]	76.37%

Table 5.3: PR-AUC on the test set of the real dataset(training on real dataset)

Detection Strategy	Validation	Test
Full-band 1Ch-DNN	97.50%	86.18%
Full-band 3Ch-DNN	97.00%	81.72%
Half-band 1Ch-DNN	91.63%	84.53%
Half-band 3Ch-DNN	96.29%	81.28%
SE-DNN	97.09%	87.28%
Raboshchuk et al. [70]	92.89%	74.96%

that encompasses the data acquisition issues, a further improvement of detection performance is expected.

Synthetic dataset validation

Concerning the validation process based on the synthetic dataset, the best performing strategies are the Signal Enhanced approach proposed in our previous work, and the multi-channel approach applied to the full-band feature set, which achieve almost identical results of about 90.55%. The second best is the half-band multi-channel approach since it achieves a 89.12% rate detection, whereas the worst performing is the half-band single-channel approach with 82.97% detection rate. The test of the approach proposed by Raboshchuk et al. [70] over the validation set achieves a score of 76.37%.

Real dataset training

From the performance rating obtained through the validation over the real dataset, we observe that the full-band single-channel DNN is the best performer with a score of 97.50%. The full-band multi-channel network and the signal enhanced network attain almost identical results with a 97.00% rate and 97.09% rate respectively. The half-band multi-channel network achieves slightly lower results with a 96.29% detection rate, whereas the half-band single-channel network is the worst performer with a 91.63% score. The approach proposed by Raboshchuk et al. [70] scores 92.89% and thus superior than the half-band single-channel DNN.

By comparing these results against the ones provided by the test over the real dataset, the performance drop is evident. The highest drop regards the multi-channel networks, both with a PRC-AUC below 82%. The full-band single-channel network appears to be slightly more robust, with a 86.18% detection rate. The signal enhancement network is the best performer with an 87.28% detection rate and a drop of roughly 10 percentage points (p.p.). The half-band single-channel network, although not the best performer with a score of 84.53%, shows the lowest performance drop, while achieving a PRC-AUC 2.75 p.p. lower than the best performer. The approach proposed by Raboshchuk et al. [70] appears to be the least robust with a drop of about 18 p.p.

Concerning the validation and test over the real dataset, can be concluded that among the DNN based approach, the multi-channel one is the least robust notwithstanding the size of the feature vector, whereas the single-channel DNN is the most robust, with a performance drop varying from 11 p.p. to 7.p.p.

Synthetic dataset training

The test on the overall real dataset results in a different scoring. The best performer is the half-band single-channel DNN, with 83.25%, with a small margin with respect to the validation result. The other investigated methods, on the other hand, show a performance drop. The full-band single-channel cry detection network achieves a score of 80.80% with almost a five point drop. The Signal Enhanced drop at 74.06% and

Table 5.4: PR-AUC on real dataset (training on synthetic dataset - test on the **overall** real dataset and on the **test set** of the real dataset)

Detection Strategy	Overall	Test
Full-band 1Ch-DNN	80.80%	76.77%
Full-band 3Ch-DNN	77.95%	72.71%
Half-band 1Ch-DNN	83.25%	80.48%
Half-band 3Ch-DNN	69.15%	59.05%
SE-DNN	74.06%	70.61%
Raboshchuk et al. [70]	54.12%	46.18%

the multichannel networks drop at 77.95% in the full-band case, and at 69.15% in the half-band case. As such, the full-band multi-channel DNN appears to be the most robust, whereas the half-band multichannel network is the least effective.

The most obvious conclusion is that the signal enhancement based solution, as well as the multi-channel networks, are affected by the deviation of the microphone array from its reference position, whereas single-channel networks are not. Also, it should be noted that while the full-band multi-channel network is the least affected, proving a higher degree of robustness with respect to the Signal Enhancement approach, which depends on a beamformer, the half-band multi-channel network is the most affected, leading to the conclusion that halving the informational content of the input impairs the detection abilities of the system.

The single-channel networks are not affected as much. Rather, the main difference lies in the fact that the full-band single-channel network shows a drop in performance, whereas the half-band single-channel network shows an improvement. This result seems to suggest that in the case of the single-channel networks the reduction of informational content, due to the halving of the feature vector, improves the detection abilities.

If the total parameter count of each network topology is taken into account, it is possible to observe that the half-band single-channel network has one hundredth of the parameters amount with respect to the full-band single-channel network. Similarly the Signal Enhanced network has less than one tenth of the parameters amount of the multi-channel

network. With regard to this, since a smaller network usually has better generalization capabilities than a larger network, the experiments results suggest that while the half-band single-channel network can take advantage of a lower informational content by discarding the lower part of the frequency spectrum of the signal, the multi-channel network can not, since the increased informational content is required to correlate the information from the three input.

The third column of Table 5.4 reports the performance of the different strategies limited to the test set of the real dataset. Regarding these results, although a further performance drop affects all the evaluated strategies, the overall situation remain unchanged. Although the half-band single-channel network is subject to a score drop, it still remains the best performer, which also shows the lowest performance drop among the different approaches.

By comparison, by testing the method proposed by Raboshchuk et al. [70] over the validation set of the synthetic dataset, we obtained a 76.37% detection score, which appear to be slightly below the half-band single-channel network. The test over the whole real dataset reveals a performance drop, which is further intensified if the experiment is carried out over the test set of the real dataset.

Remarks

The comparison of the scores over the test set of the real dataset, that is the results reported in the third column of Table 5.3 and the ones reported in the third column of Table 5.4, reveals that the real dataset, which presents occurrences where the microphone array does not targets the intended subject, represents a better training set with respect to its synthetic counterpart from a theoretical standpoint. However, although each approach achieves a better performance if the training is carried out over the real dataset, the improvement margin highly depends on the strategy of choice, with the signal enhanced network improving the most, with a performance increase of about 17 points and the half-band single-channel network improving the least with an increase of the detection rate of about 4 points.

Although an improvement of about 17 points appear significant, if the actual scores are taken into account, it means that the signal en-

hancement network is the approach that benefits the most from a real dataset. In fact, the half-band single network, which is the approach that benefits the least from the replacement of a synthetic dataset with a real one, actually achieves an 80.48% detection score on the synthetic dataset.

From a pragmatic standpoint, if the issues and the effort required to properly prepare a dataset based on on-field acquired samples are taken into account, although an approach such as the half-band single-channel network may appear sub-optimal, it might be the most cost effective. The same conclusion holds also if the hardware requirements are considered.

From a different perspective, it should be also noted that, by taking into account the crib structure, the microphone array deviation issues may be simulated, as well, thus better simulating a real dataset.

Similarly to the evaluated strategies, even the reference approach achieved a better result when trained over the training set of the real dataset. The test over the validation set also revealed an interesting score of about 92.89% which is above the half-band single-channel approach. The evaluation over the test set of the real dataset, however, shows a performance drop that reveals the complexity of the microphone array deviation issues. Nonetheless, a comparison against the results from Table 5.4, seems to suggest that in the case of the reference method, a synthetic dataset may not be a suitable replacement for a real one.

Chapter 6

Other contributions

6.1 Activity of Daily Living Recognition

The recent advances in the field of pervasive and ubiquitous computing have made possible the development of many applications that base their operation on the recognition of activity of daily living (ADL). This task will be named as activity recognition (AR) in the following, for the sake of brevity. Some examples of these applications are the health care systems for the elderly and disabled people [75, 76], context-aware prompting systems [77, 78], surveillance systems [79] and interactive gaming interfaces [80]. These systems need to recognize the activities that a person is carrying out in a non invasive manner, by observing the behaviour of people and the environment from sensor readings.

In this research area, several studies exist that deal with the problem using different methods and approaches. These works can be classified on the basis of their characteristics. A first classification criterion is to distinguish between knowledge-driven and data-driven approaches. Data-driven techniques use the information provided by the sensors to calibrate or build models of human activities in a supervised or unsupervised manner [81,82]. Differently, knowledge-driven techniques approach to the AR task from a different point of view, i.e., by creating a priori models of the activities based on the knowledge of the problem, according to a logical formalism [83, 84]. AR approaches can also be classified with respect to the employed sensor technology, which can be grouped into three main categories: computer vision technologies, wearable sensors techniques and passive sensors techniques.

Systems based on computer vision techniques [85, 86], usually have at their disposal highly informative data from which to extract the features

necessary to the AR. However, there are many challenges with video based AR such as illumination variations, occlusion and background changes. Moreover, their usability in the context of smart homes for monitoring the activities of residents is questionable since several studies showed that people consider these solutions too intrusive [87]. The techniques based on wearable sensors represent a well explored research area [75, 88]. Initially, dedicated wearable motion sensors were used to recognize different physical activities [89]. In the recent years, there has been a shift towards mobile phones, since they are equipped with various sensors (GPS, accelerometer, gyroscope, etc.) and because they have become more powerful in terms of CPU, memory and battery. The technologies based on passive sensors can rely on a wide range of sensors: reed sensors, RFID tags, and PIR sensors are just some examples of acquisition devices field-tested [90, 91]. Novel AR systems based on power consumption readings also fall in this category [92, 93].

Whatever the system used, many real-world applications that focus on addressing needs of a human require information about the activities being performed by the users in real-time [94–96]. In this case the selected approach to perform AR must work in an online or streaming fashion and recognizing activities as and when new sensor events are recorded. There is the need for online AR techniques that can classify data as they are being collected. This is a challenging problem as data that completely describe an activity are not generally available in such situations and the algorithm has to rely on the partially observed data along with other contextual information to make a decision on the activity being performed.

In [97], Wang *et al.* propose a real-time hierarchical model to recognize both simple gestures and complex activities using a wireless body sensor network. In this model, they first use a fast algorithm to detect gestures at the sensor node level, and then a pattern based real-time algorithm to recognize complex, high-level activities at the portable device level. Krishnan & Cook [98] proposed a system for the online AR from data obtained from a network of binary wireless sensors installed in a smart home and demonstrated that the technique can be used to reach interesting results for online detection, in a real life scenario. They faced various aspects of the online AR issue, like the determination of

the optimal size for the used feature extraction windows and the exploitation of the space-time dependences existing between the various events sensors and the various windows. Yala *et al.* [99] have extended the previous work by introducing two new features extraction methods.

In this contribution, a system for the AR on streaming/online sensor data is proposed and evaluated. A sliding window based approach capable to perform AR in real-time, is the baseline method. This approach is designed to process streams of binary sensor events characterized by a non-constant data rate. In the experiments, the publicly available CASAS dataset has been employed [100]. To account for the fact that the sensor events corresponding to a transition between two activities could be found within the same window, three distinct formulation of a mutual information based weighting of sensor events have been incorporated. Additional contextual information in the form of the previous activity and the activity of the previous window is also appended to the feature describing a window. The baseline approach is then extended in two different methods. First using the states of the binary sensors within the detection window. The second using the sequences of the last N sensors that have generated an event in the windows. A Support Vector Machine (SVM) with a radial basis kernel is used for the recognition task.

The innovations introduced by this contribution to the works on which get inspired [98, 99], consists in the introduction of three new methods of feature extraction that best represent the information contained in the data stream. This allows the classifier to recognize the activities more easily without increasing the computational burden during the recognition stage.

The results shows that the proposed approach performs well the AR, reaching an F1-Score of 58.6% and increasing the baseline approach F1-Score of the 5%. The activities that most benefit from these improvements are the ones which in previous works were more confused by the classifier.

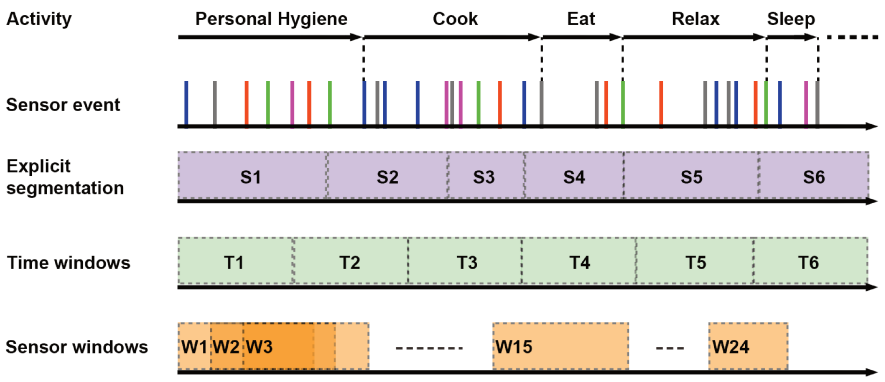


Figure 6.1: Illustration of the different approaches for stream processing.

6.1.1 Background on online recognition of activities of daily living

The scenario taken into consideration is a smart home equipped with several binary sensors (e.g., PIR motion, REED, light switches). During their everyday routines, the residents of the home interact with the sensors that react by changing their state. Every time this occurs, an event is emitted, resulting in a stream of sensor events produced at variable rate. Figure 6.1 is an illustration of a sequence of these sensor firings (represented as the vertical lines). The underlying activity sequence that results in these sensor firings is "Personal Hygiene", "Cook", "Eat", "Relax", and "Sleep". Notice that each activity results in different number and type of sensor firings.

The AR system described in this contribution operates in three steps: first, the data is segmented with an event based windowing approach. In the second step, a feature vector is extracted from each window. Finally the feature vectors are divided into training sets and test sets in order to learn the activity models and then evaluate the recognition scores.

Data segmentation

Three different strategies can be adopted in order to segment the data. The last is the chosen one for the proposed approach.

- **Explicit segmentation**: The data stream is split up into chunks

that likely include the sensor events belonging to a single activity. A pre-processing step is necessary for learning the appropriate chunk sizes needed to perform the segmentation. When this method is used for segmenting real-world data, since in this scenario the activity boundaries are generally not well distinct, the resulting chunks not always identify a single activity. Moreover, the approach has to wait for future data to make a decision on past data rendering it a somewhat non-streaming approach.

- **Time based windowing**: The data streaming is split up into windows of fixed duration. This approach is widely used because its simplicity make it suitable for data flow gathered from sensors with a constant acquisition rate, like accelerometers, gyroscope and so on. For this splitting method the window size plays a key role. If a very small interval is chosen, there is a possibility that it will not contain relevant activity information for recognition. If the time interval is too wide, then information pertaining to multiple activities can be embedded into it and the activity that dominates the time interval will have a greater influence in the AR.

- **Sensor event based windowing**: This segmentation approach produces windows containing equal number of sensor events. Every time a sensor detects an event, a windows is defined. As can be noted in Figure 6.1, the windows appear to vary in their duration. This prevents generating empty windows, also when constant acquisition rate sensors are present in the environment. For a single window, the last sensor event belongs to the activity to recognize. The others sensor-events define the contextual information needed to perform the recognition. This method has some inherent drawbacks. If there is a significant time lag between an event and preceding one, the relevance of all the sensor events in this window may be small respect the last event. Thus treating all the sensor events with equal importance is not a good approach. Furthermore, in presence of multiple residents, sensor firings from two different activities performed by different persons will be grouped into a single window, thereby introducing conflicting influences for the AR regarding the last sensor event. While by itself this

approach may not be alluring, modifying it to account for the relationship between the sensor events is a good method to process the stream of sensor events. This approach offers computational advantages over the explicit segmentation process and does not require future sensor events for recognizing the present activity.

.

Feature Extraction

Formally, the entire sensor events sequence $\{e_1, e_2, \ldots, e_n\}$, is divided to obtain the windows sequence $\{W_1, W_2, \ldots, W_m\}$, where the W_i window is represented by the sequence $[e_{i-\Delta_e+1}, e_i]$ and Δ_e is the window size, i.e., the number of events in a window. The optimal Δ_e depends on the experimental setup and could be derived in a validation set. The windows should be sized to include enough elements to best define the contest for the last sensor event. Once the window size is defined, the next step consists in extracting the feature vectors from the windows sequence for capturing their information content. The feature extraction approaches that will be presented are based on the sensor event windowing. Previous works describe a statistical approach for determining the optimal window size for each sensor, an approach of sliding windows in time domain and an alternative feature extraction method based on sensor states. For the sake of conciseness, these methods are not considered in this work, since their performance with respect to the following methods is lower.

- **Baseline Approach**: The feature vector consists of a fixed dimensional vector X_i which includes the time of first sensor event, the time of last sensor-event, the temporal span of the window W_i and a simple count of the different sensor events within the window. For example, with 31 different sensors, the size of X_i will be 34. For the training phase of the AR algorithm, each vector X_i is labeled with the identifier y_i of the last event of the window. Each label y_i corresponds to an activity class. A collection of X_i and the corresponding y_i then constitutes the training data that is fed into a classifier to learn the activity models in a discrimina-

```
2011-06-15  06:49:48.158    M012    ON
2011-06-15  06:49:50.787    M012    OFF
2011-06-15  06:49:51.320    M012    ON
2011-06-15  06:49:52.178    M021    ON
2011-06-15  06:49:53.966    M012    OFF
2011-06-15  06:49:54.640    M021    OFF
2011-06-15  06:49:58.739    M021    ON  Sleep="end"
2011-06-15  06:53:36.683    M013    ON
2011-06-15  06:53:38.663    M013    OFF
2011-06-15  06:53:42.441    M004    ON
2011-06-15  06:53:46.764    M004    OFF
2011-06-15  06:53:46.931    MA022   ON  Cook="begin"
2011-06-15  06:53:47.031    M026    ON
2011-06-15  06:53:53.036    MA022   OFF
2011-06-15  06:53:53.577    MA022   ON
2011-06-15  06:53:54.714    MA022   OFF
2011-06-15  06:53:55.059    M026    OFF
2011-06-15  06:53:59.023    M026    ON
2011-06-15  06:54:00.731    MA022   ON
```

Figure 6.2: An example of a sequence of sensor events.

tive manner. This will be the Baseline approach against which the proposed enhancements are compared.

- **Sensor Windows Mutual Information (SWMI)**: A critical point of the baseline approach can be found in situations where the sensor events correspond to a transition between two activities or, in a multi-resident scenario, when more people are doing different activities at the same time. Most of sensor events in the window might not be related with the last event in the window. An example of this case is shown in Figure 6.2. This particular sequence of events represents the transition from the activity "Sleep" to the activity "Cook". Note that all the detected events at the top of the window originate from a particular functional area of the apartment (Bedroom), while the second set comes from another area (Kitchen). As long as the sensors of a particular activity dominate the window, the chances of an incorrect recognition of the last activity are high. Therefore, the baseline scheme can be extended by including a measurement of mutual information between the sensors in order to reduce the influence of the events detected by functional areas very different and which fall within the same window. Mutual information is typically defined as the quantity that measures the mutual dependence between two random variables. In the current context, each sensor is considered as a random variable that can take on the binary states On/Off. Mu-

tual information between two sensors is defined as the likelihood of a change in their status consecutively in the flow of events. If S_i and S_j are two sensors, then the mutual information $M(i,j)$ between them is given by:

$$M(i,j) = \frac{1}{N} \sum_{m=1}^{N-1} \delta(s_m, S_i)\delta(s_{m+1}, S_j), \tag{6.1}$$

where

$$\delta(s_m, S_i) = \begin{cases} 0 & if\ s_m \neq S_i, \\ 1 & if\ s_m = S_i. \end{cases} \tag{6.2}$$

The generic element of the summation has the value of 1 when the current sensor is S_i and the next S_j. If two sensors are adjacent to each other in such a way that the activation of one is probably subsequent to activation of the other one, then the mutual information will be high. Note that the calculation of mutual information, using this bi-gram model, depends on the order of activation of the sensors. The matrix of mutual information is calculated off-line by using the training sequences and it is used to weigh the feature vector elements respect to the last event detected. The simple count of the different sensor events within a window is now replaced by a sum of contributions dependent on the mutual information.

- **SWMI Extension (SWMI Ext)**: The mutual information previously described only examines the relationship between sensor couples immediately following in the sequence of events. However, given a sensor event sequence, it can detect a certain amount of mutual information between the latter sensor and the sensors previous to the very next one. Consider the two hypothetical sequences of events sensor $S_1 \Rightarrow S_2 \Rightarrow S_3 \Rightarrow S_4$ and $S_1 \Rightarrow S_3 \Rightarrow S_2 \Rightarrow S_4$ that identify the same activity, but performed in two different ways. Assuming that the first path is statistically less used than the second path, it's clear that there is a dependency between sensors S_1 and S_2 whatever path is used. Based on these assumptions Yala *et al.* [99] proposed a method for calculating the mutual information that offers better performance then SWMI. It

consists in obtaining the mutual information between two sensors S_i and S_j by computing their frequency of occurrence in space of n sensor events along the entire data stream, as defined by the following equation:

$$M(i,j) = \frac{1}{\mathcal{W}} \sum_{m=0}^{\mathcal{W}} \{(S_i, S_j) \in W_m\}, \tag{6.3}$$

$$W_m = [e_{m \cdot n+1}, \cdots, e_{m \cdot n+n}]. \tag{6.4}$$

where $\mathcal{W} = \{W_1, W_2, \ldots, W_{|\mathcal{W}|}\}$ is the set of windows, $|\mathcal{W}|$ is the total number of windows, n is the number of events in the window. The parameter n is selected empirically. Feature vector is then computed as the original method.

- **Past contextual information (PWPA)**: The methods described above only take into account events detected in the analysis window, but they do not consider past information. The activities performed in previous windows, or the previous occurrence of an activity, are important factors related to the event detected in the current window. In the considered dataset, there are some activities that have a definite past activity associated: "Enter Home" always takes place after the "Leave Home". Adding the information of the previous feature vector, it is possible to improve the context description of last event. By relying on the activity models obtained for the previous window to get information about past activities, a two steps semi-supervised learning paradigm is obtained (see Figure 6.3). In the first step, activity patterns are learned by the classifier through the baseline approaches described above. Each of the learning instances are fed into this activity model to obtain the recognition probabilities relating to the different activity classes. In the second step, the recognition probabilities of the previous window are appended to the feature vector describing the current window along with the last activity (not the activity in the immediate preceding window). Figure 6.3 summarizes the PWPA approach. Consider for example the feature vector X_{i+2} extracted from windows W_{i+2}. This feature vector is completed

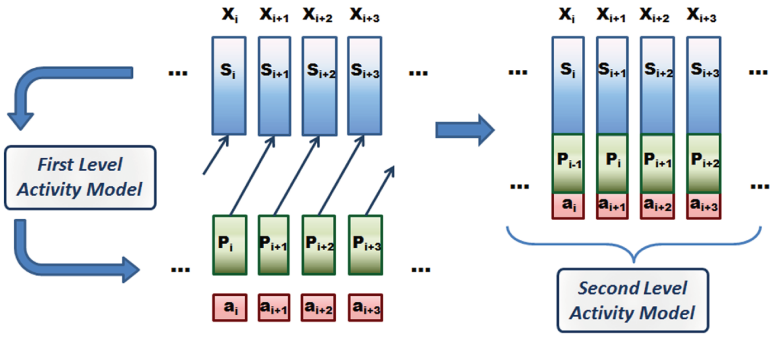


Figure 6.3: Two phase learning process that includes past contextual information.

with the recognition probabilities of the window W_{i+1}, along with the activity that occurred last, in this case a_{i+2}. Then, this new expanded vector is used to train another activity model. During the test phase, the feature vector that describes the test window is processed by the first activity model; the recognition probabilities on output are then appended to the feature vector, which is employed in the second model to recognize the activities.

Support Vector Machines

SVMs are binary classifiers that discriminate whether an input vector $\mathbf{x}$ belongs to class $+1$ or to class -1 based on the following discriminant function:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d, \tag{6.5}$$

where $t_i \in \{+1, -1\}$, $\alpha_i > 0$ and $\sum_{i=1}^{N} \alpha_i t_i = 0$. The terms $\mathbf{x}_i$ are the "support vectors" and d is a bias term that together with the α_i are determined during the training process of the SVM. The kernel function $K(\cdot, \cdot)$ can assume different forms. In this work, the radial basis function (RBF) kernel $K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)$ has been employed. The input vector $\mathbf{x}$ is classified as $+1$ if $f(\mathbf{x}) \geq 0$ and -1 if $f(\mathbf{x}) < 0$.

In this contribution, the multiclass problem has been addressed using the "one versus one" strategy. LIBSVM [101] has been employed both in the training and testing phases of the SVM. The feature vectors are

scaled using a Min-Max strategy which maps the vector elements into the interval $[-1, 1]$ Following the settings of previous work [99], the penalty parameter C was set to 100, while the width parameter γ to 1.

6.1.2 Proposed approach

Previous approaches were designed to extract the information contained in the sequence of events with focus on the spatial properties. However, some information such as the sensor states and the temporal sequence of the sensor events are still ignored by the state of the art methods described in Section 6.1.1.

In this contribution, two new feature extraction methods are proposed. These methods derive from the baseline approach, i.e., "Baseline approach plus sensors state" (BSS) and "Last N sensor sequence" (LNSS) which respectively rely on the sensor states and on the temporal sequence of the sensor events. The third approach proposed in this contribution, named "Activity Based SWMI" (SWMI Act) modifies the SWMI Ext weighting scheme to calculate the mutual information in the subsets of sensors involved in each different activity.

Baseline approach plus sensors state (BSS)

The CASAS project dataset used in this work are primarily derived from a network of PIR binary motion sensors that provide an output value interpretable with the On/Off states. In particular, the On state indicates the presence of a person near the sensor. This state information, which could facilitate the task of the classifier to discriminate the various activity classes, is neglected by the other approaches evaluated. It is possible to embed the information about the state of the sensors on the baseline approach by applying a simple weighting scheme. If the final state assumed by a sensor within the window is On, a weight equal to 1 is associated to the sensor, while, if the final state is Off, the same weight is set to -1. Once calculated, the weights are applied to the sensor event count obtained from the baseline approach.

Last N sensor sequence (LNSS)

The aim of LNSS approach, is to exploit the information content about the sequence of sensor events. This approach aims to capture the space-time information from the sensor event series by encoding the identification pattern of the movements of a person inside the house. In fact, this information is not captured by the simple counting of sensor events provided by the baseline approach. Like BSS, this approach is an extension of the baseline one. A sub-window of the last N sensor events is extracted from the feature vector obtained to the baseline method from a certain window. Only those sensor events generating from door sensors or corresponding to On⇒Off transitions of motion sensors are then appended to the end of feature vector. Specifically, the sensor identification numbers are appended followed by as many zeros as is required to obtain features vectors of length equal to the original vector size plus N. This approach will be evaluated at varying the values of the parameter N.

Activity Based SWMI (SWMI Act)

The weights calculation for the SWMI Ext scheme is based of fixed length windows obtained by splitting the sensor events sequence. We propose to use the data chunks identified by the sensor events sequences corresponding to the predefined activities (except class "Other") instead of a fixed length window. The equation (6.3) is replaced by:

$$M(i,j) = \frac{1}{|\mathcal{A}|} \sum_{m=1}^{|\mathcal{A}|} \{(S_i, S_j) \in A_m\}, \tag{6.6}$$

where $\mathcal{A} = \{A_1, A_2, \ldots, A_{|\mathcal{A}|}\}$ is the set of activities, $|\mathcal{A}|$ is the total number of activities, and (S_i, S_j) indicates that the event emitted by sensor S_j follows the event emitted by sensor S_i. The author believe that this mutual information formula, compared to the previous two, better characterizes not only the activities performed to a well defined functional area but also the others. In fact, this new approach, besides discovering the relations of mutual information between sensors, isolates the subsets of sensor used for each activity. Note that excluding non predefined activities from computation of the mutual information, we

deal with the class unbalance of a dataset.

6.1.3 Experimental set-up

Below, will be presented the characteristics of the dataset used in the experiments, the experiments conducted to assess the performance of the algorithm and then, the obtained results. The performance of the different feature extraction methods will be compared with the scores obtained from the baseline approach.

Dataset

One of the real-world datasets of CASAS project corpus [100] has been used to testing the proposed approach. The selected dataset, named HH104, contains sensor data that was collected in a home where a single person is present. The dataset was obtained using PIR motion sensors, wide area PIR motion sensors and door sensors. It consists of approximately two months of labelled activities and over 2 years of raw data. Only the first 4 weeks of the annotated dataset part has been used for testing purposes. This limitation is imposed due the long computation times required to extract the patterns of activity on large amounts of data. Table 6.1 shows the activity classes, the number of related instances found in the dataset and the number of associated sensor events. The table shows also the class "Other" that incorporates all the sensor events that are not comprised in the other classes. It is worth highlighting that the dataset is highly unbalanced, since about 48% of the activities, corresponding to 18% sensor events, is labelled as "Other".

Experiments

The algorithm has been assessed using a 5-fold cross-validation strategy and evaluated using the F1-Score due to the class unbalance of the dataset. Firstly, the performance of the baseline segmentation approach have been evaluated by varying the length of the window used to extract the feature vectors. Windows with 5, 10, 15 and 20 sensor events have been assessed in all the feature extraction approaches. Table 6.2 shows the values of average F1-Scores obtained for the different methods. The last three are the proposed ones. Note that for the baseline method the

Table 6.1: Dataset HH104 statistics.

Activity	**# Activity**	**# Sensor Event**
Bathing	180	5550
Bed toilet transition	134	4757
Cook	148	37728
Eat	103	19108
Enter Home	93	575
Leave Home	93	471
Personal Hygiene	64	3352
Relax	226	21541
Sleep	182	19820
Take Medicine	48	1234
Work	187	19575
Other	1366	28857

highest score (53.1%) has been obtained with a window length equal to 10.

The second tested approach is the PWPA. Compared with the baseline method, the F1-Score improves by 3%. It should be noted that the time required for the calculation of the activity models can be significantly longer for this method. In fact the two-level scheme of this approach requires training of two distinct SVM models. Moreover, in order to compute recognition probabilities for the activity classes, the application

Table 6.2: Average F1-Score (%).
Beside each method, the reference paper is indicated.

Window ⇒	**5**	**10**	**15**	**20**
Baseline [99]	51.8	53.1	51.1	49.4
PWPA [99]	56.1	54.0	51.6	49.5
SWMI [99]	54.0	54.5	54.7	54.5
SWMI Ext [98]	56.6	57.4	57.1	55.6
SWMI Act	57.0	57.9	58.0	58.0
BSS	56.9	57.0	55.9	54.3
LNSS 5	58.0	58.1	57.9	56.8

Table 6.3: Average F1-Score (%) for LNSS.

Window ⇒	**5**	**10**	**15**	**20**
N = 1	55.2	54.6	52.8	51.2
N = 5	58.1	58.1	57.9	56.8
N = 10	—	57.5	57.3	56.6
N = 15	—	—	55.4	55.4

of Platt's scaling, a particularly time consuming procedure, is needed [102]. Figure 6.4 shows the F1-Scores calculated for each activity class, for the methods described in [98] and for the SWMI Ext approach [99]. It should be noted that the PWPA approach, with respect to the baseline method, shows some improvements, particularly evident for the activity "Leave Home". Conversely, the features extracted with this approach increase the classifier confusion in the recognition of the "Cook", "Eat", "Relax" and "Sleep" activities.

After PWPA, the approaches based on mutual information have been compared. Figure 6.6 shows the matrices of mutual information calculated using the SWMI, SWMI Extend and SWMI Activity methods. As can be observed, the SWMI matrix (Figure 6.6a) appears to have higher values along the principal diagonal. This can be explained considering that the dataset employed in the experiments mainly consists of sensor events produced by short firings of PIR sensors. This means that, in equation (6.1), the self-transitions have a significant weight. The matrices of the SWMI Ext (Figure 6.6b) and SWMI Act (Figure 6.6c) approaches are appreciably more sparse, revealing a more "smooth" weighting scheme that takes into account the relationships between sensors from different functional areas of the smart home. The author expect that this facilitates the recognition of the activities that include walking movements in the home, like "Bath to Toilet", "Enter Home" or "Leave Home" or that can be performed on different rooms like "Relax". This is confirmed by Figure 6.4 showing the F1-Score calculated for each single activity class and for each of the three weighting schemes.

Please note that the window size n, shown in equation (6.3), has to be indicated to use SWMI Ext approach. In this work, four different values for n have been tested, precisely 5, 10, 15 and 20 and the best

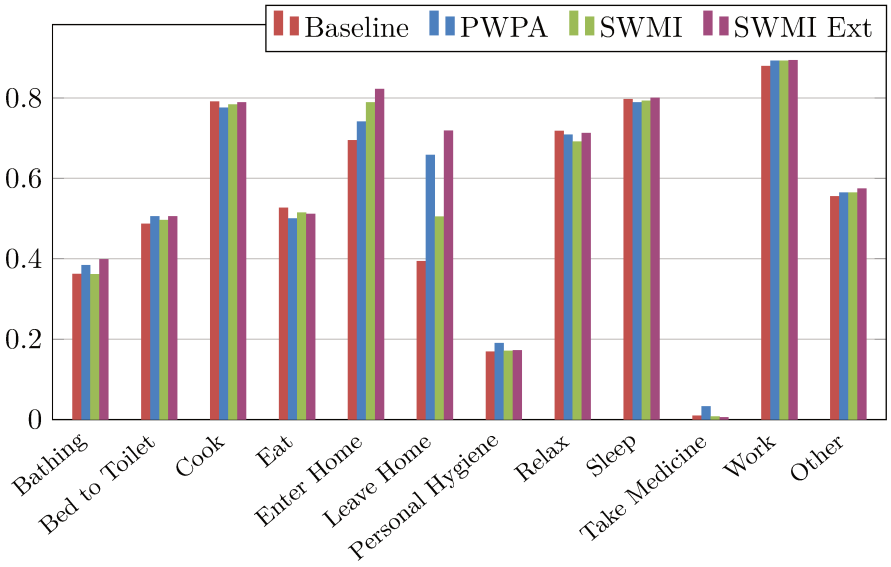


Figure 6.4: F1-Scores for the individual activities obtained by the different original approaches.

performance was obtained with $n = 5$. The SWMI, SWMI Ext and SWMI Act approaches yield a maximum F1-Score equal to 54.7%, 57.4% and 58.0%, respectively.

Regarding the BSS feature extraction method, the improvement with respect to the baseline approach is 3.9% on average.

In Figure 6.5, the F1-Scores of the three novel approaches are depicted in details for each activity. For a comparison, in the same figure are also reported the F1-Scores of the best previous method. It can be noticed that the BSS method provides the greatest improvements for the activities that involve the use of contact sensors. For example, while a person performing the "Personal Hygiene" activity, the sensor called "D003" remains active for a long time. This generates very distinctive feature vectors that can by easily recognized by the classifier.

The last tested approach is the LNSS. As explained in Section 6.1.2, for this method the length N of the sensor sequence to append at the end of feature vector extracted with the baseline approach must be defined. In Table 6.3, is reported the average F1-Score obtained by varying the parameter N. For window size equal to 10 and $N = 5$ the LNSS method improves by 5% with respect to the baseline approach and is the best

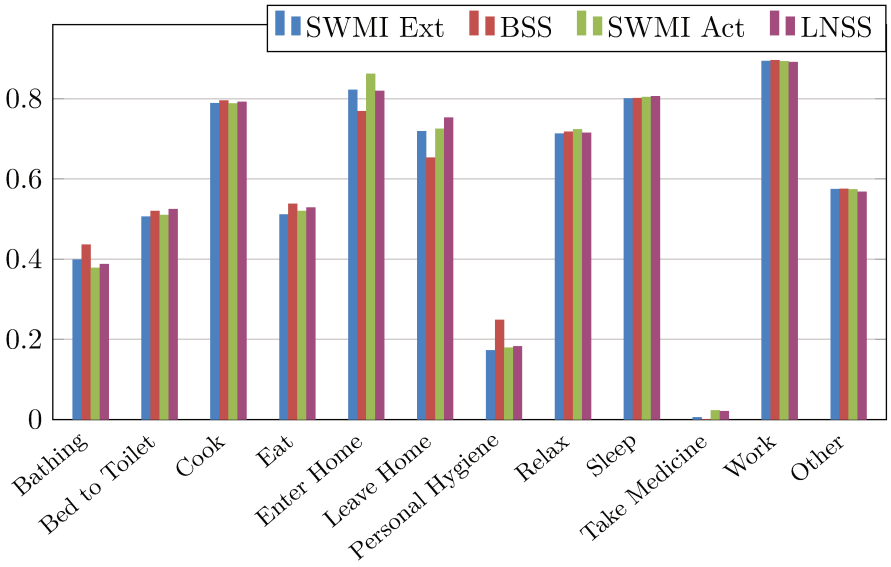
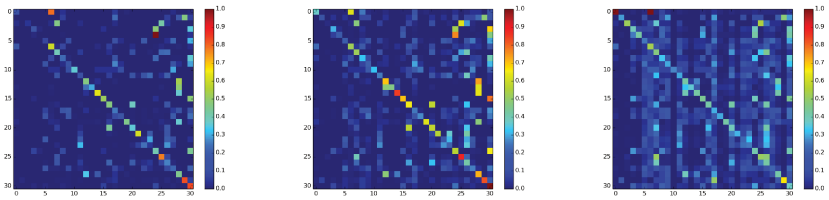


Figure 6.5: F1-Scores for the individual activities obtained by the different proposed approaches and by best original original approach.

performing method among the ones tested. In Figure 6.5, it can be observed that the activities presenting the greatest improvements are "Enter Home" and "Leave Home" due to the fact that their sequences of sensor activations are very distinctive.

As a final remark, it should be noted that almost all methods exhibit the best performance when the window size is 10. It is clear that, on average, this is the size that best describes the context for the last sensor event in the considered dataset. For SWMI and SWMI Act methods, the size of the optimal window increases to 15, therefore, the context of the activities characterized by many sensor events are better defined. At the same time, the weighting scheme introduced by this two approaches prevents that the activities characterized by a few sensors events are penalized by the ones that are not part of the context but fall within the window.

Finally, the author tried to boost the performance of the AR algorithm by searching the optimal SVM parameters on a suitable grid of values [101]. Specifically, the C and the γ parameters have been varied within the $[2^{-3}, 2^{-1}, \ldots, 2^{11}]$ and $[2^{-11}, 2^{-9}, \ldots, 2^{3}]$ ranges, respectively. Being the grid search procedure a time-consuming, we applied it only to the



(a) SWMI. (b) SWMI Extend. (c) SWMI Activity.

Figure 6.6: Mutual information matrices computed with 3 different approaches.

two best performing methods, LNSS and SWMI Act, and to the baseline approach.

As regards the baseline and SWMI Act methods, not appreciable improvements have been found, thus $C = 100$ and $\gamma = 1$ are the most performing parameter values. On the contrary, for the LNSS the best parameters are $C = 2$ and $\gamma = 2$, that increase the F1-Score from 58.1% to 58.7%.

6.1.4 Remarks

In this contribution an extension of a system for AR on streaming/online sensor data [100] has been proposed and evaluated. A sliding window based approach capable to perform AR in real-time is the baseline method. This approach operates on binary state sensor networks that are characterized by a non-constant data rate. Thus, the recognition of activities is performed only when are emitted new sensor events.

To account for the fact that the sensor events corresponding to a transition between two activities could be found within the same window, three distinct formulation of a mutual information based weighting of sensor events have been incorporated. Additional contextual information in the form of the previous activity and the activity of the previous window is also appended to the feature describing a window.

The baseline method has been extended in two different ways. The first one uses the state assumed by binary sensors within the detection window, whereas the second makes use of the sequences of the last N sensors that have generated an event in the considered window. All

these feature extraction approaches are evaluated on a real-world smart home dataset.

The results shows that the three novel approaches are more performing with respect to the others. The BSS method improves the recognition F1-Score of the activities "Bathing", "Cook", "Eat" and "Personal Hygiene" which involve the activation of the door sensors. The average improvement of those activities is equal to 5% respect the baseline approach and 3.8% respect the SWMI Ext. The LNSS approach performs better on the recognition of activities which involve walking movements inside the home, since it uses the sensors activation sequences produced during the execution of these activities. It is worth noting that LNSS shows improvements, although minimal, on the majority of the activities, making the approach the best performing among the methods assessed, with an F1-Score equal to 58.1% and an absolute improvement of 5% over the baseline approach.

Finally, the SWMI Act shows performance close to LNSS and better than the SWMI Ext from which it originated. Compared to it, SWMI Act exhibits a general performance improvement, in particular for the activity "Enter Home" that is characterized by rapid firing of a small set of sensors. Calculating the mutual information as indicated by the formula equation (6.6) facilitates the recognition during the transition between this activity and the subsequent.

A grid search for finding optimal parameters of the SVM classifier has been performed on the baseline, SWMI Act and LNSS approaches. The only approach which took advantage of the grid search was LNSS, with an absolute improvement of 0.6%.

6.2 Fall Detection

The ageing of population represents a major challenge for the immediate future of both industrialized and developing countries. Estimates show that by 2050 the elderly proportion will tend to double from 11% to 22% [103]. The increase in life expectancy joint to a getting worse ratio between the active and inactive people, will increase the relative socio-economic burden for health care and services for elders [104]. The strategy adopted to reduce the impact of this demographic change on

the society is to invest in intelligent technologies able to support the elderly directly in their homes [105]. In this context, an important topic is represented by falls detection. It was observed that about 62% of injury-related hospitalizations for the people over 65 years are the result of a fall. Instead a prompt detection of falls reduces the correlated risks of morbidity and mortality [106]. Being the primary cause of injury-related death for the elders [107], human fall detection has been a major research topic in the last years. Several works appeared in the literature that present different solutions for a prompt detection of a human fall.

Fall detection approaches can be distinguished based on their sensing technologies and on the algorithm that discriminates falls from non-falls [107–109].

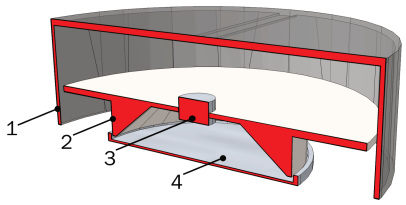
The sensors at their basis are either "environmental" if they are placed in the environment, or "wearable" if they are worn by the monitored person [110]. Among the former there are passive infrared sensors, vibration, and pressure sensors, cameras, and microphones since they are located on the environment where the fall event takes place. On the contrary, belong to the "wearable" class accelerometers, heart rate, electrocardiogram (ECG), and body temperature sensors since they are embedded in a device worn by the monitored person.

The algorithms can be distinguished between "analytical methods", that base their decision on thresholding the acquired signals or the related features, and machine learning methods that "learn" the characteristics of the fall signal directly from the data [108]. The methods proposed in [111–114] are "analytical methods"that employ wearable devices and decide whether a fall occurred or not by applying a decision threshold on the captured signals or on related features. The disadvantage of this solution is that it requires an a-priori knowledge on the fall signal characteristics and manual tuning of the parameters of the algorithm, something that can be difficult to perform due to the variability of the operating conditions and of the subjects. Machine learning techniques have, thus, been adopted in several recent works to overcome this drawback. Supervised approaches train the learning algorithm on a large dataset where all the classes of interest are represented. In [115], single-tree complex wavelet transform features are extracted from a floor vibration sensor and classification is performed by using a multiclass SVM.

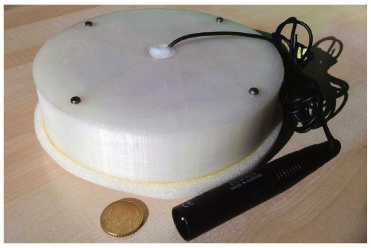
The training dataset comprises human falls, walking/running records, sitting on the floor, slammed door, and fallen book. Approaches based on audio signals are based on one or more microphones placed on the ceiling, on the walls, or on the floor. In [116, 117], an acoustic sensor that operates similarly to stethoscopes has been employed to capture the acoustic waves that are transmitted through the floor. The algorithm is based on MFCCs and GMSs as features, and on multiclass SVM trained on recordings of the falls of a human mimicking doll and of several objects. In [118], the authors employed one aerial microphone, and Perceptual Linear Predictive (PLP) coefficients as features. Classification is based on GMSs and SVM with a Kullback-Leibler divergence kernel that is trained to discriminate between falls and nine classes of non-fall events. In [119], the authors employed a circular array of eight microphones to determine the height of the sound source and to filter falls from non-falls. MFCCs are use as features and the k-Nearest Neighbour (k-NN) classifier perform the final fall/non-fall discrimination. The classifier is trained on human falls and non-fall events comprising dropping of objects, walking, speech and other sounds related to normal human activities. Li *et al.* [120] proposed a multi-channel blind source separation technique based on Non-negative Matrix Factorization (NMF). For additional ambient noise reduction a delay-and-sum beamformer has been used. Then, the MFCC features are extracted from the enhanced audio and finally a k-NN classifier is employed to discriminate the fall event from non-falls. Differently, the system proposed in [121] captures the audio signal by using a smartphone placed on the table. Four different machine learning classifiers (k-NN, SVM, least square method, and neural network) are tested with four different type of features: spectrogram, MFCCs, linear predictive coding (LPC), and matching pursuit (MP). The best performance is achieved by using spectrogram features with ANN classifier with sensitivity, specificity, and accuracy all above 98%. Acoustic signals have been also employed in combination with signals acquired with different sensors. In [122], the authors combined features from sound and vibration sensors that are then employed by a naive Bayes classifier for classification. The experiments were conducted on a dataset containing falls of the "Rescue Randy" human mimicking doll and four objects, and the resulting sensitivity and specificity were

respectively 97.5% and 98.6%. Motion, sound and video signals are employed in [123]. Signals are captured both from environment sensors and from body sensors. A fall is detected by analysing sounds and motion information, while visual and motion behaviour indicates the severity of the fall. The work by Toreyin and colleagues [124] combines PIRs, microphones and vibration sensors. Signals are processed to extract features in the wavelet domain and HMM classifier is then employed to detect falls. The authors showed the using PIR signals 100% accuracy can be obtained. The approach proposed in [125] is based on video signals acquired from the cameras of Microsoft Kinect. The algorithm comprises a first stage where features are extracted from important joints of human skeleton and a second stage where an SVM is trained on the features extracted from the tracking of the joints.

The problem with supervised approaches is that they require that each class of interest is represented in the training dataset. However, with real human falls the variability of the environmental conditions and of the subjects makes difficult or impossible to collect a sufficient number of examples that allow the algorithm to generalise well on unseen conditions [108]. Unsupervised approaches tackle the problem as a novelty detection task [126, 127], i.e., by learning a normality model from data not related to human falls. Among approaches using wearable sensors, Zhou *et al.* [128] propose a fall detection algorithm based on activity transition extrapolated from accelerometers and gyroscopes. The main idea is to extract features from transition data between adjacent activities to recognise various kinds of normal and abnormal activities by means of an OCSVM. Popescu and Mahnot [129] evaluate three unsupervised methods for acoustic fall detection: Gaussian Mixture Models, nearest neighbour and OCSVM. The acoustic signal is acquired with a single aerial microphone and the MFCCs contained in a window of 1 s are used for classification. The experiments are conducted on a dataset comprising falls and non-falls represented by dropping objects, knocking, clapping, and phone-calling related. A two microphones approach has been presented in [130], where the algorithm first processes the stereo signal with a source separation stage to remove background noises. The classification algorithm is based on OCSVM and MFCCs as in [129]. In the dataset, normal events comprise sound originating from walking,



(a) 1. The outer container, 2. The inner container, 3. The microphone slot, 4. The membrane touching the floor.



(b) A picture of the floor acoustic sensor used during the recordings.

Figure 6.7: The floor acoustic sensor scheme (a), picture of the prototype (b).

bending, lying, and sitting. The authors did not consider falls of other objects that could significantly confuse the classifier, however they considered the presence of a television that produced the interfering sound. The results in terms of Area Under Curve are 0.9928 without interference and 0.9738 with 75% interference.

6.2.1 The Floor Acoustic Sensor

For the study of the detection algorithms of human falls, a dataset was collected. The sensor employed to capture the sounds produced by a fall is shown in Figure 6.7: it is composed of a resonant enclosure and a microphone located inside. The acoustic coupling with the floor surface is guaranteed by a membrane that lays on it. As demonstrated in [116, 117, 131], compared to microphones placed on walls or on the ceiling, this solution is better able to isolate the sounds produced by a fall from external interferences (e.g., voice, music). The enclosure has been manufactured in Polylactic Acid with a 3-D printer, its diameter is 16.5 cm and its height 5.5 cm. Regarding the microphone, an AKG C 400 BL[1] has been inserted in the enclosure. The AKG C 400 BL is characterized by an hypercardiod directivity pattern, thus it has been oriented so that the maximum gain is towards floor.

[1] http://www.akg.com/pro/p/c400-bl

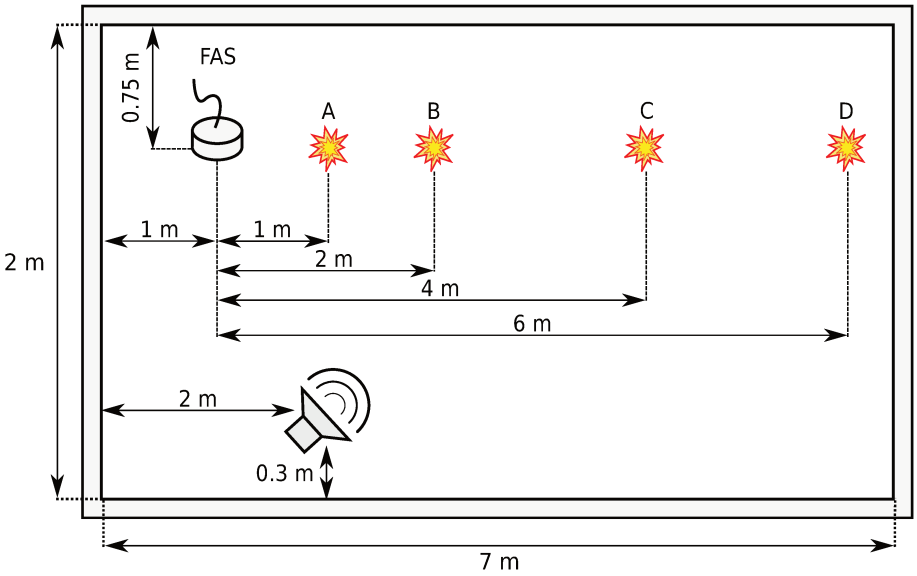


Figure 6.8: The recording setup: the letters A, B, C and D indicate the positions of fall events.

6.2.2 The human fall dataset

The dataset[2] is composed of audio events corresponding to falls of humans, objects, sounds of normal activities (voices, footsteps, etc.), and music [117]. Acquisitions have been performed in a rectangular room measuring about $7\,\text{m} \times 2\,\text{m}$ using a Presonus AudioBox 44VSL sound card and the FAS positioned on the floor (Figure 6.8).

Human falls have been simulated by means of "Rescue Randy", a human-mimicking doll employed in water rescues. The doll has been dropped from upright position and from a chair, both forward and backward, for a total of 44 events, all included in the "Human fall" class. Regarding falls of objects, a ball, a metal basket, a book, a metal fork, a plastic chair, and a bag have been used to reproduce sounds similar to human falls that could produce false detections. Each fall event has been performed at four distances from the FAS, i.e., 1, 2, 4 and 6 m (Figure 6.8). Furthermore, for each distance, the basket and the chair have been overturned from their natural position, while the other objects have been dropped at two heights, i.e, 0.5 m and 1 m. Normal activi-

[2] http://www.a3lab.dii.univpm.it/research/fasdataset

Table 6.4: Composition of the dataset.

Class	Nr. of occurrences	Total length (s)
Basket	64	86
Fork	64	82
Ball	64	129
Book	64	63
Bag	64	57
Chair	96	157
Human Falls	44	76
Human Activity	665	1218
Music	776	1498

ties sounds have been recorded while persons were performing common actions, such as walking, talking, and dragging chairs. Finally, three musical tracks have been played from a loudspeaker and acquired back with the FAS. The first track contained classical music[3], while the second[4] and the third[5] rock music. Musical tracks and normal activities sounds have been divided in segments whose lengths have mean and standard deviation estimated from instances of fall events. In addition, they have been employed alone and to create noisy versions of human and object falls occurrences in order to assess the algorithm in presence of interferences. The total number of occurrences for each class is reported in Table 6.4.

Acquisitions have been originally performed with a sampling rate equal to 44.1 kHz and 32 bit depth [117]. In the experiments, signals have been downsampled to 8 kHz and the resolution has been reduced to 16 bit. The choice of the sampling frequency is motivated in [117] where it was shown that the signals recorded with the FAS have the majority of the energy concentrated at frequencies below 1 kHz.

[3]W. A. Mozart, "Piano trio in C major"
[4]Led Zeppelin, "Dazed and confused"
[5]Led Zeppelin, "When the levee breaks"

6.3 Fall detection with OCSVM and Template Matching

As aforementioned at the beginning of this chapter, machine learning techniques can be divided in supervised and unsupervised methods. Unsupervised methods have been proposed since human falls are "rare" events, and it would be difficult to capture a sufficient amount of examples for representing them in different operating scenarios (e.g., rooms, floor material) and subjects. Unsupervised methods, on the contrary, consider a human fall as an event that deviate from normality, and they are based on one-class classifiers. However, their weakness is that also certain events differ from the "normality" as human falls, and they may induce the classifier to produce false alarms. As an example, Figure 6.9a and Figure 6.9b show respectively the waveform and the spectrogram of a segment of "normal" human activity (footsteps and speech) Figure 6.9c and Figure 6.9d show the waveform and the spectrogram of a segment of human fall, and Figure 6.9e and Figure 6.9f the waveform and the spectrogram of a book fall. The figures show clearly that both falls signals differ significantly from the human activity one, thus a classifier may be induced to consider the fall of a book as the fall of a person. An ideal algorithm should be able to detect actual human falls and simultaneously avoid the detection of false events. The approach proposed in this contribution for reducing the problem of detection of false events consists of a combined One-Class Support Vector Machine (OCSVM) [132] and template-matching classifier that operate in cascade. The general idea is that a human fall produces a sound considerably different from the ones commonly occurring in a home (e.g., voices, sounds from electronic devices, footsteps, etc.). The OCSVM is trained on a large set of "normal" sounds to detect acoustic events that deviate from normality. However, it is expected that certain acoustic events are as abnormal as a human fall (e.g., the fall of book, a chair, etc.), thus they could raise false alarms. The template-matching classifier operates in a user-aided supervised manner and it is employed to reduce such errors by using a set of templates that represent these events. Templates are identified by the user that marks the occurrence of a false positive instead of a true human fall event. The fall detector operates on a environmental sensor,

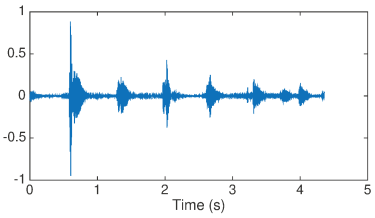
i.e., on the signals captured by a Floor Acoustic Sensor (FAS), and it extracts Mel-Frequency Cepstral Coefficients (MFCCs) [133] and Gaussian Mean Supervectors (GMSs) [134] for classification by the OCSVM and template matching classifier. The performance of the algorithm has been assessed on a large corpus of fall events created by the authors. The corpus contains human fall events reproduced by employing the "Rescue Randy" human mimicking doll[6] [122, 135, 136], and non-fall events represented by dropping of objects, music, and sounds related to common human activities. The experiments have been conducted in clean and noisy conditions in three scenarios: the first comprises human falls, human activity, and music; the second, human falls and object falls; the third represents the most realistic scenario and comprises all the classes of the first and second sets. The significance of the proposed method has been evaluated by implementing and assessing the algorithm with the OCSVM only and GMSs as input, and the algorithm described in [129] based on OCSVM and with MFCCs as input.

6.3.1 Proposed approach

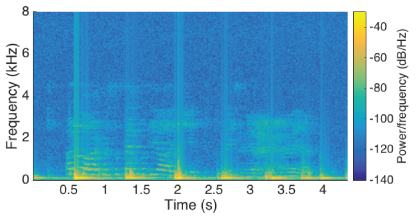
The proposed approach is composed of three stages Figure 6.10: the first ("Feature Extraction") extracts MFCCs from the input audio signal and then GMSs to describe the entire audio segment. The second stage ("Abnormal Event Detection") consists of a One-Class SVM classifier that discriminates between normal and abnormal sounds. Up to the author's knowledge, OCSVM together with GMSs have never been jointly used for acoustic fall detection.

The third stage represents the innovative element of this contribution for reducing false alarms in unsupervised approaches: it consists of a "Template-Matching" block that refines the output of the OCSVM and classifies the input data as fall or non-fall. The OCSVM is trained unsupervisedly on a large dataset of everyday sounds with the objective of discriminating normal from abnormal sounds. As aforementioned, the basic assumption is that the acoustic events related to human falls are "rare" respect to sounds normally occurring inside a home. The template-matching stage, on the other side, requires a set of "template" instances that represent rare events that can be confused with a fall.

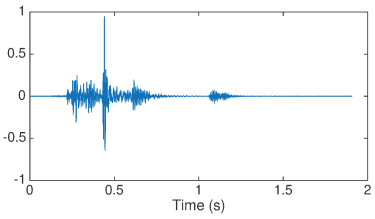
[6] `http://www.simulaids.com/1475.htm`



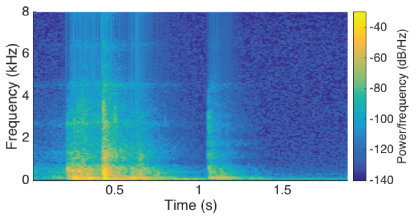
(a) Normal human activity signal in the time domain.



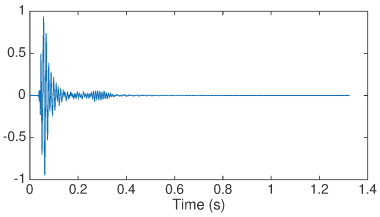
(b) Normal human activity signal in the frequency domain.



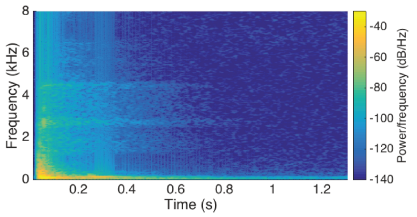
(c) Human fall signal in the time domain.



(d) Human fall signal in the frequency domain.



(e) Book fall signal in the time domain.



(f) Book fall signal in the frequency domain.

Figure 6.9: Time domain (on the left) and frequency domain (on the right) representation of a normal human activity signal (a-b), human fall signal (c-d), and book fall signal (e-f).

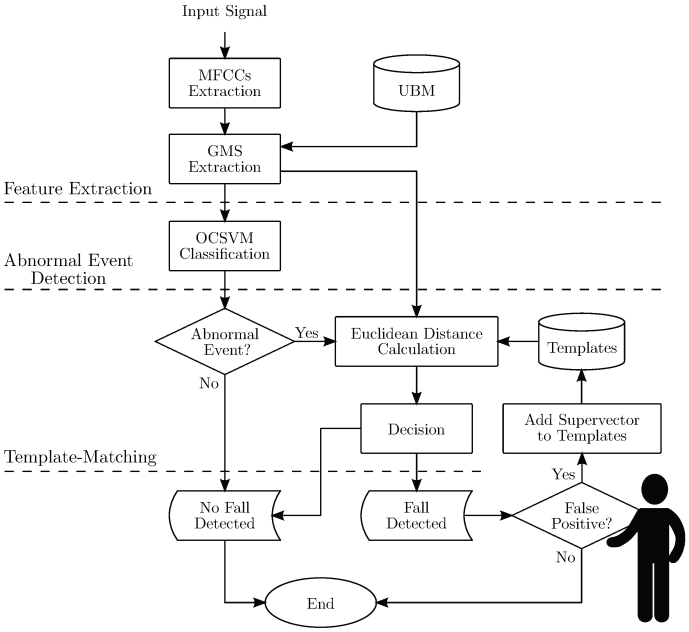


Figure 6.10: The block scheme of the proposed approach.

Referring to Figure 6.10, the "Template-Matching" stage is composed of a set of "Templates", a block that calculates the distance between the input GMS and the templates ("Euclidean Distance Calculation"), and a "Decision" block the decides whether the event is a fall or a non-fall by evaluating the magnitude of the distance. The rationale here is that certain acoustic events are as abnormal as falls and confuse the OCSVM: the template-matching stage reduces false positives by using a set of examples related to the most confusing classes. In this contribution, the algorithm is "user-aided", i.e., templates are indicated by the user each time the OCSVM produces a false positive. This is shown in Figure 6.10 with the person silhouette near the block that decides whether a detected fall is a false positive or not ("False Positive?"). In general, however, it is possible to create the templates set a-priori by recording several instances of possible false alarms events. Although rare, false alarm events (e.g., falls of objects) are certainly easier to reproduce in laboratory respect to human falls.

Feature extraction

The feature extraction stage extracts low-level acoustic features represented by Mel-Frequency Cepstral Coefficients from the input audio signal. These are then employed to calculate Gaussian Mean Supervectors (GMS), which represent higher level descriptors employed for the actual classification. MFCCs have been originally developed for speech recognition and speaker verification tasks, however they have been successfully exploited also for classifying falls [117, 122]. As shown in Figure 6.11, extracting MFCCs involves pre-emphasizing the input signal and filtering the output with a set of filters equally spaced in the mel space. After taking the logarithm of the energy in each band, the final coefficients are calculated by applying the Discrete Cosine Transform (DCT). In this contribution, pre-emphasis has not been applied, since the energy of the signals acquired with the FAS is concentrated at frequencies below 1 kHz and pre-emphasis would reduce the discriminative capabilities of the algorithm [117]. For further details on the MFCCs extraction procedure, please refer to [117, 133].

GMSs are higher level features composed of the means of a Gaussian mixture model (GMM) adapted with maximum a posteriori (MAP) algorithm [134, 137]. The GMM models a Universal Background Model (UBM) and is trained on a large set of audio data by using Expectation Maximization (EM) algorithm [138]. Then, a GMS is calculated by adapting the GMM with the MAP algorithm [139] and concatenating the adapted GMM mean values (Figure 6.12b).

More in details, consider a sequence of L MFCC vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_L\}$, where each $\mathbf{x}_l$ has size $D \times 1$. The GMM representing an UBM is given by

$$p(\mathbf{x}_l|\lambda) = \sum_{j=1}^{J} w_j p(\mathbf{x}_l|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \tag{6.7}$$

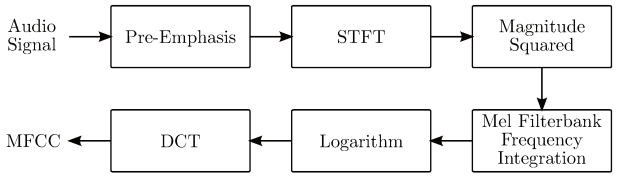
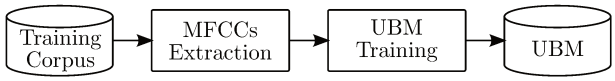
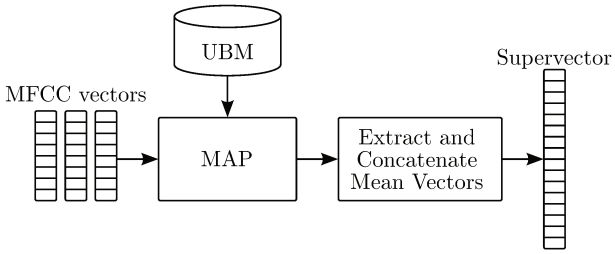


Figure 6.11: The MFCC feature extraction pipeline.



(a) UBM training scheme.



(b) The GMS extraction procedure.

Figure 6.12: Training of the Universal Background Model from MFCCs (a) and extraction of Gaussian mean supervectors (b).

where $\lambda = \{w_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j | j = 1, 2, \ldots, J\}$, w_j are the mixture weights, and $p(\cdot|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is a multivariate Gaussian distribution with $D \times 1$ mean vector $\boldsymbol{\mu}_j$ and $D \times D$ diagonal covariance matrix $\boldsymbol{\Sigma}_j$.

The GMS $\mathbf{M}$ of the sequence $\mathbf{X}$ is obtained by adapting the means of the UBM model with maximum a posteriori (MAP) algorithm and then concatenating the mean vectors:

$$\mathbf{M} = [\boldsymbol{\mu}_1^T, \boldsymbol{\mu}_2^T, \cdots, \boldsymbol{\mu}_J^T]^T, \tag{6.8}$$

where T denotes the transpose operator. Regardless the number of vectors in the sequence $\mathbf{X}$, $\mathbf{M}$ is a $DJ \times 1$ vector.

The number of Gaussians J can be determined on a validation set.

One-Class SVM

A One-Class SVM consists in a discriminant function that takes the value $+1$ in a small region that captures the majority of the data points of a set and -1 outside that region [132]. The discriminant function has the following expression:

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_i \alpha_i \cdot k(\mathbf{x}_i, \mathbf{x}) - \rho\right), \tag{6.9}$$

where $\mathbf{x}_i$ denotes the i-th support vector, and $k(\cdot,\cdot)$ represents the kernel function, e.g., the radial basis function $k(\mathbf{x},\mathbf{y}) = \exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2)$. The position of the hyperplane, thus, defines the region that represents normal data points. For each point $\mathbf{x}$ that lies outside this region, the function $f(\mathbf{x})$ takes the value -1, whereas for point inside the region, it takes the value $+1$.

The terms α_i can be found by solving the solution to the dual problem:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{ij}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j), \tag{6.10}$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq \frac{1}{\nu l}, \quad \sum_i \alpha_i = 1. \tag{6.11}$$

The term $\nu \in (0,1]$ is an hyperparameter of the algorithm that is determined on a validation set.

The offset ρ can be obtained from the Karush-Kuhn-Tucker (KKT) condition with the expression [140]:

$$\rho = \sum_j \alpha_j k(\mathbf{x}_j,\mathbf{x}_i), \tag{6.12}$$

which is satisfied for any α_i that is not at the upper or lower bound.

Template Matching

The template-matching classifier operates on a set of templates, i.e., supervectors, that can be defined a-priori or selected by the user when the OCSVM detects an abnormal sound that is not a human fall. Denoting with $\mathbf{x}$ the supervector of the input signal and with $\mathcal{Y} = \{\mathbf{y}_1,\ldots,\mathbf{y}_N\}$ the set of templates, the algorithm operates by calculating the Euclidean distance $D^{(i)} = \|\mathbf{x}-\mathbf{y}_i\|$ between the supervector to be classified and all the templates in the set. Indicating with $D_{min} = \min_i D^{(i)}$, the supervector $\mathbf{x}$ is classified as a fall if $D_{min} > \beta$ and as non-fall otherwise. The threshold β is a hyperparameter of the algorithm that can be determined on a validation set.

Table 6.5: Composition of the training-set.

Class	Nr. of occurrences	Total length (s)
Human Activity	320	593
Music	627	1180
Total	947	1773

Table 6.6: Data used in "Set 1".

(a) Composition of "Set 1".

Class	Nr. of occurrences
Human Falls	44
Human Activity	15
Music	29

(b) Templates of "Set 1".

Class	Nr. of templates	
	Clean	Noisy
Human Activity	13	11
Music	8	16
Total	21	27

6.3.2 Experimental set-up

The dataset described in Section 6.2.2 has been divided in one set for training the UBM and the OCSVM and three sets for evaluating the performance.

Training has been performed on the set shown in Table 6.5 composed of 947 occurrences (1773 s) of human activities, classical music and rock music. The assessment of the algorithm has been performed on the following datasets:

- Set 1 (Human fall and background sounds): this set comprises 44 examples of human fall sounds and 44 examples of human activity and music sounds (Table 6.6a).
- Set 2 (Human fall and object fall sounds): this set comprises 44 examples of human fall sounds and 44 examples of object fall sounds (Table 6.7a).
- Set 3 (Human fall, object fall and background sounds): this set comprises 44 examples of human fall sounds, 22 examples of background sounds and 22 examples of object fall sounds (Table 6.8a).

Table 6.7: Data used in "Set 2".

(a) Composition of "Set 2".

Class	Nr. of occurrences
Human Falls	44
Basket	7
Fork	7
Ball	8
Book	7
Bag	8
Chair	7

(b) Templates of "Set 2".

Class	Nr. of templates	
	Clean	Noisy
Basket	55	57
Fork	39	55
Ball	11	52
Book	26	57
Bag	26	56
Chair	86	89
Total	243	366

Table 6.8: Data used in "Set 3".

(a) Composition of "Set 3".

Class	Nr. of occurrences
Human Falls	44
Basket	3
Fork	4
Ball	4
Book	3
Bag	4
Chair	4
Human Activity	8
Music	14

(b) Templates of "Set 3".

Class	Nr. of templates	
	Clean	Noisy
Basket	52	57
Fork	57	57
Ball	19	55
Book	53	57
Bag	50	56
Chair	89	89
Human Activity	11	4
Music	4	11
Total	335	386

For each set, the data have been divided in four folds, each composed of 11 human falls and 11 non-falls. Then, one fold has been used for estimating the hyperparameters of the algorithm and three for calculating the performance. The final performance is calculated by using the cumulative true positives, false positives, and false negatives obtained by varying the test folds. The validation phase consisted in searching for the number of components of the UBM, the values of ν and γ of the OCSVM, and the value of the threshold β in the template-matching stage. The values assumed by these variables are summarised in Table 6.9. The method employed for the template-matching decision threshold is explained in the following section.

Table 6.9: Hyperparameters of the algorithm and search space explored in the validation phase. The search space of the template-matching threshold β is not reported, since is determined with the procedure described in Section 6.3.2.

Stage	**Hyperparameter**	**Range**
UBM	J	$1, 2, 4, \ldots, 64$
OCSVM	ν	$0.1, 02, \ldots, 1.0$
	γ	$2^{-15}, 2^{-13}, \ldots, 2^{3}$
Template-matching	β	See Section 6.3.2

All the aforementioned datasets require a set of templates for the template-matching stage of the algorithm. In the case of object falls, the set of templates has been created by classifying a set of 372 object falls with the OCSVM and selecting the occurrences misclassified as human falls. In the case of background sounds, the set of templates has been created by calculating the Euclidean distance between each occurrence of the development-set and each occurrence of a set of 470 background signals and then selecting the segment whose distance is minimum. Details on the templates sets are shown in Table 6.6b, Table 6.7b, and Table 6.8b respectively for "Set 1", "Set 2", and "Set 3".

The proposed approach has been compared to the algorithm presented in [129] based on OCSVM. The same algorithm has also been employed in [130] with a multi-microphone acquisition setup and a source separation stage. As in [129], the audio signals are divided in windows of the same lengths, and the related MFCCs are used for training the OCSVM and for classification. In [129], 7 MFCCs were extracted from audio signals sampled at 20 kHz and the length of the window was set to 1 s. Here, the feature vectors are the same of the proposed approach, i.e., they are composed of the first 13 MFCCs and their first and second derivatives. The same window length of [129] cannot be employed here, since the dataset used in this paper comprises signals with lengths less than 1 s. Thus, the length of the window corresponds to the duration of the shortest event in the dataset, and it is equal to 576 ms (71 frames). Windows are overlapped by 50%, and, as in [129], an event is classified as fall if at least two consecutive frames are classified as novelty by the OCSVM. The same grid search procedure of the proposed approach has

been adopted to search for the optimal values of ν and γ of the OCSVM.

The performance has been evaluated in terms of F_1-Measure calculated as:

$$F_1\text{-Measure} = \frac{2 \cdot tp}{2 \cdot tp + fn + fp}, \tag{6.13}$$

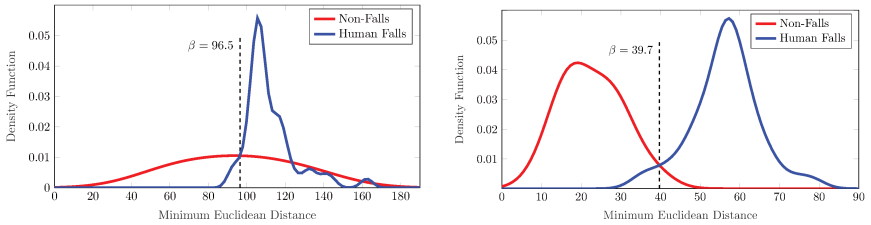
where tp is the number of correctly classified falls, fn is the number of falls misclassified as non-falls, and fp is the number of non-falls misclassified as falls.

Choice of the template-matching decision threshold

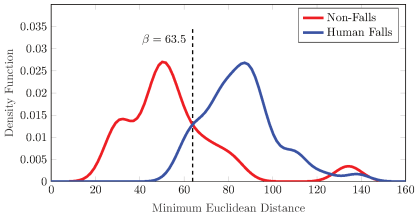
A key point of the proposed approach is the decision threshold β in the template-matching stage. Choosing a too low value would result in a low number of false negatives and a high number of false positives. On the contrary, a too high value would result in a high number of false negatives and a low number of false positives. The choice of β has been performed by calculating the minimum Euclidean distance between each fall and non-fall event in the validation set and the set of templates. Figure 6.13 and Figure 6.14 show respectively the probability distributions for the three sets in clean and noisy conditions. The decision threshold β has been chosen at the intersection point between the distribution of fall and non-fall distances. This choice represents a compromise that balances false positives and false negatives.

Observing clean condition distributions, in "Set 1" the two density are considerably overlapped, while in "Set 2" the overlap is modest. It is expected that the possible improvement of the template-matching stage will be more consistent for "Set 2" respect to "Set 1". "Set 3" contains human activity and music occurrences as "Set 1" and object falls as "Set 2": indeed, the probability distributions (Figure 6.13c) are more distinct respect to the ones of "Set 1", but not so much as the ones of "Set 2".

Noisy condition distributions, shown in Figure 6.14, are in general less distinct compared to clean condition ones. The effect of noisy is to flatten the distances of the fall and non-fall classes, thus resulting in a less discriminative capabilities of the classifier. Thus, it is expected that the performance improvement in noisy conditions will be more modest respect to the one obtained in clean condition.



(a) Probability distributions of "Set 1". (b) Probability distributions of "Set 2".

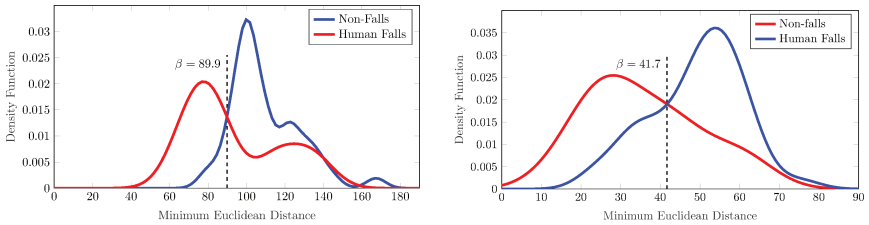


(c) Probability distributions of "Set 3".

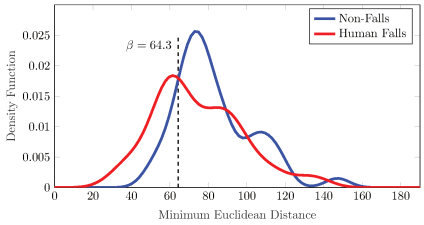
Figure 6.13: Probability distributions of the minimum Euclidean distances among the template sets, and human falls and non-falls in *clean* acoustic condition.

6.3.3 Results and remarks

Figure 6.15 shows the results in clean conditions obtained with and without the template-matching stage, respectively denoted as "OCSVM+Template-Matching" and "OCSVM". The results obtained with the method proposed in [129] are denoted with "Popescu (2009)". Observing the figure, it is evident that in all the three cases the template-matching approach is able to improve the performance with respect to "Popescu (2009)" [129] and the OCSVM only approach. In particular, in "Set 1", that comprises human falls, human activities and music, the performance improves by 2.03% with respect to OCSVM and by 19.64% with respect to "Popescu (2009)". This case can be considered as the least challenging of the three, since non-falls events are considerably different from falls ones. Conversely, "Set 2" comprises both human falls and object falls, thus it includes abnormal events whose pattern is similar to the one of human falls. Indeed, without the template-matching stage, the performance with respect to "Set 1" is 17.91% lower, mostly due the increased false positives rate that goes from 13.64% to 50.76%. The introduction of the template-matching stage considerably reduces the number of false



(a) Probability distributions of "Set 1". (b) Probability distributions of "Set 2".



(c) Probability distributions of "Set 3".

Figure 6.14: Probability distributions of the minimum Euclidean distances among the template sets, and human falls and non-falls in *noisy* acoustic condition.

positives, leading to an overall performance improvement of 20.76%. Regarding "Popescu (2009)" [129], the F_1-Measure is below both OCSVM and the proposed approach, however it is less affected by the presence of object falls, since the F_1-Measure decreases only by 0.64% . "Set 3" comprises human falls, human activities, music and object falls and represents the most realistic test condition of the three. The results obtained by using the OCSVM classifier alone is 82.25%. As expected, this result is lower than "Set 1", since object falls are also present, and higher than "Set 2", since human activities and music segments are easier to discriminate. Introducing the template-matching stage, the performance improves by 7.64%, leading to an F_1-Measure equal to 89.89%. Differently, the approach by Popescu and Mahnot [129] degrades by 5.25% with respect to "Set 1", and by 4.61% with respect to "Set 2", demonstrating that it is less robust to the concurrent presence of object falls and daily human activities sounds.

Figure 6.16 shows the results obtained for the three cases in noisy conditions. As expected, the performance decreases in all the three evaluated methods. In "Set 1", the performance decrease is modest

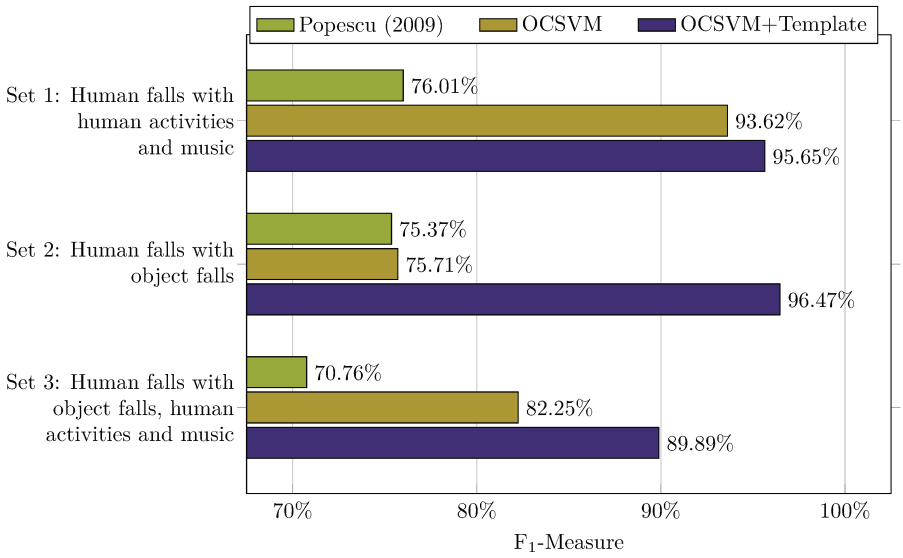


Figure 6.15: Results in *clean* conditions for the three test cases. "Set 1" comprises human falls, human activities and music. "Set 2" comprises human falls and object falls. "Set 3" comprises human falls, object falls, human activities, and music.

(2.32% for the OCSVM, 2.63% for the proposed approach, and 1.44% for "Popescu (2009)"), demonstrating that the OCSVM is able to effectively reject non-fall events corrupted by music interference. The use of the template-matching stage increases the performance by 1.72%, thus providing a significant improvement also in noisy conditions. In "Set 2", the presence object falls corrupted by music significantly decreases the performance of the OCSVM, reducing the F_1-Measure by 12.74% with respect to the clean "Set 2". Template-matching provides a performance improvement of 8.02%, leading to an F_1-Measure higher than 70%. The improvement is lower with respect to the clean "Set 2", since the variability of the music interference makes the Euclidean distances of fall and non-fall classes more similar. The method by Popescu and Mahnot [129] achieves the highest F_1-Measure in this case, confirming the good capabilities of rejecting dropping objects sound events observed in clean conditions. In "Set 3", the proposed approach improves the performance by 4.77% with respect to OCSVM and by 8.68% with respect to "Popescu (2009)", confirming that it is able to achieve the highest

performance in the most realistic scenario of the three.

In summary, the results demonstrated that the introduction of a template-matching stage significantly improves the performance both of the OCSVM only approach and of the method by Popescu and Mahnot [129]: averaging the results over "Set 1", "Set 2", and "Set 3", the absolute improvement with respect to the former is 10.14% in clean conditions and 4.84% in noisy conditions. With respect to the latter [129] the improvement is 19.96% in clean conditions and 8.08% in noisy conditions. As shown in Figure 6.15 and Figure 6.16, both in clean and noisy conditions the F_1-Measure of the method by Popescu and Mahnot [129] is close to 75% in "Set 1" and "Set 2", and close to 71% in "Set 3". The different behaviour compared to the OCSVM only approach can be attributed firstly to the different feature representation of the audio signal (MFCCs instead of supervectors). Secondly, to the strategy adopted for classification: in [129], signals are divided in windows and a fall is detected if at least two consecutive windows are classified as fall. Differently, in the proposed algorithm, the overall signal is represented by a single supervector and classified as fall or non fall.

Comparing the results in clean (Figure 6.15) and noisy (Figure 6.16) conditions, it is evident that techniques for reducing the impact of additive noise are needed. Additionally, the proposed solution requires the intervention of the user for selecting the templates after the first classification stage performed by the OCSVM. This aspect will be addressed in future works in order to make the algorithm completely unsupervised.

Summing up, this contribution proposed a combined OCSVM and template-matching classifier to discriminate human falls from non-falls in a semi-supervised framework. Fall signals are captured by means of a Floor Acoustic Sensor, then MFCCs and GMSs are extracted from the acquired signal. The OCSVM discriminates between normal and abnormal acoustic events and the template-matching stage performs the final fall/non-fall decision. This stage employs a set of template supervectors represented by the events detected as abnormal by the OCSVM and marked as false positives by the user. The performance of the algorithm has been evaluated on a corpus containing human falls reproduced by a human-mimicking doll and non-falls represented by sounds of falling objects, human activities, and music. In order to confirm the signifi-

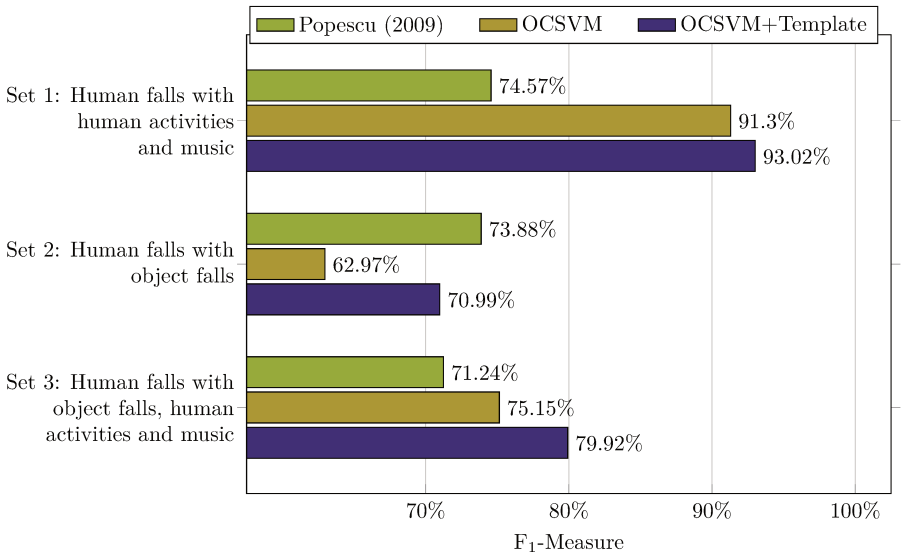


Figure 6.16: Results in noisy conditions for the three test cases. "Set 1" comprises human falls, human activities and music. "Set 2" comprises human falls and object falls. "Set 3" comprises human falls, object falls, human activities, and music.

cance of the approach, it has been compared to the method proposed in [129] and to the OCSVM only approach. The results showed that in the most realistic scenario, the proposed solution provides a performance improvement equal to 7.64% in clean conditions and equal to 4.77% in noisy conditions with respect to the OCSVM only approach, and equal to 19.13% and to 8.68% with respect to [129].

6.4 Fall detection with End-To-End CNN Autoencoders

As aforementioned in Section 6.2, in "machine learning" methods, the algorithm learn from the data how to discriminate falls from non-falls adopting a "supervised" or "unsupervised" strategy. Regardless of the used approach, machine learning tasks require that the inputs are mathematically and computationally convenient to process, so researchers have traditionally relied on a two-stage strategy: some features are extracted from the raw signals of dataset and are then used as input for the suc-

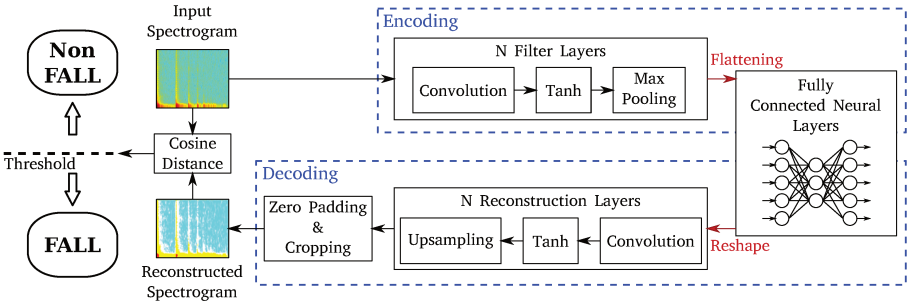


Figure 6.17: The proposed approach scheme.

cessive tasks. The choice and design of the appropriate features requires considerable expertise about the problem and constitutes a significant engineering effort.

In recent years, thanks to the success of deep learning methods, feature learning approaches have become increasingly popular for transforming the raw data inputs to a representation that can be exploited in machine learning tasks, minimizing the need of prior knowledge of application domain. Furthermore, such approaches are often able to generalize well real-world data compared to traditional hand-crafted features, resulting in an increase in performance of classification or regression tasks. The end-to-end learning is a particular example of feature learning, where the entire stack, connecting the input to the desired output, is learned from data [141]. As in feature learning, only the tuning of the model hyperparameters requires some expertise, but even that process can be automated [142].

In this second contribution on fall detection, an end-to-end acoustic fall detection approach is presented. A deep convolutional neural network autoencoder is trained with the signals, gathered by a FAS, corresponding to sounds that commonly occurring in a home (e.g., voices, footsteps, music, etc.). Since the sound produced by a human fall should be considerably different from the ones used for the training, it will be recognized as "novelty" by the network and classify as Fall.

6.4.1 Proposed Approach

In the state of the art the majority of fall detection systems are logically designed as a cascade of elements which perform some sub-task

(e.g. feature extraction, modelling, classification, etc.). Each task is independently developed and generally requires the tuning of a number of hyperparameters using an experimental procedure and some a prior knowledge about the domain of the problem.

The proposed approach, showed in Figure 6.17, is designed according to the end-to-end paradigm and then, the entire stack, connecting the input to the desired output, is learned from data. End-to-End is a feature learning strategy that, in presence of sufficient training data, may result in better performance than systems based on handcrafted features, since the training procedure automatically select the salient information. Therefore, if were possible to analyze the feature learned by the network with end-2-end strategy, there should be clues about what kind of information is important for a specific task.

The system core is a deep convolutional neural network autoencoder. Some exhaustive discussion about this type of network can be easily found in literature [18, 143, 144]. The network input consists of the normalized log-power spectrograms of the signals calculated with a STFT on windows of 32 ms and overlapped by 50%. Due to the presence of fully-connect neural layers, the input dimensions must be fixed. After having identified the widest spectrogram extracted from the dataset, the other ones have been extended with some AWGN frames added at the end. Each input consist in a $f \times t$ matrix, where f are the positive points of discrete Fourier transform and t are the number of windows considered in time. The output of the autoencoder are the reconstructed spectrograms. To classify an event, a distance measurement between input and output must be made with some heuristic. If the distance exceeds a certain threshold, automatically defined by the algorithm during the training phase, the system label the output as "Fall" or as "Non Fall" otherwise. In this work the cosine distance has been used:

$$D_C(v,u) = 1 - \frac{u \cdot v}{\|u\| \, \|v\|} = 1 - \frac{\sum_{k=1}^{n} u(k)v(k)}{\sqrt{\sum_{k=1}^{n} u(k)^2}\sqrt{\sum_{k=1}^{n} v(k)^2}}, \qquad (6.14)$$

where u and v are the the vectors obtained flattening the input and the output spectrograms and n are the length of this vectors. According to the cosine definition, the value of the distance always has a value between

-1 and $+1$, where $+1$ indicates two equal vectors while -1 indicates two opposite vectors. The added AWGN part of the spectrums was not considered to calculate the distance. The choice of this heuristic allowed to make distance measurements independents of the size of the initial spectrum. The structure of the autoencoder is not defined a priori, but it is chosen through a phase of cross-validation during which the network parameters are varied with a random search strategy.

6.4.2 Experimental set-up

In this contribution a subset of the human fall dataset described in Section 6.2.2 has been used. It was composed by only the human fall sounds, namely novelty and the background sounds, i.e. the sounds of normal activities (voices, footsteps, etc.) and of the three musical tracks.

Since in this work a novelty approach is presented, the dataset has been divided in two groups: the former composed only of background sounds used for the training; the latter composed of both background sound and novelty sounds, i.e, the human falls, used in development and test phase. In order to assess the classification accuracy in noisy conditions, a second version of human fall sounds were crated in which a musical background was recorded and then digitally added to the fall events.

The input spectrograms of the audio signals has been calculated with a fft point number of 256 and a windows size of 256 samples (32 ms at sample rate of 8 kHz). The longest spectrogram present in the dataset is composed of 197 frames. Therefore the resulting input matrix dimension $f \times t$ is 129×197. The optimization of the experiment hyper-parameters has been carried out using the random-search technique. Table 6.10 shows the parameters used in the random-search, and their ranges. The parameters of the network architecture are related only to the encoding part of autoencoder since the decoding part is its mirrored version. Instead other parameters, described below, have been set to the same value for all experiments. The activation function for each layer, whether they are convolutional or fully connected, have been set to *tanh*. "Adam" [145] has been used as optimization algorithm for the traing phase. The loss function used was *mlse*. The initialization algorithm for the weight of the autoencoder was Glorot Uniform [146].

Table 6.10: Hyper-parameters optimized in the random-search phase, and their range.

Parameter	Range	Distribution	Parameter	Range	Distribution
Cnn layer Nr.	[1-3]	uniform	Batch size	[10%-25%]	log-uniform
Kernel shape	[3x3-8x8]	uniform	Max pool shape	[1x1-5x5]	uniform
Kernel Nr.	[4-64]	log-uniform	Max Pool	All[7]-Only end[8]	uniform
MLP layers Nr.	[1-3]	uniform	Dropout	[Yes-No]	uniform
MLP layers dim.	[128-4096]	log-unifom	Drop rate	[0.5-0.6]	normal
Stride	[1x1-3x3]	uniform	Learning rate	$[10^{-4}$-$10^{-2}]$	log-unifom

The number of epoch has been set to 1000, while the patience, that is the number of epoch without an Auc improvement on a devset to wait before stopping the training phase, has been set to 40.

In order to implements a 4 fold cross-validation, the signals not being part of training-set have been divided in four folds, each composed of 11 human falls and 11 non-falls signals. Then, one fold has been used as validation-set and the remaining three for calculating the performance in test phase. In cross-validation phase the scores have been evaluated in term of AUC. Here also the optimal thresholds have been infer by searching points on ROC curves closest to the $(0, 1)$: $d_{min} = \sqrt{(1 - fpr)^2 + (1 - tpr)^2}$. At the end the final performance has been evaluated in term of $F_1 - Measure$ by mediating the results obtained on individual folds.

The proposed approach has been compared with 2 algorithms both based on OCSVM.

The first is the approach presented in Section 6.3. The second algorithm, presented in [129], the audio signals are divided in windows of the same lengths, and the related MFCCs are used for training the OCSVM and for classification. Both for comparison purposes and for different composition of the dataset, we have introduced some changes to the original approach: we employee the same MFCCs used in [147]. The window length used for the analysis corresponds to the duration of the shortest event in our dataset, and it is equal to 576 ms (71 frames). Windows are overlapped by 50%, and, as in [129], an event is classified as fall if at least two consecutive frames are classified as novelty by the OCSVM. On both the target algorithms, the grid search procedure has

[7] After each Conv. layer
[8] At the end of cnn part

Table 6.11: Best hyper-parameters found in random-search phase for *clean* and *noisy* condition

	Clean				Noisy			
Parameter	Fold1	Fold2	Fold3	Fold4	Fold1	Fold2	Fold3	Fold4
Cnn layer Nr.	3	3	3	2	3	3	3	3
Kernel shape	8x8	7x7	5x5	8x8	8x8	7x7	8x8	8x8
Kernel Nr.	[16,16,8]	[32,16,16]	[8,8,8]	[32,16]	[32,32,8]	[32,32,8]	[32,32,32]	[8,8,8]
Max Pool Position	only end	only end	all	all	only end	only end	all	only end
Max pool shape	5x5	3x3	5x5	4x4	3x3	5x5	5x5	3x3
Stride	3x3	3x3	1x1	3x3	3x3	3x3	1x1	3x3
MLP layers Nr.	2	1	1	1	2	2	1	3
MLP layers dim.	[16,231]	96	32	32	[48,153]	[16,2084]	128	[48,1952,1952]
Learning rate($\times 10^{-4}$)	4.89	4.08	15.09	15.44	1.56	4.46	1.01	1.00
Batch size	11.26%	10.81%	13.59%	21.10%	20.06%	12.55%	13.51%	13.13%
Drop rate	0.64	0.57	0.53	0.55	0.58	0.53	0.55	0.59

been adopted to find the optimal values ν and γ of the OCSVM and the number of mixture of the GMM-UBM.

6.4.3 Results an remarks

The results for both clean and noisy are reported in Figure 6.18. The comparative algorithms are denoted with "Popescu (2009)" and "OCSVM" respectively, while the proposed approach is named "Autoencoder". It is immediately clear that the proposed approach outperforms the other in both conditions. In fact, in clean condition, it gains about 1% compared to "OCSVM" and about 18.6% compared to "Popescu (2009)". Moreover the proposed algorithm results to be very robust in noisy condition, where the score of "OCSVM" falls down while the score of "Popescu (2009)" remains around the previous case. In particular the performance improves by 32.05% with respect to "OCSVM" and by 21.14% with respect to "Popescu (2009)". Clearly the end-to-end method seems insensitive with respect to the corrupted human fall signals when the novelty sounds are dissimilar respect to normality model learned from the background sounds. In Table 6.11 are reported the hyperparameters that have led to the results discussed above.

Furthermore other experiments were made up with a manual tuning of parameters. Particularly have been investigated deeper architectures composed up to 5 convolutional layer. We found that increasing the depth on cnn part (5 layers) with a different kernel number for each layers of [32,32,16,16,8] and a kernel dimension for all layers of 4x4, a max pooling after only the first three convolutional layer of 2x2 and two MLP layer of 1024 and 512, leads to considerable improvements.

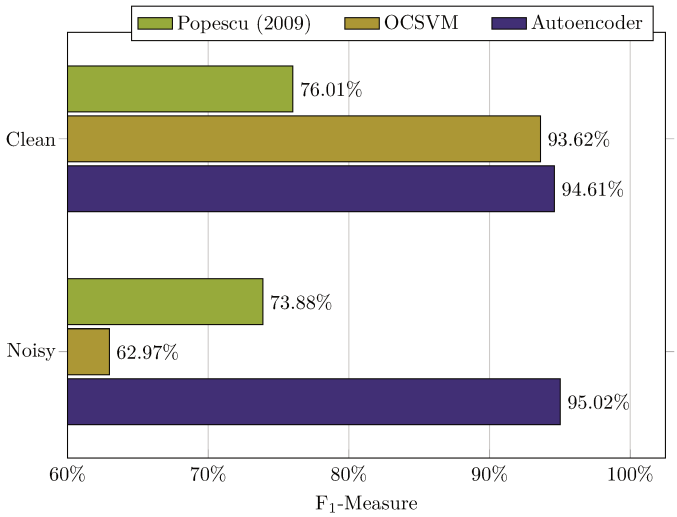


Figure 6.18: Results in *clean* and *noisy* conditions for the three test cases.

In effect the final $F_1 - Measure$ increases to 95.42% both in clean and noisy condition.

Summing up, in this contribution the author proposed an end-to-end approach composed of a deep-convolutional-autoencoder with a downstream threshold classifier, that is a purely unsupervised approach to acoustic fall detection. The new method exploits the reconstruction error of the autoencoder. When a sound that the network has never seen in training phase occurs, the reconstruction error increases allowing the recognition of a novelty. The algorithm has been trained on a large corpus of background signals, that is human activity noise and music, and evaluated with human-fall sound and other instances of background sounds. It has been evaluated in two different conditions: the first with a clean version of human fall sounds and the second with corrupted version of the same. Moreover a comparison was made with two different algorithms, one proposed in [129] and the other based on OCSVM [147]. The results showed that the proposed solution leads to an average improvement of about 20% with respect to [129] and of about 16.5% compared to the OCSVM based approach.

Chapter 7

Conclusions

In this thesis, the issues related the Infant Cry Detection in adverse acoustic environments have been addressed by means of algorithms for digital signal processing and machine learning systems. The case examined is that of Neonatal Intensive Care Units (NICUs). In this context the use of devices for automatic detection of crying is very useful for monitoring patients.

Due the sensitivity of the data handled and to privacy concerns, researchers generally do not disclose their datasets. Therefore, a dataset has been acquired in the NICU of the Salesi Hospital of Ancona. More than 900 hours of raw multichannel audio data has been acquired and over an half of this data has been manually labeled in order to distinguish between "cry" and "non-cry" segments. Moreover, in order to avoid issues that usually arise when NICUs are involved, such as sanitary concerns resulting from researchers accessing a NICU environment, and bureaucratic concerns to obtain the authorization to make audio recording in NICU, a synthetic dataset has been created by means of a suitable acoustic scene simulation procedure. The rationale here was to explore the possibility that an expert system trained by means of a synthetic dataset could represent a viable alternative to the employment of real-life dataset. This can be very useful, especially when the operating conditions make the acquisition of training data a big issue. With regard to the real dataset, it present very high noise levels, typical of the NICUs and which hinder the detection. To overcome this problem it is necessary to pre-process the acquired data with audio enhancement strategies. Given the multi-channel nature of the acquired dataset, the most natural choice was to make use of audio beamforming techniques.

In Chapter 4 has been presented a preliminary study on neural beam-

forming approaches for audio enhancing. In particular, an algorithm was proposed for the neural estimation of the Direction of Arrival (DOA). The output of the algorithm has been used to modify adaptively the beam pointing direction of a beamformer by modifying its filters weights. The experiments reported in Section 4.5 show that neural DOA estimation exhibits excellent performance with respect to a reference technique such as MUSIC. When used in conjunction with a classic beamforming algorithm, its higher precision also improves its capability in enhancing the quality of speech affected by noise and reverberation with respect to a MUSIC DOA estimator in conjunction with the same beamforming algorithm. The performance, evaluated in terms of both PESQ and Itakura-Saito distance, is further increased in conjunction with a well-known speech algorithm by Ephraim *et al.* [35].

In Chapter 5, a deep neural network based approach for infant cry detection has been proposed. It makes use of a Convolutional Neural Network having as input Log-Mel features extracted from the audio signals. Log-Mels are extracted from audio pre-processed version of signals. The pre-processing stage is characterized by a LVCM beamformer and a speech enhancement algorithm (OMLSA). The investigation of the collected audio signals, however, revealed a few concerns relating the microphone array position and orientation. To this purpose, some alternatives to the original approach have been investigated. One of the revised approach operates on a single-channel, without additional preprocessing. The second approach exploits three channels, among the eight provided by the array, as input of a DNN, thus incorporating the multi-channel processing directly into the DNN.

The evaluations carried out in the experimental phase reveal that although the original approach, when trained over the real life dataset, remains the best performer with a 87.28% detection rate, it may show a performance drop if trained over a synthetic dataset. At the same time has been identified a more robust approach that, although it achieves a second-best result with a detection score of 84.53% when trained over a real-life dataset, it is the best performer when trained over a synthetic dataset with a 80.48% detection score. This allows to confirm about the effectiveness of the data-driven approach, full trained by means of simulated acoustic data. To have a performance comparison with respect

to State of Art, the proposed approaches have been compared to the work by Raboshchuk *et al.* [70]. The obtained detection performance is remarkably superior compared to the one achieved by the comparative method, thus supporting the effectiveness of the proposed approaches.

The achieved results prove that a synthetic dataset can be a useful replacement with respect to a real-life dataset, at least in the early design process. Indeed, it permits to lower the interaction with a sensitive environment such as a NICU, to the bare minimum. Moreover it can be adjusted to include changes to the environment as needed, without requiring an additional acquisition session.

7.1 Future research topics

Infant cry detection in noisy acoustic environments is a field of research that is not much explored and therefore it present many aspects to investigate.

Wanting to carry on the way indicated by this thesis, a first aim for future research could be to integrate the study presented in Chapter 4 in the proposed basic approach described in Section 5.2.4. This could solve the problems deriving from accidental misalignment of the device by "tracking" the monitored subject.

It should be noted that as discussed in Section 5.5, DNN also seems able to handle these accidental shifts if they are appropriately represented in the training set. To this purpose, another strategy to handle the problem could be to revise the simulation dataset, to include some occurrences where the microphone array does not target the intended subject, to bridge the still existing gap between the DNN trained by using synthetic and the one trained by using real data.

Another way to close this gap could be to adopt strategies for the on-site adaptation of models. The network trained on the synthetic dataset could exploit domain-specific data to refine its models with reinforcement learning strategies [148].

One aspect to consider is that into a NICU there are many cribs and it is therefore probable the presence of multiple detection devices. These could be interconnected to share data in a kind of data fusion strategy. For example, the outputs of the DNN of each device present in the NICU

could be collected on a server and revalued off-line.

Another research aim is to adapt detection algorithms in embedded devices to make them work in real time, with good performance. This would be the first step towards the realization of a concrete device for infant cry detection.

List of Publications

[1] D. Ferretti, E. Principi, S. Squartini, and L. Mandolini, "An experimental study on new features for activity of daily living recognition," *in Proc. of International Joint Conference on Neural Networks (IJCNN),* 2016, pp. 3958–3965.

[2] D. Droghini, D. Ferretti, E. Principi, S. Squartini, and F. Piazza, "A combined one-class svm and template matching approach for user-aided human fall detection by means of floor acoustic features," *Computational Intelligence and Neuroscience,* vol. 2017, 2017, Article ID 1512670.

[3] D. Droghini, D. Ferretti, E. Principi, S. Squartini, and F. Piazza, "An End-To-End Unsupervised Approach employing Convolutional Neural Network Autoencoders for Human Fall Detection," *Neural Advances in Processing Nonlinear Dynamic Signals,* pp. 185–196, Springer International Publishing, Cham, 2019.

[4] S. Tomassetti, L. Gabrielli, E. Principi, D. Ferretti, and S. Squartini, "Neural Beamforming for Speech Enhancement: Preliminary Results," *Neural Advances in Processing Nonlinear Dynamic Signals,* pp. 37–47, Springer International Publishing, Cham, 2019.

[5] F. Vesperini, D. Droghini, D. Ferretti, E. Principi, L. Gabrielli, S. Squartini, and F. Piazza, "A hierarchic multi-scaled approach for rare sound event detection," *Detection and Classification of Acoustic Scenes and Events (DCASE), IEEE Audio and Acoustic Signal Processing Challenge,* Ancona, Italy, 2017, Technical Report.

[6] D. Ferretti, M. Severini, E. Principi, A. Cenci, and S. Squartini, "Infant cry detection in adverse acoustic environments by

using deep neural networks," *in Proc. of European Signal Processing Conference (EUSIPCO),*, Rome, Italy, Sept. 3-7 2018, pp. 997–1001.

[7] M. Severini, D. Ferretti, E. Principi, and S. Squartini, "Automatic detection of cry sounds in nicus by using deep learning and acoustic scene simulation," *Journal of Biomedical and Health Informatics (JBHI),* submitted.

Bibliography

[1] György Várallyay Jr, "The melody of crying," *international journal of pediatric otorhinolaryngology*, vol. 71, no. 11, pp. 1699–1708, 2007.

[2] MD Livera, B Priya, A Ramesh, PN Suman Rao, V Srilakshmi, M Nagapoornima, AG Ramakrishnan, M Dominic, et al., "Spectral analysis of noise in the neonatal intensive care unit," *The Indian Journal of Pediatrics*, vol. 75, no. 3, pp. 217, 2008.

[3] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] SRN Reddy, Dinesh Kumar, et al., "Review of smart health monitoring approaches with survey analysis and proposed framework," *IEEE Internet of Things Journal*, 2018.

[5] CF Zachariah Boukydis and Barry M Lester, *Infant crying: Theoretical and research perspectives*, Springer Science & Business Media, 2012.

[6] A. Chittora and H.A. Patil, "Classification of normal and pathological infant cries using bispectrum features," in *Proc. of European Signal Processing Conference (EUSIPCO)*, Nice, France, Aug. 31 - Sept. 4 2015, pp. 639–643.

[7] Orion Fausto Reyes-Galaviz, Sergio Daniel Cano-Ortiz, and Carlos Alberto Reyes-García, "Evolutionary-neural system to classify infant cry units for pathologies identification in recently born babies," in *Proc. of Mexican International Conference on Artificial Intelligence (MICAI)*, Atizapán de Zaragoza, Mexico, Oct. 27-31 2008, IEEE, pp. 330–335.

[8] Z Benyó, Z Farkas, A Illényi, G Katona, and G Várallyay Jr, "Information transfer of sound signals. a case study: The infant cry. is it noise of an information?," in *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, Prague, Czech Republic, 2004, pp. 2774–2781, Citeseer.

[9] Vinay Kumar Mittal, "Discriminating features of infant cry acoustic signal for automated detection of cause of crying," in *Proc. of International Symposium on Chinese Spoken Language Processing (ISCSLP)*, Tianjin, China, Oct. 17-20 2016, pp. 1–5.

[10] S. Ntalampiras, "Audio pattern recognition of baby crying sound events," *Journal of the Audio Engineering Society*, vol. 63, no. 5, pp. 358–369, 2015.

[11] Anshu Chittora and Hemant A Patil, "Newborn infant's cry analysis," *International Journal of Speech Technology*, vol. 19, no. 4, pp. 919–928, 2016.

[12] Susan M. Ludington-Hoe, Xiaomei Cong, and Fariba Hashemi, "Infant crying: nature, physiologic consequences, and select interventions," *Neonatal network*, vol. 21, no. 2, pp. 29, 2002.

[13] 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.

[14] Sahar MA Hassanein, Nehal M El Raggal, and Amani A Shalaby, "Neonatal nursery noise: practice-based learning and improvement," *The Journal of Maternal-Fetal & Neonatal Medicine*, vol. 26, no. 4, pp. 392–395, 2013.

[15] Karen A Thomas and Patricia A Martin, "Nicu sound environment and the potential problems for caregivers," *Journal of Perinatology*, vol. 20, no. S1, pp. S94, 2000.

[16] Helen Shoemark, Edward Harcourt, Sarah J Arnup, and Rod W Hunt, "Characterising the ambient sound environment for infants

in intensive care wards," *Journal of paediatrics and child health*, vol. 52, no. 4, pp. 436–440, 2016.

[17] Simon S Haykin, *Neural networks and learning machines*, vol. 3, Pearson, 2009.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[20] Li Deng, Dong Yu, et al., "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[21] Yoshua Bengio, Aaron Courville, and Pascal Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[22] Sharon Gannot and Israel Cohen, "Adaptive beamforming and postfiltering," in *Springer Handbook of Speech Processing*, pp. 945–978. Springer, 2007.

[23] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M.L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. of ICASSP*, May 2016, pp. 5745–5749.

[24] B. Li, T.N. Sainath, R.J. Weiss, K.W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Proc. of Interspeech*, Sep. 8-12 2016, pp. 1976–1980.

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

[26] Hui Zhang, Xueliang Zhang, and Guanglai Gao, "Multi-channel speech enhancement based on deep stacking network," in *Proc. of the 4th CHiME Speech Separation and Recognition Challenge*, San Francisco, CA, USA, 2016.

[27] Hakan Erdogan, Tomoki Hayashi, John R Hershey, Takaaki Hori, Chiori Hori, Wei-ning Hsu, Suyoun Kim, Jonathan Le Roux, Zhong Meng, and Shinji Watanabe, "Multi-channel speech recognition: LSTMs all the way through," in *Proc. of the 4th CHiME Speech Separation and Recognition Challenge*, San Francisco, CA, USA, 2016.

[28] Israel Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Transactions on speech and audio processing*, vol. 11, no. 5, pp. 466–475, 2003.

[29] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," in *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, Canada, Apr. 15-20 2018.

[30] Tony Robinson, Jeroen Fransen, David Pye, Jonathan Foote, and Steve Renals, "Wsjcamo: a british english speech corpus for large vocabulary continuous speech recognition," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on.* IEEE, 1995, vol. 1, pp. 81–84.

[31] Emanuele Principi, Stefano Squartini, Roberto Bonfigli, Giacomo Ferroni, and Francesco Piazza, "An integrated system for voice command recognition and emergency detection based on audio signals," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5668–5683, 2015.

[32] Jinyu Li, Li Deng, Reinhold Haeb-Umbach, and Yifan Gong, *Robust Automatic Speech Recognition: A Bridge to Practical Applications*, Academic Press, 2015.

[33] Ralph Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[34] Xiong Xiao, Shengkui Zhao, Xionghu Zhong, Douglas L Jones, Eng Siong Chng, and Haizhou Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2814–2818.

[35] Yariv Ephraim and David Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[36] Jacob Benesty, Jingdong Chen, and Yiteng Huang, *Microphone array signal processing*, vol. 1, Springer Science & Business Media, 2008.

[37] Jack Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[38] L.J. Griffiths and C.W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 1, pp. 27–34, 1982.

[39] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function gsc and postfiltering," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 6, pp. 561–571, 2004.

[40] A. Hussain, M. Chetouani, S. Squartini, A. Bastari, and F. Piazza, *Nonlinear Speech Enhancement: An Overview*, pp. 217–248, Springer Berlin Heidelberg, 2007.

[41] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Using neural network front-ends on far field multiple microphones based speech recognition," in *Proc. of ICASSP*, Florence, Italy, May 4-9 2014, pp. 5542–5546.

[42] S. Renals and P. Swietojanski, "Neural networks for distant speech recognition," in *Proc. of HSCMA*, 2014, pp. 172–176.

[43] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, 2014.

[44] Y. Hoshen, R.J. Weiss, and K.W. Wilson, ," in *Speech acoustic modeling from raw multichannel waveforms*, 2015, pp. 4624–4628.

[45] W Knecht, M. E. Schenkel, and George S. Moschytz, "Neural network filters for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 6, pp. 433–438, 1995.

[46] V Yoganathan and TJ Moir, "Multi-microphone adaptive neural switched Griffiths-Jim beamformer for noise reduction," in *Proc. of the 10th International Conference on Signal Processing*, 2010, pp. 299–302.

[47] S. Araki, T. Hayashi, M. Delcroix, M. Fujimoto, K. Takeda, and T. Nakatani, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proc. of ICASSP*, Aug. 2015, pp. 116–120.

[48] Charles Knapp and Glifford Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[49] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung, "A statistical model-based voice activity detection," *IEEE signal processing letters*, vol. 6, no. 1, pp. 1–3, 1999.

[50] J.B. Allen and D.A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoustic Society of America*, p. 943, April 1979.

[51] Keisuke Kinoshita, Marc Delcroix, Takuya Yoshioka, Tomohiro Nakatani, Armin Sehr, Walter Kellermann, and Roland Maas, "The REVERB challenge: A common evaluation framework for

dereverberation and recognition of reverberant speech," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2013, pp. 1–4.

[52] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, May 2015.

[53] Philipos C Loizou, *Speech enhancement: theory and practice*, CRC press, 2007.

[54] B. Reggiannini, S.J. Sheinkopf, H.F. Silverman, X. Li, and B.M. Lester, "A flexible analysis tool for the quantitative acoustic assessment of infant cry," *Journal of Speech Language and Hearing Research*, vol. 56, no. 5, pp. 1416–1428, 2013.

[55] S. Orlandi, P.H. Dejonckere, J. Schoentgen, J. Lebacq, N. Rruqja, and C. Manfredi, "Effective pre-processing of long term noisy audio recordings: An aid to clinical monitoring," *Biomedical Signal Processing and Control*, vol. 8, no. 6, pp. 799–810, 2013.

[56] María A Ruíz Díaz, Carlos A Reyes García, Luis C Altamirano Robles, Jorge E Xalteno Altamirano, and Antonio Verduzco Mendoza, "Automatic infant cry analysis for the identification of qualitative features to help opportune diagnosis," *Biomedical Signal Processing and Control*, vol. 7, no. 1, pp. 43–49, 2012.

[57] Nemir Ahmed Al-Azzawi, "Automatic recognition system of infant cry based on f-transform," *International Journal of Computer Applications*, vol. 102, no. 12, 2014.

[58] Rami Cohen and Yizhar Lavner, "Infant cry analysis and detection," in *Proc. of Electrical & Electronics Engineers in Israel (IEEEI)*, Eilat, Israel, Nov. 14-17 2012, pp. 1–5.

[59] L. Abou-Abbas, C. Tadj, C. Gargour, and L. Montazeri, "Expiratory and inspiratory cries detection using different signals' decomposition techniques," *Journal of Voice*, vol. 31, no. 2, pp. 259.e13–259.e28, 2017.

[60] Gaurav Naithani, Jaana Kivinummi, Tuomas Virtanen, Outi Tammela, Mikko J. Peltola, and Jukka M. Leppänen, "Automatic segmentation of infant cry signals using hidden markov models," *EEURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, no. 1, pp. 1, 2018.

[61] Lina Abou-Abbas, Hesam Fersaie Alaie, and Chakib Tadj, "Automatic detection of the expiratory and inspiratory phases in newborn cry signals," *Biomedical Signal Processing and Control*, vol. 19, pp. 35–43, 2015.

[62] Rafael Torres, Daniele Battaglino, and Ludovick Lepauloux, "Baby cry sound detection: a comparison of hand crafted features and deep learning approach," in *Proc. of International Conference on Engineering Applications of Neural Networks (EANN)*. 2017, pp. 168–179, Springer.

[63] Silvia Orlandi, Carlos Alberto Reyes Garcia, Andrea Bandini, Gianpaolo Donzelli, and Claudia Manfredi, "Application of pattern recognition techniques to the classification of full-term and preterm infant cry," *Journal of Voice*, vol. 30, no. 6, pp. 656–663, 2016.

[64] Maŕıa Antonia Rúız, Carlos Alberto Reyes, and Luis Carlos Altamirano, "On the implementation of a method for automatic detection of infant cry units," *Procedia Engineering*, vol. 35, pp. 217–222, 2012.

[65] Y. Lavner, R. Cohen, D. Ruinskiy, and H. Ijzerman, "Baby cry detection in domestic environment using deep learning," in *Proc. of International Conference on the Science of Electrical Engineering (ICSEE)*, Eilat, Israel, Nov. 16-18 2016, pp. 1–5.

[66] Kathleen Wermke, W Mende, C Manfredi, and P Bruscaglioni, "Developmental aspects of infant's cry melody and formants," *Medical Engineering & Physics*, vol. 24, no. 7-8, pp. 501–514, 2002.

[67] Paolo Vecchiotti, Fabio Vesperini, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "Convolutional neural networks

with 3-d kernels for voice activity detection in a multiroom environment," in *Multidisciplinary Approaches to Neural Computing*, vol. 69, pp. 161–170. Springer, 2018.

[68] Y. Wang, L. Neves, and F. Metze, "Audio-based multimedia event detection using deep recurrent neural networks," in *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Shanghai, China, Mar. 20-25 2016, pp. 2742–2746.

[69] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio, *Deep learning*, vol. 1, MIT press Cambridge, 2016.

[70] Ganna Raboshchuk, Climent Nadeu, Sergio Vidiella Pinto, Oriol Ros Fornells, Blanca Muñoz Mahamud, and Ana Riverola de Veciana, "Pre-processing techniques for improved detection of vocalization sounds in a neonatal intensive care unit," *Biomedical Signal Processing and Control*, vol. 39, pp. 390–395, 2018.

[71] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[72] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al., "The htk book," *Cambridge university engineering department*, vol. 3, pp. 175, 2002.

[73] James Bergstra and Yoshua Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research (JMLR)*, vol. 13, no. Feb, pp. 281–305, 2012.

[74] Kendrick Boyd, Kevin H. Eng, and C. David Page, "Area under the precision-recall curve: Point estimates and confidence intervals," in *Machine Learning and Knowledge Discovery in Databases*, Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, Eds., Berlin, Heidelberg, 2013, pp. 451–466, Springer Berlin Heidelberg.

[75] Saisakul Chernbumroong, Shuang Cang, Anthony Atkins, and Hongnian Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1662–1674, 2013.

[76] Bin Huang, Guohui Tian, Hao Wu, and Fengyu Zhou, "A method of abnormal habits recognition in intelligent space," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 125–133, 2014.

[77] Barnan Das, Chao Chen, Adriana M Seelye, and Diane J Cook, "An automated prompting system for smart environments," in *Proc. of Toward Useful Services for Elderly and People with Disabilities*, pp. 9–16. Springer, 2011.

[78] Yi Chu, Young Chol Song, Richard Levinson, and Henry Kautz, "Interactive activity recognition and prompting to assist people with cognitive disabilities," *J Ambient Intell Smart Environ*, vol. 4, no. 5, pp. 443–459, 2012.

[79] Sarvesh Vishwakarma and Anupam Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," *The Visual Computer*, vol. 29, no. 10, pp. 983–1009, 2013.

[80] Bobak Mortazavi, Suneil Nyamathi, Sunghoon Ivan Lee, Thomas Wilkerson, Hassan Ghasemzadeh, and Majid Sarrafzadeh, "Near-realistic mobile exergames with wireless wearable sensors," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 2, pp. 449–456, 2014.

[81] Dorra Trabelsi, Sabah Mohammed, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat, "An unsupervised approach for automatic activity recognition based on hidden markov model regression," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 829–835, 2013.

[82] Eunju Kim, Sumi Helal, and Diane Cook, "Human activity recognition and pattern discovery," *Pervasive Computing, IEEE*, vol. 9, no. 1, pp. 48–53, 2010.

[83] Natalia Díaz Rodríguez, Manuel P Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores, "A survey on ontologies for human

behavior recognition," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, pp. 43, 2014.

[84] Oliver Brdiczka, James L Crowley, and Patrick Reignier, "Learning situation models in a smart home," *IEEE Trans. Syst., Man, Cybern. B*, vol. 39, no. 1, pp. 56–63, 2009.

[85] Shian-Ru Ke, Hoang Le Uyen Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, pp. 88–131, 2013.

[86] Daniel Weinland, Remi Ronfard, and Edmond Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[87] Veerle Claes, Els Devriendt, Jos Tournoy, and Koen Milisen, "Attitudes and perceptions of adults of 60 years and older towards in-home monitoring of the activities of daily living with contactless sensors: An explorative study," *International Journal of Nursing Studies*, vol. 52, no. 1, pp. 134–148, 2015.

[88] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[89] Liming Chen, Jesse Hoey, Chris D Nugent, Diane J Cook, and Zhiwen Yu, "Sensor-based activity recognition," *IEEE Trans. Syst., Man, Cybern. C*, vol. 42, no. 6, pp. 790–808, 2012.

[90] Can Tunca, Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy, "Multimodal wireless sensor network-based ambient assisted living in real homes with multiple residents," *Sensors*, vol. 14, no. 6, pp. 9692–9719, 2014.

[91] Hande Alerndar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy, "Aras human activity datasets in multiple homes with multiple residents," in *Proc. of PervasiveHealth*, 2013, pp. 232–235.

[92] Corinne Belley, Sebastien Gaboury, Bruno Bouchard, and Abdenour Bouzouane, "An efficient and inexpensive method for activity recognition within a smart home based on load signatures of appliances," *Pervasive and Mobile Computing*, vol. 12, pp. 58–78, 2014.

[93] Simin Ahmadi-Karvigh, Burcin Becerik-Gerber, and Lucio Soibelman, "A framework for allocating personalized appliance-level disaggregated electricity consumption to daily activities," *Energy and Buildings*, vol. 111, pp. 337–350, 2016.

[94] Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *Proc. of ARCS*, 2010, pp. 1–10.

[95] Martha E Pollack, Laura Brown, Dirk Colbry, Colleen E McCarthy, Cheryl Orosz, Bart Peintner, Sailesh Ramakrishnan, and Ioannis Tsamardinos, "Autominder: An intelligent cognitive orthotic system for people with memory impairment," *Robotics and Autonomous Systems*, vol. 44, no. 3, pp. 273–282, 2003.

[96] Walter S Lasecki, Young Chol Song, Henry Kautz, and Jeffrey P Bigham, "Real-time crowd labeling for deployable activity recognition," in *Proc. of CSCW 2013*. ACM, 2013, pp. 1203–1212.

[97] Liang Wang, Tao Gu, Xianping Tao, and Jian Lu, "A hierarchical approach to real-time activity recognition in body sensor networks," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 115–130, 2012.

[98] Narayanan C Krishnan and Diane J Cook, "Activity recognition on streaming sensor data," *Pervasive Mob Comput*, vol. 10, pp. 138–154, 2014.

[99] Nawel Yala, Belkacem Fergani, and Anthony Fleury, "Feature extraction for human activity recognition on streaming data," in *Proc. of INISTA*, 2015, pp. 1–6.

[100] Diane J Cook, Aaron S Crandall, Brian L Thomas, and Narayanan C Krishnan, "CASAS: A smart home in a box," *Computer*, vol. 46, no. 7, 2013.

[101] Chih-Chung Chang and Chih-Jen Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.

[102] John Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[103] World Health Organization et al., "World health day 2012," *World report on ageing and health*, 2012.

[104] G. Carone and D. Costello, "Can europe afford to grow old?," *Finance and Development*, vol. 43, no. 3, pp. 28–31, 2006.

[105] Ger van den Broek, Filippo Cavallo, and Christian Wehrmann, *AALIANCE Ambient Assisted Living Roadmap*, vol. 6 of *Ambient Intelligence and Smart Environments Series*, IOS press, Amsterdam, The Netherlands, 2010.

[106] R Jan Gurley, Nancy Lum, Merle Sande, Bernard Lo, and Mitchell H Katz, "Persons found in their homes helpless or dead," *New England Journal of Medicine*, vol. 334, no. 26, pp. 1710–1716, 1996.

[107] Muhammad Mubashir, Ling Shao, and Luke Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[108] Norbert Noury, Anthony Fleury, Pierre Rumeau, AK Bourke, GO Laighin, Vincent Rialle, and JE Lundy, "Fall detection-principles and methods," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. IEEE, 2007, pp. 1663–1666.

[109] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors (Switzerland)*, vol. 14, no. 7, pp. 12900–12936, 2014.

[110] C.-C. Yang and Y.-L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *Sensors*, vol. 10, pp. 7772–7788, 2010.

[111] Paola Pierleoni, Alberto Belli, Lorenzo Maurizi, Lorenzo Palma, Luca Pernini, Michele Paniccia, and Simone Valenti, "A wearable fall detector for elderly people based on AHRS and barometric sensor," *IEEE Sensors Journal*, vol. 16, no. 17, pp. 6733–6744, 2016.

[112] B. Andò, S. Baglio, C.O. Lombardo, and V. Marletta, "A multisensor data-fusion approach for ADL and fall classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, pp. 1960–1967, 2016.

[113] Liang-Hung Wang, Yi-Mao Hsiao, Xue-Qin Xie, and Shuenn-Yuh Lee, "An outdoor intelligent healthcare monitoring device for the elderly," *IEEE Trans. Consum. Electron.*, vol. 62, no. 2, pp. 128–135, 2016.

[114] L. Palmerini, F. Bagalà, A. Zanetti, J. Klenk, C. Becker, and A. Cappello, "A wavelet-based approach to fall detection," *Sensors (Switzerland)*, vol. 15, no. 5, pp. 11575–11586, 2015.

[115] Ahmet Yazar, Furkan Keskin, B. Ugur Töreyin, and A. Enis Çetin, "Fall detection using single-tree complex wavelet transform," *Pattern Recognition Letters*, vol. 34, pp. 1945–1952, 2013.

[116] Emanuele Principi, Paolo Olivetti, Stefano Squartini, Roberto Bonfigli, and Francesco Piazza, "A Floor Acoustic Sensor for Fall Classification," in *Proc. of The 138th International AES Convention*, Warsaw, Poland, May 7-10 2015.

[117] Emanuele Principi, Diego Droghini, Stefano Squartini, Paolo Olivetti, and Francesco Piazza, "Acoustic cues from the floor: a new approach for fall classification," *Expert Systems with Applications*, vol. 60, pp. 51–61, 2016.

[118] Xiaodan Zhuang, Jing Huang, Gerasimos Potamianos, and Mark Hasegawa-Johnson, "Acoustic fall detection using Gaussian mix-

ture models and GMM supervectors," in *Proc. of ICASSP*, Taipei, Taiwan, Apr. 19-24 2009, pp. 69–72.

[119] Yun Li, KC Ho, and Mihail Popescu, "A microphone array system for automatic fall detection," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 5, pp. 1291–1301, 2012.

[120] Yun Li, K C Ho, and Mihail Popescu, "Efficient source separation algorithms for acoustic fall detection using a Microsoft Kinect," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 3, pp. 745–755, 2014.

[121] M. Cheffena, "Fall detection using smartphone audio features," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 4, pp. 1073–1080, 2016.

[122] Yaniv Zigel, Dima Litvak, and Israel Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound—proof of concept on human mimicking doll falls," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 12, pp. 2858–2867, 2009.

[123] Charalampos N Doukas and Ilias Maglogiannis, "Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 2, pp. 277–89, Mar. 2011.

[124] B Toreyin, A Soyer, Ibrahim Onaran, and E Cetin, "Falling person detection using multi-sensor signal processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, pp. 7, 2008.

[125] T.-T.-H. Tran, T.-L. Le, and J. Morel, "An analysis on human fall detection using skeleton from microsoft kinect," in *Proc. of IEEE 5th International Conference on Communications and Electronics*, Danang, Vietnam, 2014, pp. 484–489.

[126] M. Markou and S. Singh, "Novelty detection: a review – part 1: statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[127] M. Markou and S. Singh, "Novelty detection: a review – part 2: neural network based approaches," *Signal processing*, vol. 83, no. 12, pp. 2499–2521, 2003.

[128] Min Zhou, Shuangquan Wang, Yiqiang Chen, Zhenyu Chen, and Zhongtang Zhao, "An activity transition based fall detection model on mobile devices," in *Human Centric Technology and Service in Smart Space*, pp. 1–8. Springer, 2012.

[129] M. Popescu and A. Mahnot, "Acoustic fall detection using one-class classifiers," in *Proc. of the Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Minneapolis, MN, USA, 2009, pp. 3505–3508.

[130] Muhammad Salman Khan, Miao Yu, Pengming Feng, Liang Wang, and Jonathon Chambers, "An unsupervised acoustic fall detection system using source separation for sound interference suppression," *Signal Processing*, vol. 110, pp. 199–210, 2015.

[131] Paolo Olivetti, "Sistema per la rilevazione e prevenzione di caduta anziani, mediante cassa di risonanza a pavimento," Italian Patent 0001416548, July 1, 2015.

[132] Bernhard Schölkopf, Robert C. Williamson, Alexander J. Smola, John Shawe-Taylor, and John C. Platt, "Support vector method for novelty detection," in *Advances in Neural Information Processing Systems*. 2000, vol. 12, pp. 582–588, MIT Press.

[133] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 4, pp. 357–366, 1980.

[134] Tomi Kinnunen and Haizhou Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[135] Majd Alwan, Prabhu Jude Rajendran, Steve Kell, David Mack, Siddharth Dalal, Matt Wolfe, and Robin Felder, "A smart and passive floor-vibration based fall detector for elderly," in *Proc. of Inf. Commun. Technol.*, 2006, vol. 1, pp. 1003–1007.

[136] F. Werner, J. Diermaier, S. Schmid, and P. Panek, "Fall detection with distributed floor-mounted accelerometers: An overview of

the development and evaluation of a fall detection system within the project eHome," in *Proc. of the 5th Int. Conf. on Pervasive Computing Technologies for Healthcare and Workshops*, Dublin, Ireland, May 23-26 2011, pp. 354–361.

[137] W.M. Campbell, D.E. Sturim, and D.A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[138] Jeff A Bilmes et al., "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *International Computer Science Institute*, vol. 4, no. 510, pp. 126, 1998.

[139] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[140] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[141] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann L Cun, "Off-road obstacle avoidance through end-to-end learning," in *Advances in neural information processing systems*, 2006, pp. 739–746.

[142] James Bergstra, Daniel Yamins, and David Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *International Conference on Machine Learning*, 2013, pp. 115–123.

[143] Andrew Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[144] Erik Marchi, Fabio Vesperini, Stefano Squartini, and Björn Schuller, "Deep recurrent neural network-based autoencoders for acoustic novelty detection," *Computational intelligence and neuroscience*, vol. 2017, 2017.

[145] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of International Conference on Learning Representations (ICLR)*, Banff, Canada, Apr. 14-16 2014.

[146] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks.," in *Aistats*, 2010, vol. 9, pp. 249–256.

[147] Diego Droghini, Daniele Ferretti, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "A combined one-class svm and template matching approach for user-aided human fall detection by means of floor acoustic features," *Computational Intelligence and Neuroscience*, vol. 2017, 2017, Article ID 1512670.

[148] Richard S Sutton and Andrew G Barto, *Introduction to reinforcement learning*, vol. 135, MIT press Cambridge, 1998.