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

---

Deep Neural Networks for Speech Detection and Speaker Localization in Reverberant Environments

Ph.D. Dissertation of:
Paolo Vecchiotti

Advisor:
Prof. Stefano Squartini

Curriculum 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

---

Deep Neural Networks for Speech Detection and Speaker Localization in Reverberant Environments

Ph.D. Dissertation of:
Paolo Vecchiotti

Advisor:
Prof. Stefano Squartini

Curriculum 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

Acknowledgments

I would like to thank my supervisor, Prof. Stefano Squartini, for giving me this opportunity and believing in me during these years. He trained me, with infinite patience, in the art of being pragmatic about research, focusing on realistic goals and doable things. A special thank to Emanuele Principi for his help and his experience, and to Fabio Vesperini, who has been a colleague and a friend at the same time.

I am very grateful to Diego Zallocco, for his important teachings, and to the colleagues from Elettromedia, for all the time spent together. I would like to thank Ning Ma and Prof. Guy Brown, it was really a pleasure to work with you at the University of Sheffield. Thanks to Ferdinando and Giovanni: I truly enjoyed your collaboration and your passion for audio and for engineering. My gratitude also goes to Alessandro, Linda and Massimo, the friends who shared with me this study experience.

I am grateful to my family, to support me and to give me all the things one can wish for. Thanks to Sarah, for always being there, to Lucia, for pushing me forward, to Arianna, Armando, Federico, Natasha, Silvio, Susanna, for all the love you give me. Too many people I would like to thank for being part of these three twisted years, but they already know they are important to me.

Ancona, Novembre 2018

Paolo Vecchiotti

Abstract

This thesis addresses the tasks of Voice Activity Detection (VAD) and Speaker LOCalization (SLOC) in reverberant environments. A data-driven approach characterizes this work, where Deep Neural Networks (DNN) are largely employed and investigated. Indeed, although VAD and SLOC have been assessed by classical algorithms for a long time, the new breakthrough of machine learning for audio processing has lead to encouraging results into the addressed tasks. Hence, this thesis proposes several reliable DNN-based strategies for VAD and SLOC, which act more robustly when tested against classical algorithms. Furthermore, DNNs are a powerful tool to develop human-inspired systems and joint VAD and SLOC frameworks, reason why they are the interest of this work.

Initially, VAD and SLOC are analysed separately, in order to properly focus on novel approaches for audio processing by means of DNNs. In particular, this work is driven by an extensive employment of Convolutional Neural Networks (CNNs). Indeed, a virtuous exploitation of data captured by multiple microphones and a temporal evolution of the signal is possible by means of CNNs convolutional kernels. A multi-room environment is chosen to assess the performance of the proposed algorithms, since it shows a high degree of similarity with a real world scenario. There issues such as reverberation, cross-talk through multiple rooms and a wide range of background noise must be dealt with. Along with this, studies focus on binaural sound localization, which is addressed by means of models inspired by the human hearing systems. In particular, the tasks of determining the azimuth and the elevation of a speaker are separately addressed. The first case study is solved by means of an end-to-end approach, which learns to localize sounds similarly to human beings. After that, elevation is estimated from the frequency domain amplitude and phase of the signals, outperforming the state-of-the-art models present in literature.

Finally, VAD and SLOC are jointly performed by means of a unique framework, whose purpose is to increase the overall performance over the two tasks. Indeed, a CNN-based model capable of virtuously exploiting localization and detection related features, achieves remarkable results in terms of VAD. In addition, a novel data augmentation technique is proposed in this study, where the acoustic scenes of two different rooms are simulated.

Sommario

In questa tesi vengono affrontate le tematiche del Voice Activity Detection (VAD) e dello Speaker LOCalization (SLOC) in ambiente riverberante. Un approccio data-driven caratterizza questo lavoro, e per questo motivo reti neurali deep vengono ampliamente sfruttate e analizzate. Sebbene diversi algoritmi classici siano stati utilizzati per VAD e SLOC per lungo tempo, le recenti scoperte nel campo del machine learning applicato all'audio hanno portato a risultati incoraggianti per quanto concerne VAD e SLOC. Di conseguenza, questa tesi propone numerose strategie vincenti per VAD e SLOC basate su reti neurali, che si dimostrano piú performanti e piú robuste quando paragonate ad algoritmi classici. In aggiunta, le reti neurali risultano un ottimo strumento per sviluppare modelli matematici ispirati dal sistema uditivo umano, o per studiare approcci capaci di fare rilevamento e localizzazione di un parlatore in modo simultaneo; per questo motivo vengono quindi sfruttate in questo lavoro.

Inizialmente le tematiche di VAD e SLOC vengono affrontate separatamente, in modo da potersi focalizzare accuratamente su nuovi approcci basati su reti neurali. In particolare, questa tesi fa affidamento su un impiego estensivo di reti neurali convoluzionali (CNN). Infatti, questa architettura neurale permette uno sfruttamento intensivo di segnali audio catturati da diversi microfoni, insieme alla possibilitá di impiegare un'evoluzione temporale del segnale. Per testare gli algoritmi proposti si sceglie un ambiente caratterizzato da piú stanze, in quanto mostra un alto grado di somiglianza con uno scenario reale. In particolare questo ambiente é soggetto a problematiche come riverbero, individui parlanti contemporaneamente e una grossa varietá di rumore di sottofondo. Insieme a questo viene affrontata la tematica della localizzazione del suono da udito binaurale, tramite modelli neurali ispirati dall'apparato uditivo umano. Nel dettaglio, ci si pone l'obiettivo di stimare separatamente l'azimuth e l'altezza di un parlatore. Nel primo caso, viene proposto un approccio end-to-end per la stima dell'azimuth, il quale si dimostra capace di imparare a localizzare il suono in maniera simile all'essere umano. Dopo di ció, l'altezza del parlatore dal suolo viene stimata per mezzo di un sistema che sfrutta l'ampiezza e la fase del segnale nel dominio della frequenza, il quale ottiene prestazioni migliori dei sistemi presenti in letteratura.

Infine viene proposto un sistema capace di eseguire VAD e SLOC allo stesso tempo, il cui obiettivo é di migliorare l'accuratezza del sistema stesso. Per

questo motivo si sviluppa un modello basato su CNN capace di sfruttare in maniera virtuosa due diverse features audio mirate al rilevamento e alla localizzazione del parlatore, rispettivamente. Insieme a questo, viene proposta una nuova tecnica di data augmentation, che permette di simulare le scene acustiche di due diverse stanze.

Contents

List of Figures

List of Tables

List of Acronyms

AMS Amplitude Modulation Spectrograms.

ANN Artificial Neural Network.

APF All-Pass Filter.

ASR Automatic Speech Recognition.

BLSTM Bidirectional Long Short Time Memory.

BRIR Binaural Room Impulse Response.

CCF Cross-Correlation Function.

CNN Convolutional Neural Network.

CSP Cross Spectrum Phase.

CSPCM Crosspower Spectrum Phase Coherence Measure.

DBN Deep Belief Network.

DCT Discrete Cosine Transform.

Del Deletion rate.

DFT Discrete Fourier Transform.

DNN Deep Neural Network.

DOA Difference Of Arrival.

DRR Direct-to-Reverberant Ratio.

DSP Digital Signal Processing.

DWT Discrete Wavelet Transformation.

EM Expectation-Maximization.

ERB Equivalent Rectangular Bandwidth.

EVM Envelope-Variance Measure.

FA False Alarm rate.

FD Fractional Derivatives.

FDC Fractional Derivative Constraint.

FIR Finite Impulse Response.

FPE Forward Prediction Errors.

GCC Generalized Cross Correlation.

GCF Global Coherence Field.

GMM Gaussian Mixture Model.

HATS Head And Torso Simulator.

HPF High-Pass Filter.

HRIR Head Related Impulse Response.

HRTF Head Related Transfer Function.

IIR Infinite Impulse Response.

ILD Interaural Level Difference.

IPD Interaural Phase Difference.

ITD Interaural Time Difference.

KEMAR Knowles Electronic Manikin for Acoustic Research.

LPE Linear Prediction Error.

LPEF Linear Prediction Error Filter.

LPF Low-Pass Filter.

LSTM Long Short Time Memory.

MAA Minimum Audible Angle.

MCT Multi-Conditional Training.

MFCCs Mel-Frequency Cepstral Coefficients.

MLP Multi-Layer Perceptron.

MLS Maximum Length Sequence.

MSE Mean Square Error.

PDF Probability Density Function.

PHAT Phase Transform.

PLP Perceptual Linear Prediction.

PSD Power Spectral Density.

PSO Particle Swarm Optimization.

RASTA RelAtive SpecTrAl.

RBM Restricted Boltzmann Machines.

RIR Room Impulse Response.

RMS Root Mean Square.

RMSE Root Mean Square Error.

RNN Recurrent Neural Network.

SAD Speech Activity Detection.

SHS Sub-Harmonic-Summation.

SI Swarm Intelligence.

SLOC Speaker LOCalization.

SNR Signal-to-Noise ratio.

SRC Stochastic Region Contraction.

SRP Steered Response Power.

STFT Short-time Discrete Fourier Transform.

SVM Support Vector Machine.

TDOA Time Difference Of Arrival.

VAD Voice Activity Detection.

WC Wavelet Coefficient.

Chapter 1

Introduction

Pronounced speech plays an extremely relevant role in communication between human beings, and it has always been a subject of study and interest. For this reason, much effort has been recently spent for machine systems capable of automatically understanding and reproducing speech, being referred to as the field of *speech processing*. Although these tasks are straightforwardly accomplished by humans, they result largely complex to be reliably transferred to machines. Nevertheless, the recent development of mathematical models inspired by the animal brain and being capable of learning, combined with the last advancements in hardware present in modern computers, consists in a powerful tool for the processing of speech data.

Within this research field, the elaboration of human speech is generally dealt with as a set of distinct sub-problems. For example, a system performing Automatic Speech Recognition (ASR) is commonly composed at least by four different algorithms, which are an initial acoustic preprocessing of the audio signal, followed by a decoder relying on an acoustic model, a pronunciation model and a language model. While dividing a complex problem into smaller ones allows to easily tackle each one of them, it requires the strong assumption of their mutual independence, which is not always correct. Indeed, this aspect will be deeply discussed in one chapter of this thesis, where it will be shown that combining multiple tasks instead of addressing them separately allows to achieve more accurate performance.

In the research community, the detection and the localization of a speaker are two relevant fields of interest, which find applications in audio surveillance, human hearing modelling, speech enhancement, human and robot interaction and so forth [1, 2, 3, 4]. These tasks, referred to as Voice Activity Detection (VAD) and Speaker LOCalization (SLOC), are the focus of this thesis work, and their deployment will target a domestic environment scenario. While the investigation of VAD and SLOC has been tackled for several years by means of classical algorithms, the recent breakthrough of machine learning has heavily influenced this research field. For this reason, Deep Neural Networks (DNNs) are largely investigated in this work. Indeed, they do not require a dedicated fine tun-

ing typically necessary for state-of-the-art classical algorithms, furthermore it has been observed across different applicative fields that robust performance can be achieved due to their better capability of generalizing even in unknown environments.

In addition, DNNs are extremely powerful tools to simulate and explore the hearing process of the human being, being the interest of this thesis as well. Last but not least, in this work the development of a joint DNN-based system simultaneously acting as VAD and SLOC is addressed, while such a task results extremely complex to pursue by means of classical algorithms.

1.1 Goals and Methodology

Goals

This thesis work proposes novel DNN-based models for VAD and SLOC. In the recent years, data-driven approaches have achieved encouraging results in the addressed tasks, nevertheless much effort still requires to be spent in order to enhance the systems proposed in literature. The believe behind this thesis is that classical algorithms for VAD and SLOC will be substituted by more reliable machine learning approaches in the next years.

One of the main contribution of this work concerns the employment of Convolutional Neural Networks (CNNs) for audio processing. In particular, few works have been proposed for performing VAD and SLOC by means of CNNs. However, this type of DNN allows to accurately exploit the correlation present in audio signals in terms of time domain, frequency domain and signals captured by multiple microphones. For this reason this thesis work heavily relies on CNNs, where remarkable results are achieved by systems capable of properly using this correlation.

Furthermore, the research presented here also aims to simulate the human hearing system in order to understand certain mechanisms ruling this complex system. As a consequence of that, human binaural sound localization in reverberant environments is addressed with human-inspired DNN models. In conclusion, the proposed systems behave extremely similar to how the human hearing system localizes sound in presence of strong reverberation.

Finally, this work focuses on the comprehension of the proposed approaches, conducting extended studies with regards to particular methodology and neural models discussed here. Indeed, the objective is to highlight advantages and disadvantages of each addressed framework, and to give the tools to easily understand the behaviour of certain important mechanisms.

Methodology

In order to pursue these goals, in this thesis, a methodology relying on a data-driven approach is adopted. The main reason driving this strategy is motivated by the encouraging results achieved by DNN-based models for audio-related tasks.

Basically, a data-driven approach allows to predict an output from unknown input data, by training the neural model with known data. Although this *black box* description of a data-driven methodology seems to trivialise the methodology itself, this process deserves particular attention in several key points. For example, input data are generally represented by means of hand-crafted features, in order to reduce their complexity and to highlight specific characteristics of the signals themselves. However, the choice of the most adequate features set for the addressed task is not trivial, and may heavily influence the performance of the model itself. Indeed, this issue will be often raised within this thesis, by comparing multiple features or investigating brand new features extracted directly from the neural model.

Furthermore, even the choice of the correct DNN architecture may result extremely complex. In details, some architectures show the capability of better exploiting a temporal evolution of the signal, while other ones may be able to focus most on the cross correlation between signals recorded by multiple audio channels. For this reason different neural architectures are explored and tested in this work.

Last but not least, the procedure of correctly designing a DNN is an art too, since well known problems such as overfitting and missing convergence of a model are extremely easy to come across to. Even this aspect is in the interest of this work.

1.2 Thesis Outline and Main Contribution

This section provides a chapter by chapter overview, summarizing the main contributions of this work. References to the publications that have been produced in the course of the work are provided at the end of the thesis.

In Chapter 2 a review of classical models and DNN-based approaches present in literature for VAD and SLOC is discussed. This review introduces the common strategies adopted for these tasks, and motivates the main choices pursued in this work in terms of employed DNNs, features and neural architectures.

After that, Chapter 3 describes the common tool employed in the various studies conducted in this thesis. In particular, details of DNNs such as architectures, non-linear activations and the model training are discussed. In the following Chapter 4, audio signals and features employed for their repre-

sentation are described. In addition, audio corpora addressed this work are reported.

The first study targeting VAD is then presented in Chapter 5. A preliminary research targeting a multi-room environment is initially addressed, where several DNNs architectures are compared and multiple features are employed. After that, the most reliable model goes through further investigation, with the purpose of jointly exploiting audio captured by multiple microphones.

Later, neural approaches for SLOC are proposed in Chapter 6. A preliminary study is firstly presented, where a simple DNN localizes the speaker in terms of coordinates in a multi-room environment. This research is then extended by considering a more complex neural architecture, and exploring the importance of a temporal context, whose purpose is to increase the amount of input audio processed by the DNN. After that, apart from the research conducted for SLOC in a multi-room environment, two works address binaural sound localization by means of CNNs. The focus now goes on the mechanisms ruling the human hearing system. In particular, the azimuth and the elevation of the speaker are separately estimated by two different models exploiting directly the raw audio or its Short-time Discrete Fourier Transform (STFT), respectively.

The development of a joint system capable of simultaneously detecting and localizing a speaker in multi-room environment is presented in Chapter 7. A first study discusses a brand new DNN model, and performs a comparison to the most performing VAD and SLOC models proposed in the previous chapters of this thesis. After that, an extensive comparison with the only framework present in literature for detecting and localizing a speaker in a multi-room environment is discussed. Novel data augmentation techniques are proposed along with this research.

Chapter 8 discusses another contribution always relying on artificial intelligence, but not concerning the tasks of VAD and SLOC. Indeed, a novel algorithm for designing Infinite Impulse Response (IIR) filters characterized by quasi-linear phase is presented, where the work targets the development of an audio crossover.

Finally, Chapter 9 concludes the thesis and provides further directions for extending this work.

Chapter 2

Literature Review

2.1 Overview

This thesis chapter goes through some of the models, the methods and the approaches proposed in the last decades for detecting and localizing a sound source. In the early years, these tasks have been assessed by means of so called classical algorithms, which typically rely on specific signal characteristics. Indeed, these approaches aim to replicate or exploit well-know mathematical laws ruling sound propagation. However, a dedicated fine tuning of these algorithms is generally required, furthermore, their inability of generalizing leads to poor performance of these models when tested against novel and unknown environments.

On the other hand, the recent investigation of new DNN-based methods, in addition to the development of new hardware and software suitable for training complex DNN models, has shown promising results in the field of audio processing.

Following that, the review discussed here, divided into the three main fields of interest of this work, firstly addresses classical algorithms proposed in literature for VAD and SLOC, and then focuses on the new DNN-based methods.

2.2 Related Works

2.2.1 Voice Activity Detection

One of the first classical VAD algorithm was standardized in 1997 in [5]. It evaluates four parameters, which are the differential power in the 0–1 kHz band, differential power over the whole band, differential zero crossing rate and spectral distortion. After that a multi-boundary decision procedure is applied in the region defined by these four parameters. The work in [6] proposes a different approach called Spectral Autocorrelation Peak Valley Ratio (SAPVR). This method performs the autocorrelation on the magnitude spectrum and then determine the ratio of the sum of the peaks in the spectral autocorrelation domain

over the value of the first valley.

Differently, the models described in [7, 8], assign a statistical model to the noise and speech data, and then determine the presence of speech by applying a threshold to the likelihood of the actual frame to these statistical models.

VAD is performed by more complex systems relying on Gaussian Mixture Model (GMM) in [9] and Support Vector Machine (SVM) in [10].

DNN-Based VAD

Two different neural architectures have been successfully proposed for VAD in reverberant environment in [11], being a Deep Belief Network (DBN) exploiting multiple domain feature fusion, and a Bidirectional Long Short Time Memory (BLSTM). Advancements are then discussed in [12]. In [13], a comparison between two VADs based on DNN and GMM demonstrates that the neural approach is able to outperform the GMM-based VAD. Similarly, a Recurrent Neural Network (RNN) achieves better detection performance in [1] when it is again compared to a GMM system. Numerous DNN-based VADs for a multi-room domestic scenario are discussed in [14], where a DBN achieves the highest accuracy compared to a Multi-Layer Perceptron (MLP) and a BLSTM. A later work [15] addresses the same acoustic scenario, where a VAD system relying on MLP is discussed. It consists in a multi-channel speech segmentation performed for each room, a time-alignment of the detected speech segments and a room assignment method applied to each speech event. Also CNNs have been exsploited for speaker detection in [16]. This kind of neural network is also employed in [2], where is used to directly process the audio spectrogram, outperforming the state-of-the-art VADs.

As result, several DNN architectures have been employed in literature for VAD. Hence, in this thesis an initial comparison of some of them is performed, in order to assess the most performing and reliable approach. Along with this, the novelty of extensively exploiting the temporal evolution of the signal will be addressed.

2.2.2 Speaker Localization

A review of classical localization algorithms is presented in [17]. In general, the main categories in which the algorithms can be grouped are: Time Difference Of Arrival (TDOA)-based locators, steered beamformer based locators and spectral estimation-based methods such as the multiple signal classification algorithm (MUSIC) [18].

The first technique consists in the estimation of the TDOA from the Generalized Cross Correlation (GCC) of the signals; after that, from TDOAs, the hyperbolic curves representing the signal direction of arrival Difference Of Ar-

rival (DOA) are determined. However, this method is subject to a severe performance degradation in noisy and reverberant conditions [19]. In [20], the authors present a method exploiting the crosspower spectrum phase and relying on the GCC computation, for sound localization in indoor environment. In this case, advancements are obtained with the help of additional de-reverberation techniques.

A more robust algorithm for low Signal-to-Noise ratio (SNR) scenarios is based on the Global Coherence Field (GCF), also known as Steered Response Power (SRP) [21]. It is a one-stage method where the cost function represents the probability of a given point of the considered space to be the signal source point [22]. This algorithm is not suitable to a real time application due to its severe computational burden related to the high number of local maxima in the SRP space [23]. For this reason, many strategies to optimize the global-maximum research have been proposed. An example is given in [24], where the authors use the Stochastic Region Contraction (SRC) to decrease the computational cost. A framework suitable for acoustic event localization with a microphone array in far-field context is proposed in [25], where a combination of GCC-PHAse Transform (GCC-PHAT) and SRP-PHAse Transform (SRP-PHAT) methods is employed. The resulting spatial likelihood function is spatially filtered and smoothed, significantly outperforming the reference algorithm, by means of an onset event detection technique.

DNN-Based SLOC

It has been observed that localization accuracy achieved by classical algorithms degrades severely in presence of strong reverberation. For this reason, new SLOC data-driven models are discussed in this thesis. In recent literature, the development of DNN-based models for sound localization has mainly addressed two different case studies. The first one concerns machine-oriented systems, where algorithms are fed with audio data recorded by microphone arrays. These studies aim mostly to industrial and domestic applications, where the installation of microphone arrays is generally the most suitable solution. Alternatively, the second case study focuses on human-oriented system, where sound localization is performed from binaural data. Here the research is more oriented to the simulation and understanding of the human hearing system.

The first work targeting sound localization for industrial applications was initially discussed in 1991 [26], but the obtained results were not suitable for real world application due to the computational limitations of that time. The approach employed two feedforward Artificial Neural Networks (ANNs) composed of one hidden layer to determine the position (width and depth) of an acoustic source in a waveguide. A later work [27] performs SLOC from data recorded by microphone arrays. There a pruning algorithm for the neural ar-

chitecture based on compressive sampling theory is discussed, aiming to speed up the convergence of a DNN with a sparse structure. CNNs have been used for localizing a speaker in terms of azimuth in [28], where the authors presents a special technique exploiting the phase of audio signals recorded by a linear array. Similarly, CNNs have been employed in [29] for the more general task of sound source localization.

On the other hand, binaural sound localization was firstly addressed in [30], where a three-layer MLP is trained according to the multiple extended Kalman algorithm, with the purpose of estimating the DOA of a sound source captured by two directional and spatially separated receivers. Lately, in [31] and similarly in [32], a MLP model following a spectral analysis step (i.e., based on GCC spectrum or pinna related transfer function) is used for a talker-following robot, which requires the estimation of the elevation and the DOA of the speaker. The search for a balance between computational complexity and the neural network design has been always an issue to be addressed in these tasks, in particular in the past years when the computing power was limited. Extensive studies targeting human inspired machine systems have been discussed in [33, 34], where a MLP predicts the speaker azimuth with the help of a simulated head movement of the listener. A different type of DNN is employed in [35], where multiple speakers localization is performed by a robot, which predicts the speaker azimuth in a indoor environment by using a CNN fed with Mel-dependent GCC-PHAT features. Although several studies have been proposed for estimating the azimuth of a sound source from binaural data, spare works target the estimation of the elevation of the sound source [36, 37]. In addition, elevation is rarely estimated by means of DNNs [31, 32]. A MLP capable of elevation estimation using the Cross-Correlation Function (CCF) is presented in [38]. In this studies the employment of spectral features along with CCF-based features allow to reduce the elevation estimation errors in reverberant environments. Further improvements with respect to [38] have been observed in [39], where Mel-Frequency Cepstral Coefficients (MFCCs) features are integrated to CCF and spectral features.

Nonetheless these extensive studies, no works target the development of a SLOC systems for multi-room environment by means of DNNs fed with data recorded by multiple microphone arrays. For these reason, models based on MLPs and CNNs are taken into account in this thesis. Furthermore, only one work [35] in literature performs binaural sound localization by means of CNNs. However this task can be accurately accomplished, both in terms of azimuth and elevation, as addressed later in this work.

2.2.3 Systems for Joint VAD and SLOC

Although promising results have been achieved with the new DNN based VAD and SLOC algorithms, few works target the development of a reliable framework performing speaker detection and localization at the same time. To solve this task, two main strategies can be followed. The first one relies on cooperative but distinct VAD and SLOC algorithms, while the second one counts on only one unique model acting simultaneously as detector and localizer. Last but not least, the latest proposed frameworks simultaneously accomplishing VAD and SLOC, rarely make use of new DNN approaches.

In terms of frameworks counting on distinct VAD and SLOC algorithms, a binaural model for speaker detection, localization and recognition is presented in [40]. This work localizes the speaker by means of a GMM classifier elaborating binaural features extracted through gammatone filters. After that a speech detection module applies a binary mask to GMM azimuth predictions. Similarly, in [41] a microphone array beamforming technique divides the room under study into a fixed number of cells, from which features are extracted and classification takes place by means of a GMM. For the audio surveillance purpose, in [42] a first detection stage is employed, where two separated GMMs classify scream and gunshot signals, respectively. Then localization is performed by means of cross-correlation based TDOA estimation. An ensemble of SLOC and VAD algorithms is studied in [3], where the focus goes on the interaction between the employed classical algorithms. Furthermore, an integration architecture based on DNN or GMM is there proposed, leading to a higher overall accuracy.

With regards to unique VAD and SLOC models, a modified version of the SRP-PHAT is proposed in [4], where the SRP-PHAT algorithm processes both speech and noise data, and a rest position is predicted when the latter occurs. An unique DNN-based capable of detecting and localizing multiple sound source simultaneously is presented in [43]. In this work, a recurrent CNN is fed with both the phase both the magnitude of the spectrogram of the audio signals captured by microphone arrays. After that, the network detects and localizes each audio event by means of two sets of outputs, where the first detects a specific event and the second estimates the related DOAs.

Hence, the deployment a unique framework capable of joint VAD and SLOC for a multi-room environment is targeted in this thesis since it has never been addressed in literature. For this purpose a particular DNN architecture will be proposed.

Chapter 3

Deep Neural Networks

3.1 Fundamentals of Neural Networks

The birth of ANNs was inspired by the biological neural networks composing the animal brain. The key element of the biological neural network is the *neurone*, which, as depicted in figure Fig. 3.1, is mainly composed by three parts: *soma*, *dendrites* and *axon*. In details, the soma is able to process inputs coming from each dendrite, and to send a electrical message along the axon. Hence, the axon will be the connected through dendrites to other neurons, and so on. Similarly, the artificial neuron is expressed by a variable function producing an output value depending on a set of inputs.

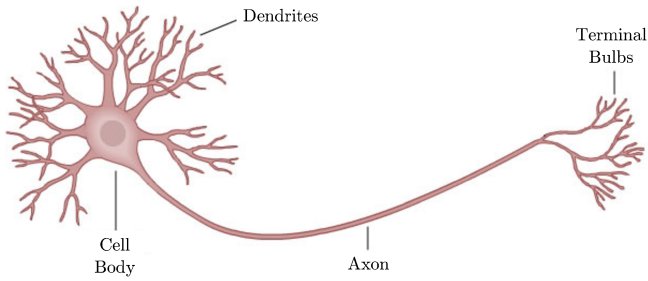


Figure 3.1: The biological neurone

An artificial neuron is defined by a pair of mathematical expressions, where all the inputs are initially summed together, and then a non-linear activation function is then applied. In details the summation is performed by:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{3.1}$$

while the non-linearity comes as:

$$y_k = f(u_k + b_k) \tag{3.2}$$

Where $x_1, x_2, \ldots, x_m$ are the input signals, $w_{k1}, w_{k2}, \ldots, w_{kw}$ are the respective synaptic weights of the neuron k, u_k is the linear combiner output due to the input signals, b_k is the bias, f is the non-linear activation function and y_k is the output signal of the neuron. A graphical representation is depicted in Fig. 3.2.

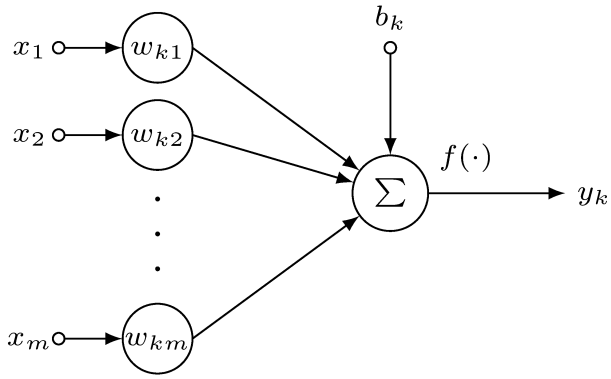


Figure 3.2: The artificial neuron

The will of replicating the learning capability of the animal brain is driving the research community to the study of ANNs. In details, the science dedicated to the development of computing system capable to learn is generally referred to *machine learning*. Specifically, the animal brain is an extremely complex organ. Indeed, it counts a huge number of deeply connected biological neurons, plus it is continuously trained during the animal life. Moreover, most of the mechanisms behind the animal brain are partially unknown. On these premises, machine learning sounds unattainable. In the research field, however, it was possible to tackle complexity of the animal brain by proposing simplified mathematical models emulating the brain itself. Hence, in the last decades ANNs have achieved encouraging results in numerous application fields [44].

In details, ANNs have been gathered in two main groups depending on the typology of connections present in the neural network itself. The two categories are *feedforward neural network* and *recurrent neural network*, which differs in the absence or presence of recursive loops connecting the units, respectively. In this thesis work feedforward neural networks are generally considered, with an exception for a specific case study. In the following paragraphs a detailed discussion of the employed ANNs is provided.

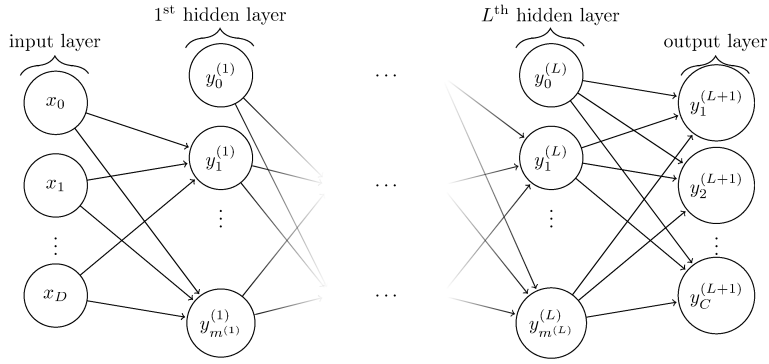


Figure 3.3: Network graph of a $(L+1)$-layer perceptron with D input units and C output units. The l^{th} hidden layer contains $m^{(l)}$ hidden units.

3.2 Neural Network Architectures

Multilayer Perceptron

The MLP is a class of ANNs introduced in 1986 [44]. It is characterized by groups of neurons having a non-linear activation, which are arranged in separated layers. Each unit of the actual layer is connected to each unit of the following and the previous layer, but no connections are possible between units belonging to the same layer. The information is allowed to travel in only one direction, from input to output. These statements respect the condition of the feedforward neural network. Due to the non-linear activation function present in the units, this topology of neural network is able to distinguish data which is not linearly separable. Three or more layers characterizes the MLP, which are the input layer, the output layer and at least one *hidden layers*. The activation function is applied to each one of these layers, except for the input one. A general $(L+1)-layer$ MLP is depicted in Fig. 3.3.

Deep Belief Networks

A DBN [45] is a probabilistic generative model obtained by stacking several simpler learning modules: Restricted Boltzmann Machines (RBM). The network topology characterizing RBM (and consequently DBN) follows the rules described in the previous paragraph for MLP, hence no connections take place between units of the same layer. The absence of inter-units connection distinguishes RBM from general Boltzmann Machines. However, even if DBN and MLP share the same architecture, they differs for the *pre-training* phase that characterizes DBN. In this phase, a greedy layer-by-layer unsupervised training algorithm called Contrastive Divergence (CD-k) is exploited, which is the peculiarity of RBM. In details, CD has pointed out to be a fast method to

approximate the gradient of the log likelihood $\log p(\mathbf{v};\theta)$, with respect to the model parameters θ. Afterwards, a supervised learning procedure fine-tunes the whole network. Compared to MLP network, DBNs can prevent overfitting and significantly speed-up the discriminative supervised learning convergence.

Restricted Boltzmann Machine

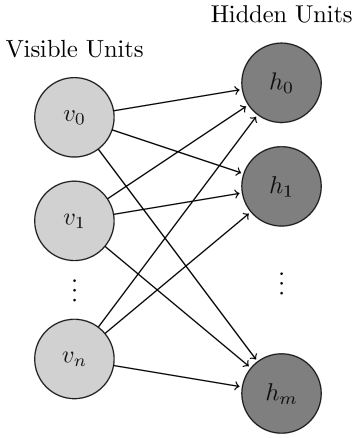


Figure 3.4: Network graph for the generic RBM.

The standard RBM is characterized by m hidden units h_j and n visible units v_i, as depicted in Fig. 3.4. The relationship between the hidden and visible units is given by the weights matrix $W = (w_{ij})$ of size $m \times n$, as well the bias weights for hidden and visible units, respectively a_j and b_j. Hence, the *energy* of the configuration (v, h) is given as:

$$E(v,h) - \sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j} h_j \tag{3.3}$$

Then, the energy function is employed for describing the probability distributions over hidden and/or visible vectors:

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)} \tag{3.4}$$

Where Z is a normalizing constant given as the sum of $e^{-E(v,h)}$ over all possible configuration, so that the probability distribution sums to 1. Similarly, the marginal probability of a visible input vector is given by the sum of all the possible hidden layer configuration:

$$P(v) = \frac{1}{Z} \sum_h e^{-E(v,h)} \tag{3.5}$$

The training of RBM has the purpose of maximizing the product of probabilities assigned to some training set V:

$$\arg\max_W \prod_{v \in V} P(v) \tag{3.6}$$

or equivalently, to maximize the expected log probability of a training sample v selected randomly from V:

$$\arg\max_W \mathbb{E}\left[\log P(v)\right] \tag{3.7}$$

Bidirectional Long Short Time Memory

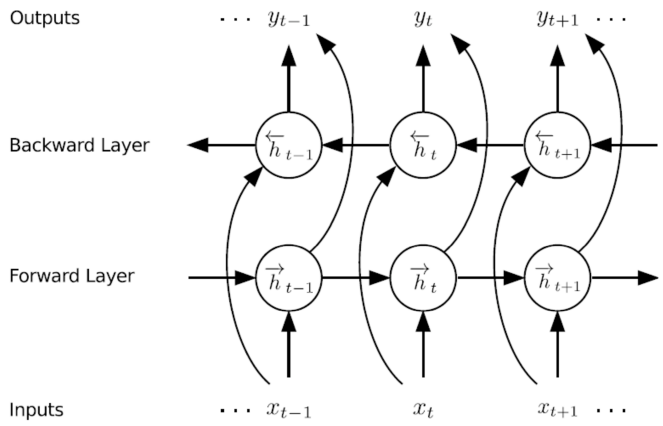


Figure 3.5: Network graph for the generic layer of a BLSTM.

The BLSTM [46] is a recurrent neural network in which the hidden units are replaced by the long short-term memory blocks. Each memory block consists of a memory cell and three gates: input gate, output gate and forget gate. The memory cell can store informations for a long time while its content is controlled by the three gates which act as the memory write, read and reset operations. In this way, the network exploits long-range temporal context. A bidirectional recurrent neural network is able to access context from both temporal directions, so here the input data are processed in both directions with two separate hidden layers.

A BLSTM layer is depicted in Fig. 3.5. It is composed by the forward and the backward layer. At the time instant t the input x_t is processed by the neurons belonging to the forward and backward layer, being indicated as $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$, respectively. The output y_t is the given as the combination of these two neurons. In addition, the forward layer sends information to the next time

instant ($\overrightarrow{h}_{t+1}$), while the backward accesses data from the previous instant ($\overleftarrow{h}_{t-1}$).

Convolutional Neural Networks

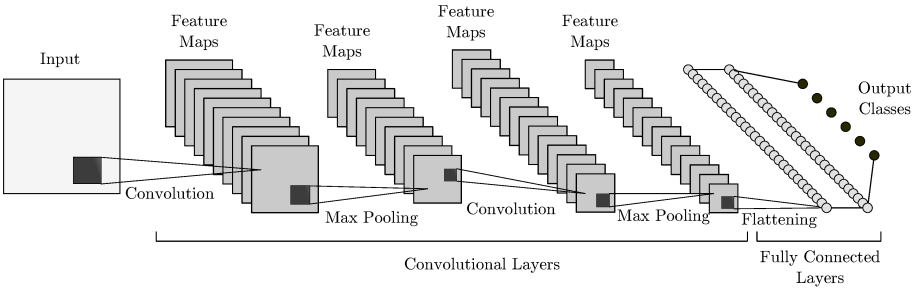


Figure 3.6: Network graph for the generic CNN. A first set of convolutional layers act as feature extractor, while fully connected layers map the extracted features to the classes to predict.

The birth of CNNs [47] finds inspiration from the ruling mechanisms behind the animal virtual cortex. As a natural consequence, the first applicative field where CNNs have been employed is image processing. After that, this kind of feed-forward neural network have found extensive applications in other fields, such as audio processing [16, 48].

In details, individual cortical neurons are sensible to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. A similar process is pursued by the convolutional kernels of the CNN, which process a 2-D input matrix by repetitively applying a special convolution operation across its sub-regions. Practically, this convolution performs a dot product between the kernel and a portion of the input data, then sums the each dot products and applies a non-linear activation function, leading to an output defined as *feature map*.

In details, denoting the m-th feature map at a given i-layer as $h_{m,i}$, the m-th *kernel* is composed by the weights $W_{m,i}$, $\mathbf{u}_j[n]$ the input data, hence:

$$h_{m,i} = f(W_{m,i} * \mathbf{u}_j[n]) \tag{3.8}$$

where $*$ represent the 2-D convolution operation, which relies on a dot product. These kernels are arranged in order to compose a convolutional layer, and multiple convolutional layers may be present in a CNN. The different feature maps obtained from each kernel of a convolutional layer are generally summed (other strategies have been proposed in literature) to compose the input data

for the following layer.

Commonly, a kernel layer is coupled with a pooling layer in order to introduce robustness against patterns shifts in the processed data. The CNN structure is completed by neuronal dense layers which map the output class or predict a set of continuous values.

An example of a CNN is reported in Fig. 3.6. In details, two convolutional layers perform feature extraction from the input matrix, where each layer is followed by max pooling. After that, fully connected layers map the extracted features to the classes to predict. The dark squares denoted with *convolution* represent the convolutional kernels moving along the input data or the feature maps; for each position a dot product between the kernel and a sub-region of the processed data is performed, leading to the next feature maps. The dark squares denoted as *max pooling* reduce the dimension of the processed feature maps by taking the maximum value over a sub-region of that input feature map. After that, the final feature maps are flattened and given as input to neuronal dense layers.

Although the pure convolution operation deals with a 2-D kernel and a 2-D input matrix, special solutions have been proposed for tensors of bigger dimensions. The first one is the 3-D convolution, where the kernel moves even on the third dimension, and the output for each point of the feature maps is given as the sum of all the dot products along the three dimensions. Nonetheless, this process is complex and requires specific software, furthermore it aims mostly to the task of video processing.

Another solution is generally preferred and is normally implemented in available tools [49]. A 3-D matrix is processed by a 3-D kernel, where the size of the third dimension is the size for the two tensors. Hence, a 2-D convolution is performed for the j-th 2-D matrix of the 3-D input and 3-D kernels, where j moves along the third dimension. After that, all the 2-D feature maps obtained are summed over the third dimension, leading to a final 2-D feature maps. The latter procedure will be deeply explored in this thesis work.

3.3 Activation Function

An activation function is generally applied at each node of a neural network. Its purpose is to process the sum of the weighted input values in a non-linear manner, so that a non-linear representation of the output is achieved. Several activation functions have been proposed in literature, however in this section only the employed ones are described.

Sigmoid

The Sigmoid function, depicted in Fig. 3.7, is defined by:

$$f(x) = \frac{1}{1 + exp(-x)} \tag{3.9}$$

This function is especially used for models which have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice. The function is differentiable.

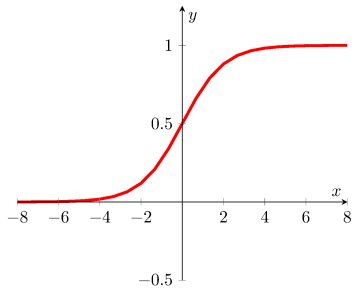


Figure 3.7: The Sigmoid activation function.

Tanh

Tanh is generally considered an evolution of the *sigmoid* activation function [50], which ranges in $[0, 1]$, and is given by:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.10}$$

Employing the sigmoid function raises training issues, since the neural network may get stuck in a condition where the neurons have a zero output, or, in other words, they do no activate. Conversely, tanh, depicted in Fig. 3.8, extends the sigmoid working range and deals with training problems. It is described by:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.11}$$

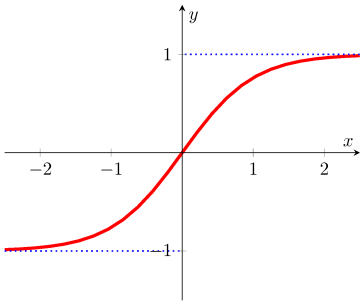


Figure 3.8: The Tanh activation function.

Hard Tanh

This activation function is similar to tanh, but it is linear in the range $[-1, 1]$. Hence, a more efficient computation than tanh is achieved. A graphical rapresentation is given in Fig. 3.9, while its mathematical behaviour is presented by the following formula:

$$f(x) = \begin{cases} 1, & x \geq 1 \\ x, & -1 < x < 1 \\ -1, & -1 > x \end{cases} \tag{3.12}$$

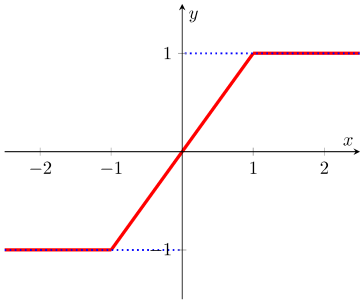


Figure 3.9: The Hard Tanh activation function.

ReLU

The Rectifier Linear Unit function was introduced in 2000 [51]. Its mathematical formula is given by:

$$f(x) = \max(0, x) \tag{3.13}$$

The ReLU activation, compared to activation such as Tanh, shows several advantages. In details, since the biological neuron is not capable of emitting a

negative value, the ReLU behaviour strictly respects the biological world. Furthermore, the training of the neural network does not come across to issues such as the vanishing or exploding gradient, plus the ReLU linear behaviour in the positive region guarantees the neural network independence from any adopted scaling of the input values. Finally, since no addition or multiplication are required, its computation is efficient. The generic ReLU is plotted in Fig. 3.10.

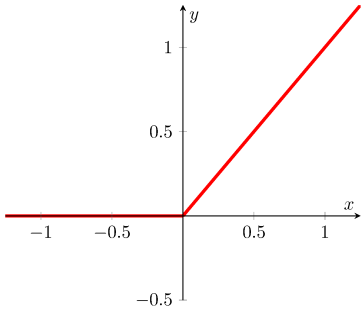


Figure 3.10: The ReLU activation function.

Softmax

Given a set of target classes, softmax calculates the probability of each target class over all possible classes [52]. Hence, it is required that each target class ranges in $[0, 1]$, plus the sum of all the probabilities of the target classes must be equal to 1. Straightforwardly, this activation function is generally used for the classification task. In details, given a K-dimensional vector $\boldsymbol{z}$, it is mapped to a K-dimensional vector $\sigma(\boldsymbol{z})$ ranging in $[0, 1]$ by means of:

$$\sigma(\boldsymbol{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad \text{for } j = 1, \ldots, K \tag{3.14}$$

3.4 Training Algorithm

DNNs map an input to an output. This procedure is performed by the artificial neurons composing the DNN, which represent a complex mathematical function. In details, each neuron is characterized by a set of parameters, and the total of the parameters of a DNN is generally referred to as *weights*. Hence, the proper value of these parameter must be set in order to accurately represent the correct mathematical function. The network training consists in the process of tuning the DNN weights. For this purpose, the analytical method of *steepest descent* is employed, being also referred to as *gradient descent*.

Gradient Descent

This method is a first-order iterative optimization algorithm that allows to find the minimum of a function, by calculating the gradient of the function at the current point and then updating the function parameters by a step proportional to the negative of the gradient. Gradient descent is expressed by:

$$W_{n+1} = W_n - \mu \bigtriangledown F(W_n) \tag{3.15}$$

where W_n is the set of the function $F()$ parameters at current point n, μ is the learning rate, and $\bigtriangledown F(W_n)$ is the gradient of the function. Reiterating this equation for each current point leads to minimum of the function $F()$. In details, when training a neural network, the function to minimize is the mismatch between the output predicted by the DNN and the ground truth, also known as the error produced by the network. This error, referred to as *loss*, may be evaluated in different manners, such as the mean squared error, the mean absolute error or the categorical crossentropy, dependently on the case under study.

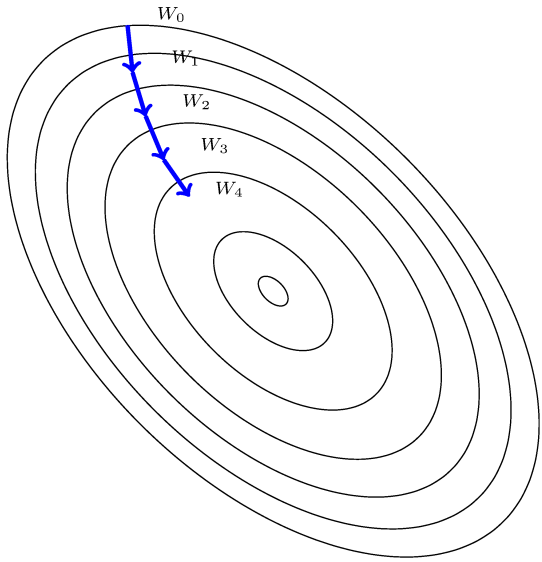


Figure 3.11: An example of the application of gradient descent to search for minimum of a function in a figurative 2-D plane. The process is iterated four times, while the small central ellipse corresponds to the minimum of the function.

Following the definition given in Equation 3.15, the gradient must be evaluated for each current point, or, in other words, for each input data feeding the

network. Nonetheless, since DNNs have a huge number of parameter, the operation of evaluating the gradient results in a heavy computational cost, leading to an extremely slow training of the network. This issue is tackled by the stochastic mini-batch gradient descent, which does not evaluate the gradient for each single input data, but for a batch of them. The batch is randomly selected, reason why the process is defined as stochastic. Furthermore, diverse strategies have been proposed and adopted for the weights update, by considering for example a momentum. Some of the most famous are Adagrad [53], Adadelta [54] and Adam [55].

Evaluating the Gradient

The key point for training an ANN is the calculation of the gradient. In particular, a simple case study of a MLP is addressed in this paragraph, which allows to easily explain the procedure necessary for evaluating $\bigtriangledown$.

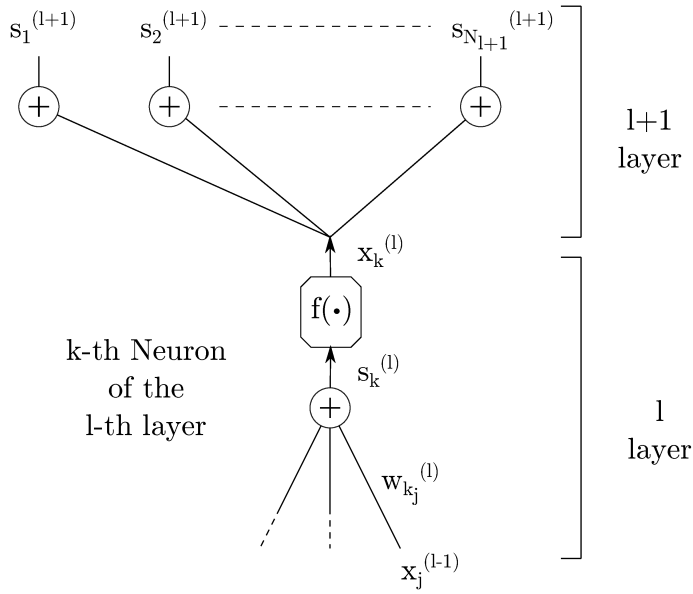


Figure 3.12: Details of the k-th Neuron of the l-th layer of a generic MLP.

Considering a network of M layers, where a neuron of a layer is connected with all the neurons of the previous layer, as shown in the Fig. 3.12, the following values are defined: M is the number of layers, indicated then with the l index, N_l is the number of neurons at the l-th layer, $s_k^{(l)}$ is the number of connections of the k-th neuron at the l-th layer, $x_k^{(l)}$ is the output of the k-th neuron at the l-th layer, $w_{kj}^{(l)}$ is the weight of the k-th neuron at the l-th layer related to the j-th input, $w_{k0}^{(l)}$ is the bias weight.

Hence, given the inputs x_k with $k = (1, \ldots, N_0)$, the outputs y_k with $k = (1, \ldots, N_M)$ are obtained as follows:

$$s_k^{(l)} = \sum_{j=0}^{N_{l-1}} w_{kj}^{(l)} x_j^{(l-1)} \tag{3.16}$$

$$x_k^{(l)} = f(s_k^{(l)}) \tag{3.17}$$

where the inputs are initially summed together, and then the non-linear function $f(\cdot)$ is applied. $x_0^{(l)}$ is the bias, and other indices range into $l = 0, \ldots, M-1$; $k = 1, \ldots, N_l$; $l = 1, \ldots, M$. That leads to the total of outputs:

$$y_k = x_k^{(M)}, \qquad k = (1, \ldots, N_M) \tag{3.18}$$

At this point, an error ε must be evaluated to adapt the weights, as stated in the previous paragraph. Hence, the network has to approximate the desired outputs, defined as $d_k[n]$ with $k = 1, \ldots, N_M$ and $n = 1, \ldots, Q-1$, where Q denotes the number of output patterns, or, in other words the samples considered in the batch. After that, choosing Mean Square Error (MSE) as cost function, the error is given as:

$$\varepsilon = \frac{1}{2Q} \sum_{n=0}^{Q-1} \sum_{k=1}^{N_M} \left(d_k[n] - y_k[n]\right)^2 \tag{3.19}$$

Since ε depends on all the outputs of the $l+1$ layer, the gradient is evaluated as:

$$\begin{aligned} \frac{\partial \varepsilon}{\partial w_{kj}^{(l)}} &= \sum_{n=1}^{N_{l+1}} \frac{\partial \varepsilon}{\partial s_n^{(l+1)}} \cdot \frac{\partial s_n^{(l+1)}}{\partial x_k^{(l)}} \cdot \frac{\partial x_k^{(l)}}{\partial s_k^{(l)}} \cdot \frac{\partial s_k^{(l)}}{\partial w_{kj}^{(l)}} = \\ &= \sum_{n=1}^{N_{l+1}} \frac{\partial \varepsilon}{\partial s_n^{(l+1)}} \cdot w_{nk}^{(l+1)} \cdot f'(s_k^{(l)}) \cdot x_j^{(l-1)} \end{aligned} \tag{3.20}$$

In addition:

$$\frac{\partial \varepsilon}{\partial s_k^{(l)}} = \sum_{n=1}^{N_{l+1}} \frac{\partial \varepsilon}{\partial s_n^{(l+1)}} \cdot w_{nk}^{(l+1)} \cdot f'(s_k^{(l)}) \tag{3.21}$$

Therefore, starting from the $l = (M-1)$ layer, since $\frac{\partial \varepsilon}{\partial s_n^{(M)}}$ are known values, it is possible to calculate the derivatives recursively:

$$\frac{\partial \varepsilon}{\partial w_{kj}^{(l)}}, \qquad l = (M-1), (M-2), \ldots, 1 \tag{3.22}$$

For this reason, the procedure is called *back propagation.*

In details, for the MLP case, defining $e_k^{(l)} = \frac{\partial \varepsilon}{\partial x_k^{(l)}}$ and $\delta_k^{(l)} = \frac{\partial \varepsilon}{\partial s_k^{(l)}}$, the error

is given by:

$$e_k^{(l)} = \begin{cases} (d_k - y_k), & \text{for } l = M; k = 1, \dots, N_l \\ \sum_{n=1}^{N_{l+1}} w_{nk}^{(l+1)} \delta_n^{(l+1)} & \text{for } l = (M-1), (M-2) \dots, 1; \ k = 1, \dots, N_l \end{cases} \tag{3.23}$$

where:

$$\delta_k^{(l)} = e_k^{(l)} f'(s_k^{(l)}) \tag{3.24}$$

Hence, the weight update is given by:

$$w'^{(l)}_{kj} = w_{kj}^{(l)} + \mu \delta_k^{(l)} \cdot x_j^{(l-1)} \qquad \text{for } k = 1, \dots, N_l; j = 0, \dots, N_{l-1} \tag{3.25}$$

where $w'^{(l)}_{kj}$ is the updated weight, $w_{kj}^{(l)}$ is the old weight, and μ is the learning rate.

Gradient Main Issues

In order to evaluate the gradient, each function present in the network must be derivative. Nonetheless, this behaviour is not always guaranteed. For example, the well-known activation function ReLU, defined in Equation 3.13, has its derivative non-defined in $x = 0$. However, to solve this issue, it is assumed that its derivative in $x = 0$ is given by:

$$f'(x) = 0 \tag{3.26}$$

so that ReLU can be normally used in ANNs.

Another well-known problem of ANNs is the *vanishing gradient*. The issue is that in some cases (e.g., extremely deep networks, sigmoid employed as activation function), the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training. The main cause of this behaviour relies in the choice of the activation function, indeed some functions tends to squash their input into a very small output range due to their strong non-linearity. As a result, there are large regions of the input space which are mapped to an extremely small range. In these regions of the input space, even a large change in the input will produce a small change in the output, hence the gradient is small. This phenomena heavily increases when multiple layers characterized by these activation function are stacked together, since each layer tends to squash its input to a small range output. A common solution is the employment of ReLU activation function, where for inputs $x > 0$, the output is not squashed but maintains its linearity.

Chapter 4

Speech Features and Datasets

4.1 Signals and Representations

In the audio processing field, data-driven models are not directly fed with captured signals, but with *features* extracted from captured data, whose purpose is to represent the signals themselves. Indeed, although the audio signal is extremely rich in terms of carried information, its direct employment implies a severe computational cost. This issue is dealt with by extracting features from the signal. In addition, features aim to enhance particular characteristics and behaviours of the signals, which are related to the task under study, while tend to hide unessential information.

In this section a brief introduction of audio signals is initially presented, after that, employed features are described.

4.1.1 Signals

Sound is generally defined as a vibration that typically propagates as an audible wave of pressure, through a transmission medium such as a gas, liquid or solid. The branch of physics dealing with the study of mechanical waves is defined as Acoustics.

Acoustics covers an extremely wide set of different field of studies, such as aeroacoustics, acoustic signal processing, architectural acoustics, bioacoustics, electroacoustics, environmental noise and soundscapes, speech, underwater acoustics, vibration and dynamics and so forth. However, the objective of this section is not to cover all these aspects, but to give a brief overview of some crucial details of Acoustics which are concerned in this thesis.

Speech

Speech is human vocal communication using language. Each language uses phonetic combinations of a limited set of perfectly articulated and individualized vowel and consonant sounds that form the sound of its words, and using those words in their semantic character as words in the lexicon of a language

according to the syntactic constraints that govern lexical words' function in a sentence. In speaking, speakers perform many different intentional speech acts, e.g., informing, declaring, asking, persuading, directing, and can use enunciation, intonation, degrees of loudness, tempo, and other non-representational or paralinguistic aspects of vocalization to convey meaning. In their speech speakers also unintentionally communicate many aspects of their social position such as sex, age, place of origin, physical states, emotions, education or experience, and the like.

Due to the undeniable importance of speech communication, machine systems capable of automatically capture speech have been heavily studied by scientists in the last decades.

Psychoacoustics

Psychoacoustics is the scientific study of sound perception, or, in other words, how humans perceive various sounds. More specifically, it is the branch of science studying the psychological and physiological responses associated with sound. It is an interdisciplinary field of many areas, including psychology, acoustics, electronic engineering, physics, biology, physiology, and computer science. For this reason it is in the interest of this thesis.

Hearing is a sensory and perceptual event. Indeed, when a person hears something, that something arrives at the ear as a mechanical sound wave travelling through the air, but within the ear it is transformed into neural action potentials. In particular, the human hearing system is composed by several different stages, which influence the perception of sounds. In details, sound is initially captured by the outer ear, which has a directivity pattern and is more sensible to certain frequencies. After that, another filtering behaviour characterizes the external auditory channel and the subsequent eardrum. Finally, the inner ear, converts sound waveforms into neural stimuli by means of hair cells located in the cochlea, which respond differently for signal frequency and phase.

The human ear can nominally hear sounds in the range 20 Hz to 20000 Hz. The upper limit tends to decrease with age; most adults are unable to hear above 16 kHz. Frequency resolution of the ear is 3.6 Hz within the octave of 1000 – 2000 Hz. In addition, even smaller pitch differences can be perceived through other means, for example, the interference of two pitches can often be heard as a repetitive variation in volume of the tone. The semitone scale used in Western musical notation is not a linear frequency scale but logarithmic. Other scales have been derived directly from experiments on human hearing perception, such as the Mel scale and Bark scale, which are approximately logarithmic in frequency at the high-frequency end, but nearly linear at the low-frequency end. The intensity range of audible sounds is enormous. Human

ear drums are sensitive to variations in the sound pressure, and can detect pressure changes from as small as a few micropascals to greater than 100 kPa.

Reverberation

Reverberation, in psychoacoustics and acoustics, is a persistence of sound after the sound is produced. Reverberation is created when a sound or signal is reflected causing a large number of reflections to build up and then decay as the sound is absorbed by the surfaces of objects in the space, such as furniture, people and air. This is most noticeable when the sound source stops but the reflections continue, decreasing in amplitude, until they reach zero amplitude. Reverberation is frequency dependent, being influenced by the room shape and its size, plus the materials present in its surfaces. The length of the decay, or reverberation time, receives special consideration in the architectural design of spaces which need to have specific reverberation times to achieve optimum performance for their intended activity. In comparison to a distinct echo, that is detectable at a minimum of 50 to 100 ms after the previous sound, reverberation is the occurrence of reflections that arrive in a sequence of less than approximately 50 ms. As time passes, the amplitude of the reflections gradually reduces to non-noticeable levels. Moreover, reverberation is not limited to indoor spaces as it exists in forests and other outdoor environments where reflection exists.

4.1.2 Features

Mel Frequency Cepstral Coefficients

MFCCs is a well-known set of features widely employed in audio applications [56], especially for the purpose of representing speech data. Indeed, the weighting operation performed by the Mel bands emulates the frequency response of the human hearing organ, which is sensible at its most to speech frequencies.

The extraction procedure requires few stages. An excerpt of the signal is transformed in the frequency domain by means of STFT. The obtained spectrum is hence mapped to the mel scale by using triangular overlapping windows. For each mel frequency the logs of the powers are considered, to whom the Discrete Cosine Transform (DCT) is applied. The resulting spectrum are the MFCCs. Furthermore, a common procedure consists in concatenating MFCCs with their first and second derivatives, in order to provide a temporal evolution of the signal. The two derivatives are referred to as Δ and $\Delta\Delta$, respectively.

LogMel

LogMel features have been recently applied in the field of acoustic modelling and music structure analysis [57, 58], leading to encouraging results. The procedure for LogMel extraction shares several aspects with the one described for MFCCs features. In details, a set of mel-band filters is applied to the spectrogram of the signal, from which the logarithm of the power spectrum for each band is considered. However, due to the absence of the DCT, no spatial compression is performed to the features, which remain correlated in the frequency domain. In this work, the employment of LogMel matches the choice of using some of the systems proposed in the next chapters (e.g., CNNs), where the objective is to exploit the intrinsic correlation of input features in order to highlight repetitive patterns present in the features.

Envelope-Variance Measure

In a closed environment, the evolution of the dynamic range of an audio signal is deeply affected by reverberation. The Envelope-Variance Measure (EVM) estimates the fading behaviour of the intensity envelope. The feature extraction firstly requires the application of a set of Mel sub-bands which filter the audio signal. For each of them the energy is computed in the log domain, considering a sliding window in the time domain. Finally, the EVM is evaluated as the variance of these sub-band Mel energies.

Pitch

As the name states, the *pitch* feature describes the main tone present in an audio excerpt, which is highly characteristic in the case of human speech. The extraction procedure relies on the Sub-Harmonic-Summation (SHS) method described [59]. In details, the audio signal is framed, and for each frame the frequency transformation in log-domain is applied. Hence, along the log-frequency axis the amplitude spectrum is shifted, where a shift correspond to a compression on a linear scale. For each shift the spectrum is scaled and then summed. This procedure creates the sub-harmonic summation spectrum, where peak picking is applied to determine pitch.

Wavelet Coefficient and Linear Prediction Error

The Wavelet Coefficient (WC) and Linear Prediction Error (LPE) feature set relies on the non-stationary components of the audio signals; it has been recently employed for the boundary detection in [60]. Initially, the framed audio signal undergoes the Discrete Wavelet Transformation (DWT), from which 6

sub-bands are obtained. Subsequently, each wavelet-domain sub-band is filtered by a set of Linear Prediction Error Filters (LPEFs) in order to extract Forward Prediction Errors (FPE). In addition, the first derivatives may be obtained from wavelet coefficients and then added to the features.

Relative Spectral transform - Perceptual Linear Prediction

The feature set relying on the RelAtive SpecTrAl (RASTA) transform and the Perceptual Linear Prediction (PLP) is generally referred to as RASTA-PLP [61]. In the research field, it has been assessed the suitability of this feature in order to represent the speech signal. The feature extraction procedure initially computes the amplitude spectrum of the audio signal in the log frequency domain. Subsequently, the obtained spectral components are separately filtered in the time domain by considering the previous frames. The spectrum is then transformed from the log frequency domain to the linear one. The PLP curve is finally multiplied in order to simulate the corresponding curve of the human hearing.

Amplitude Modulation Spectrum

The Amplitude Modulation Spectrograms (AMS) is a spectro-temporal feature set introduced in [62], with the purpose of dealing with extremely noisy and reverberant environments. The extraction procedure firstly requires the audio signal to be processed by means of STFT. From the resulting spectrogram the envelope is evaluated by squaring the complex values magnitude. Furthermore, the Bark scale decomposition is applied, which relies on a set of 9 filters targeting critical bands. Thus, for each sub-band the long-term spectral envelope is computed by a second STFT. As result, the complex AMS coefficients are obtained, in which features regarding time, acoustic and modulation frequencies are carried.

Generalized Cross Correlation with Phase Transform

This feature set is strictly related to the localization task. In details, due to sound wave propagation, a time delay occurs between a sound source and and a listening microphones. Furthermore, when a microphone pair is considered, a time delay $\Delta\tau$ generally occurs between the two relative recorded signals. Once the time delay $\Delta\tau$ is known, it is possible to calculate the DOA by the following equation:

$$\theta = \arctan\left(\frac{c \cdot \Delta\tau}{d}\right) \tag{4.1}$$

where θ is the DOA, c the speed of sound and d the distance between the two microphones. In real world application noise and reverberation yield to a difficult evaluation of the precise time delay $\Delta\tau$, hence several algorithms have been proposed for this purpose. One of them relies on the GCC function, plus the weighting procedure Phase Transform (PHAT). As result, the features named GCC-PHAT is obtained [63].

In particular, considering two different microphones i, j, the extraction process relies on the Crosspower Spectrum Phase Coherence Measure (CSPCM):

$$C_{ij}(t,\tau) = \int_{-\infty}^{+\infty} \frac{X_i(f,t)X_j^*(f,t)}{|X_i(f,t)||X_j(f,t)|} \cdot e^{j2\pi f\tau} df, \tag{4.2}$$

where $X_i(f,t)$ is the Short-Time Fourier Transform of the signal $x_i(t)$ coming from the i-th microphone. In details, the numerator consists in the cross-correlation of the two signals, while the denominator act as a weighting factor. In many applications, such as [64], it is sufficient to consider the maximum of Equation 4.2 for estimating the time delay:

$$\Delta\tau_{ij} = \arg\max_{\tau} C_{ij}(\tau,t). \tag{4.3}$$

however, in the case of reverberant and noisy environment this value is not reliable.

Hence, in this thesis work the GCC-PHAT Patterns, which consists in Equation 4.2, is employed as features, following the approach described in [65]. Regarding to the proposed approach, preliminary experiments demonstrated that TDOA estimation is not sufficiently reliable as input feature, thus GCC-PHAT Patterns are selected, which have been previously exploited in [65].

4.2 Datasets

In this thesis the proposed VAD and SLOC algorithms addresses indoor reverberant environments. These scenarios deserve much interest since they are part of our day life, and they are subjected to numerous issues which may degrade the performance of the proposed frameworks. In details, discussed models must deal with reverberation, which is strictly dependent to the observed room, cross-talk from the room under study and adjacent areas if present, and noise generated from different sources. To better simulate this context, two main scenarios are here considered. A multi-room scenario is discussed in Section 4.2.1, which is characterized by several rooms and results extremely complex. It is suitable for developing strategies relying on the parallel exploitation of signals recorded from different microphones installation, in order to enhance the accuracy and generalization capability of the model. On the other hand, single-room recordings are also discussed in Section 4.2.2, which combined with speech data described in Section 4.2.3 are suitable to simulate a simpler case study compared to the multi-room one. This strategy aims to give the tools for better understanding such mechanisms ruling binaural sound localization. Indeed, this task will be addressed with novel neural models, whose purpose is to replicate the human auditory system.

4.2.1 Multi-room Environment

Most of the research conducted in this thesis targets a multi-room scenario. Indeed, considering a multi-room and a single-room scenario, they undoubtedly share some common aspects, however the first can be considered closer to a real-world application. In particular, both scenarios are subjected to cross-talk between multiple speakers, however in the multi-room environment this event even occurs between speakers located in different rooms. Hence, a model for speaker detection and localization must be robust against utterances pronounced in rooms different from the one under observation. A similar issue raises for background noise. Indeed, even noise coming from other rooms must be dealt with by VAD and SLOC algorithms. Last but not least, room-dependent reverberations affect signals in different manners. In conclusion, considering a real world application where noise and speech signals are present inside and outside the room under study, a multi-room scenario succeeds in replicating this working condition.

DIRHA

The DIRHA dataset [66] is the multi-room environment considered in this thesis work. It is characterized by diverse scenes, rooms, microphones and noise

conditions[1]. In details, the apartment where the dataset has been recorded consists in five rooms, each equipped with several microphones, for a total amount of 40. The microphones deployment changes from room to room, as depicted in Fig. 4.1: both linear and circular arrays are present, with the linear ones placed on the walls of all rooms, while the circular ones are placed on the ceiling of the living room and of the kitchen only.

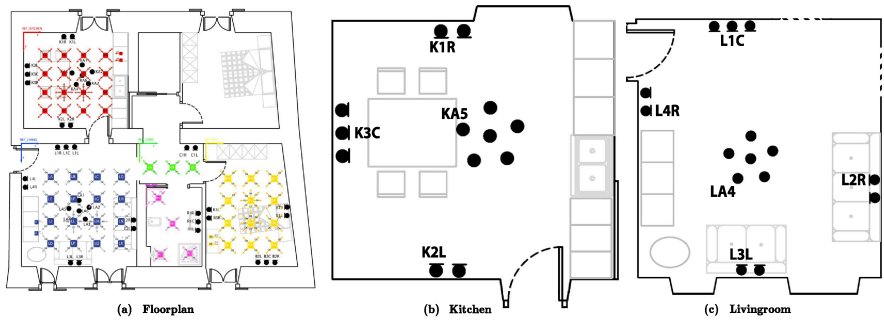


Figure 4.1: The map of the apartment used for the DIRHA project (a). Figures (b) and (c) show the rooms taken into account in this thesis work, with the disposition of their relative microphones.

The dataset is composed of two subsets, named *Simulated* and the *Real*. For each of them several *scenes* have been recorded, composed of typical situations observable in a domestic context. As reported in Table 4.1, the two subsets differ in terms of scenes and total length: in the Simulated set the scenes length is fixed to 60 seconds, while it varies in the Real set. In addition, the latter has been recorded with persons moving in the rooms and speaking towards different directions throughout the scenes, whilst the Simulated has been obtained by convolving a fixed set of measured Room Impulse Responses (RIRs) with recorded signals. Furthermore, the Simulated dataset lasts almost four times the Real one. The Simulated subset is also characterized by a lower SNR compared to the Real one, plus overlapping speech does not occur in the latter.

Two rooms of the dataset are addressed in this thesis work, i.e. the Kitchen and the Living Room, due to three main aspects. First of all, these rooms consist in the area of a home-environment where most of the events take place. In addition, being the widest rooms of the apartment, the localization task is more challenging. Finally, the available microphones are higher in number, with both wall and ceiling installations, making possible advanced multi-channel systems.

Two different versions of the Simulated DIRHA dataset are available and

[1] `http://dirha.fbk.eu/simcorpora`

	Real	Simulated
Nr. of Scenes	22	80
Total Duration	21.5 min.	80 min.
Speech Percentage	12.9% 2.8 min.	23.6% 18.9 min.
Source	human (moving)	loudspeaker (static)
Background	quiet	various
Noise Source Rate	low	high
Overlapping Events	no	yes

Table 4.1: Main differences between the DIRHA Real and Simulated subsets.

are then taken into account in this work. The *EVALITA* contains scenes of Italian spoken utterances. Diversely, the *HSCMA* dataset counts folders equally divided in Italian, Greek, German and Portuguese languages.

4.2.2 Single-room Environment

The multi-room environment described in Section 4.2.1 deserves numerous studies, which target the employment of multi-channel or multi-room data. However, due to its complexity, a high number of variables influences the performance of the proposed models. Hence, when novel studies such as end-to-end sound localization are addressed, it is reasonable to focus on a simpler case study, in order to reduce that number of variables. Along with this, the interest of this thesis goes to binaural sound localization, where the objective is to simulate and understand certain aspects of the human hearing system. Nevertheless, binaural sound localization is not possible within the DIRHA corpus, and makes necessary to address another acoustic scenario. Thus, the single room environment will be taken into account, where Binaural Room Impulse Responses (BRIRs) are employed to build the training and testing datasets. In particular, the localization task concerns the azimuth (DOA) of the speaker, or its elevation, while the distance from the speaker and the listener is fixed.

Surrey database

The Surrey database [67] contains BRIRs captured from real rooms. These responses have been recorded at the University of Surrey from four rooms of different sizes that exhibit a range of acoustical characteristics. A Cortex (MK.2) Head And Torso Simulator (HATS) and Genelec 8020A loudspeaker have been used to capture the responses. Sine sweeps signals have been played by the loudspeaker, being then deconvolved to produce the impulse responses. For the anechoic condition, a similar procedure have been used and impulse responses were obtained using a pseudo-anechoic approach whereby the responses

were captured in a large room and truncated before the first reflection. BRIRs available within these recordings do not concern the elevation of the speaker, which is assumed to be at the same height from the ground of the listener.

SADIE database

Since no recordings are available in the Surrey database concerning different speakers elevation, an alternative set of recording is considered. The SADIE database [68] contains a set of Head Related Impulse Responses (HRIRs) measured on a Knowles Electronic Manikin for Acoustic Research (KEMAR) 45BC binaural mannequin. The database contains measurements spanning across many different azimuth and elevation locations, distributed in steps of 5° in the azimuth plane and 10° in the elevation plane. All the measurements are taken with an Equator D5 loudspeaker positioned 1.5 m from the centre of the KEMAR head. Measurements take place in anechoic environment.

OpenAir

OpenAIR library [69] is an online resource which allows users to share impulse responses and related acoustical information. An open-source software plus tools and guidelines are provided within this project, with the purpose of rendering the captured RIRs and to spread common practice for recording. The database accommodates impulse response datasets captured according to different measurement techniques and relies on robust spatial audio coding formats for better distributing this information.

RIRs provided within this library will be combined with SADIE HRIRs in order to simulate a reverberant environment.

4.2.3 Speech Corpora

Here a brief description of the pure speech datasets employed in this thesis is given. Indeed, some case studies addressed in this work, such as binaural sound localization or data augmentation technique, require the development of brand new datasets. This operation is generally pursued by convolving measured or generated RIRs with speech data present in publicly available corpora.

TIMIT

The TIMIT corpus [70] has been largely used by the speech processing community [28, 33, 34, 39, 48]. It was released in 1993 and it consists in read speech designed to provide speech data for acoustic-phonetic studies and for the development and evaluation of automatic speech recognition systems. 630 speakers of eight major dialects of American English have been employed for

the dataset development, and each speaker reads ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic and word transcriptions as well as a 16-bit, 16 kHz speech waveform file for each utterance. Corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI). The speech was recorded at TI, transcribed at MIT and verified and prepared for CD-ROM production by the National Institute of Standards and Technology (NIST).

LibriSpeech

The Librispeech corpus [71] was released in 2015. It is a publicly available corpus suitable for training and evaluating speech recognition systems. The LibriSpeech corpus is derived from audiobooks that are part of the LibriVox[2] project, and contains 1000 hours of speech sampled at 16 kHz. In [71] a *Kaldi* [72] based speech recognition system trained on the LibriSpeech corpus outperforms the same system trained on the Wall Street Journal (WSJ) test sets.

This corpus is mainly created for the speech recognition task, and relies on two alignment stages with the purpose of accurately align speech with text, a data segmentation stage dealing with long silences, and a final selection and partition stage dividing the corpus into different subsets.

[2] `https://librivox.org`

Chapter 5

Voice Activity Detection

This chapter proposes several DNN-based models for VAD. Two main studies are conducted. The first one, discussed in Section 5.1, compares several neural architectures plus diverse audio features in order to highlight pros and cons of each one of them. After that, in Section 5.2, the most reliable model previously observed is taken into account for further advancements. This chapter gives the theoretical background and assesses the winning strategy necessary to the research then conducted in Chapter 7.

5.1 Comparison of several neural architectures

5.1.1 Preliminaries and Problem Statement

In recent times, promising VAD approaches take advantage of deep neural networks. In [11] two different architectures have been successfully used for this task, in particular a DBN exploiting multiple domain feature fusion, and a BLSTM recurrent neural network. Advancements are then discussed in [12]. Furthermore, in [73] an Long Short Time Memory (LSTM)-VAD using RASTA-PLP features outperforms three different VAD algorithms applied to speech recognition of Hollywood-movies audio. Last but not least, CNNs have been recently compared to other DNNs for VAD task [16].

These works drive the research of this section. Indeed, the intent here is to analyse the application of several DNNs for VAD to show advantages and disadvantages of each DNN. For this purpose, two different datasets are taken into account, in order validate the achieved results.

5.1.2 Proposed Method

The block diagram of the proposed DNN-mVAD is shown in Fig. 5.1. The first stage of the algorithm consists in the features extraction from the input audio signals. These features then feed the classifier, which is based on a DNN. The investigated networks have an input layer with the same dimension

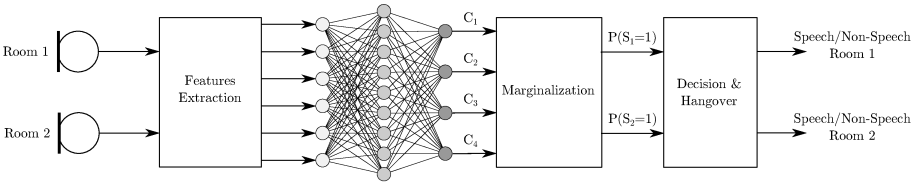


Figure 5.1: Block diagram of the proposed Deep Neural Network Multi-Room VAD in a 2 rooms application.

of the feature-set, followed by one or more hidden layers. In the case of n rooms, the output layer is a top discriminative layer with $K = 2^n$ units and softmax activation function, in order to perform a multi-class classification task. Hence, the networks have 2^n output classes, one for each condition of speech/non-speech in every considered rooms. In particular, the outputs of the softmax layer represent the joint probability of the presence or the not-presence of speech in a frame of all the selected rooms. For example, considering two rooms ($n = 2$) and denoting with $S_i = 1$ the presence of speech in the room i and with $S_i = 0$ its absence, the outputs of the four softmax neurons are:

$$C_1 = P(S_1 = 0, S_2 = 0), \tag{5.1}$$
$$C_2 = P(S_1 = 0, S_2 = 1), \tag{5.2}$$
$$C_3 = P(S_1 = 1, S_2 = 0), \tag{5.3}$$
$$C_4 = P(S_1 = 1, S_2 = 1). \tag{5.4}$$

For simplicity of notation, the frame index has been omitted.

The joint probabilities are then marginalized in order to obtain the separate speech probabilities of each room. As last stage, a thresholding block plus an hangover scheme is applied in order to handle isolated speech detections and to reduce the early non-speech classification.

Feature Extraction

The feature extraction stage operates on signals sampled at 16 kHz and frame rate equal to 100 Hz (10 ms). Six features are employed in this research, being previously described in Section 4.1.2. In particular, the feature sets are EVM, Pitch, WC-LPE, MFCCs, RASTA-PLP, AMS. MFCCs are extracted along with their first derivative. Further details are reported in Table 5.1. Finally, features are concatenated as a unique vector.

Index	Name	Feature size	Acronym	Frame Size (ms)
1	EVM	1	Ev2	25
2	Pitch *	1	Pi	50
3	WC-LPE	24	Wc	25
4	MFCC *	26	Mf2	25
5	RASTA-PLP *	54	Ra2	25
6	AMS	135	Am	25

Table 5.1: Indexed list of features, their dimensionality and the acronym used during the experiments. The * indicates that the features are extracted using the openSMILE toolkit [74].

Neural Networks

A total of 4 different neural networks is investigated in this research. They are a DBN, a MLP, a BLSTM and a CNN, which have been previously introduced in Section 3.2. In particular, for the DBN, the MLP and the BLSTM, the following network topologies have been explored, being composed by 1 or 2 hidden layers with respectively $4, 8, 10, 15, 20, 25, 40$ units per layer. With regards to CNN, a temporal context is employed for better representing the input signal. This procedure is implemented in order to fairly compare with the BLSTM-based model, which also makes use of the temporal evolution of the signal. In this case study, the time context is created by concatenating the feature vectors of a certain amount of consecutive frames. This yields a 2-D matrix of feature values related to a single room. Then, the final input matrix is obtained by stacking the single room matrices. Furthermore, several CNN parameters have been investigated. In details, convolutional kernel size using of shape 3×3, 4×5, 4×9, 6×6. and pooling equal to 1×1, 2×2, 3×3 have been analysed. After that, number of kernels is varied in the range 8, 16, 24, 32, 40, 64. Architectures with 1 or 2 layers are explored. The latter is investigated by making use of combinations of the number of kernels in the range listed above. Finally, the neuronal dense layer is tested with 50, 100, 200, 500 or 1000 neurons. Deep architectures exploiting two or three neuron layers were explored too, but no improvements were observed.

Marginalization

In the third stage of the algorithm the joint probabilities given by the network are marginalized. In the case of two rooms:

$$P(S_1 = 1) = P(S_1 = 1, S_2 = 0) + P(S_1 = 1, S_2 = 1), \tag{5.5}$$

$$P(S_2 = 1) = P(S_1 = 0, S_2 = 1) + P(S_1 = 1, S_2 = 1). \tag{5.6}$$

The output class with all non-speech probabilities $P(S_i = 0), \forall S_i$ is discarded, while all the other conditional probabilities are summed in order to obtain the probability of speech condition for a specified room $P(S_i)$.

Decision and Hangover

$P(S_1 = 1)$ and $P(S_2 = 1)$ are compared to a threshold in order to obtain a binary signal. In order to reduce false or failed speech recognitions, a simple smoothing algorithm has been employed in this work. It is called *hangover*, and it relies on a counter. In particular, if two speech frames are consecutive, the counter is set to a predefined value. On the contrary, for each non-speech frame, the counter is decreased by 1. The actual frame is classified as speech only if the counter is positive. The value of the counter is set equal to 8.

5.1.3 Experimental Setup

Experiments are conducted by means of the k-fold cross-validation technique, thus for each fold three subsets are obtained, being the *training*, the *validation* and the *test* set. The first two are employed for the network training, while the last one for its testing. In details, k is set to 10 for the Simulated subset, leading to a 64-8-8 scenes split, while $k = 7$ come for the Real subset, with a 16-3-3 split. A common optimization strategy is employed for the 4 DNNs, which relies on a first features selection, a network size selection, a second features selection, a microphone selection and a final features selection. The performance has been evaluated using the False Alarm rate (FA), the Deletion rate (Del) and the overall Speech Activity Detection (SAD) defined as follows:

$$\text{Del} = \frac{N_{del}}{N_{sp}}, \quad \text{FA} = \frac{N_{fa}}{N_{nsp}}, \quad \text{SAD} = \frac{N_{fa} + \beta N_{del}}{N_{nsp} + \beta N_{sp}}, \tag{5.7}$$

where N_{del}, N_{fa}, N_{sp} and N_{nsp} are the total number of deletions, false alarms, speech and non-speech frames, respectively. The term $\beta = N_{nsp}/N_{sp}$ acts as regulator term for the class unbalancing. In Table 5.2 training parameters for the four networks are reported. Different GPU-based toolkits have been employed for the experiments: *CURRENNT* [75] for BLSTM-mVAD, a custom version of GPUMLib [76] for DBN-m/MLP-mVAD and *Keras* (*Theano-based*) for CNN-mVAD [49].

DIRHA Dataset

The multi-room environment [66] previously described in Section 4.2.1 has taken into account for testing the proposed method. Experiments takes place in the kitchen and in the living room of the DIRHA dataset. In this case study, both Simulated both Real subset are considered.

DNN	Training algorithm	Weight initialization	Epochs	Momentum & Learning rate
MLP	BP	Gaussian distr. $\mu = 0, \sigma = 0.1$	ES	$m = 0.9$ $lr = 0.01$
DBN pre-training	CD-1	Gaussian distr. $\mu = 0, \sigma = 0.1$	200	$m = 0.3$ $lr = 0.01$
DBN fine-tuning	BP	DBN pre-training	ES	$m = 0.7$ $lr = 0.1$
BLSTM	BPTT	Gaussian distr. $\mu = 0, \sigma = 0.1$	ES	$m = 0.9$ $lr = 10^{-5}$
CNN	BP	Random	ES	$lr = 2.5 \cdot 10^{-3}$

Table 5.2: Comparison of training algorithm parameters. BP stands for "back-propagation", BPTT indicates "backpropagation through time" and ES "early stopping".

5.1.4 Main Results

Simulated Subset

In this section, the results obtained for the Simulated dataset are discussed. The first features selection makes use of K2L and L1C microphones, for kitchen and living room, respectively. The neural network layout is fixed, in particular two hidden layers of 10 units each are used for DBN/MLP/BLSTM-mVAD. In this stage, the temporal context exploited by CNN is investigated for the values 9, 13, 15, 17, 21 frames, being finally set to 13 frames. The CNN here employed counts 16 convolutional kernel and 100 hidden nodes. The first part of Table 5.3 shows the best performing feature set in term of SAD for the different mVADs. After that, the neural network architectures have been investigated, where more than 50 layouts have been tested for each neural classifier. In the case of CNN, the best network counts two convolutional layers, the first having 4×5 convolutional kernels with 3×3 pooling, and the second with 3×3 convolutional kernels and no pooling. The most performing networks are reported in the second part of Table 5.3.

After that, employed features goes again under study, resulting in new configuration compared to what observed in the first stage. Hence, the new neural architectures are capable of better exploiting a larger amount of features. Microphones are then varied, ending in different microphone pairs employed by each network. Indeed, in Fig. 5.2, the box-plot with mean, standard deviation, maximum and minimum values of SAD is reported for all the microphones pairs. Please note that the DBN-mVAD is highly sensible to microphone positioning, although reaching the absolute lowest SAD. On the contrary, the best result in terms of mean and standard deviation is observed for the CNN-mVAD. Furthermore, the convergence of the DBN-mVAD is not always guaranteed, being the lowest value of achievable SAD equal to 50%. Finally, the last feature

selection stage mostly confirms the features sets previously achieved. As result, DBN shows to be the most performing network, with a final score of 5.8% SAD.

1st Feat. Sel.	Feature-set	Del (%)	Fa (%)	SAD (%)
DBN	Mf2Ra2WcEv2	5.8%	9.1	7.4
BLSTM	Mf2WcAm	9.1	19.4	14.3
MLP	Mf2Ra2AmEv2	6.5	6.6	**6.5**
CNN	PiMf2Ev2	10.7	12.8	11.8
Net. Size Sel.	**Layout**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	20,20	7.1	6.5	6.8
BLSTM	40,40	5.8	19.5	12.6
MLP	25,4	5.9	6.2	**6.0**
CNN	16,24 + HN 100	7.3	11.1	9.2
2nd Feat. Sel.	**Feature set**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	PiMf2WcAmEv2	5.1	6.5	**5.8**
BLSTM	Mf2WcAmEv2	8.0	8.0	8.8
MLP	Mf2Ra2AmEv2	5.9	6.2	6.0
CNN	PiMf2Ra2WcAmEv	7.2	9.1	8.2
Mic. Sel.	**Microphones**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	K2L, L1C	5.6	6.5	**5.8**
BLSTM	KA5, LA4	6.4	7.7	7.0
MLP	K2L, L1C	5.9	6.2	6.0
CNN	K2L, LA4	5.9	7.2	6.5
3rd Feat. Sel.	**Feature set**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	PiMf2WcAmEv2	5.1	6.5	**5.8**
BLSTM	PiMf2WcAm	5.3	8.3	6.8
MLP	Mf2Ra2AmEv2	5.9	6.2	6.0
CNN	PiMf2Ra2WcAmEv2	5.9	7.2	6.5

Table 5.3: Comparison between different DNN-mVAD percentage of deletion rate (Del), false alarm rate (FA) and overall speech activity detection (SAD) for Simulated dataset. Marked in bold is the lowest SAD for each optimization step.

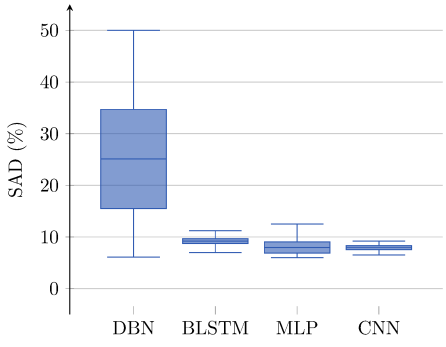


Figure 5.2: Box-plot of the resulting SADs for the microphone selection experiment in the case of the Simulated dataset.

Real Subset

Here the results for the Real dataset are reported. The initial feature selection takes place following the same neural network configurations considered for the Simulated subset. Selected microphones are K2L and L1C. Results are reported in Table 5.4. Please note that BLSTM and especially MLP reach poor performance within this stage, meaning that the two networks do not properly converge. The temporal context employed by the CNN is investigated in this stage consistently with the values used for the Simulated subset. As result, this value is set to 15 frames. The network size selection stage does not improve significantly the mVAD. Although several network architectures have been tested, the layout chosen for the first feature selection is still the best performing among the ones tested. Regarding the CNN, the best performing network has only one layer with 4×5 convolutional kernels plus 2×2 pooling. The second feature selection has been performed only on the MLP-mVAD, since no improvements are observed with the other models. A proper convergence of the network is here achieved.

1st Feat. Sel.	Feature-set	Del (%)	Fa (%)	SAD (%)
DBN	PiMf2Ra2WcAmEv2	2.8	2.9	**2.8**
BLSTM	Ra2Am	6.7	32.3	19.5
MLP	Mf2Ra2	38.8	48.3	43.6
CNN	AmEv2	2.9	6.7	4.8
Net. Size Sel.	**Layout**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	10,10	2.8	2.9	**2.8**
BLSTM	10,10	6.7	32.3	19.5
MLP	4	33.1	53.3	43.2
CNN	16 + 100 HN	2.9	6.7	4.8
2nd Feat. Sel.	**Feature set**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	PiMf2Ra2WcAmEv2	2.8	2.9	**2.8**
BLSTM	Ra2Am	6.7	32.3	19.5
MLP	PiMf2WcAmEv2	3.3	3.6	3.5
CNN	AmEv2	2.9	6.7	4.8
Mic. Sel.	**Microphones**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	K2L, L3L	2.9	2.3	**2.6**
BLSTM	KA5, LA4	8.2	25.8	17.0
MLP	K2L, L1C	3.3	3.6	3.5
CNN	K1R, LA4	2.9	5.0	4.0
3rd Feat. Sel.	**Feature set**	**Del (%)**	**Fa (%)**	**SAD (%)**
DBN	PiMf2Ra2WcAmEv2	2.9	2.3	**2.6**
BLSTM	Ra2Am	8.2	25.8	17.0
MLP	PiMf2WcAmEv2	3.3	3.6	3.5
CNN	AmEv2	2.9	5.0	4.0

Table 5.4: Comparison between different DNN-mVAD percentage of deletion rate (Del), false alarm rate (FA) and overall speech activity detection (SAD) for Real dataset. Marked in bold is the lowest SAD for each optimization step.

After that, the microphone selection stage takes place, where the best overall

performance for the Real dataset is reached. Here it is possible to observe the robustness of the CNN against the position of the microphones, shown in Fig. 5.3. Indeed, although DBN and MLP reach a lower SAD, their behaviour is not reliable in terms of mean and standard deviation. The last features selection stage confirms the results previously achieved.

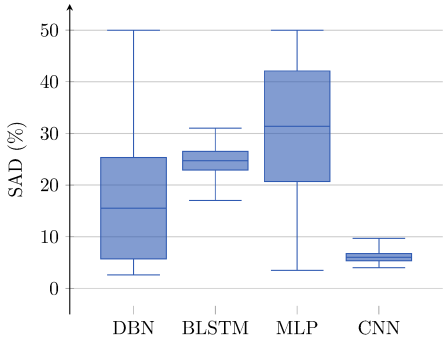


Figure 5.3: Box-plot of the resulting SADs for the microphone selection experiment in the case of the Real dataset.

Conclusions

In general, the superiority of the DBN-mVAD is due to the ability of the pre-training phase to speed up the convergence during the training of the network. The performance of the MLP-mVAD is always below the DBN-mVAD, as proof of the benefits of non-random initialization of the weights of these network architectures. Interesting results are observed in the microphone selection stage. The CNN and BLSTM architectures do not reach results comparable to the DBN and MLP, but they have a more reliable response to the sensor placement. The independence from a particular microphone position is a remarkable aspect in real application scenarios. In particular, the CNN-mVAD guarantees the lowest mean and standard deviation for both the datasets. In addition, there is evidence that the knowledge of the temporal context, as in the case of memory-enhanced units of the BLSTM or the CNN, means robustness for the system, especially with respect to the spatial origin of the input signals and to the used features.

As result, CNNs are selected for further investigation in terms of speaker detection. Indeed, although this neural architecture does not achieve the best result in absolute sense, several advancements are still possible, targeting employed features, features organization, training data and training procedure.

5.2 Further Advancements in CNN-based VAD

In this section further advancements of the models discussed in Section 5.1 are proposed. The focus now goes on the employment of data captured from multiple microphones. Extracted features are then arranged in a 3-D tensor, so that their cross-correlation can be properly processed by CNNs.

5.2.1 Preliminaries and Problem Statement

From the research conducted in Section 5.1, remarkable results have been observed when the mVAD model relies on CNNs. Indeed, although this model does not reach the lowest SAD, its stability in terms of microphone placement drives this new research, which targets the simultaneous employment of data recorded by multiple microphones. In details, the objective of this work is the employment of 3-D kernels of CNNs. This procedure is the same one adopted for image processing with CNN, where the three separated image channels (e.g., RGB) are commonly processed by the network. It is accurately discussed in Section 3.2. This strategy matches the multi-room environment, where due to speech signal degradation caused by background noise and reverberation, a multiple sensor (i.e., microphone arrays) deployment is generally necessary. Hence, CNNs make possible the virtuous employment of data coming from different microphone installation through its 3-D kernels.

Along with this, only one features set feeds the DNN models. Indeed, the previous study makes use of a unique matrix obtained by stacking different features. Nevertheless, this procedure leads to a hybrid matrix of input features, which cannot be properly exploited by a CNN, since convolutional kernels are forced to process adjacent features where related patterns may be extremely variable. Furthermore, the employment of multiple features requires several time-consuming feature selection stages. For this reason, a new a single feature set is here employed. A comparative model based on MLP is considered for evaluating the model performance. In this case study, only the Simulated dataset is taken into account for testing the models. Indeed, consistent results have been observed in the previous research between the Simulated and Real subset, however the latter is characterized by a smaller amount of data, which can be insufficient for a proper training of the CNN-based model, which has a considerable number of weights to train, compared to the MLP-based one.

5.2.2 Proposed Method

In order to be consistent with the previous research, a model similar to the one described in Section 5.1.2 is here taken into account. It consists in an initial features extraction stage, a DNN classifier, a marginalization procedure

dealing with the multi-class problem, and a final post processing stage. This model is referred to as *one network per two rooms, N microphones per room* (2R *N*MxR). Several differences are introduced here compared to the previous work. In details, only two DNNs are here taken into account, being the MLP and the CNN. In addition, *LogMel* features feed the network, instead of a set of 6 different features. They have been previously discussed in Section 4.1. Their extraction relies on a total of 40 mel-band, while frame size is equal to 25 ms and the frame step is equal to 10 ms, on a 16 KHz sampled audio. These features have been selected with the purpose of exploiting the convolution process performed by convolutional kernels. Furthermore, it is desirable to be independent from an extensive features selection stage.

The organization of the input features has two main novelties. The first one is the use of *strides* combined with frame context. In particular, this parameter pilots the frame selection for building the input matrix. When strides is equal to one, adjacent frames are considered, while a jump in the selection process is introduced for strides bigger than one. Hence, a 2-D matrix is obtained for each microphone, by combining context and features, after that, microphone-dependent matrices are stacked in a parallel manner, leading to a 3-D matrix.

In addition, a VAD model for single room is considered in this research. In details, this model is referred to as *one network per room, one microphone per room* (1R 1MxR). It relies on features extracted by only microphone present in the room under study, plus it acts as a VAD only for the considered room. As result, two outputs are present in the model, being the speech and non-speech probability; no marginalization is present here. The purpose of this model is to directly test the reliability and effectiveness of the employment of data recorded in different environments. Finally, marginalization is applied coherently to the previous model, plus smoothing of the network predictions is performed with *hangover* technique, being described in Section 5.1.2, and of which counter is set to 8.

5.2.3 Experimental Setup

The analysis of the proposed method relies on a two-stage strategy: a network size selection and a microphone combination selection. Each one of the proposed models goes through these two optimizations. Metrics and cross-validation are consistent with the one employed in the previous section and described in Section 5.1.3. MLP networks are trained with a fixed momentum of 0.9, learning rate equal to 0.01 and a Gaussian distribution with zero mean and standard deviation of 0.1 for weight initialization. For the CNN networks a fixed learning rate of $2.5 \cdot 10^{-3}$ and a random weight initialization is used. Simulations take place against the Simulated subset of the DIRHA dataset,

previously addressed in Section 4.2.1.

5.2.4 Main Results

Initially the network size selection is performed. MLP-mVAD network topologies are explored by means of 1 or 2 hidden layers with respectively 4, 8, 10, 15, 20, 25, 40 units per layer and all their combinations. For CNN-mVAD, due their greater number of hyperparameters and increased training time, a comprehensive grid search is not reasonable, hence a progressive strategy based on intermediate results is adopted. After that, audio channels are selected. An initial subset of 9 microphones: 4 in the kitchen (i.e., K2L, K1R, K3C, KA5) and 5 in the living room (i.e., L1C, L2R, L3L, L4R, LA4) is considered. For the 2R NMxR model, when multiple microphones are employed, only combinations obtained with the best performing ones are analysed. A maximum of $N = 3$ is selected for the 2R NMxR model.

CNN						
		1R 1MxR		2R 1MxR	2R 2MxR	2R 3MxR
		Kitchen	Living Room			
Input	Strides	8	10	8	8	8
Params	Context	17	23	25	23	23
First	N Kern	16	16	32	128	256
Convolutional	Size	6×6	6×6	4×4	4×4	4×4
Layer	Pooling	2×2	2×2	2×2	-	-
Second	N Kern	24	16	64	64	32
Convolutional	Size	4×4	4×4	3×3	3×3	3×3
Layer	Pooling	-	-	-	-	-
Third	N Kern	24	16	128	32	32
Convolutional	Size	3×3	3×3	3×3	3×3	3×3
Layer	Pooling	-	-	-	-	-
Fully Connected	Num. of	100	100	500	250	500
Layers	Units	20	20	100	100	100
SAD Min (%)		9.0	10.7	9.3	8.1	7.0
MLP						
Fully Connected	Num. of	10	15	10	8	8
Layers	Units	-	-	-	-	-
SAD Min (%)		11.8	13.3	11.7	8.8	7.4

Table 5.5: Network topology parameter for CNN- and MLP-mVAD.

One network per room, one microphone per room (1R 1MxR). In the network size selection, the best MLP-VAD resulted to have one layer with 10 units and 8 units respectively for the kitchen and the living room. In the second stage, the best performing microphone for the kitchen was the KA5, while for the living room the LA4: both of them are placed at the center of the room ceiling and the averaged SAD was equal to 12.5%. The two networks exploited for the CNN-VAD are reported in Table 5.5. As for MLP-VAD, best microphones are KA5 and LA4, with an average 9.9% SAD.

One network per two rooms, one microphone per room (2R 1MxR). First of all one channel per room is considered: the best MLP-mVAD has one layer with 15 units and the audio captured by the pair KA5, LA4 (confirming the result of the previous step), leading to a SAD equal to 11.7%. For the CNN-mVAD, SAD equal to 9.3% is again obtained with the pair of microphones KA5 and LA4. CNN topology is reported in Table 5.5.

One network per two rooms, two microphones per room (2R 2MxR). Compared to the previous step, the best configuration for MLP-mVAD has only one hidden layer with 8 neurons. In the microphone selection, on the basis of the above analysis, the 12 combinations of double pairs of channels is explored, achieving with the couple KA4, K1R (from the kitchen) and LA4, L2R (from the living room) an absolute improvement of −2.9% of SAD in respect to the case with one microphone per room. Settings of the CNN-mVAD are shown in Table 5.5. Again, best microphones are the same of the MLP-VAD: KA4, K1R, LA4, L2R. The resulting SAD is 8.1%.

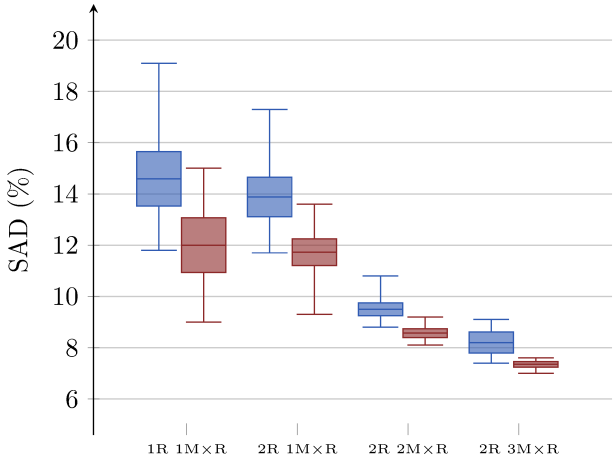


Figure 5.4: Box-plot of the resulting SADs for the microphone selection experiments in the different steps, where blue is the MLP-mVAD and red the CNN-mVAD. Evident is the improvement given by increasing the microphone number, and, for the CNN-mVAD, the related statistical robustness.

One network per two rooms, three microphones per room (2R 3MxR). For the MLP-mVAD the network topology remains the same as in the case with two microphones per room and the best result (SAD = 7.4%) is obtained with the combination K1R, K2L, KA5, L1C, L2R, LA4. The CNN-mVAD achieves 7.0% SAD with topology shown in Table 5.5, selected microphones are: K1R, K3C, KA5, L2R, L4R, LA4.

Finally, the mean and standard deviation of the models tested during the

microphone selection is reported in Fig. 5.4. CNN-based models show a high robustness compared to MLP in terms of microphone placement, consistently with the previous research.

Conclusions

The employment of input features extended with a temporal context and multiple microphones, along with CNNs ad hoc neural architectures, allows to reach better performance compared to the baseline MLP. Furthermore, a remarkable aspect of the CNN mVAD is the robustness to the microphone choice, with lower mean and standard deviation. The independence from the audio source positioning is an interesting applicative result, being consistent to what observed in the previous research. In addition, the models exploiting data recorded in different rooms perform generally better compared to the ones relying on single-room data. As result, the study will move upon the development of CNNs relying on multiple microphones.

Chapter 6

Speaker Localization

This chapter discusses several DNN-based approaches for SLOC in reverberant environments. Two main case studies in terms of sound localization are addressed. The first targets a multi-room environment, where machine systems exploiting signals captured from multiple microphone arrays are developed. In particular, a preliminary model is discussed in Section 6.1, while its advancements are investigated in Section 6.2. After that, the second study focuses on binaural sound localization, performed in terms of azimuth in Section 6.3 and in terms of elevation in Section 6.4. These last two researches proposes human inspired localization frameworks, whose purpose is to simulate the human hearing system.

6.1 Multi-room Environment - Preliminary Study

The research discussed in this section addresses the task of localizing a speaker in a multi-room reverberant environment by means of DNNs. Due to the novelty of the study, targeting localization in terms of coordinates in an indoor environment, the proposed method is directly compared with on of the state-of-the-art classical algorithm recently employed in the same environment under study.

6.1.1 Preliminaries and Problem Statement

Several speaker localization techniques have been proposed in literature, of which a main review is given in [77]. One of this methods [20], based on TDOA estimation, has been tested against the DIRHA dataset, which is largely employed in this thesis. That work relies on the calculation of the time delay present between signals captured by multiple microphones, from which hyperbolic curves are determined in a 2-D or 3-D space. In particular, it is based on Cross Spectrum Phase (CSP) computation, to which advancements are obtained with the help of additional filtering techniques.

On the other hand, few works address SLOC by means of DNNs. One model is discussed in [31], where a robot for a talker-following task is developed. In particular, that algorithm relies on a VAD stage, a signal pre-processing, a feature extraction step and finally the MLP model, achieving good performance in noisy condition. Another similar work is proposed in [65], where DOA estimation is performed by means of MLP fed with features based on the GCC-PHAT.

Although few DNN based systems for SLOC have been recently proposed, none of them aims to localize the speaker in terms of coordinates, reason why it is proposed in this research. In details, here a regressive model based on MLP and relying on features derived by GCC-PHAT is proposed, which directly predicts the speaker coordinates in the 2-D plane of the room under study. The model is tested again against a multi-room environment, where background noise and reverberation are expected to heavily affect the localization accuracy. The state-of-the-art method addressed in [20] is developed to compare the proposed method.

6.1.2 Proposed Method

In this section the description of the proposed model is provided. It consists in an initial feature extraction stage, which extracts localization based features. After that, an MLP processes these features and predicts the speaker position in the 2-D plane of the room. The model, depicted in Fig. 6.1, will be referred to as MLP-SLOC.

GCC-PHAT Patterns are employed as input features, of which purpose is to estimate the TDOA between microphone pair recordings in presence of a sound event. Their detailed description is given in Section 4.1.2. To calculate GCC-PHAT Patterns an assumption is made, based on the microphones displacement. In particular, only microphone pairs taken from the same array are considered. Supposing that the maximum distance between two sensors is 50 cm and the sample rate f_s is equal to 16 kHz, the maximum time delay (in samples) between 2 microphone is:

$$\Delta\tau_{max} = \frac{d_{max}}{c} \cdot f_s \approx 24$$

where $d_{max} = 50\,\text{cm}$ is maximum the distance between the microphones and c the sound speed (assumed to be 340 ms). Hence, the GCC-PHAT Patterns are extracted as follows: for each considered microphone pair the GCC-PHAT is computed with a frame size and an hop size respectively equal to 480 ms and 160 ms, having previously circularly shifted one of the two signals by 24 samples. Then, the first 50 values of the GCC-PHAT are selected. Finally, the

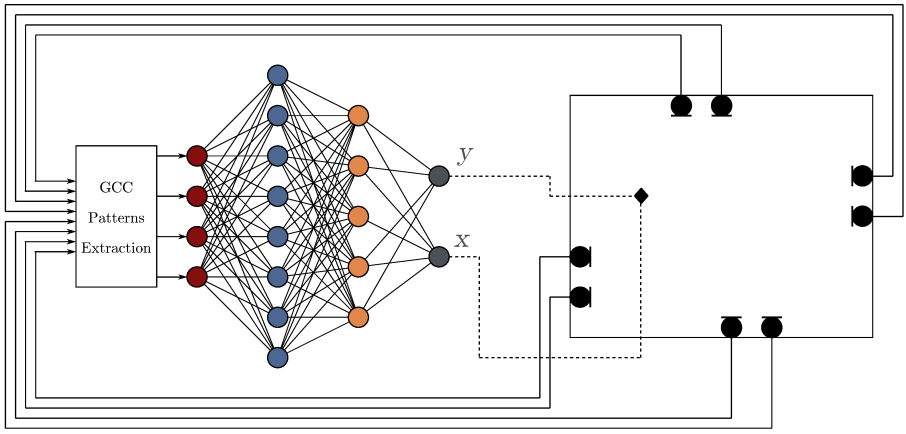


Figure 6.1: Block diagram of the proposed MLP-based system for speaker localization.

GCC-PHAT Patterns of the different M microphone pairs are concatenated for each frame, leading to a feature vector of size $50 \cdot M$.

Localization is then performed by an MLP (Section 3.2) employed as a regressive model. The network architecture is composed of an input layer of fixed dimensions equal to the feature vector size, one or more hidden layers with fully feed forward connections from one layer to the next and an output layer of two units. The activation function at the output of each layer is ReLU, discussed in Section 3.3. The network predicts the x and y coordinates of the speaker inside the room, scaled in the $[0, 1]$ range. The supervised network learning is accomplished by using the Adam algorithm [55] for the stochastic gradient-based optimization and a feature wise batch normalization [78].

6.1.3 Experimental Setup

The dataset provided by the DIRHA project [66] is used to test the proposed algorithm. DIRHA project was previously described in Section 4.2.1. The *simulated* subset is chosen for this purpose, since it contains more data and the background noise is higher and more various. Two rooms are selected for evaluating the proposed algorithm, being the Kitchen and the Living Room, for several reasons. Firstly, main events are expected to occur in those areas of an home-environment. Moreover, they are the wider rooms, leading to a more challenging source localization task. Finally, the number of available microphones is greater, since they contain a circular microphone array on the ceiling.

Training of the MLP is performed by means of an Oracle VAD, while its testing takes place on speech detected by the same Oracle VAD or by the multi-

room VAD addressed in Section 5.1. The experiments are conducted by means of the k-fold cross-validation technique in order to reduce the performance variance. A validation set in the training procedure is employed in order to perform an early-stopping strategy on the training epochs. In this work k is equal to 10, thus 64-8-8 scenes respectively compose the training, validation and test sets. The localization accuracy is evaluated in terms of Root Mean Square Error (RMSE). In addition, the performance has been evaluated in terms of $P_{cor} = \frac{N_{FINE}}{N_{TOT}}$, where N_{FINE} is the number of frames with RMSE less than 500 mm and N_{TOT} is the total number of frames. These two metrics are averaged over all the predicted outputs. After a series of preliminary tests, the learning rate, β_1, β_2 and ϵ parameters of the Adam optimizer have been set respectively to 0.001, 0.9, 0.999 and 10^{-8}. The network weights are initialized with a normal Gaussian distribution, while the batch normalization has been employed [78] with ϵ and momentum respectively to 10^{-6} and 0.9. In addition, the MLP-SLOC have been tested on signals pre-processed by cepstral pre-filtering, as described in the following section. However, no improvements were observed, hence this filtering technique was not considered.

Comparative Model

The proposed method is tested against a DOA estimation approach, being the CSP speaker localization algorithm [20]. It will be referred to as CSP-SLOC. It relies on the TDOA evaluated as in Equation 4.3, from which is it possible to evaluate the DOA by means of:

$$\frac{d\cos\theta}{c} = \Delta\tau_{ij} \Rightarrow \theta = \cos^{-1}\left(\frac{c\Delta\tau_{ij}}{d}\right), \tag{6.1}$$

where θ denotes the DOA angle, $\Delta\tau_{ij}$ is the TDOA between the *i-th* and *j-th* microphones, d is the distance between the microphone pair and c the sound speed.

In details, in a 2-D plane (Fig. 6.2), the DOA is the line connecting the estimated sound source and the middle of the segment between the microphone pair. After that, a generic point $\boldsymbol{a}$ is considered, whose distance from the k-th DOA is denoted as $D_k(\boldsymbol{a})$. Thus, the error to minimize with a least mean square strategy is defined as:

$$E(\boldsymbol{a}) = \sum_{k=1}^{M} {D_k}^2(\boldsymbol{a}), \tag{6.2}$$

where M is the total number of DOA. This procedure is applied per one room at time. The TDOA estimation highly depends on the reverberation time, the noise level and the orientation of the speaker. In order to reduce the

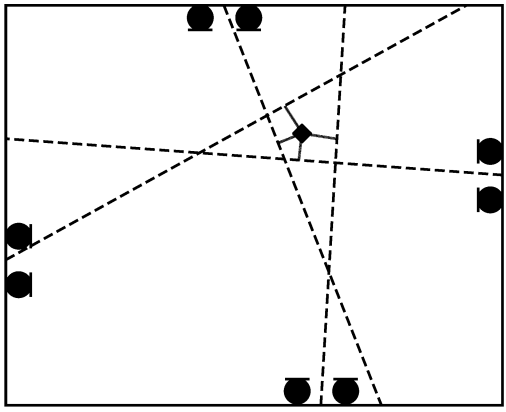


Figure 6.2: An example of the CSP Speaker Localization Algorithm (CSP-SLOC). The black points are the microphones installed in the room walls. The docked lines are the estimated DOA, the diamond is the point whose coordinates are given by the algorithm.

reverberation effect, in [20] the cepstral filtering technique described in [79] is employed for pre-processing the microphone signals. The same strategy is here adopted. The algorithm is highly dependent on the fine tuning of its parameters. After several listening tests, frame size was set to 2048 samples with no overlap, the exponential window scaling factor α was set to 0.9985 and the averaging weighting coefficient μ to 10^{-4}.

6.1.4 Main Results

A three-stage optimization strategy leads the tuning of the proposed system. The first stage consists in a variation of the network layout while keeping fixed the input feature set. The second one is a microphone pairs selection stage which aims to find the more reliable GCC-PHAT Patterns. Here combinations of signals coming from the available arrays have been gradually tested. As last step, a second network size selection is explored in order to assess or consolidate the resulting setup. An initial test is performed by comparing the proposed method at the first optimization stage against the CSP-SLOC. Considering the latter, it relies only on signals coming from the wall arrays. Hence, for a fair comparison, the first network selection is performed by means of the same audio data (i.e., 4 microphone pairs for the k itchen and 5 microphone pairs for the living room). Around 30 different network topologies combinations composed of 1, 2 or 3 hidden layers with 4, 8, 16, 32, 256, 512, 1024 units are explored. These preliminary results of the MLP-SLOC outperforms the comparative algorithm (Table 6.1), leading to an RMSE equal to 710 mm.

Room	Algorithm	RMSE (mm)	Pcor (%)
Kitchen	CSP-SLOC	1280	8.2
	MLP-SLOC	**680**	**44.6**
Living Room	CSP-SLOC	1650	7.8
	MLP-SLOC	**750**	**55.4**

Table 6.1: Comparison of the localization accuracy between the MLP-SLOC and the CSP-SLOC algorithm after the first optimization stage.

The second optimization stage targets the GCC-PHAT Patterns feeding the model. With respect to the ceiling array, 10 possible microphone pairs are available since the central microphone is excluded. In addition, the ceiling array is considered as a unique set, and it will be the starting set employed for this optimization stage. Following that, combinations of signal pairs coming from the wall arrays are added to the ceiling set. It is interesting to note that the best performance is obtained with the same number of GCC-PHAT Patterns in both rooms: 10 combinations from the ceiling array and 4 from the wall arrays, for a total feature size equal to 700. With this set up, accuracy is improved, reaching an RMSE equal to 529 mm. Furthermore, as shown in Fig. 6.3, the proposed algorithm has a significant solidity in terms of microphone positioning. In particular a standard deviation equal to 14 mm and 26 mm for the Kitchen and the Living Room is respectively obtained.

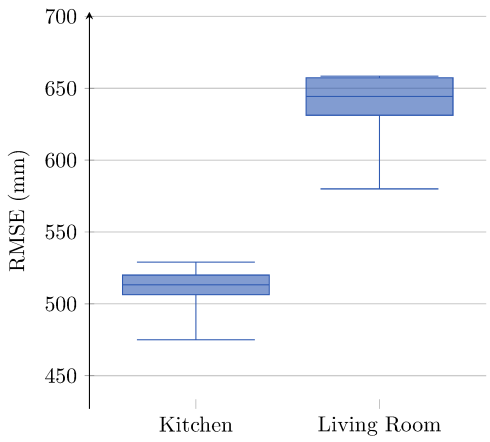


Figure 6.3: Box-plot of RMSE results for both rooms in the microphone pair selection stage.

As a further refinement, the last network size selection is performed, by using as input the set of GCC-PHAT Patterns that provided the best performance in the previous stage. For both rooms, the best localization accuracy is given

by a network with a single hidden layer with 512 units. The resulting averaged RMSE is equal to 525 mm. Details of the two best performing configurations are provided in Table 6.2.

Room	Network Layout	Microphone Pairs	RMSE (mm)	P_{cor} (%)
Kitchen	700-512-2	Ceiling Array (K3L, K3C) (K3C, K3R) (K2L, K2R) (K1R, K1L)	475	60.3
Living room	700-512-2	Ceiling Array (L4L, L4R) (L3L, L3R) (L2L, L2R) (L1R, L1C)	575	64.0

Table 6.2: Best performing setups of MLP-SLOC with the Oracle-mVAD for the two rooms after the third optimization stage.

Results with Neural-mVAD

Finally, the most performing MLP-SLOCs is tested over the predictions provided by the neural network multi-room VAD (Neural-mVAD) addressed in Section 5.1. The Neural-mVAD is based on a DBN of 2 hidden layers of 20 units each, and it achieves a SAD of 5.8% in the Simulated dataset. In this case, the MLP-SLOC performance depends on the Neural-mVAD errors, thus a strategy to evaluate false negatives and false positives must be provided. In both cases, the central point of each room is considered as a reference for the RMSE evaluation. In details, for the false negative decision of the Neural-mVAD, it is supposed that the MLP-SLOC outputs the reference position. In case of false positive decisions, it is supposed that the ground truth position corresponds to the reference one. Concluding, the integrated system Neural-mVAD+MLP-SLOC leads to a RMSE equal to 730 mm and a P_{cor} equal to 42.4% for the Kitchen and a RMSE equal to 810 mm and a P_{cor} equal to 42.7% for the Living Room.

Conclusions

A neural network approach for speaker localization in a domestic environment is discussed in this research, relying on MLP and GCC-PHAT Patterns as input features (MLP-SLOC). The approach is fully data-driven, therefore the speaker position is directly estimated without additional processing. Results

are compared with a state-of-the-art algorithm (TDOA-based), which has been recently proposed for the same multi-room scenario. As conclusion, the novel approach significantly outperforms the CSP-SLOC algorithm, achieving an averaged RMSE equal to 525 mm when preceded by an Oracle-VAD. In addition, the MLP-SLOC is integrated with a VAD algorithm previously discussed in this thesis. In this experiment, the Neural-mVAD prediction errors are dealt by the MLP-SLOC, nevertheless the localization accuracy still outperforms the CSP-SLOC, leading to an averaged RMSE equal to 770 mm. However, this strategy penalizes the accuracy of SLOC algorithm, reason why further investigation must be conducted in this sense.

In conclusion, the following research (Section 6.2) of this thesis will be oriented on the exploitation of a more complex DNN, with a major focus on extending the input data along with its temporal evolution, in order to increase the robustness of the algorithm. Furthermore, even the option of exploiting audio data recorded in different rooms will be addressed with special solutions.

6.2 Multi-room environment - Further Advancements

The promising results achieved in Section 6.1 drive the research conducted in this section, where numerous advancements with respect to the previous contribution are proposed. In particular, this section focuses on the employment of CNNs in addition to MLPs, and on the joint exploitation of data recorded from multiple microphones. Furthermore, simulations now consider even another case study in terms of multi-room environment. Last but not least, a deep investigation concerns the temporal excerpt feeding the neural models.

6.2.1 Preliminaries and Problem Statement

This work addresses the development of a completely data-driven approach for SLOC in a multi-room environment. The purpose of the data-driven strategy is to avoid a dedicated fine-tuning of parameters, which is typical and highly specific for the state-of-the-art algorithms, as discussed in Section 2.2.1. In details, the proposed algorithm is composed of a feature extraction stage and an artificial neural network, which together lead to the DNN-SLOC. A preliminary version of this model achieves remarkable results in Section 6.1, where a MLP fed by GCC-PHAT based feature is introduced. Several advancements are proposed here with respect to the model discussed in the previous section. In particular, CNNs are investigated in addition to MLPs, plus the concurrent processing of audio data coming from single or multiple rooms is addressed. In addition, specific studies target the dependence on the microphone position and the importance of a temporal context. Last but not least, while in Section 6.1 a single comparative method being the CSP speaker localization algorithm [20] is considered, in this study the proposed models are compared with a further state-of-the-art approach [24], based on SRP. Finally, experiment takes place even in the Real subset of the DIRHA corpus [66].

6.2.2 Proposed Method

The proposed multi-room speaker localization algorithm is composed of a features extraction stage and an artificial neural network. The first stage extracts GCC-PHAT Patterns features from each input frame by using pairs of microphone signals. After that, the feature matrices of previous and future frames are joined in a chunk with the purpose of exploiting the temporal evolution of the data. Hence, the ANN is trained on labelled data to estimate the coordinates (χ, ψ), i.e., the position of the speaker inside the target room. A block diagram of the algorithm based on a CNN is shown in Fig. 6.4.

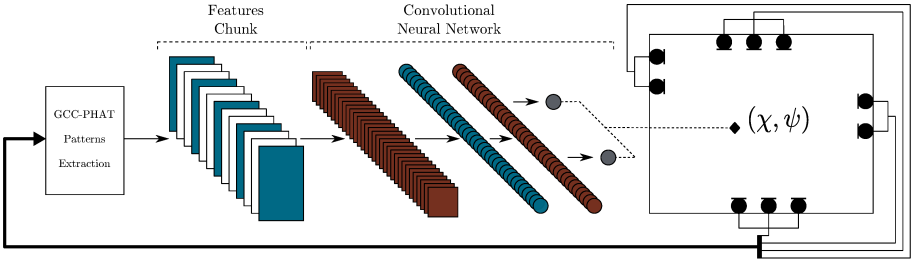


Figure 6.4: Block diagram of the proposed DNN algorithm for Speaker Localization. In this figure, the full CNN architecture is depicted.

In the multi-room scenario, speakers positions are estimated by means of a feature extraction stage and a neural network per room. Two different architectures have been investigated in this contribution: in the first, each neural network is dedicated to processing the audio signals coming from a room and it estimates the position of the speaker in that room. This architecture will be denoted as 1Rx1N-SLOC in the following. In the second architecture, the neural network jointly processes audio coming from different rooms, while estimating the speaker position only in one room. This architecture will be denoted as KRx1N-SLOC in the following, where K denotes the number of rooms. Fig. 6.5 shows the differences between the two architectures in the two rooms case study ($K = 2$).

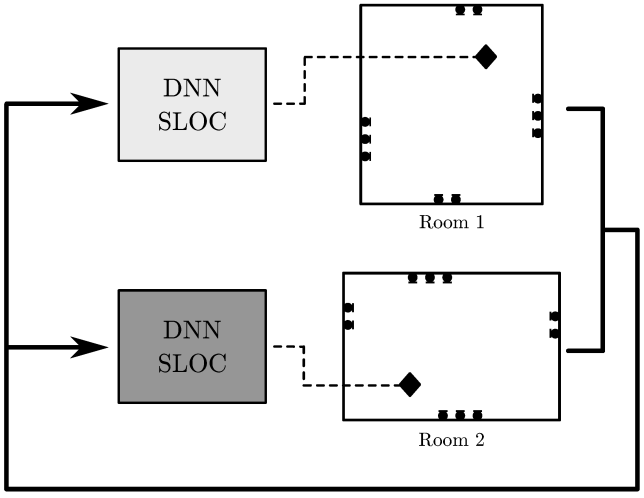


Figure 6.5: Block diagram of the proposed 2Nx1R approach. Both the DNN-SLOC algorithms localize the speaker in a single room by jointly exploiting the audio coming from both rooms.

Features based on GCC-PHAT Patterns

This set of features, accurately described in Section 4.1.2 and previously employed in Section 6.1, aims to estimate the time delay present between the audio signals captured by a microphone pair. Features are computed with a frame size and a hop size respectively equal to 480 ms and 160 ms. The first 50 values of the GCC-PHAT are selected, as in Section 6.1.2. Furthermore, features extracted from the different microphone pairs are standardized to have zero mean and unitary standard deviation. After that, all possible combinations for each microphone array are considered. In particular, the feature matrix related to the array i composed of $N^{(i)}$ microphones assumes the following form:

$$\mathbf{X}^{(i)}[n] = \begin{bmatrix} \tilde{\mathbf{x}}_{12}^{(i)}[n] \\ \tilde{\mathbf{x}}_{13}^{(i)}[n] \\ \vdots \\ \tilde{\mathbf{x}}_{1N^{(i)}}^{(i)}[n] \\ \tilde{\mathbf{x}}_{23}^{(i)}[n] \\ \vdots \\ \tilde{\mathbf{x}}_{2N^{(i)}}^{(i)}[n] \\ \vdots \\ \tilde{\mathbf{x}}_{(N^{(i)}-1)N^{(i)}}^{(i)}[n] \end{bmatrix}. \tag{6.3}$$

where $\tilde{\mathbf{x}}_{jk}^{(i)}[n]$ is GCC-PHAT Pattern evaluated from the signal captured by the j-th and the k-th microphones of the i-th array at the n-th frame. Finally, from all the M considered arrays, the input matrix $\mathbf{X}[n]$ is given by

$$\mathbf{X}[n] = \begin{bmatrix} \mathbf{X}^{(1)}[n] \\ \mathbf{X}^{(2)}[n] \\ \vdots \\ \mathbf{X}^{(M)}[n] \end{bmatrix}, \tag{6.4}$$

In Fig. 6.6 an example of the $\mathbf{X}[n]$ matrix is depicted. Colors represent the amplitude of GCC-PHAT Patterns, where orange tones denote the lowest values and blue tones the highest values in the considered range. In the exposed case, the $\mathbf{X}[n]$ is composed of 10 GCC-PHAT Patterns, belonging to a subset of the possible pairs originated from a circular ceiling array. It is possible to note that the maximum arrival time difference is equal to 9 samples (corresponding of around 0.2 seconds), observed in the GCC-PHAT Pattern of the fourth microphone pair.

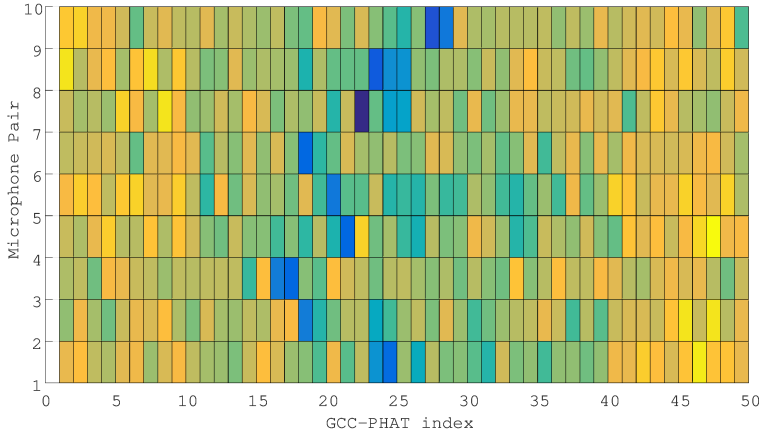


Figure 6.6: An example of GCC-PHAT Pattern matrix $\mathbf{X}[n]$ obtained from a combination of microphones belonging to a ceiling array. Blue tones represent maximum amplitude values of $C_{ab}(n, \tau)$ in the range of considered τ.

In addition, the extracted features are extended by means of a temporal context, composed by the frames of the signal adjacent to the one being processed. In order to exploit this information, the input of the neural network is augmented with the $(C-1)/2$ GCC-PHAT Patterns preceding and following the current GCC-PHAT Pattern matrix $\mathbf{X}[n]$. The network, thus, estimates the speaker position by employing a *chunk* of feature matrices defined as:

$$\overline{\overline{\mathbf{X}}}[n] = \begin{bmatrix} \mathbf{X}[n - \frac{C-1}{2} \cdot s] \\ \vdots \\ \mathbf{X}[n - s] \\ \mathbf{X}[n] \\ \mathbf{X}[n + s] \\ \vdots \\ \mathbf{X}[n + \frac{C-1}{2} \cdot s] \end{bmatrix}, \tag{6.5}$$

where C is the total length of the chunk, and s denotes the stride, which defines the temporal extension of the chunk. Fig. 6.7 shows an example with $s = 2$.

In cases where the selected frames do not contain speech, the GCC-PHAT Patterns matrices related to the first or the last frame of the segment are replicated accordingly.

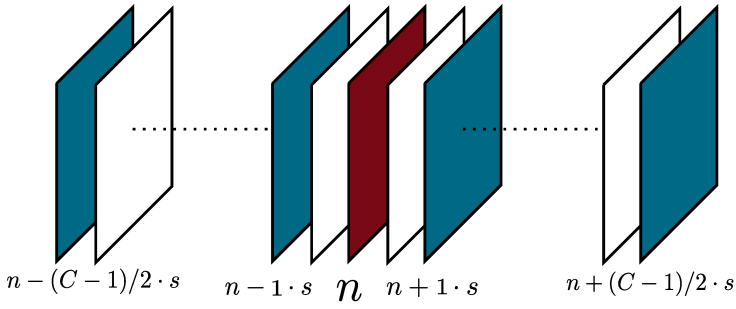


Figure 6.7: A scheme showing the GCC-PHAT Patterns matrices composing the temporal context. The GCC-PHAT Patterns matrix of current frame $\mathbf{X}[n]$ is shown in red, while the matrices included in the final chunk are shown in blue. The value of the stride s is 2.

DNN-based SLOC

The DNN composing the two proposed models are here discussed. The first one consists in a MLP, being similar to the model discussed in Section 6.1.2, trained as a regressive model and making use of ReLU activations. The feature extraction stage arranges GCC-PHAT Patterns as 3-D tensor, however MLP takes as input a mono-dimensional vector. Hence, matrices are flattened to one dimension and the network input layer consists of a number of units equal to $C \cdot M$.

The second model relies on a CNN, always treated as a regressive model. Similarly to Section 5.2, its first convolutional layer deals with a 3-D input, being then followed by others convolutional layers and then fully connected layers. ReLU is chosen as activation function. In particular, the 3-D matrix is obtained by stacking the 2-D matrices described in Equation 6.3 by means of the temporal context. Indeed, as shown in Fig. 6.6, it is reasonable to expect that this procedure leads to specific patterns related to localization, reason why CNNs have been employed.

6.2.3 Comparative Methods

The neural network localization algorithm has been compared with two state-of-the art methods, described as follows.

Crosspower Spectrum Phase Speaker Localization

The first algorithm taken as reference is the CSP speaker localization algorithm (CSP-SLOC) [20], previously employed and described in Section 6.1.3. It is composed of two consecutive steps and due its structure, the algorithm is evaluated per one room at time. It relies on the estimation of the TDOAs

present from several pairs, from which an hyperbolic curve in a 2-D plane is constructed. Several physical aspects affect the accuracy of the CSP-SLOC, such as the orientation of the speaker, the noise level and reverberation. A pre-processing cepstral filtering technique provided in [79] is employed, being the cepstral dereverberation algorithm, which relies on no-overlapping frames of 2048 samples, to which is applied an exponential window with scaling factor $\alpha = 0.9985$ and averaging weighting coefficient $\mu = 10^{-4}$. With regards to the CSP-SLOC, in order to be consistent with the proposed method, the TDOAs are computed with a rate of 100 frames/second and a frame overlap equal to 66%.

Steered Response Power Using the Phase Transform

Another state-of-the-art algorithm has been considered for comparison purpose. It has been already discussed in Section 2.2.2. It consists in a modification of the SRP method, based on the SRC approach, as described in [24]. SRC avoids the complete fine grid-search, by applying an iterative process that progressively contracts the search volume for local maxima, thus reducing the overall computational cost.

As first step, a delay-and-sum beamformer is steered in the considered volume for each n-th frame of length T, leading to the SRP function for the spatial vector $\mathbf{s}$:

$$P_n(\mathbf{s}) \equiv \int_{nT}^{(n+1)T} \left| \sum_{a=1}^{M} w_a s_a(t - \tau(\mathbf{s}, a)) \right|^2 dt, \tag{6.6}$$

where $s_a(t)$ is the signal from a generic microphone a, w_a its weight, $\tau(\mathbf{s}, a)$ the distance in the time domain between $\mathbf{s}$ and that microphone. Practically, Equation 6.6 is evaluated in the frequency domain, scaled by the PHAT weighting factor. The SRC is iteratively applied: $J_0 = 5000$ points are randomly evaluated, $N_0 = 20$ points maximizing Equation 6.6 are selected and the search volume is restricted to a smaller region that contains the N_0 points. In the following, this algorithm will be referred to as SRP-SLOC.

6.2.4 Experimental Setup

The performance of the two proposed methods, which are the 1Rx1N-SLOC and the 2Rx1N-SLOC, described in Section 6.2.2, are investigated in the two DIRHA subsets, the Real and the Simulated, being described in Section 4.2.1. Both approaches exploit MLP and CNN as neural networks. The evaluated SLOC algorithms work with the assumption of the presence of an Oracle VAD, which selects only the speech portions of audio signals, consistently with the

research conducted in Section 6.1. Nevertheless, a Neural VAD, differently from the previous chapter, is not taken into account here. In details, the objective is to investigate the most reliable and performing SLOC algorithm in an absolute sense, independently from the errors produced by a real VAD.

Experiments are conducted by means of a k-fold cross-validation technique in order to reduce the performance variance, and an early-stopping training strategy has been employed to prevent overfitting. In the *Simulated* dataset k is equal to 10, thus 64-8-8 scenes respectively compose the training, validation and test sets. In the *Real* dataset, due to the absence of speech in certain scenes, only 11 of them were suitable for each room. Thus, a leave-2-out cross validation has been adopted, where, for each fold, 7 scenes compose the training set, 2 the validation set and 2 the test set. This configuration has been previously employed in Section 5.1.3.

The selection of the most performing DNN-SLOC architecture has been carried out by means of a four-stage optimization strategy, described as follow:

- **I - Network Size Selection.** It consists in varying the network layout while keeping fixed the input signals (i.e., 4 microphone pairs for the Kitchen and 5 microphone pairs for the Living Room). Concerning the MLP architecture, 30 different network topologies have been investigated, composed of 1, 2 or 3 hidden layers with 4, 8, ..., 1024 units. On the contrary, in the case of CNN architecture a higher number of parameters must be considered, which have been reported in Table 6.3 for the sake of conciseness. No temporal context is employed for the CNN.

- **II - GCC-PHAT Patterns Selection.** This stage aims to find the most performing GCC-PHAT Patterns matrix $\mathbf{X}[n]$ between a subset of the available $\mathbf{X}^{(i)}[n]$ microphone combinations belonging to different arrays. The starting point was the circular array placed on the room ceilings, which is composed of $N = 6$ microphones. Excluding the central one, it leads to 10 possible pairs. Then, combinations of signal pairs $\tilde{\mathbf{x}}_{ab}^{(i)}[n]$ coming from the wall arrays have been gradually added to the ceiling array signals with a sequential forward selection strategy, in order to arrange the evaluated $\mathbf{X}[n]$.

- **III - Network Size Selection.** Another network size selection is then performed, having as input features the set of GCC-PHAT Patterns providing the best results in the previous step.

- **IV - Temporal Context Selection.** Here the objective is evaluating the effects of the temporal context, by varying the strides values, i.e. $s = \{1, 3, 4, 5\}$, and the context dimensions, i.e. $C = \{3, 7, 11, 13, 15, 17, 19, 21\}$.

CNN							
First Convolutional Layer			Second Convolutional Layer			Neurons	
Nr. of Kernels	Size	Pooling	Nr. of Kernels	Size	Pooling	I Layer	II Layer
16 24 48	3 × 3 5 × 5	2 × 2 -	16 24 48	2 × 2	-	64 128 256 512	128 256 512
MLP							
Fully Connected Layers	Nr. of Units	4, 8, 16, 32, 256, 512, 1024		Nr. of Layers	1, 2, 3		

Table 6.3: Network topology parameter explored during the optimization stages.

In addition, the parameters of the training optimizer (i.e., Adam, batch normalization [55, 78]) are set as shown in Table 6.4. The performance of the models are evaluated in RMSE and P_{cor}, defined in Section 6.1.3.

	Weight initialization	**Epochs**	**Optimizer parameters**
MLP	Gaussian distr. $\mu = 0$ $\sigma = 0.1$	500 50 E.S. patience	learn. rate = 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
CNN	Gaussian distr. $\mu = 0$ $\sigma = 0.1$	200 50 E.S. patience	learn. rate = 0.025, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$
Batch Normalization: $\epsilon = 10^{-6}$, $\mu = 0.9$			

Table 6.4: Parameters of the Adam optimizer selected for the neural network training. "E.S." denotes early stopping strategy, evaluated on the validation loss.

6.2.5 Main Results

Results on Simulated Dataset

The best results obtained by the different addressed algorithms in the Simulated case study are reported in Table 6.5. As regards the DNN-SLOC, both the values are reported at the end of the third stage of optimization (i.e., without temporal context), and at the end of the fourth stage, in which performance

have been studied varying the parameters of C and s. For instance, the first two rows report the results related to the MLP-SLOC algorithm, of type 1Rx1N, without and with the temporal context, respectively.

In this dataset both the CSP-SLOC and the SRP-SLOC yield a low localization performance, with an RMSE averaged equal to 1464 mm and 981 mm, in the kitchen and living room respectively. This performance is remarkably worse in comparison to what obtained with the DNN architectures.

SIMULATED								
ROOM	Kitchen			Living Room			Average	
	Context	RMSE	P_{cor}	Context	RMSE	P_{cor}	RMSE	P_{cor}
MLP 1Rx1N	/	475	60	/	575	64	525	62
MLP 1Rx1N	19 - 4	370	69	17 - 4	442	72	406	71
MLP 2Rx1N	/	453	59	/	571	62	512	61
MLP 2Rx1N	17 - 3	375	67	19 - 3	455	70	415	68
CNN 1Rx1N	/	529	57	/	625	63	577	60
CNN 1Rx1N	21 - 4	309	75	21 - 3	358	78	**333**	**77**
CNN 2Rx1N	/	522	57	/	635	61	579	59
CNN 2Rx1N	21 - 4	331	73	17 - 5	374	75	353	74
CSP-SLOC	/	1281	8	/	1648	8	1464	8
SRP-SLOC	/	1005	22	/	958	37	981	30

Table 6.5: Comparison of best results of SLOC algorithms in terms of RMSE (mm) and P_{cor} (%) in the Simulated case study.

The focus now goes on the results attained at the end of the third stage. The best DNN configuration is the 2Rx1N MLP-SLOC. In particular, with both 1Rx1N and the 2Rx1N configurations, the MLP performs slightly better than the CNN. Furthermore, even if only a slight improvement is observable in terms of RMSE for the 2Rx1N case, the advantage of this setting lies in the statistical behavior, as reported in Fig. 6.8. The boxplots show in terms of mean and standard deviation a narrow dispersion of results. Indeed, the exploitation of audio from both rooms reduces the dependence on the microphones location inside the room. In details, the best MLP layouts for the Kitchen and Living room are two single layer networks of 256 and 1024 units, respectively. With regards to the CNN-SLOC, the two best layouts are the following: for the Kitchen, a single convolutional layer of 48 kernels of size 5×5 with no pooling, followed by two feed-forward layers with 512, 512 units, while for Living room, a single convolutional layer of 24 kernels of size 5×5 with no pooling, followed by two feed-forward layers with 256, 512 units.

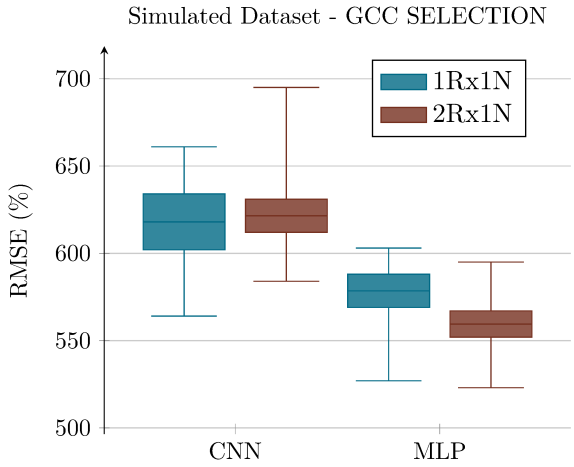


Figure 6.8: Boxplot for the Simulated dataset, for MLP and CNN neural network, comparing 1Rx1N and 2Rx1N setups. This evaluation is performed at the end of the II stage of optimization, considering all the tested GCC Patterns configurations. The results are averaged over the two target rooms.

The effect of temporal context is considered with the final optimization stage, in which the network size determined after the third stage is kept fixed.

The best localization performance has been obtained by using the CNN-SLOC having as input the microphone signals coming only from the target room. The resulting averaged RMSE is equal to 333 mm and the highest P_{cor} is 77%. Details of the configuration parameters for the best performing setups on Simulations subset are shown in Table 6.6. The performance improvement at this stage is evident, whereas the employment of multiple room audio features does not seem to have a beneficial effect for CNN-based algorithms.

The results obtained at the very end of the optimization procedure show the ability of the CNN architecture to efficiently exploit the contextual information of adjacent frames, with a reduction of 42.2% of the localization error with respect to the configuration with $(C, s) = (1, 1)$, as shown in Fig. 6.9. The best resulting values of context and strides are $(C, s) = (21, 4)$ and $(C, s) = (17, 5)$ for Kitchen and Living Room, respectively, which means processing a segment of duration approximatively equal to 8.5 s.

The introduction of the temporal context has beneficial effects also with the MLP, but with a lower error reduction (equal to 21.5%). The results obtained in the investigation of the audio excerpt are reported in Fig. 6.10, where the different strides s are plotted while varying the temporal context C.

It can be noticed that a similar trend of performance with respect to temporal resolution values is registered.

	Room	Kitchen	Living Room
Features Settings	Configuration	1Nx1R	1Nx1R
	Context	21 - 4	21 - 3
Microphones	Ceiling	Circular Array	Circular Array
	Wall	K3L,K3C K2L,K2R K1R,K1L	L4L,L4R L3L,L3R L2L,L2R L1R,L1C
Convolutional Kernels	Number	48	24
	Size	5×5	5×5
Feed Forward Layers	First Layer	512	256
	Second Layer	512	512
Results	RMSE (mm)	309	358
	P_{cor}	75	78

Table 6.6: Results for Convolutional Neural Networks with the best performing configurations in the Simulated case study.

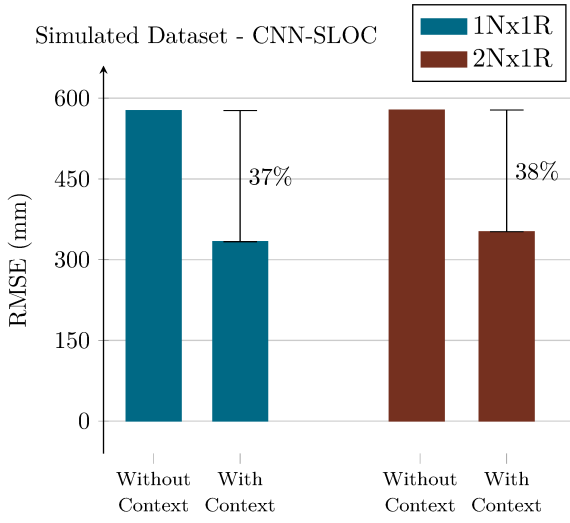


Figure 6.9: Improvements on Simulated dataset for CNN-SLOC when temporal context is considered.

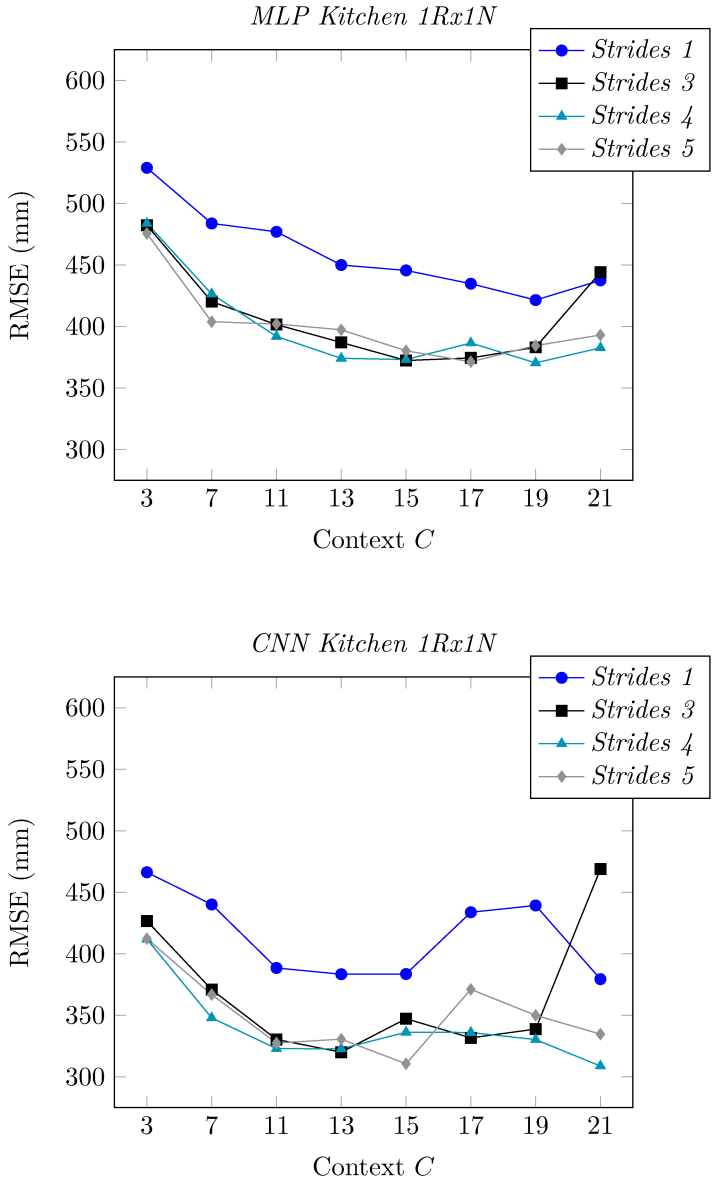


Figure 6.10: Performance trend in the Simulated dataset for different *strides* at growing *context*. The considered room is Kitchen with 1Rx1N configuration, the two DNN are plotted.

Results on Real Dataset

The main difference between the Real and the Simulated dataset lies in the speaker position, which is not fixed during the scene. Thus, the speaker moves within the room while pronouncing the sentence. Furthermore, in this dataset

overlapping events are not present.

Table 6.7 reports a comparison among the evaluated algorithms with the best results obtained for each configuration. As for the results obtained for the Simulated dataset, the proposed models are reported with and without the temporal context.

The localization performance obtained by the CSP-SLOC is quite similar to the one in the Simulated case study, with an average RMSE equal to 1280 mm. The SRP-PHAT algorithm attains an averaged RMSE equal to 792 mm. Such a performance achieved by the comparative methods are significantly superior than the one obtained in the Simulated case study. The motivation likely relies on the higher SNR level characterizing the audio files in the Real dataset.

REAL								
ROOM	Kitchen			Living Room			Average	
	Context	RMSE	P_{cor}	Context	RMSE	P_{cor}	RMSE	P_{cor}
MLP 1Rx1N	/	789	39	/	688	43	644	42
MLP 1Rx1N	19 - 3	498	60	21 - 4	446	63	472	61
MLP 2Rx1N	/	710	38	/	619	45	664	41
MLP 2Rx1N	21 - 4	494	59	17 - 4	445	63	470	61
CNN 1Rx1N	/	706	37	/	583	47	686	42
CNN 1Rx1N	21 - 3	460	64	15 - 5	349	78	405	71
CNN 2Rx1N	/	687	40	/	552	54	470	61
CNN 2Rx1N	19 - 4	425	72	17 - 3	350	75	**387**	**74**
CSP-SLOC	/	1394	9	/	1166	11	1280	10
SRP-SLOC	/	895	30	/	690	53	793	42

Table 6.7: Comparison of best results of SLOC algorithms in terms of RMSE (mm) and P_{cor} (%) in the Real case study.

As shown in Fig. 6.11, in this case the 2Rx1N architecture produces a more significant improvement of performance both for the MLP-SLOC and the CNN-SLOC, and consistently with what observed for the Simulated dataset, the variance with the microphone position decreases. This behaviour may be motivated by the fact that the Simulated dataset is noisier compared to the Real one, hence feeding the DNN model with data coming from multiple rooms may lead to a too noisy input tensor, which could become deceiving for the DNN. On the other hand, stacking multiple room data does not create a so noisy input in the Real dataset.

As result of the third optimization stage, the best performing MLP layout in the 2Rx1N case is composed of a single layer of 16 units both for the Kitchen

and the Living Room. Regarding the CNN-SLOC, the most performing configuration is composed by a convolutional layer with 24 5×5 kernels without pooling, followed by a two layers MLP with respectively 256, 256 units and 256, 512 units for Kitchen and Living Room.

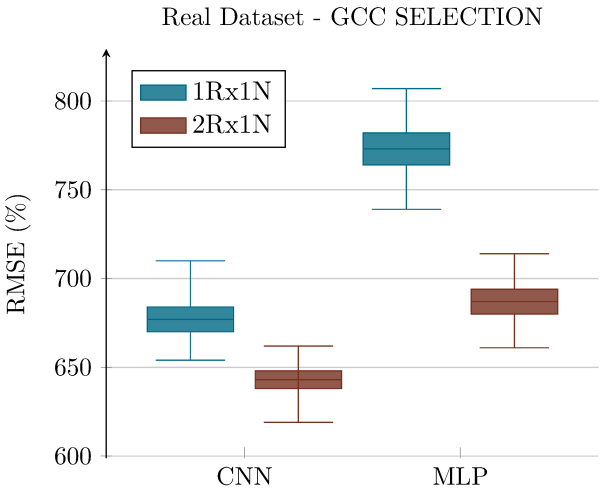


Figure 6.11: Boxplot for the Real dataset, for MLP and CNN neural network, comparing 1Rx1N and 2Rx1N setups, after the II optimization stage. Averaged results are shown.

As performed with the Simulated dataset, the effects of the temporal context have been studied after the third optimization stage. Results at the end of the fourth optimization stage are reported in Table 6.8. The introduction of the temporal context leads to an average localization accuracy of 387 mm with the CNN-SLOC for the 2Rx1N configuration. As highlighted in Fig. 6.12, the RMSE reduces by 37.2% for the 1Rx1N configuration and by 37.5% for the 2Rx1N configuration. In concordance with the results obtained with the Simulated dataset, the MLP architecture benefits by the introduction of the temporal context, with a resulting error reduction of 36% for the 1Rx1N and of 29% for the 2Rx1N configuration.

For the CNN-SLOC applied in the Real dataset, the best resulting values of context and strides are $(C, s) = (19, 4)$ and $(C, s) = (17, 3)$ respectively for Kitchen and Living Room, corresponding to a segment of length about 7.6 s and 5.1 s.

Fig. 6.13 report the RMSE for different context sizes C and strides s in the case of 1Rx1N applied to the Kitchen room. Similarly to the *Simulated* dataset, the variation of the temporal resolution produces a similar performance trend for the two neural architectures. Details of the best performing configurations for the CNNs are provided in Table 6.8.

	Room	Kitchen	Living Room
Features Settings	Configuration	2Nx1R	2Nx1R
	Context	19 - 4	17 - 3
Microphones	Ceiling	Circular Array (K)	Circular Array (L)
	Wall	K1R,K1L K3L,K3C K3L,K3C L1C,L1L L2L,L2R L3L,L3R	 K2L,K2R K3C,K3R L2L,L2R L1R,L1C
Convolutional Kernels	Number	24	24
	Size	5×5	5×5
Feed Forward Layers	First Layer	256	256
	Second Layer	256	512
Results	RMSE (mm)	425	350
	P_{cor}	72	75

Table 6.8: Results for Convolutional Neural Networks with the best performing configurations in the Real case study.

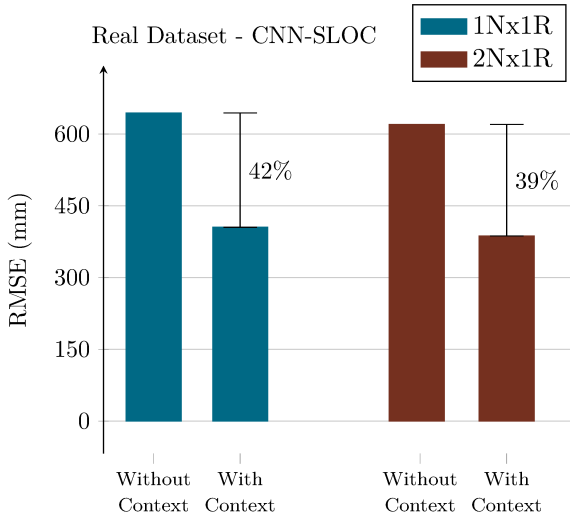


Figure 6.12: Improvements on Real dataset for CNN-SLOC when temporal context is considered.

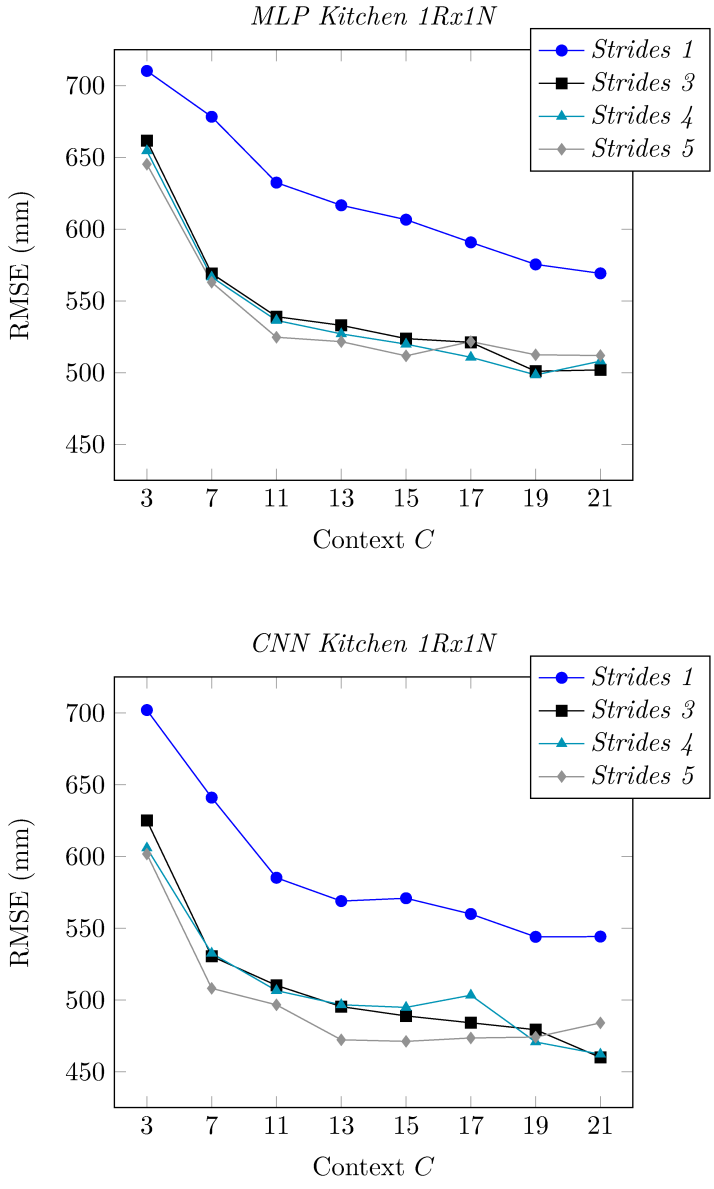


Figure 6.13: Performance trend in the Real dataset for different *strides* at growing *context*. The considered room is Kitchen with 1Rx1N configuration, the two DNN are plotted.

Conclusion and Outlook

This research proposes numerous advancements to the DNN-based system for SLOC discussed in Section 6.1. The data-driven SLOC approach is tested against a multi-room environment. Two architectures are investigated, based

on MLP and CNN, respectively, and being fed with GCC-PHAT Patterns. The coordinates of a speaker inside the target room are directly estimated. A particular effort has been directed to the evaluation of a spatial and temporal context, revealing the latter to be extremely decisive. In details, audio coming from one or two rooms has been jointly exploited (1Rx1N, 2Rx1N), while the temporal evolution has been tested by means of close in time frames. The algorithm implicitly requires an Oracle VAD in order to process only human speech. Results are evaluated on the DIRHA Dataset in comparison with two state-of-the-art algorithm, based respectively on CSP estimation and on SRP. As result, the proposed system based on CNNs improves the performance by 66% and 51% respectively for the Simulated and the Real dataset with respect to the SRP-PHAT approach, which proofs to be the most effective classical algorithm. Furthermore, the CNN with 3-D kernels, previously addressed in Section 5.2 for VAD, is able to exploit the temporal context information more efficiently respect to the MLP network both in terms of RMSE and P_{cor}.

Future works will target the development of a unique system capable of simultaneously detect and localize a speaker. Indeed, promising results have been achieved for VAD by using multiple channels and an extended temporal context, as in Section 5.2, plus similar results have been obtained here for SLOC. Hence, a combination of VAD and SLOC systems relying on multiple microphones and time-extended data may lead to promising results.

6.3 End-to-end Azimuth Localization

The focus of this research is to develop a machine learning framework inspired by the human hearing system for localizing a speaker in reverberant environment. Although this study shares many aspects to the previous work of this chapter, several differences are introduced with respect to Section 6.1 and Section 6.2. In details, since the focus now goes to localization performed by the human being, a different case study is taken into account. Indeed, here a totally novel end-to-end approach concerning binaural localization is considered, being largely different from Section 6.2, where microphone arrays are employed. This strategy is adopted since binaural sound localization allows to address the azimuth localization task independently from other phenomena which strongly affect a multi-room environment. Furthermore, within this study the employment of a human like mannequin allows to better simulate the human hearing system, while this is not possible when linear or circular arrays are exploited.

6.3.1 Preliminaries and Problem Statement

In the last years, localization systems based on DNNs have shown promising performance. In [80], probabilistic neural networks are used to estimate the DOA in an indoor environment using GCC-based features. Binaural cues are employed in [81], where the CCF is used as features in a DNN to estimate the azimuth of a sound source with simulated head movement. CNN architectures are used in [28, 43] using frequency-domain features such as the phase or the magnitude of the signal. A similar scenario has also been previously studied in this thesis in Section 6.1 and Section 6.2, where a CNN predicts the speaker coordinates.

All of the approaches so far are based on hand-crafted features explicitly extracted from the waveform. Such a feature extraction process may lead to a loss of information which can affect the performance of the SLOC algorithm. Human listeners, on the other hand, are able to use waveforms from just two ears to reliably determine the location of a sound source [82]. It is well known that this ability is largely based on both binaural cues, such as the Interaural Time Difference (ITD) and the Interaural Level Difference (ILD), and monaural spectral cues created by direction-dependent filtering of the outer ears. However, it is less clear how these cues are seamlessly combined and processed by the auditory cortex for sound localization [83].

Furthermore, much effort has been recently spent in the development of end-to-end systems for many audio applications. For example, a model for end-to-end ASR is proposed in [84], which combines localization, beamforming, acoustic modelling and speech enhancement in a unified DNN. In audio generation, several end-to-end methods were proposed to directly generate waveforms

from text [85, 86].

Contribution

Within this research, a novel end-to-end approach for sound localization, referred to as *WaveLoc*, is proposed. One of the main objective is to avoid an explicit feature extraction stage, which may introduce an information loss in the input signals. Hence, the proposed approach uses a CNN with a cascade of convolutional layers to implicitly extract features directly from the raw waveform for sound localization. One of the key stages in the network is the initial frequency analysis, being investigated by means of two different approaches. Indeed, the first one is auditory-inspired and uses a convolutional layer based on the gammatone filterbank [87]. The gammatone filter is a widely-used model of auditory frequency analysis, with bandwidths set to reproduce human critical bandwidths [88]. The second model relies on a standard convolutional layer which is intended to learn how to perform frequency analysis along with the training process of the entire network. By analysing these two opposite strategies important observations can be made with respect to frequencies and binaural cues useful for sound localization. After frequency analysis, further convolutional layers with 2-D kernels operates directly on the signals from both ears to extract features that are similar to the binaural cues used by the auditory system. The extracted features are finally concatenated and used as input to a DNN with fully connected layers, in order to map them to the corresponding source azimuth.

The following simulations show that the proposed WaveLoc systems are able to accurately estimate the azimuth of a sound source in the anechoic condition. However, the performance of the data-driven WaveLoc approach is poor in reverberant conditions when trained only on anechoic signals. This leads to a detailed investigation of the benefits of Multi-Conditional Training (MCT), following which robust performance of both the wave-based approaches are achieved across a range of challenging reverberant conditions.

6.3.2 Proposed Method

Overview

The proposed end-to-end sound localization approach is illustrated in Fig. 6.14. The CNN can be broadly divided into three stages: (i) a frequency analysis stage that takes the framed binaural ear signals as input, (ii) a feature extraction stage with a cascade of convolutional layers to extract suitable features for sound localization, and (iii) a sound localization stage based on several dense layers to perform sound localization as a classification task.

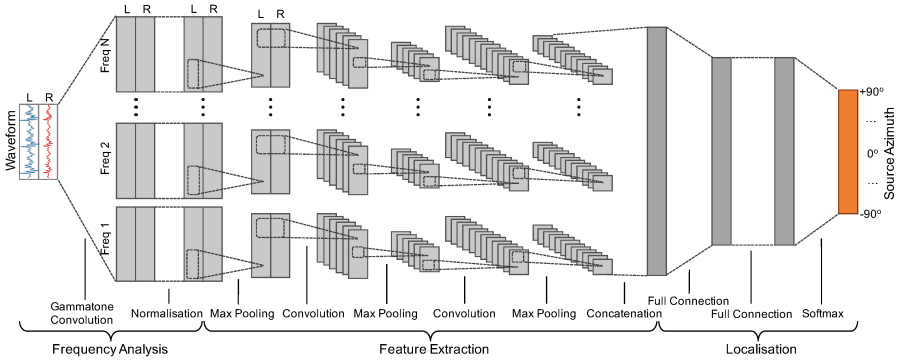


Figure 6.14: The proposed end-to-end WaveLoc-GTF system using convolutional neural networks for binaural sound localization.

The raw waveforms of the left and right ear signals, as indicated by 'L' and 'R' in Fig. 6.14, are directly used as inputs to the proposed CNNs. The ear signals are sampled at 16 kHz and framed with 20 ms window size with 10 ms overlap. In each frame the left and right channels are stacked together to form an input matrix of size 2×320.

It is well established that the auditory system performs a frequency analysis that divides the ear signal into frequency bands, and then does analysis on the fine time signal in each band [89, 90]. Such processing has been shown to improve the robustness when exploited in a binaural sound localization system, particularly in reverberant environments [34]. To simulate this operation, the first stage of the CNN performs a frequency analysis which filters the ear signals in the time domain with convolutional kernels.

Two frequency analysis strategies are investigated in this study. In the first system, named *WaveLoc-GTF*, the frequency analysis is performed by a convolution layer which is broadly based on a gammatone filterbank [87]. As shown in Fig. 6.14, the frequency analysis layer consists of a number of frequency channels. The following convolutional layers in each frequency channel elaborate upon the frequency analysis output, in order to extract frequency-dependent features. The second system, named *WaveLoc-CONV* imposes no constraint on frequency analysis. Instead, a convolutional layer with 1-D convolutional kernels is exploited to analyse frequency, with parameters learned from the data as part of the network training process.

In both systems, the frequency analysis is followed by a layer of 2-D convolutional kernels to extract features based on correlations of the left and the right channels. In WaveLoc-GTF these kernels are applied separately for each output of the gammatone filters, hence each frequency band is elaborated within an independent channel, while in WaveLoc-CONV they are applied to the sin-

gle frequency analysis layer. The correlation-based features are closely related to ITD and ILD cues, which are further processed by another convolutional layer with 1-D kernels in order to search for specific patterns that are related to the localization task. Finally, the features produced by the convolutional layers are flattened and concatenated, before being passed to two dense layers. A softmax activation function is used in the output layer in order to perform sound localization as a classification task.

WaveLoc-GTF

Fig. 6.14 illustrates the first proposed CNN: WaveLoc-GTF. As discussed, the frequency analysis is performed by a gammatone filter bank, which consists of 32 filters spanning between 70 and 7000 Hz with peak gain set to 0 dB. These filters are directly coded into *non-trainable* CNN kernels of size 1×320, with a linear activation function. The gammatone impulse response is given by:

$$w[t] = at^{n-1}\cos(2\pi ft + \phi)e^{-2\pi bt} \tag{6.7}$$

where t is time, a is the amplitude, f is the centre frequency, ϕ is the phase of the carrier, n is the filter's order, and b is the filter's bandwidth. The necessity of flipping the 1-D kernel raises. Indeed, A 1-D convolutional kernel performs the convolution operation following:

$$y[t] = \sum_{m=-M}^{M} x[m]w[t+m] \tag{6.8}$$

where x is the input signal, w the weights of the filter, t is the index of the actual value and M is the filter length. However, the time domain convolution is ruled by:

$$y[t] = \sum_{m=-M}^{M} x[m]w[t-m] \tag{6.9}$$

In other words, the 1-D kernel performs a time domain cross-correlation between the filter and the input signal. Nevertheless, here the objective is to perform a time domain convolution, hence, the design procedure of the filter must lead to $w[t-m]$, reason why the filter is designed in the opposite time direction of the binaural features.

In each frequency band, the resulting feature maps share the same dimensions (2×320) of the input matrix. A normalisation layer is then applied which looks for the maximum absolute value across all the gammatone channels before dividing them by this value. Hence, the output feature values range between [-1,1], which are further processed with 1×2 max pooling.

A separate stack of two further convolutional layers processes each normalised

channel, searching for specific patterns related to localization. The first convolutional layer has 2-D kernels of size 2×18 and the second layer has a set of 1-D kernels of size 1×6. Both convolutional layers are followed by 1×4 max pooling and employ *ReLU* activation. Finally, the processed channels are concatenated and fed into two fully connected dense layers. Each dense layer consists of 1024 hidden units with *ReLU* activation and a dropout rate of 0.5.

The output layer consists of 37 nodes corresponding to the 37 azimuth classes, with *softmax* activation.

WaveLoc-CONV

The neural architecture of the second system, WaveLoc-CONV, employs a single convolutional layer dedicated to frequency analysis. Its key difference from WaveLoc-GTF is that the frequency analysis of this model is learnt during the training process together with other parameters of the network. A convolutional layer with 64 1-D kernels of shape 1×256 is employed as time domain filters for frequency analysis. It is reasonable to expect that the shape of a convolutional kernel directly trained on a raw waveform will be similar to all the sinusoidal components that form the waveform itself. In other words, the convolutional kernels are characterised by a set of sinusoidal functions, which lead to a particular frequency response of the kernel itself. This result has been previously observed in [84].

The convolutional layer is followed by 1×2 max pooling with a linear activation function applied. As in WaveLoc-GTF, two more convolutional layers are employed to search for features suitable for localization. However, instead of acting separately for each channel as in WaveLoc-GTF, they now jointly process all the output of the frequency analysis stage. The first of the two layers uses 64 2-D kernels of size 2×18 to look for correlations between the left and right channels. The second uses 64 1-D kernels of size 1×6. Both layers use the ReLu activation function and are followed by 1×4 max pooling. Finally, the outputs are flattened and fed into a two fully-connected hidden layers with 1024 units each. The output layer uses softmax activation with 37 neurons.

All the hyperparams or both end-to-end architectures are chosen based on an optimisation process using a development dataset.

6.3.3 Experimental Setup

Binaural simulation

Binaural signals are simulated by convolving speech recordings with the Surrey BRIR database [67], previously addressed in Section 4.2.2. The Surrey BRIRs were captured using a Cortex HATS in both anechoic and reverberant rooms. A

total of 37 azimuth angles are used, ranging from [-90°, 90°] in steps of 5°, where 0° is located exactly in front of the head. Four reverberant rooms are employed, denoted A–D. The reverberation time (T_{60}) and Direct-to-Reverberant Ratio (DRR) of each room is shown in Table 6.9.

	Room A	Room B	Room C	Room D
T_{60} (s)	0.32	0.47	0.68	0.89
DRR (dB)	6.09	5.31	8.82	6.12

Table 6.9: Room characteristics of the Surrey BRIR database [67].

Speech signals belonging to the DARPA TIMIT database [70], described in Section 4.2.3, are convolved with each BRIRs. The initial and final frames of each speech utterance are truncated if silence is present. The training dataset is obtained by randomly selecting 24 sentences per azimuth from the TIMIT training subset, while another 6 sentences composes the validation dataset. 15 more sentences per azimuth are selected from the TIMIT test subset to create the test dataset.

Experimental setup

For training the *Adam* optimizer with a learning rate of 1e−3 and a batch size of 128 samples is employed. The training process lasts for 50 epochs, but early stopping is applied if no improvement is observed on the validation set for more than 5 epochs. A decreasing learning rate is employed to improve training, being multiplied by 0.2 if no lower error is achieved after 2 epochs.

The networks are trained in two acoustic room conditions: (i) using anechoic signals only for training; (ii) multiconditional training, in which the networks are trained using data from all the reverberant rooms apart from the one used for test.

The evaluation results are reported based on chunks. Each chunk is 250 ms long (25 frames). The prediction made for each frame in a chunk is averaged to report a single azimuth location for the chunk. Chunk-based evaluation is adopted in order to avoid the issue that a speech signal typically includes short pauses where there is no directional sound source. The accuracy of the models is finally measured in terms of RMSE given in degrees.

Baseline system

The baseline system is a state-of-the-art DNN-based localization system using GCC-PHAT features as inputs [65], as also tested in Section 6.1 and in Section 6.2. GCC-PHAT features are computed as the inverse transform of the

frequency domain cross-correlation of two audio signals captured by a microphone pair. The binaural signals sampled at 16 kHz are framed at 20 ms, with 10 ms overlap. Since a distance of 18 cm occurs between the two microphones, the first 37 values are selected from the inverse transform. Unit variance and zero mean normalization is then applied. The baseline network consists of an input layer, two hidden layers of 1024 units each and an output layer of 37 classes. Dropout equal to 0.5 is applied after the two hidden layers. Softmax is selected as the activation function for the output layer, while a sigmoid activation function is used for the hidden units. All the hyperparameters are optimised using the development dataset.

6.3.4 Main Results

Anechoic training

Table 6.10 shows results using systems trained in the anechoic condition. The best overall performance is achieved by the baseline GCC system. The proposed WaveLoc-GTF performs slightly worse compared to the baseline, while the localization errors for WaveLoc-CONV are considerably larger across all reverberant conditions.

Room	Anechoic	A	B	C	D
Baseline	0.1°	**2.6°**	**9.3°**	2.6°	**10.1°**
WaveLoc-GTF	**0°**	9.1°	10.7°	**1.6°**	10.5°
WaveLoc-CONV	**0°**	37.7°	41.8°	37.3°	44.4°

Table 6.10: localization RMSE results in degrees for the models trained in anechoic environment.

It appears that the WaveLoc-CONV system has a tendency for overfitting compared to the other two systems. Fig. 6.15 shows the log-power spectra of all the 64 kernels in the first convolutional layer in WaveLoc-CONV. It is clear that the kernels, when trained in the anechoic condition, act largely as a set of band pass filters, mostly enhancing the frequency bands between 300–600 Hz and between 2300–2800 Hz. It is widely known that binaural features such as ITDs are more reliable in the low frequency region below 1600 Hz while others such as ILDs become more robust in the high frequency region above 1600 Hz [82]. It is possible that the network extracts related binaural features which are most effective in these two bands for sound localization in the anechoic condition. Such behaviour, however, fails to generalise to unseen reverberant conditions as these frequency bands could become unreliable due to reverberation. The WaveLoc-GTF model, on the other hand, performs frequency analysis with the

gammatone filterbank layer which forces the system to exploit all frequency bands and thus extract the most effective localization features in each band.

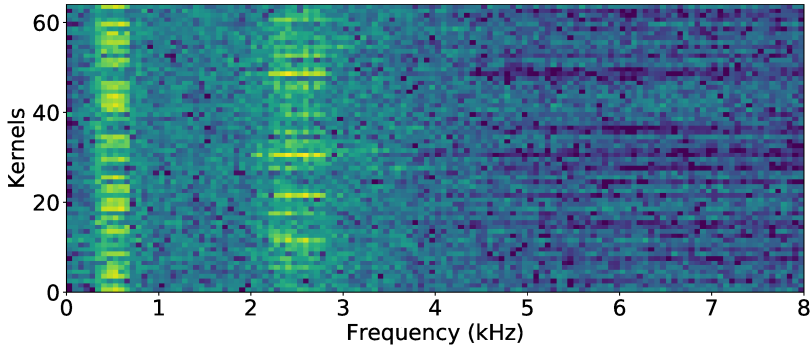


Figure 6.15: Log-power spectra of the kernels in the first convolutional layer of WaveLoc-CONV when trained in the anechoic environment.

Multiconditional training

It has been shown in the past that MCT can mitigate overfitting and increase the robustness of sound localization in reverberant conditions [81, 91]. This can be done by adding either diffuse noise or reverberation to the training signals. In this study, a reverberant training approach has been adopted as preliminary experiments showed it to be more effective. Specifically, the anechoic training dataset was supplemented with reverberant versions by convolving it with various BRIRs. The evaluation room is excluded for building the new training datasets, but for each room all the remaining three rooms are included for MCT.

Room	A	B	C	D
Baseline	2.7°	3.3°	3.1°	5.2°
WaveLoc-GTF	**1.5°**	3.0°	1.7°	3.5°
WaveLoc-CONV	1.7°	**2.3°**	**1.4°**	**2.4°**

Table 6.11: localization RMSE results in degrees using MCT.

Table 6.11 lists the results of all the models. The anechoic condition was excluded in this study, as all the models performs well even without MCT. All the models benefit from MCT, especially the proposed WaveLoc models. The best overall performance in reverberant conditions is achieved by the WaveLoc-CONV model, which has an average localization RMSE less than 3° compared to over 30° without MCT.

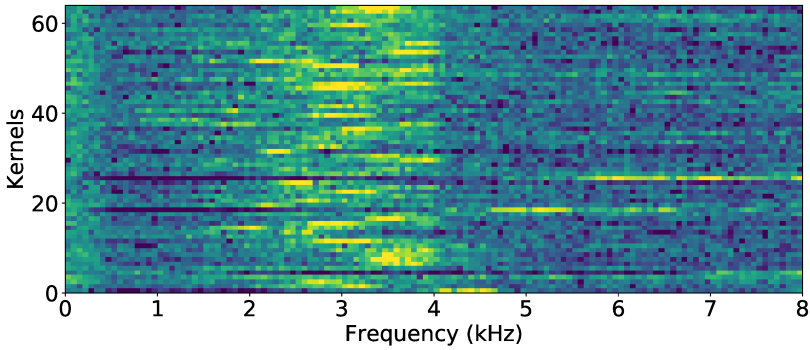


Figure 6.16: Log-power spectra of the kernels in the first convolutional layer of WaveLoc-CONV when trained using MCT for the model tested in room B.

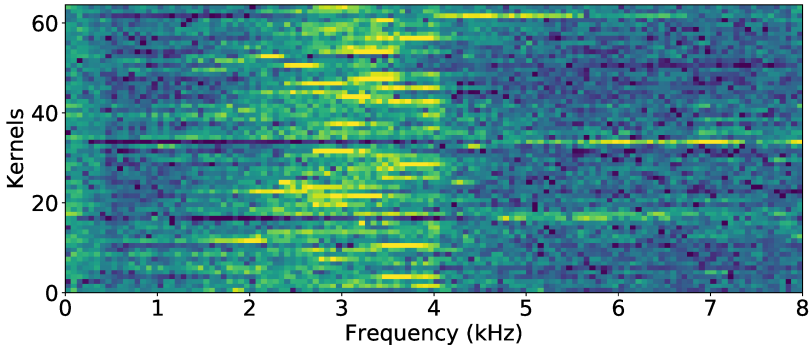


Figure 6.17: Log-power spectra of the kernels in the first convolutional layer of WaveLoc-CONV when trained using MCT for the model tested in room D.

Again, to investigate the effect of MCT on the convolutional kernels, the log-power spectra of all the 64 kernels in the first convolutional layers of the WaveLoc-CONV model are plotted. Plots for rooms B and D are shown in Fig. 6.16 and in Fig. 6.17, respectively; plots for the remaining two rooms are similar. It can be seen that the first convolutional layer is now composed of a set of distributed bandpass filters emphasising mainly the 1500-4000 Hz range, with some kernels stretching up to 6–7 kHz. The low frequencies below 1500 Hz are less exploited by the WaveLoc-CONV model. It is interesting to notice that the data-driven model learns to use more high frequency cues in a reverberant environment, which suggests ILD become more useful than ITD. It is reasonable to expect that the ITD is more affected by reverberation, while the ILD, created by the head shadowing effect mainly for frequencies

higher than 1600 Hz, is more robust to reverberation. Indeed, psychophysical cue-trading studies find that human listeners give ILD more weight than ITD when localising sounds in reverberant conditions [92].

Conclusions

This research describes a new approach for localising a sound source directly from the waveform, by proposing two novel end-to-end CNN systems. Machine localization systems typically employ hand-crafted features, such as the ITD and ILD, or GCC based features, as discussed in Section 6.1 and Section 6.2. Such explicit feature extraction may limit the model performance since it implies a lossy transformation of the input signals. Instead, the proposed end-to-end approach employs a cascade of convolutional layers to extract features directly from the waveform, that are suitable for localization in reverberant environments. When MCT is used across reverberant conditions, both end-to-end systems outperform a state-of-the-art DNN system using conventional features.

Two CNN-based systems are introduced. The first system, WaveLoc-GTF, is inspired by the auditory system and employs a convolutional layer that is largely based on a gammatone filterbank. The second system, WaveLoc-CONV, employs a data-driven approach, where a convolutional layer with trainable 1-D kernels is dedicated for frequency analysis. Although the gammatone filterbank is in some sense more 'principled', since it approximates the filtering characteristics of the human auditory system, it does not work as well as a system that is trained (i.e., finds its own filters) across a number of reverberation conditions. One reason for this is that the system may elect to emphasise frequency regions during training that provide more robust cues to localization.

Indeed, when MCT is used, the WaveLoc-CONV model is better able to exploit features in the high frequency regions above 2 kHz, which tend to be less corrupted by reverberation. This mirrors findings from human perception suggesting that ILD (which is primarily available at high frequencies) is more robust than ITD when reverberation is present.

Future work, not addressed in this thesis, will focus on improving the ability of end-to-end systems to generalise to unseen room conditions and multiple sources. Another possible direction is to combine sound identification with sound localization within an end-to-end system. Finally, conducting 'psychophysical' studies on trained networks will allow to fully understand their underlying mechanisms, e.g. by using the cue trading protocol described in [92].

6.4 Estimate Sound Source Elevation using Phase and Magnitude Spectra

The focus of this research is to develop a machine learning system for estimating the elevation of a sound source. For the same reasons addressed in Section 6.3, the multi-room environment considered for the task of localizing a speaker in Section 6.1 and Section 6.2 is no more taken into account. Alternatively, binaural localization is now addressed, with the purpose of investigating a novel approach independently from issues related to a multi-room environment.

6.4.1 Preliminaries and Problem Statement

Human beings determine both the azimuth of a sound source in the horizontal plane and its elevation in the vertical plane by using two binaural sensors [82]. Horizontal sound localization is largely based on binaural cues such as the ITD, or the related Interaural Phase Difference (IPD), and the ILD, which encode an azimuth location in terms of the difference between the left and right ears in both phase and magnitude [82, 83]. In addition, the human outer ear, together with the head, shoulders and torso, form direction-selective filters. Within this process, the two ears receive sound going through direction-specific frequency responses, which are referred to as spectral cues. These cues are responsible for vertical sound localization in the median plane (directly in front of and behind the listener), where binaural cues provide little information [82, 93, 94]. When a source is located away from the median plane, the binaural cues become more useful for perceiving its elevation, as different elevation angles will cause a disparity in the frequency responses of the left and right ears [36, 95]. Based on these principles, several machine systems for sound localization in the horizontal plane have been proposed [81, 96, 97]. Few works target localization in the vertical plane [36, 37], where the pursued strategy consists in concatenating binaural cues and monaural spectral cues in a single feature vector. Rodemann et al. [98] added binaural hearing to a robotic head in order to make use of spectral cues for elevation localization. They showed that by combining binaural cues (ITDs and ILDs) and spectral cues, localization accuracy improved in both azimuth and elevation. O'Dwyer et al. [38] used the CCF, which was previously used for azimuth localization [81], for elevation estimation using a DNN system. Their studies suggest that using the CCF can greatly reduce elevation estimation errors in reverberant environments when combined with spectral features. Their later study in [39] further improved the system by integrating MFCCs features.

Contribution

This research proposes a novel binaural machine system that robustly estimates the elevation of a speech source using a CNN framework. The approach here discussed differs in two important respects from previous studies. First, instead of using explicitly extracted IPDs and ILDs as features, the proposed system uses a convolutional layer with 2-D kernels that operate directly on the phase spectrum and the magnitude spectrum of the binaural ear signals. Such operations extract binaural features that are similar to IPDs and ILDs, but have better robustness, particularly in reverberant environments. Secondly, this approach combines monaural and binaural features with the same DNN architecture. The 2-D kernels also operate along the frequency dimension, and therefore combine binaural features with monaural spectral features. Features extracted from both the phase spectrum and the magnitude spectrum are concatenated and used as input to a DNN with fully connected layers in order to map them to the corresponding source elevation. Evaluation shows that the proposed system is able to accurately estimate the elevation of a speech source, even in challenging reverberant conditions, and substantially improves upon the performance of previous approaches.

6.4.2 Proposed Method

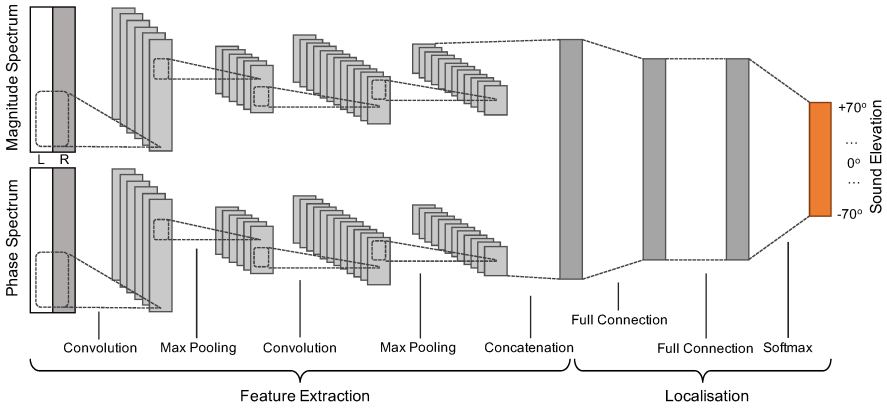


Figure 6.18: A convolutional neural network using phase and magnitude spectra for binaural sound localization. 'L' and 'R' represent the left and the right channels, respectively.

The proposed CNN system for binaural sound localization is illustrated in Fig. 6.18, which can be broadly divided into two stages. First, a feature extraction stage with a cascade of convolutional layers is designed to extract suitable

features for localization. Two feature pathways are considered: a *phase pathway* operating on the phase spectrum and a *magnitude pathway* operating on the magnitude spectrum. The extracted localization features are passed to the second stage for sound localization, which uses several fully connected layers to perform elevation localization as a classification task.

Frequency Analysis

The input to the CNN system includes both the phase spectrum and the magnitude spectrum. The binaural ear signals are framed using a 20 ms window with 10 ms overlap. At a 16 kHz sampling rate each frame contains 320 samples. A STFT with 512 points is applied to each frame after zero-padding with a Hamming window, and then the phase spectrum and the magnitude spectrum are extracted. The phase spectrum is wrapped to the range $[-\pi, \pi]$ and the magnitude spectrum is converted to log-magnitude in dB. Finally, the left and right channels are stacked together so that both phase and magnitude features are combined in a matrix of size 2×256, with the left and right channels indicated by 'L' and 'R' respectively in Fig. 6.18. The input phase is normalised to the range [-1, 1] and the magnitude is normalised to zero mean and unit variance.

Feature Extraction Layers

It is not clear how best to combine binaural cues and monaural spectral cues in a machine system for binaural sound localization. Most systems simply concatenate all the features as one feature vector to be used as input to a classifier, such as a DNN [38, 81]. In this study, a cascade of convolutional layers is applied to the phase spectrum and magnitude spectrum with the intention of extracting features that are closely related to both the binaural features (IPDs and ILDs) and the monaural spectral features.

The first convolutional layer consists of 32 2-D kernels. The size of each kernel is $[2 \times 9]$, where 2 corresponds to the binaural channels and 9 corresponds to 9 FFT bins ($\sim$281 Hz with a 16 kHz sampling rate). When applied to the phase spectrum, the 2-D kernels model the phase correlation not only between the left and the right channels, extracting features similar to the IPD, but also the spectral correlation across the frequency. Similarly, the magnitude pathway produces features similar to the ILD as well as spectral features.

Next, in each pathway, the correlation-based features are down-sampled by a $[1 \times 4]$ max pooling layer, which helps to reduce over-fitting and also reduces the computational cost. The resulting features are further elaborated by another convolutional layer with 32 1-D kernels of size $[1 \times 3]$, in order to identify specific patterns that are related to the localization task. The output features are again

down-sampled with a [1 × 2] max pooling layer. In both convolutional layers, the activation function is the ReLU.

Finally, the features produced by the convolutional layers from each pathway are flattened and the phase features and magnitude features are concatenated to form the input to the localization layers.

Localization Layers

The localization stage maps the correlation-based features to sound elevation angles using two fully connected hidden layers. Each of the layers has 512 hidden nodes, with the ReLU as the activation function and a dropout rate of 0.5. Finally, the output layer uses the softmax activation function. In this study the elevation angles range from -70° (below the head) to 70° (above the head) with a step of 10°. Thus the output layer consists of 15 neurons.

Training

The network is trained using the Adam optimiser with a learning rate of 5e-4 and a batch size of 128 samples. Categorical cross-entropy is used as the loss function. The training procedure involves 50 epochs in total, but early stopping is applied if no improvement is observed on the validation set for more than 5 epochs. A decreasing learning rate is also adopted with a decreasing factor of 0.5 when no loss reduction is achieved after 2 epochs. The learning rate is fixed once 1e-4 is reached.

6.4.3 Experimental Setup

Binaural Simulations

Binaural stimuli used in this study are created by convolving a speech database with a set of HRIRs measured on a KEMAR 45BC binaural mannequin from the SADIE database [68], previously addressed in Section 4.2.2. The database contains measurements spanning across many different azimuth and elevation locations, distributed in steps of 5° in the azimuth plane and 10° in the elevation plane. All the measurements are taken with an Equator D5 loudspeaker positioned 1.5 m from the centre of the KEMAR head.

As shown in Fig. 6.19, 15 elevation angles ranging between [-70°, +70°] in steps of 10° are selected in this study. For each elevation angle, the simulation also includes 19 azimuth locations in the frontal hemifield ranging between [-90°, +90°] in steps of 10°. Therefore the simulation contains a total of 285 source locations.

Speech sentences from the TIMIT database [70], discussed in Section 4.2.3, are used for simulating the binaural signals. 30 speech sentences are randomly

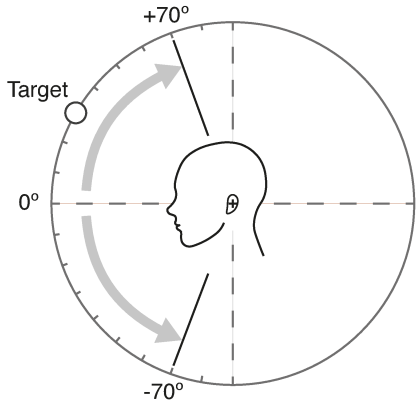


Figure 6.19: Schematic diagram of the virtual listener configuration. Target source positions were limited to the elevation range [-70°, +70°].

selected for each of the 285 source locations totalling 8,550 sentences. Since the TIMIT sentences are sampled at 16 kHz, the SADIE HRIRs are resampled to 16 kHz first. Each TIMIT sentence is then convolved with a pair of HRIRs from a source location to simulate spatialised binaural signals.

The training set includes 285 sentences per elevation angle. The networks are only trained using the anechoic dataset. The validation set includes 95 sentences per elevation angle, while the evaluation test set includes 190 sentences per elevation angle. To avoid the effect of silence that occurs at the beginning and the end of a TIMIT sentence, only the central 1-s segment of each sentence is used for evaluation [39, 81]. Care is taken to make sure there is no overlap between the datasets, and the number of sentences is evenly distributed across all azimuth and elevation locations.

Evaluation Dataset

While the networks are trained on just HRIR data, the evaluation dataset also includes RIR data to simulate the effect of reverberation. Following [38], binaural signals from the HRIR evaluation set are further convolved with four sets of RIRs from the OpenAIR library [69], of which details are discussed in Section 4.2.2. The details of these four RIRs are given in Table 6.12. Therefore the evaluation dataset includes in total five room conditions. To investigate the effect of room reverberation, each room condition uses the same 190 TIMIT sentences per elevation angle.

Label	Description	T_{60}
Room A	Domestic living room	0.2 s
Room B	A church built in the 11th century in Italy	0.53 s
Room C	A disused mine in the UK	0.71 s
Room D	A church built in the 16th century in Italy	1.16 s

Table 6.12: Details of the four reverberant room environments from the OpenAIR library [69] used in this study.

Experimental Setup

The proposed system exploiting both phase and magnitude spectra is referred to as *PHASE-MAG*. To investigate the separate contributions of the phase and the magnitude features, the proposed system is also altered so that only one feature pathway was exploited. They are referred to as *PHASE* and *MAG*, respectively.

The localization performance is measured by comparing reference source elevation angle with the estimated elevation angle using various chunk sizes. A chunk consists of a number of frames (10 ms frame rate) and the network output is averaged across a chunk to predict one elevation angle per chunk, following the method used in [81]. Two chunk size values were used for evaluation: 10 ms (single frame) and 50 ms (5 frames).

The results were reported using two metrics. The *elevation prediction accuracy* was measured by counting the number of chunks for which the predicted elevation angle matched the reference angle. The *Root Mean Square (RMS) errors* were reported in degrees by comparing the predicted and reference elevation angles.

Baseline Systems

A GCC-PHAT system is developed as a baseline. The GCC-PHAT algorithm is a popular cross-correlation based method for estimating the TDOA with a phase amplitude transform. The GCC-PHAT features are computed based on cross-correlations of binaural signals for lags between ± 1.1 ms. With the 16 kHz sampling rate this produced a 37-D feature vector for each frame. The GCC features are standardised by removing the mean and scaling to unit variance, and directly used as input to a DNN system with three fully connected hidden layers. Each hidden layers has 512 nodes with ReLU as the activation function and a dropout rate of 0.5. The learning rate is 1e-3. Otherwise the topology of the DNN is identical to the localization layers described in Section 6.4.2 (also in Fig. 6.18). This system is referred to as *GCC-PHAT*.

In addition, the evaluation setup adpoted here is broadly in line with the

one used by O'Dwyer et al. [39]. Thus the results from their best-performing system, a DNN using MFCCs and CCF features, are directly taken for comparison. Note that the CCF features are similar to the GCC-PHAT features, and the main difference is the use of MFCCs in their system. This system is referred to as *MFCC-CCF*.

6.4.4 Main Results

Model	Anechoic		Room A		Room B		Room C		Room D	
	10 ms	50 ms	10 ms	50 ms	10 ms	50 ms	10 ms	50 ms	10 ms	50 ms
MFCC-CCF [39]†	96.9%	N/A	94.4%	N/A	79.4%	N/A	62.5%	N/A	N/A	N/A
GCC-PHAT	60.9%	78.4%	44.6%	58.1%	49.4%	61.8%	54.3%	68.6%	40.7%	52.6%
PHASE	93.4%	98.9%	82.2%	92.5%	88.9%	96.8%	91.6%	97.8%	83.5%	93.9%
MAG	96.5%	99.8%	90.7%	97.9%	93.8%	99.0%	95.6%	99.5%	91.4%	98.3%
PHASE-MAG	**99.0%**	**100%**	**95.3%**	**100%**	**97.6%**	**100%**	**98.6%**	**100%**	**96.6%**	**99.8%**

† O'Dwyer et al. [39] reported results with a ±5° accuracy threshold.

Table 6.13: Elevation localization accuracy in different reverberant conditions, evaluated using 10 ms and 50 ms chunks.

Model	Anechoic		Room A		Room B		Room C		Room D	
	10 ms	50 ms	10 ms	50 ms	10 ms	50 ms	10 ms	50 ms	10 ms	50 ms
MFCC-CCF [39]	**1.59°**	N/A	**2.03°**	N/A	10.07°	N/A	12.97°	N/A	N/A	N/A
GCC-PHAT	37.74°	25.96°	45.88°	37.65°	42.05°	33.82°	40.08°	31.10°	47.05°	39.09°
PHASE	9.97°	2.21°	17.31°	10.08°	12.65°	5.62°	11.23°	4.18°	15.76°	7.83°
MAG	3.62°	0.29°	7.20°	2.37°	4.90°	1.02°	4.03°	0.61°	5.80°	1.50°
PHASE-MAG	1.78°	**0°**	4.53°	**0.02°**	**2.69°**	**0.02°**	**1.95°**	**0°**	**3.07°**	**0.07°**

Table 6.14: Elevation RMS errors in degree in different reverberant conditions, evaluated using 10 ms and 50 ms chunks.

Tables 6.13 and 6.14 show the elevation estimation accuracy rates and the RMS errors, respectively, with 10 ms and 50 ms chunk sizes. Among all the models evaluated, GCC-PHAT is the worst performing system. Using 10 ms chunks, the estimation accuracy decreases from 60.9% in the anechoic condition to 40.7% in Room D which is a reverberant church. The performance degradation in RMS errors is similar. This is not surprising, given that GCC is designed to mainly measure the ITD. However, since the evaluation dataset includes source locations spanning the frontal sphere, off the median plane the ITD will also change across elevation. Furthermore, as the entire GCC function is used as input, the DNN is able to learn systematic changes in the GCC with source elevation.

The MFCC-CCF results from O'Dwyer et al. [39] suggest that the use of MFCCss is beneficial for elevation localization as they provide spectral cues as well as the ILD information (via the use of the energy term in MFCCss) that are not available in the GCC features. However, the model was not very robust

in reverberant conditions, with only a 62.5% accuracy in Room C, even though the reported results are computed using a $\pm 5^\circ$ accuracy threshold.

The proposed PHASE-MAG model demonstrates high robustness to reverberation with a localization accuracy above 95% in all the room conditions using 10 ms chunks, and close to 100% using 50 ms chunks. When using just the phase features or magnitude features, there is a substantial performance drop across all conditions. This suggests that the two features provide synergistic information for elevation localization that is exploited in the joint CNN. Comparing the PHASE model and the MAG model, it can be seen that the magnitude spectrum provides more discriminative features for elevation estimation than the phase features, as the standalone MAG model performs significantly better than the PHASE model across all room conditions.

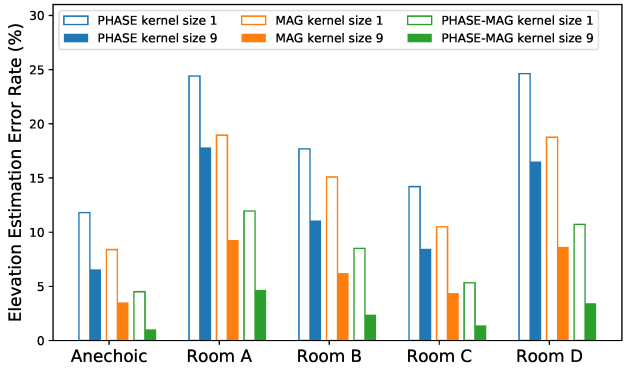


Figure 6.20: Elevation estimation error rates (10 ms chunk size) comparing various CNN models with different kernel sizes in the first layer.

It is interesting to note the relatively good performance produced by the standalone PHASE model. This is apparently due to the proposed network architecture, which is able to extract spectral cues from both the convolutional layers and the dense layers. To investigate this further, the size of the 2-D kernels in the first convolutional layer is reduced from $[2 \times 9]$ to $[2 \times 1]$ and thus the layer can only extract phase correlation between the left and right channels without correlations across frequency. The estimation error rates using 10 ms chunks are shown in Fig. 6.20, which also includes results of other CNN systems with the same modification. It can be seen that with a kernel size 1 along the frequency dimension, all the systems (the white bars) produce significantly higher error rates. Although the subsequent layers are still able to capture spectral cues to some extent, they do so less effectively.

Conclusions

This research proposes a novel binaural system that robustly estimates the elevation of a sound source. Instead of using explicitly extracted features such as IPDs and ITDs, the system exploits a CNN with 2-D kernels to extract features directly from the phase and magnitude spectra. Such operations allow binaural information and monaural spectral information to be combined effectively. Computer simulations show that neither the phase nor magnitude spectrum alone provide a robust basis for identifying the elevation of a sound source under reverberant conditions; combining the two is necessary. By doing so, the performance of the proposed system substantially improves upon that of previous approaches.

Future directions include full 3-D localization in both azimuth and elevation. In addition, further studies will address the integration of sound localization and sound identification within a common CNN architecture, similarly to the studies conducted in the next Section 7.1 and Section 7.2. However, no studies have been addressed in this direction within this thesis work.

Chapter 7

Integrating Voice Activity Detection and Speaker Localization

The research conducted here aims to the development of a unique DNN-based framework capable of simultaneous VAD and SLOC. The idea driving these works concerns the possibility of virtuously exploiting localization and detection related features to increase the overall performance of the system.

Experiments deal with the multi-room environment previously addressed in Chapter 5 and then in Section 6.1 and Section 6.2. The novel framework proposed here relies on the previous research conducted in terms of VAD and SLOC against the DIRHA environment. For this reason, several winning strategies of the previous models are adopted, such as the features considered as input, the typology of the DNN, the exploitation of a temporal context, and so forth. On the other hand, features such as the signal spectrogram and the raw waveform, plus the related DNN architectures employed for their processing, discussed in Section 6.3 and Section 6.4, are not considered. Indeed, these strategies have never been addressed in this thesis for a multi-room environment and for the purpose of VAD, hence an already established approach is preferred.

7.1 Joint VAD - Preliminary Model

The study here conducted addresses the development of a new model relying on multiple input and outputs, for the purpose of joint detection and localization of a speaker. Nevertheless, due to the novelty of the model and the particularity of a multi-room environment, a specific strategy has been adopted for properly evaluating the proposed method. In the first instance, only one framework [3] is present in literature for joint VAD and SLOC, but due to its complexity and due to the fact that it relies on a specific testing strategy not really suitable for DNN-based models, it is not considered here. This comparative model will be then dealt with in Section 7.2. Furthermore, it is in the interest of this research to compare the proposed method to a baseline data-driven model, in order to achieve the fairest comparison as possible. For this purpose, two CNN-based

models are developed for comparison to act as VAD and SLOC, being the result of the research discussed in Section 5.2 and Section 6.2, respectively. Indeed, following this strategy, it is possible to train and test all the discussed models within the same experimental set up. In addition, being the CNN architectures similar, the dependency of the results from this factor is extremely reduced.

7.1.1 Preliminaries and Problem Statement

The focus in this work is on VAD and SLOC. Different techniques have been proposed in the literature to tackle VAD problem in indoor environments. Among the most recent ones, an approach recognizing a reference *anchor* word with the help of mean subtraction is discussed in [99], the interaction between VADs based on the SNR estimate is investigated in [100]. DNNs have been employed in [14] and in Section 5.1. Further advancements have been proposed by means of CNNs in Section 5.2. At the same time, several approaches have been proposed for localizing a speaker in closed environments. In particular, the SLOC problem has been recently faced by means of neural networks in [33, 101], especially with a focus on CNN [28, 29, 35]. Similarly, Section 6.1 and Section 6.2 proposes a SLOC system based on MLP and CNN, respectively.

In the last years, some works focused on systems simultaneously addressing the speech detection and the speaker localization tasks. A common approach consists in grouping VAD and SLOC considering a cascade [3, 40, 41, 42] or a parallel [102] configuration. Up to the writers' knowledge, only two contributions investigate the cooperation between VAD and SLOC. One is the approach proposed in [3], in which an ensemble integration of speaker localization and statistical speech detection data in domestic environments is implemented. The second technique jointly performs VAD and SLOC [4] by employing a modified version of SRP-PHAT algorithm.

Contribution

Although the effort spent for developing models for joint speaker detection and localization, a single data-driven model has never been investigated within this context. Therefore, this research is intended to simultaneously exploit both VAD and SLOC data in order to improve the overall performance, both in terms of speech detection and speaker localization. DNNs are employed on purpose, for two main reasons. First, DNN have already shown remarkable performance on the two separate tasks, as mentioned above. Second, a neural architecture with its multiple inputs and outputs allows to easily make use of VAD and SLOC feature data and decision variable values.

In details, here is proposed a new model based on CNN, simultaneously operating as detector and localizer exploiting standard VAD and SLOC features.

The training of this network is performed by using both speech and non-speech signals, raising the issue of performing localization even for non-speech frames. The model is tested against a comparative framework relying on a classic cascade configuration, where a neural SLOC trained by means of an Oracle VAD is cascaded to a neural VAD, and speaker localization errors are considered only in correspondence of correctly detected speech frames (true positive).

For the proposed study, the multi-room scenario already addressed in Chapter 5 and in Chapter 6 is taken as reference, in order to have a solid experimental background for evaluating the proposed approach.

7.1.2 Proposed Method

The proposed model named *Joint VAD* is discussed in this section. Although it is capable of detecting and localizing the speaker at the same time, it is employed only for VAD. After that, the comparative model is described. It consists in the cascade of the so-called *Neural VAD* and *Neural SLOC*, which are the results of the research conducted in Section 5.2 and Section 6.2, respectively.

Joint VAD Model

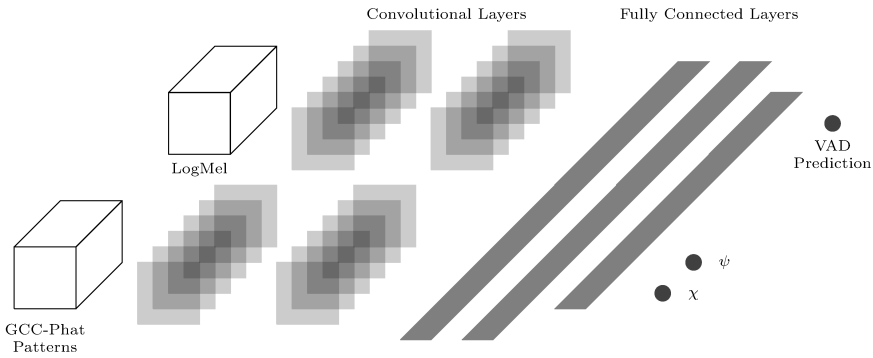


Figure 7.1: The Convolutional Neural Network employed for the Joint VAD Model. Pooling layers are absent. The outputs of the network are three neurons, one for speech detection and the other two for speaker localization.

In this neural model, the simultaneous detection of speech frames and localization of speaker position is performed. As discussed in Section 7.1.1, the objective is to exploit the synergy between these two tasks to improve their performance, and a fully data-driven technique was identified as the most viable solution to implement the idea. Several options have been investigated, and the most performing one is the model depicted in Fig. 7.1. It consists in a

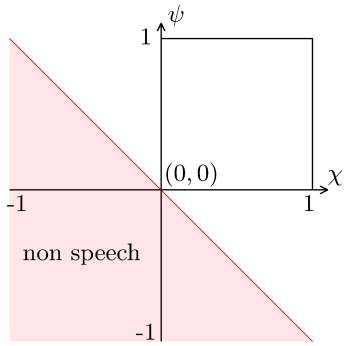


Figure 7.2: The application of the 2-D threshold. The square box is the room, in which speech is expected to be predicted. The thin red line is the threshold. If prediction lies in the red region then it is labeled as non speech, otherwise it is considered as speech.

single CNN with two standalone stacks of convolutional layers separately processing LogMel and GCC-PHAT Patterns features, being then concatenated and finally followed by a common set of standard feed-forward layers. The network ends with three outputs, being the voice activity prediction and the two speaker position coordinates.

A specific strategy has been adopted for labelling the network outputs. Indeed, a 0 or 1 label is used for speech/non-speech classification. On the other hand, localization is performed over speech frames in a 2-D plane, where the room coordinates (χ, ψ) are given in the $[0, 1]$ range. As a consequence of that, the non speech frames lack of a label for the coordinates outputs. Hence, the two (χ, ψ) coordinates are labelled as -1 for non-speech frames. This solution is basically labelling the speaker position in the non speech condition as a physical location outside the considered room, as similarly done in [4].

The result of this labelling method is that also the two coordinates are eligible to represent the speech/ non-speech condition. A specific threshold needs to be used on purpose, being the straight line depicted in Fig. 7.2 in a 2-D plane. In details, speech detection is performed by means of the room coordinates, to whom this threshold is applied, while VAD predictions are rejected.

Furthermore, this labelling strategy forces the model to range in $[-1, 1]$. Hence, a specific activation function must be selected for the network output, for its neuronal dense layers and the convolutional layers. For this purpose, the *Hard Tanh* nonlinearity has been chosen, which acts as $f(x) = x$ in $[-1, 1]$ and saturates to -1 and 1 out of this range, being previously discussed in Section 3.3. In addition, the neural network training is dealt with as a regression problem, since labels are clearly not eligible for a classification-based training.

Last but not least, a smoothing technique is employed to tackle the variability of the speech prediction. In details, the *hangover* technique is applied, being described in Section 5.1.2, and with its counter set to 8.

Features

GCC-PHAT Patterns features, described in Section 4.1.2, are extracted by considering only adjacent microphone pairs. Plus, due to the spatial disposition of the microphones, the first 51 values of the cross correlation are selected. Signals are sampled at 16 kHz, while frame size and hop size are set to 30 ms and 10 ms respectively. Zero mean and unit variance normalization is applied.

LogMel features, also reported in Section 4.1.2, are extracted by using frame size equal to 25 ms and hop size 10 ms. Zero mean and unit variance is applied for normalizing this features.

An improvement of CNN performance has been observed in the previous research described in Chapter 5 by extending the processed input data including also past and future occurrences. The same approach has been used here as well. Two are the parameters to set in this case, i.e., *context* and *strides*. The first indicates the total number of frames considered as input instead of the single actual frame, where an equal number of past and future frames is selected. The latter recursively pilots the the frames selection.

An important consideration has to be made with respect to the feature arrangement discussed in Section 6.2.2. Indeed, in this research the third dimension of the input matrix is formed by the microphone pairs (GCC-PHAT Patterns) or by the single microphone (LogMel), while the 2-D matrix, where the true 2-D convolution process takes place, is composed by the features and the temporal context. On the other hand, in Section 6.2.2 the third dimension is formed by the temporal context, so that the true 2-D convolution takes place over the matrix composed by features and microphones. This new arrangement aims to weight more the temporal evolution of the signal instead of the employed multiple microphones.

Comparative Model: Cascade Configuration

Speech detection is performed by the Neural VAD. It consists in a CNN fed by LogMel features extracted from all the available microphones. Training and testing of Neural VAD is accomplished over speech and non speech data acquired by means of environmental microphones. A boolean label is employed for representing the speech / non-speech condition, allowing to train the network as a classification problem. $ReLU$ activation, described Section 3.3, is employed as non linearity within the model. Hangover technique with counter set to 8 is used for smoothing the network prediction.

The Neural SLOC performs the speaker localization task in terms of room coordinates (χ , ψ) and employs a CNN processing GCC-PHAT Patterns. An Oracle VAD selecting only speech frames is used during the training phase of the Neural SLOC, as in Section 6.1 and Section 6.2. Also the Neural SLOC is characterized by *ReLU* activation, however its training is dealt with as a regression problem. The coordinates predicted by the network go through smoothing by means of a moving average filter of window size equal to 5.

Again, input features are presented as discussed above, differently from Chapter 6. Hence, the third dimension of the 3-D matrix consists in the considered microphones, while the 2-D convolution takes place along with features and the temporal context.

Finally, in computer simulations, as discussed later on, the Neural SLOC has been tested using only speech frames detected by Oracle VAD and by Neural VAD, i.e., considering all the available speech frames in the dataset and the true positive predictions of the Neural VAD, respectively.

7.1.3 Experimental Setup

Simulations take place over the Simulated dataset of the DIRHA dataset, previously described in Section 4.2.1. The Real dataset has not been considered due to its short amount of speech, which may be insufficient for a proper training of the models. Simulations address two of the five rooms, which are the Living Room and the Kitchen. These rooms are chosen since most of the speech events occurs there, plus a higher number of microphones is available.

In details, no microphone selection stage is adopted in this research, hence a fixed set of microphones is considered. For the Neural VAD, all the available microphones are considered, from which LogMel features are evaluated. Regarding the Neural SLOC, GCC-PHAT Patterns are extracted from all the couples of adjacent microphones installed in the wall and the ceiling array (i.e., microphones pairs distancing 50 cm). The central microphones of the ceiling arrays (KA6, LA6) are excluded. Differently, the Joint VAD relies on LogMel extracted from a reduced set of microphones which are K1R, K2L, K3C, KA5 for the kitchen and L1C, L2R, L3L, L4R, LA5 for the living room, and GCC-PHAT Patterns extracted with the same strategy adopted for the Neural SLOC.

The metrics employed for evaluating VAD accuracy are the ones described in Section 5.1.3, being the FA, Del and SAD. On the other hand, SLOC performance are tested in terms of metrics defined in Section 6.1, being the RMSE and P_{cor}.

Neural Networks details

The CNN training is performed by using standard backpropagation with the *Adam* optimizer [55]; plus, early stopping and variable learning rate are employed. Details are reported in Table 7.1.

The CNN hyper-parameters optimization is executed by random search; a total of more than 30 neural architectures is investigated for each model. Context and strides have been chosen a priori, as follows: context is set to 15 in all cases, while strides is equal to 4 for Neural VAD, 5 for Neural SLOC and 3 for the Joint VAD-SLOC Model.

	Training Epochs	Early Stopping	Learning Rate
Neural VAD	30	10	1e−5
Neural SLOC	500	50	2.5e−4
Joint VAD Model	500	50	2.5e−4

Table 7.1: CNN Training Parameters

A 10-fold cross validation is employed for testing the models, with 8 folds used of training, one for validation and one for testing. Thus, the 80 scenes of the DIRHA Simulated dataset are grouped in 10 subsets of 8 scenes each. The scene selection procedure here employed aims to balance the amount of speech between the 10 subsets. This procedure is different from the one adopted in the research conducted in Chapter 5. In particular, the scene with the maximum amount of speech is selected and allocated into the first subset, then discarded. The next scene is selected in the same way, and allocated into the second subset, and so forth. The speech balancing operated by this data folds organization has shown to improve the training convergence behaviour of the neural models.

7.1.4 Main Results

The Joint VAD Model employs the same CNN topology for the two rooms. Two separated stacks of two convolutional layers process LogMel and GCC-PHAT Patterns, respectively. Each one of the four layer is composed by 64 kernels of size 5×5. Three fully-connected layers respectively with 1024, 1024, 256 units and Hard Tanh activation function follow the convolutional layers. The two coordinates are employed for speech detection by using the threshold depicted in Fig. 7.2; VAD prediction is rejected being less accurate. In Table 7.2 the performance of the Joint VAD Model are shown for detection and localization. The proposed method acts as a remarkable detector, reaching an average SAD of 3.5%., while an average P_{cor} of 66% is achieved by the model.

	Kitchen	Living Room	Average
SAD (%)	3.8	3.1	3.5
DEL (%)	4.5	3.9	4.2
FA (%)	3.1	2.4	2.8
RMS (mm)	601	657	629
P_{cor} (%)	64	68	66

Table 7.2: Results for the Joint VAD Model.

The CNNs employed in the two rooms for the Neural VAD are similar, counting two layers of 128 kernels sized 3×3, succeeded by two layers of neurons, being 1024, 1024 for the kitchen and 1024, 256 for the living room. ReLU is employed as activation. All results discussed in this section are obtained by choosing the best threshold in the different addressed case studies. In Table 7.3 the accuracy achieved by the Neural VAD is reported. As result, the Joint VAD outperforms the Neural VAD in terms of detection, with a 3.5% of SAD against the Neural VAD 5.2%.

	Kitchen	Living Room	Average
SAD (%)	5.6	4.8	5.2
DEL (%)	6.8	5.7	6.2
FA (%)	4.5	4.0	4.2

Table 7.3: Results of the Neural VAD applied on the two considered rooms of the dataset.

The Neural SLOC relies on a CNN consisting of 128 kernels sized 7×7 for the kitchen and 5×5 for the living room. For each room, the convolutional layers are followed by two layers composed of 1024, 256 units. Results are reported in Table 7.4. The Neural SLOC is tested on speech detected by an Oracle VAD or by Neural VAD. When the latter is employed, the Neural SLOC accuracy increases, since it is tested against a reduced set of speech (true positive), instead of all the available speech. This means that the Neural VAD fails in detecting frames in which the Neural SLOC is less accurate. Furthermore, a higher accuracy is achieved by the Neural SLOC compared to the Joint VAD, when the latter localizes the speaker on its localization coordinates. This behaviour may be caused by the fact that the Joint VAD is also trained on non speech data, while the Neural SLOC is trained in presence of an Oracle VAD.

Finally, a comparison for the average results of the three models is reported in Table 7.5. The most performing configuration is obtained using the Joint VAD Model as speech detector with the Neural SLOC in cascade. In terms of detection, comparing the Neural VAD and the Joint VAD-SLOC Model, SAD is decreased from 5.2% to 3.5% when the latter is employed, corresponding to

		Kitchen	Living Room	Average
Oracle VAD	RMS (mm)	332	359	345
	P_{cor} (%)	76	77	76
Neural VAD	RMS (mm)	317	337	327
	P_{cor} (%)	77	78	77

Table 7.4: Performance of the Neural SLOC. Its test takes place over all the speech of the dataset detected by the Oracle VAD, or against the speech detected by the Neural VAD reported in Table 7.3, i.e. only for true positive.

a relative reduction equal to 33%. In addition, a lower SAD means as well that a higher number of true positive (+3.1%) is detected by the Joint VAD-SLOC Model. Then, when assessing the localization accuracy of the Neural SLOC on speech frames detected by the Joint VAD Model (Table 7.5b), a P_{cor} relative improvement of +1.3% is observed against the Neural SLOC tested on speech frames detected by Neural VAD. The average RMS reduces from 329 mm to 318 mm, i.e., a relative reduction of 3.34%.

Nevertheless, in Table 7.4 it was previously observed that the accuracy of the Neural SLOC increases when less true positive are detected, i.e., the Neural VAD is employed instead of the Oracle VAD. Hence, when detection is performed by the Joint VAD Model rather than the Neural VAD, it is reasonable to expect a decay of localization performance. Interestingly, the opposite takes place. This result shows that the Neural VAD fails to detect a subset of speech which is straightforward to localize for the Neural SLOC. Conversely, the Joint VAD Model detects those speech frames, thus proving that the proposed model is able to cooperatively exploit detection and localization data.

Conclusions

The joint speech detection and speaker localization problem is addressed within this section of the thesis. The objective of the research is the development of a model relying on DNNs capable of cooperatively exploit VAD and SLOC data in order to improve the overall performance of the system. The proposed model, namely Joint VAD Model, consists in a 3 outputs CNN processing LogMel and GCC-PHAT Patterns features. The model training makes use of non-speech frames, which requires the inclusion of a new label representing the localization of absent speakers. Computer simulations have been performed by considering a multi-room acoustic scenario and the DIRHA dataset has been used on purpose. In terms of speech detection, the Joint VAD Model is compared with the original Neural VAD system, which is the result of the research conducted in Chapter 5. As result, a relative reduction of average SAD error equal to 33% is achieved.

Detection	Neural VAD	Joint VAD Model
SAD (%)	5.2	3.5
DEL (%)	6.2	4.2
FA (%)	4.2	2.8

(a)

Localization	Neural SLOC*	Joint VAD Model	Neural SLOC†
RMS (mm)	327	629	318
P_{cor} (%)	77	66	78

(b)

Table 7.5: Comparison of the two proposed models. The shown results are averaged between the two considered rooms. In (a) the comparison in terms of detection, (b) shows localization performance. Neural SLOC* means the localizer operating on the speech frames detected by Neural VAD, whereas Neural SLOC† operates on the speech frames detected by Joint VAD Model.

Furthermore, when the Neural SLOC is employed for localizing the speaker, a slight improvement is observed in terms of RMSE and P_{cor} when speech is detected by the Joint VAD instead of the Neural VAD. This result shows that the proposed system truly exploits localization data to improve detection accuracy.

Hence, future studies, discussed in the following Section 7.2, will target further advancement of the Joint VAD Model, by augmenting the available speech frames in the original dataset. Moreover, a fair comparison with the state-of-the-art algorithm for joint detection and localization of speech in a multi-room environment will be taken into account.

7.2 Joint VAD - Further Advancements

The research presented in this section proposes further advancements for the Joint VAD Model described in Section 7.1. Indeed, this model, which is capable of virtuously exploiting detection and localization features, is now tested against the state-of-the-art method for a multi-room environment. Furthermore, specific data augmentation techniques are taken into account for an accurate training of the model.

7.2.1 Preliminaries and Problem Statement

As already addressed in Section 7.1.1, the investigation of a unique framework performing VAD and SLOC in a multi-room scenario deserves particular interest. Indeed, results achieved for the proposed Joint VAD Model in Section 7.1 show that this model truly benefits from the employment of localization features in addition to detection ones. Nonetheless, the Joint VAD has been tested against the best models for VAD and SLOC discussed in this thesis, but it has not been tested against the state-of-the-art framework for detection and localization in a multi-room environment. Furthermore, the employment of the localization outputs for the detection purpose may be deceiving, reason why an alternative is proposed. In addition, the short amount of data employed for the model training in Section 7.1 may affect the accuracy of the model, hence further investigation are here conducted. Last but not least, although the Joint VAD benefits in terms of speech detection, even SLOC is investigated by proposing a new neural model.

Contribution

This work extends and completes the research addressed in Section 7.1, where a similar CNN-based framework for VAD and SLOC is taken into account. A baseline model [3] is here directly compared with the proposed method. In particular, the framework presented in [3] is the only other one present in literature performing joint VAD and SLOC in a multi-room environment. The baseline model relies on an ensemble of the state-of-the-art classical VAD and SLOC algorithms, plus an integration stage. In order to perform a fair comparison with this baseline method, the same simulation strategies are here adopted, which differs from the ones of Section 7.1.3.

In details, the models are tested against another version of the DIRHA dataset, where spoken utterances differ in language. Furthermore, the training and testing set are now obtained by dividing the dataset into two parts of equal length. As a consequence of that, the cross-validation testing strategy adopted in Section 7.1, in Section 5.1 and in Section 5.2 is no more suitable in

this research. In addition, simulations in [3] take place at a lower frame rate from what employed in the previous section and in experiments conducted in Section 5. Here the lowest of the two frame rate is adopted. As result, less data is presented during the model training with respect to the previous case studies. To tackle the training issues raising by the new simulation setting, a data augmentation technique is taken into account. Indeed, the version of the DIRHA dataset previously employed in Section 7.1 and in Chapter 5 initially extends the training data. After that, a novel data augmentation approach specific for the addressed tasks is presented and tested. The idea behind this strategy aims to develop an additional version of the DIRHA dataset, where new speech data is employed. Nevertheless, RIRs originally recorded within the DIRHA project are not publicly available, thus a hybrid approach is necessary. Hence the RIRs related to the rooms under study are virtually emulated by means of a RIRs generator tool, relying on the only knowledge of the room dimension and the placement of the microphones installations.

7.2.2 Proposed Method

The method proposed is depicted in Fig. 7.3. In details, VAD and SLOC tasks are performed by means of distinct algorithms disposed in a cascade configuration. This approach has been successfully discussed in Section 7.1. Indeed, speech activity is predicted by the Neural VAD algorithm elaborating data recorded in the room under observation. After that, detected speech is processed by a Neural SLOC in order to estimate the speaker position. A feature extraction stage precedes the VAD and SLOC models, leading to LogMel and GCC-PHAT Pattern features feeding the two neural networks, respectively. A simple post-processing technique is employed only for localization predictions, while no processing is applied to the VAD output, since the previous *hangover* technique could introduce new errors due to lower frame rate. The Joint VAD described in the next paragraph detects the activity of the speaker also using localization information, while the SLOC localizes the speaker over speech frames detected by VAD. Coherently with the previous research, localization is performed in terms of speaker coordinates, where the height of the speaker from the ground is not taken into account. Hence, considering the 2-D top view of a room, the speaker Cartesian coordinates will be referred as χ and ψ, which ranges in [0,1] due to normalization.

Voice Activity Detection

The Joint VAD model described in Section 7.1 is used for detecting a speaker. Indeed, it has previously shown the capability of this model of simultaneously exploiting localization and detection information in order to improve VAD ac-

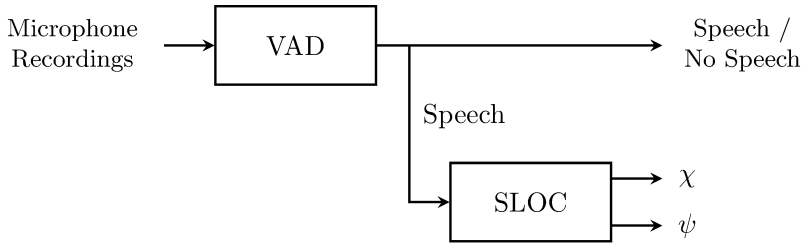


Figure 7.3: Conceptual scheme of the proposed method

curacy. The Joint VAD is depicted in Fig. 7.4. It consists in a CNN fed by LogMel and GCC-PHAT Patterns features, which detects and localizes the speaker at the same time. A dedicated stack of convolutional layers processes the two features sets, then a concatenation of the stack-dependent feature maps is performed. The result of the concatenation is then elaborated by a set of hidden neuron layers. The model ends with three outputs, where the first one predicts the speech presence, and the remaining two correspond to the speaker coordinates inside the room in a 2-D plane.

This model jointly acts as detector and localizer. The localization task is treated as a regression problem. The VAD dedicated output makes use of labels assuming 0 or 1 value. Differently, the two localization outputs are mapped in the [-1, 1] range. In details, when speech is present, the speaker is given in the range [0, 1] for both coordinates, while both labels are set to -1 in the case of speaker inactivity. Following this approach, both the detection output and the coordinates outputs are valid to predict the speaker activity. Indeed, in Section 7.1, VAD was performed by means of SLOC predictions through the threshold depicted in Fig. 7.2. Due to [-1, 1] range, *hard tanh* is employed as activation function, while *ReLU* activation is used for VAD output.

A temporal context extends the amount of data processed by the network frame-by-frame. With this procedure, previous and future frames are processed together with the actual frame, for a total of C frames, where C denotes the *context*. Consistently with the previous research, the selection of past and future frames is piloted by the integer value *strides*, although this value is here set equal to 1. In details, a 2-D matrix is obtained for each microphone for the actual frame, when the rows the features and the columns are the frames with context. Then the different microphones features are stacked, leading to a 3-D tensor. The model training is performed on speech and non speech data. Compared to the research conducted in Section 7.1, here no smoothing technique is applied to the VAD output. In addition, VAD is performed by using the VAD prediction output, differently from the previous case study, where SLOC predictions were employed for detecting the speaker activity. This

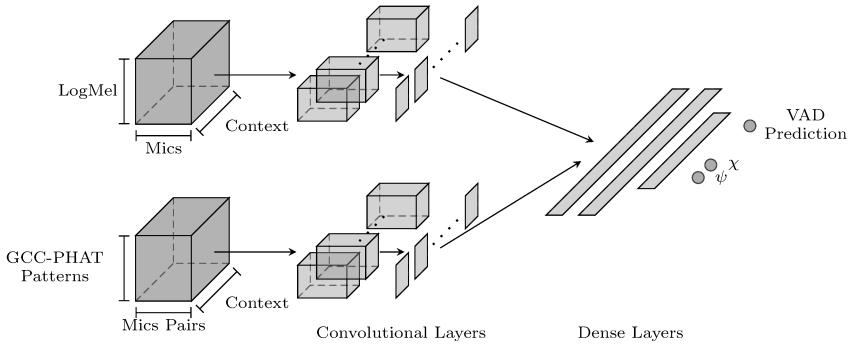


Figure 7.4: Architecture of the Joint VAD model. Two separated convolutional layers elaborate LogMel and GCC-PHAT Patterns features, respectively. Since the input matrices are 3-D, the first convolutional layer has 3-D kernels, which then become 2-D. A concatenation step joins feature maps extracted by the two convolutional stacks.

procedure has been adopted for avoiding the possible confusion between the employment of VAD or SLOC predictions for VAD purpose. Furthermore, due to the new splitting strategy for the considered dataset, less data is available for training the model. Hence, the possibility that insufficient speech data will be presented during the model training must be taken into account. As a consequence, it is reasonable to expect that this condition affects more the SLOC outputs than the VAD output, since the latter is just a binary label. For these reasons, SLOC outputs are not considered here for VAD purpose.

Speaker Localization

Localization is performed by two diverse models relying on CNNs. Both networks are trained on speech data by means of an oracle VAD, and their outputs are the room coordinates in the range [0,1]. The CNNs are trained through regression. $ReLU$ is selected as activation function.

The first model is the same discussed in Section 7.1, and it will be referred as SLOC_{SC}, which stays for Single-Channel SLOC. The GCC-PHAT Patterns feature are organized in a 3-D tensor. Further details of input tensor given to CNNs are discussed in Section 3.2. The second model differs from the previous one in terms of input features organization and elaboration. Indeed, a separated input is created for each microphone pairs. As result, a set of 2-D matrices is now presented to the network, where rows and columns of each matrix are the temporal context and the features. The CNN is then characterized by a number of inputs equal to the considered microphone pairs. The name adopted for the model is SLOC_{MC} (Multi-Channel SLOC). The main difference between the two models lies in how the first CNN layer processes the inputs. Indeed, in the

case of the SLOC_{SC}, the first CNN layer consists in a set of 3-D kernels. For each kernel a 2-D feature map is then computed, where a summation over the third dimension takes place. In details, this summation acts as a compression stage over the extracted microphone-dependent feature maps. Differently, for the SLOC_{MC}, the first CNN layer consists in 2-D kernels, which are trained over data coming from different microphones, and no feature maps compression is performed. The two models are depicted in Fig. 7.5 and in Fig. 7.6. Finally, the SLOC outputs are further processed by using a smoothing technique. In details a moving average filter of window size equal to 5 is applied to each predicted coordinated.

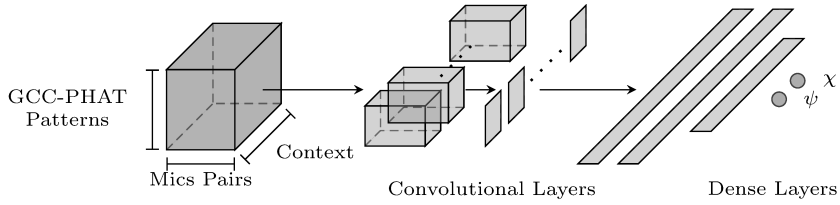


Figure 7.5: Architecture of Single-Channel SLOC

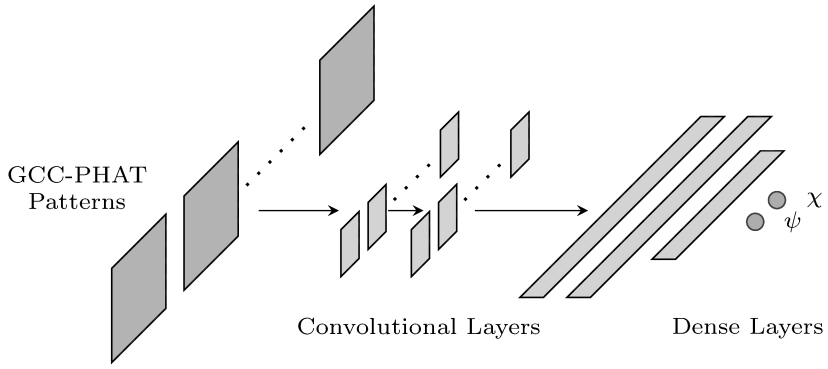


Figure 7.6: Architecture of Multi-Channel SLOC

Features Extraction

The proposed method makes use of two different features: LogMel and GCC-PHAT Patterns. The first are commonly employed for audio analysis, while the latter are specific for the localization task. Their reliability has already been assessed the previous research of this thesis discussed in Section 7.1. Their description is given in Section 4.1.2. In particular, LogMel features are extracted by applying a set of 40 Mel filters, while the signal is framed with frame size and hop size equal to 50 ms and 60 ms respectively. LogMel features go through zero

mean and unit variance normalization. On the other hand, for the GCC-PHAT Patterns only 51 values are selected from the inverse transform, since microphones pairs distancing 50 cm are considered for feature extraction. Frame size and hop size of 50 ms and 60 ms respectively are used in the features extraction stage. Finally, features are normalized in the range [0, 1].

7.2.3 Data Augmentation

Data augmentation techniques have been largely employed in the field of data-driven approaches. Indeed, in [103] it has been observed that the accuracy of a data-driven algorithm improves when extra-data is used for the model training, since chances of overfitting the model are reduced. Data augmentation was initially applied for image recognition purpose [104], where images already present in the dataset went through image processing techniques such as tilting or shifting, in order to generate new training material. The effectiveness of data augmentation has been recently assessed even in the audio field. In particular, in [105] data augmentation allows to achieve a higher accuracy when applied for the language recognition purpose. Similarly, several studies targeting speech recognition task benefit from this technique [106, 107] and for sound event detection as well [108, 109, 110].

Within this research, data augmentation targets voice detection and speaker localization purpose, although this technique has never been used for sound localization. Two main strategies are here employed. The first one relies on external data already recorded in the same conditions of the dataset under study [66], being also employed in the previous researches of this work. The other approach requires to build a new dataset from scratch. The proposed data augmentation technique generates virtual acoustic scenes using appropriate audio software and some parameters of the real scene. Further details are discussed in Section 7.2.5. As result, this second technique is suitable to be applied to different case studies, being independent from the dataset taken into account.

7.2.4 Baseline Method

A brief description of the baseline model proposed in [3] and employed here for comparison is reported in this section.

The baseline model consists in an ensemble of multiple VAD and SLOC algorithms, whose arrangement is depicted in Fig. 7.7. Indeed, in [3] two algorithms are considered for VAD, being Sohn's method and Switching Kalman Filter (SKF). Four SLOCs algorithms are evaluated, where three are derived from the CSP method, being 2D-CSP, multi-channel CSP and Template CSP, and the last SLOC algorithm is Steered Response Power (SRP-PHAT). A fi-

nal integration algorithm jointly processes VAD and SLOC predictions. Three methods are investigated: minimum cost criterion, SVM and a ANN-based classifier. A three stages selection strategy leads to the best configuration of this ensemble, which is the one reported in this work.

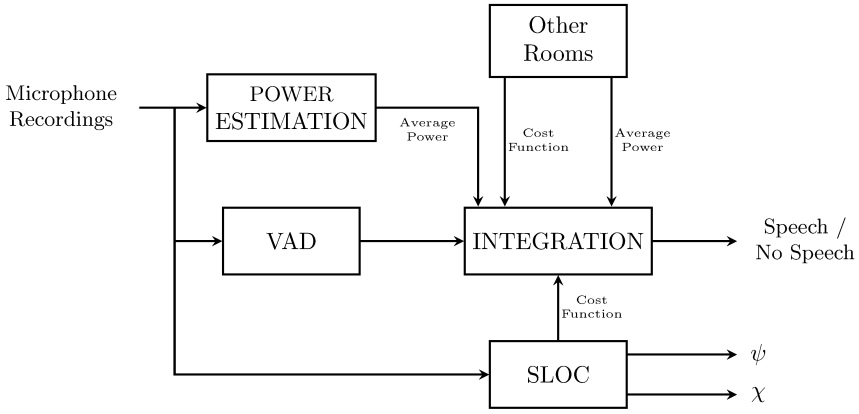


Figure 7.7: Conceptual scheme of the baseline SLOC

Voice Activity Detection

The first of the two detection algorithm is Sohn's method [7], which is based on conventional likelihood ratio test. This method assumes that the noise power spectra estimated in speech frames is conditionally independent from its observation in non-speech frames. Hence the statistical models of speech and noise are formulated, being characterized by their variance, respectively. For each frequency bin the log-likelihood ratio of the speech and noise models is computed, and the geometric mean is then computed. Finally, a threshold is applied to this mean in order to determine the class of the frame under observation.

The other detection algorithm is switching Kalman filter [111]. It relies on a prepared speech model and an on-line estimated noise model, from which the noisy speech model is finally build. This approach elaborates LogMel features by means of a GMM which is continuously updated by the Kalman filters. The model ends with a likelihood ratio, which allows to discriminate speech and noise.

Speaker Localization

A modified version of the original CSP method [63] is used for SLOC. Indeed, the original method assumes a plane sound wave, while in the reference work

the 2D-CSP is tested under the spherical wave assumption. This method estimates the TDOA between two adjacent microphones. For this purpose the frequency domain cross-correlation of the two signal is computed. From the cross-correlation the maximum of the inverse transform is taken. After that, for a speaker candidate point a cost function is defined, which consists in the difference between the theoretical and the estimated time delays related to each considered microphone pair. The speaker position is eventually given as the point minimizing this cost function.

In addition, in the baseline method [3] the multiple channel 2D-CSP (M-CSP) [112] is taken into account. This technique extends conventional CSP by using a correlation matrix of time difference of arrival.

The third SLOC algorithm discussed in [3] is the Template CSP, which is a modified version of the 2D-CSP. Indeed, since the theoretical TDOAs and the observed TDOAs differ due to reverberation present in the room under observation, the theoretical TDOAs are subjected to a correction. In details, a bias is added to the coordinates of each position, where the bias is estimated as the average difference between the theoretical and the observed TDOA in the development set.

The last algorithm used for SLOC is the SRP-PHAT [113]. This technique steers a delay-and-sum beamformer in the volume under observation. From that, an objective function is then computed, which depends on the frequency domain cross-correlation of the signals recorded by microphone pairs. Thus PHAT weighting procedure is applied. Finally, with SRC the area under observation is recursively reduced. The speaker position is finally estimated as the point maximizing the objective function.

Integration

The first integration algorithm relies on minimum cost criterion. It is applied when a speaker is detected in multiple rooms. Hence, the localization cost function is compared across the detected rooms, and the smallest one determines the room prediction. However, since the cost function depends on the room size, a tolerance parameter is introduced. This parameter associates a flag to the evaluated frame when the cost function is close to be the smallest between the detected rooms, instead of being the smallest in absolute. Finally, the utterance under study is rejected if the ratio of the flags over the total number of utterance frames is lower than a threshold.

The other two approaches are classifier-based requiring a training stage. In details, features from all the rooms are fed to the classifier, which predicts the probability of having speech only for the room under observation. This prediction is then flagged by means of a tolerance parameter, exactly as for the cost criterion. Similarly, a threshold is applied to the ratio of the recognized

frame over the total of utterance frames.

Two different classifiers are tested. Their input features are the speech powers averaged over microphones in each room and the localization function cost. The first classifier is a SVM, while the other one is a MLP consisting of two hidden layers of 15 and 10 units each. A standalone classifier is trained separately for each room.

Comparison with the Proposed Method

The baseline model is composed of a VAD, a SLOC and an integration stage. This model results to be complex for a couple of aspects. First of all, a dedicated manual tuning is required for each one of the VAD, SLOC and integration algorithm, which can be extremely time demanding. Furthemore, each room must be analysed before the single-room prediction. These issues are tackled by the proposed method. Indeed, a extensive tuning for each algorithm is not required and the other room predictions are not necessary when the model is applied to the room under study. In addition, the proposed method avoids a third integration stage.

7.2.5 Experimental Setup

Here the main aspects of the three considered dataset are discussed. Furthermore, the details of the tested neural networks are reported. The proposed models are tested by means of the metrics previously taken into account in Section 7.1.3. In particular, these metrics are the FA, Del and SAD for VAD, while RMSE and P_{cor} measure the localization accuracy.

DIRHA Dataset - HSCMA and EVALITA

Two different versions of the Simulated DIRHA dataset are taken into account in this research. They are briefly introduced in Section 4.2.1, however further details are here discussed. The EVALITA dataset contains 80 scenes of Italian spoken utterances. It has been previously employed in Chapter 5, in Chapter 6 and in Section 7.1, in which the experiments were carried out using the k-fold cross validation technique, where $k = 10$, so that 64-8-8 scenes compose the training-validation and test sets. On the contrary, the state-of-the-art classical approach [3] is tested over Simulated and Real subset of the HSCMA dataset. This dataset counts 80 folders equally divided in Italian, Greek, German and Portuguese languages. The Simulated HSCMA is split into two main subset of 40 folders each, which are the *Dev* and *Test*; the first is employed for training the model and the second for testing its performance. In order to compare with the state-of-the-art, the HSCMA is employed in this work for training and testing the models, where the same *Dev* and *Test* splits are taken into

account. In details, training and validation sets for CNN training are obtained from the *Dev* set, with a 90% and 10% split, respectively. Finally, EVALITA dataset is employed for data augmentation technique.

DIRHA-LibriSpeech

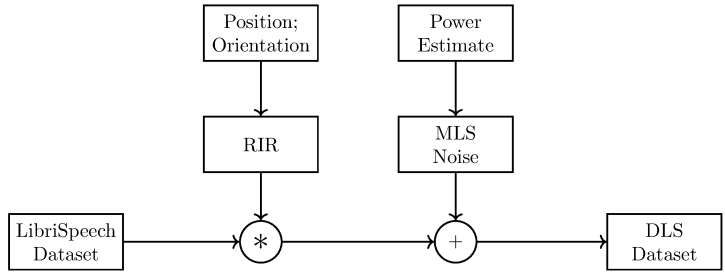


Figure 7.8: Block diagram of the algorithm used for the realization of the DLS dataset

The proposed method relies in a data augmentation stage as well, reason why an artificial dataset, being referred as DLS (DIRHA-LibriSpeech), is developed. Since the original RIRs recorded within the DIRHA project are not publicly available, a new set of RIRs must be generated consistently. For this purpose, a version of the Python Room Impulse Response generator [114] is employed, which relies on the image source model theory [115]. The artificial dataset aims to replicate the working condition of the DIRHA Simulated subset, where the speaker positions are fixed. The rooms under observation are the kitchen and the living room. In the first case, 17 positions are available for the speaker, where each position can assume 4 different orientations as shown in Fig. 7.10, in addition 13 microphones are installed in this room. As result, 884 RIRs are computed. For the living room, displayed in Fig. 7.9, the number of speaker positions and orientation is the same as the kitchen, however 15 microphones are installed, leading to 1020 RIRs.

Speech data employed for DLS is randomly selected from the Librispeech dataset [71]. Only the clean speech subset of Librispeech discussed in Section 4.2.3 is considered for this purpose. A desired SNR is then achieved by adding artificial noise created with Maximum Length Sequence (MLS) technique. MLS amplitude is calculated for each Librispeech utterance. As result, the same noise power characterizes each microphone. The block scheme of the DLS development is depicted in Fig. 7.8.

Important differences occur between DLS and DIRHA EVALITA or HSCMA. At first place, the latter is the result of the measured RIRs between the positions of the sources (using acoustic loudspeakers) and the microphones installations,

while the DLS is the result of the modeled RIRs. In addition, the language employed for the DLS is English, while EVALITA is in Italian and HSCMA has speech pronounced in Italian, Greek, Portuguese and German. Finally, the DIRHA project contains scenes and overlapping events coming from other rooms, while this option is avoided for the DLS.

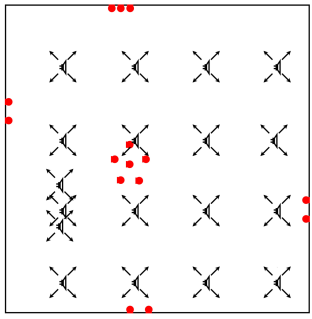


Figure 7.9: The living room designed through the data augmentation process.

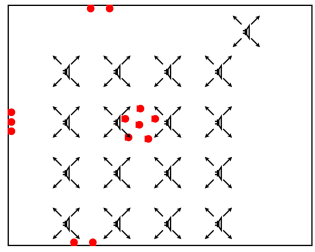


Figure 7.10: The kitchen designed through the data augmentation process.

Neural Networks Details

The GPU-based toolkit *Keras* [49] has been employed for developing and testing the DNN models. Training and testing of the proposed models rely on two different susbsets of the DIRHA Simulated dataset. In details, *Dev* subset is used for training the DNNs and for optimizing the hyper-parameters of the baseline model. Testing is executed over the *Test* subset. When data augmentation is used, the *Dev* training set is extended with new data, while *Test* is not varied.

Consistently with the research conducted in Section 7.1, no microphone selection is considered. Indeed, when LogMel features are extracted, all the available microphones are taken into account, being in total 13 and 15 for the kitchen and the living room, respectively. GCC-PHAT Patterns are extracted from adjacent microphone pairs. In details, with regards with the ceiling array, all the possible combinations have been considered, for a total of 15. Hence, a total of 19 and 20 microphone pairs is selected for the kitchen and the living room, respectively.

The DNN optimization strategy here adopted relies on two stages. In the first stage, the neural network architecture goes under investigation by means of a random search technique; after that data augmentation takes place with the most performing model resulting from the previous stage.

Training of the neural networks is performed by means of *Adam* optimizer, of

which decay parameter for momentum estimates are $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The number of training epochs is set to 500, while a batch size of 200 frames is employed. Neural network weights are initialized with a gaussian distribution with $\mu = 0$ and $\sigma = 0.1$. In addition, strides are used in convolutional kernels, in order to let the neural network process the equivalent of a larger audio excerpt. Please note that these strides are different from the *strides* defined along with the temporal context. Furthermore, convolutional kernels go through two regularizers, which take care of the activity of a kernel and the weights decay. Their coefficients *L1* and *L2* are both set to 1e−4. Two different learning rates pilot the training of the Joint VAD and the SLOC, being respectively 5e−5 and 1e−4. Finally, overfitting is prevented by applying early stopping after 5 epochs without improvement on the validation loss. Variable learning rate allows a finer tuning of the models, by decreasing of a factor scale 0.5 after 2 epochs without improvements. The *context* value is set to 15 for both models, since this parameter has been deeply investigated in the previous research of this thesis, especially in Section 6.2. Dropout equal to 0.5 is applied after each hidden layer. The investigated hyper-parameters by random search are reported in Table 7.6.

		Joint VAD	SLOC
Convolutional Layers	Number of Layers	1, 2	1, 2
	Number of Kernels	64, 128	64, 128, 256
	Kernel Size	[3, 3], [4, 4], [5, 5]	[3, 3], [4, 4], [5, 5]
	Kernel Strides	1, 2, 3, 4, 5	1, 2, 3, 4, 5
Hidden Layers	Number	1, 2, 3, 4	1, 2, 3, 4, 5, 6, 7
	Neurons	256, 512, 1024, 2048	512, 1024, 2048

Table 7.6: Hyper-parameters of the DNN models, investigated through random search in the first optimization stage.

7.2.6 Main Results

Joint VAD

The best results achieved by the Joint VAD model during the two optimization stages are reported in Table 7.7. Initially, the CNN architectures have been investigated by means of random search. The best model results to be a CNN having one convolutional layer counting 64 kernels of size [5, 5] and strides 4 for each branch, followed by 3 hidden layers with 1024, 1024 and 256 units,

respectively. With this configuration, trained on the *Dev* subset, an average SAD equal to 11% is achieved over the *Test* subset. After that, the second optimization stage takes place, where the training set contains also augmented data. When data augmentation is performed, the symbol † is appended to the name of the considered model. As expected, after the optimization performed by means of data augmentation, a higher accuracy of 5.8% SAD is achieved by the data-driven model.

		Kitchen	Living Room	Average
Joint VAD	SAD (%)	10.1	12.4	11.0
	DEL (%)	16.4	20.4	18.4
	FA (%)	4.7	2.7	3.7
Joint VAD†	SAD (%)	6.3	5.3	5.8
	DEL (%)	11.3	9.1	10.2
	FA (%)	1.3	1.5	1.4

Table 7.7: Achieved results for the Joint VAD on the *test* set. The first main line highlights the first optimization stage, where neural networks hyper-parameters are investigated. The latter shows the result of data augmentation, denoted with †.

As already addressed in the previous research, even the SLOC outputs of the Joint VAD are eligible for detecting and localizing a speaker. This procedure requires the application of a particular threshold, which corresponds to a oblique line in the 2-D plane of the room, as in Section 7.1.1. When this configuration is considered, the model will be referred as Joint SLOC instead of Joint VAD. Results achieved by means of the Joint SLOC are then reported in Table 7.8, where the same neural architecture of the Joint VAD employed in Table 7.7 is taken into account. This study is motivated by the interest in observing the dependency of the Joint SLOC from the training material.

As result, when data augmentation is employed, fairly good results are observed in terms of detection and localization by means of the Joint SLOC†. On the other hand, when few data is employed for training, the Joint SLOC truly fails in terms of VAD and SLOC, since the model is unable to fit. For this reason, a dedicated SLOC model trained on speech data and localization features is employed for SLOC task, being addressed in the next section. This result even confirms the hypothesis discussed in Section 7.2.2 for what concerns the employment of SLOC outputs instead of VAD output for VAD purpose.

SLOC

To test the CNN-based SLOCs two strategies are here adopted: the first one couples the SLOC with an Oracle VAD, reported in Table 7.9, while the latter tests the SLOC over true positive frames detected by the Joint VAD†, shown

		Kitchen	Living Room	Average
Joint SLOC	SAD (%)	49.1	49.5	49.3
	DEL (%)	90.8	92.4	91.6
	FA (%)	7.5	6.6	7.1
	P_{cor} (%)	2.9	2.2	2.6
	RMS (mm)	3518	4384	3952
Joint SLOC†	SAD (%)	8.1	5.8	6.9
	DEL (%)	15.4	10.5	12.9
	FA (%)	0.7	1.1	0.9
	P_{cor} (%)	72.2	69.3	70.7
	RMS (mm)	759	790	774

Table 7.8: Here are reported the Joint SLOC performance, in other words when the SLOC coordinates are used for detection and localization instead of the VAD output. Data augmentation † heavily influences the behaviour of these two outputs.

in Table 7.10. These two strategies are generally consistent in terms of performance, however the first assesses the localization algorithm in an absolute sense, while the second one considers the dependency from the VAD algorithm.

The first optimization stage, where CNN parameters are varied, leads to the best result of 610 mm RMS achieved by SLOC_{MC} tested with an Oracle VAD. In details, this model has only one convolutional layer counting 64 kernels sized 5×5, making use of strides value equal to 4. After that 4 hidden layers of 1024 neurons each end the network. Similarly, SLOC_{SC} achieves an almost equal RMS. It is composed of a single convolutional layer where a total of 256 kernels of size 4×4 with strides equal to 3 is employed. The convolutional layer is then followed by 4 neuronal dense layers counting 1024 units each. Subsequently, the two best models are trained by using the augmented training set. This step is denoted with †. As result, the $\text{SLOC}_{\text{SC}}{}^{\dagger}$ achieves an RMS of 405 mm, while a slightly worse performance of 478 mm characterizes the $\text{SLOC}_{\text{MC}}{}^{\dagger}$.

When tested in the presence of the Joint VAD† as reported in Table 7.10, the $\text{SLOC}_{\text{SC}}{}^{\dagger}$ and the $\text{SLOC}_{\text{MC}}{}^{\dagger}$ achieve 312 mm and 394 mm of RMS, respectively. A maximum P_{cor} of 90.6% distinguishes the $\text{SLOC}_{\text{SC}}{}^{\dagger}$. In addition, comparing these localization results with one achieved by the Joint VAD reported in Table 7.8, it is possible to state that the Joint VAD results to be less accurate in the SLOC task compared to a SLOC trained on speech data, as already discussed in Section 7.1.

SLOC performance increases in the presence of a real VAD. Indeed, since true positives consist in a subset of all the speech data present in the test set, it is possible to state that a real VAD fails to detect speech being more difficult to localize. Furthermore, although the main idea behind the SLOC_{MC} is to provide to the CNN a better capability of generalizing compared to the SLOC_{SC},

Oracle VAD		Kitchen	Living Room	Average
$SLOC_{SC}$	RMS (mm)	687	586	618
	P_{cor} (%)	57.1	65.3	62.4
$SLOC_{SC}^{\dagger}$	RMS (mm)	394	416	**405**
	P_{cor} (%)	86.0	87.9	86.9
$SLOC_{MC}$	RMS (mm)	641	612	610
	P_{cor} (%)	59.5	63.9	62.9
$SLOC_{MC}^{\dagger}$	RMS (mm)	478	433	478
	P_{cor} (%)	79.9	85.5	79.9

Table 7.9: Results for the two proposed SLOC when tested in the presence of an Oracle VAD detecting speech over the *Test* subset. The † denotes the application of data augmentation.

Joint VAD		Kitchen	Living Room	Average
$SLOC_{SC}$	RMS (mm)	605	476	512
	P_{cor} (%)	63.3	73.1	70.3
$SLOC_{SC}^{\dagger}$	RMS (mm)	303	321	**312**
	P_{cor} (%)	90.0	91.3	90.6
$SLOC_{MC}$	RMS (mm)	477	570	524
	P_{cor} (%)	73.3	65.6	69.5
$SLOC_{MC}^{\dagger}$	RMS (mm)	432	355	394
	P_{cor} (%)	78.2	87.3	82.8

Table 7.10: Performance of the two VADs when tested over true positive frames detected by the Joint VAD.

the latter provides better performance. This behaviour may be caused by the higher complexity of the $SLOC_{MC}$. Indeed, within this model the number of features maps extracted by the convolutional kernels is N_{mic} times the number of feature maps extracted for the $SLOC_{SC}$, where N_{mic} is the number of employed microphone pairs. As result, the information processed by the hidden layers and the model parameters to train are extremely more complex compared to the $SLOC_{SC}$.

Results with the Baseline Method

In Table 7.11 the best results achieved in [3] are reported. In details, in the baseline model a three stage optimization strategy has been adopted to select the most performing algorithms within the ensemble. Initially, all the four SLOCs are tested in presence of an Oracle VAD. The two more accurate techniques are then separately coupled with the Sohn's and SKF. In this stage, the less performing of the two previously selected SLOCs is rejected. Finally, the three proposed integration algorithms are applied to the remaining SLOC

coupled with the two VADs. As result, the best combination shows to be the Sohn's method acting as VAD and the Template method as SLOC, when integration is performed by SVM. Here a straightforward notation is adopted for these algorithms. Indeed, Sohn's method plus the SVM integration will be referred as VAD_B (Baseline), and the Template SLOC is referred as SLOC_B.

Furthermore, the metrics employed in [3] are converted to the ones in use in this work. Unfortunately, specific results for the kitchen and the living room are not available.

		Average
VAD_B	SAD (%)	6.7
	DEL (%)	6.1
	FA (%)	6.1
SLOC_B	RMS (mm)	961
	P_{cor} (%)	59.2

Table 7.11: Results achieved with the most performing algorithms in the baseline method

Last but not least, the result of the SLOC_B when it is coupled with an Oracle VAD instead of VAD_B are reported. Indeed, this result, being shown in Table 7.12, is important in order to analyse the baseline SLOC independently from VAD accuracy.

Oracle VAD		Average
SLOC_B	RMS (mm)	1094
	P_{cor} (%)	56.4

Table 7.12: Best performance of the baseline SLOC in the presence of an Oracle VAD.

Final Comparison

Here the overall performance of the proposed approach and the baseline model are discussed. In Table 7.13 a comparison between the two approaches for speaker localization is presented. In details, for each employed metric, Δ is defined as the subtraction of the result achieved by the baseline model from the result related to the most performing algorithm here proposed. Indeed, the data-driven $\text{SLOC}_\text{SC}{}^\dagger$ and the baseline SLOC_B are tested over speech detected by means of the Oracle VAD, hence all available speech in the *Test* subset. This comparison aims to test the SLOC accuracy in a absolute sense, independently from a VAD algorithm. As result, the data-driven model is more robust against the multi-room environment, outperforming the classical localization algorithm of more than 30% P_{cor}.

Oracle VAD		Average
Δ	RMS (mm)	-689
	P_{cor} (%)	+30.5

Table 7.13: Difference of the most performing proposed SLOC ($\text{SLOC}_{\text{SC}}{}^{\dagger}$) with the SLOC_{B} in the presence of an Oracle VAD.

Finally, in Table 7.14 the overall performance of the proposed model and the baseline framework are reported in terms of difference. The comparison is accomplished in terms of Δ defined above. In terms of detection, a reduction of 0.9% SAD and of 4.7% FA is observed when the Joint VAD† is employed, while there is an increase of 4.1% of DEL. On the other hand, when the $\text{SLOC}_{\text{SC}}{}^{\dagger}$ is tested over true positive detected by the Joint VAD†, a higher accuracy on P_{cor} of 31.4% and a reduction on RMS of 640 mm is observed with respect to the SLOC_{B}.

		Average
Δ	SAD (%)	-0.9
	DEL (%)	+4.1
	FA (%)	-4.7
	RMS (mm)	-640
	P_{cor} (%)	+31.4

Table 7.14: Differences between the proposed data-driven approach and the baseline model of [3].

Conclusions

This research addresses the tasks of VAD and SLOC in a multi-room environment by means of a CNN-based approach. Here further advancements with respect to Section 7.1 are proposed, by providing a fair comparison between the proposed model and the only other framework present in literature tackling the multi-room environment, which relies on classical VAD and SLOC algorithms. Hence, the CNN-based Joint VAD model is here employed for speech detection, being able to cooperatively exploit features for detection and localization. Two SLOC models have been evaluated to localize the speaker, in order to differently process input features. In addition, the quantity of training material is increased by applying data augmentation. In particular, two subsets extend the original DIRHA dataset: the first one is another version of the employed dataset, being used in previous research conducted within this thesis work, while the second one is the result of an ad-hoc technique here developed. In details, the RIRs of two virtual rooms equivalent in dimension to the rooms under observation have been generated, and speech data is then convolved with

them. As result, when the proposed Joint VAD model has been trained with data augmentation technique, a SAD reduction of 0.9% is observed compared to the baseline work. Similarly, the data-driven SLOC architecture here discussed outperforms the reference framework in localization of a P_{cor} 31.4% higher and with a RMS 640 mm lower. The effectiveness of data augmentation is clearly observed for VAD and SLOC. Indeed, a SAD being almost the half is achieved when data augmentation is employed, along whit a RMS reduction of more than 200 mm.

Chapter 8

Other Contributions

8.1 Quasi-Linear Phase IIR Filters for Audio Crossover

8.1.1 Preliminaries and Problem Statement

Digital Signal Processing (DSP) technologies have been widely employed in many areas including image processing, digital audio, and automation. Digital filters play a fundamental role in DSP based solutions. They are typically divided into two main categories: Finite Impulse Response (FIR) and IIR filters. Notoriously, FIR filters are generally stable, show a linear-phase behaviour, but require a high number of coefficients, which introduces a not negligible delay in the filtered signal. Conversely, IIR filters are preferred in terms of the total number of coefficients, although they have a non-linear phase. For this reason, phase response of IIR filters has been a study subject of researchers and industry experts for decades [116, 117, 118]. The first approach aiming to IIR with quasi-linear phase response relied on all-pass connected in cascade to the IIR filter, whose purpose was to equalize its phase response. Alternatively, this approach was substituted by new techniques directly designing the IIR filter in the digital domain, without any further step dedicated to the phase response equalization. This methodology, studied for several years and now well established, is based on the parallel connection of two digital all-pass sub-filters [119, 120, 121, 122], which allows to achieve a quasi-linear phase filter. The reference [119], dated 1986, already introduces this topic, showing how to obtain digital filters with *approximately linear phase*. On this same theoretical basis various techniques have been developed to refine the IIR response: an example is described in [120] where the design of digital IIR is based on the formulation of an eigenvalues problem solved with the Remez Exchange algorithm. On [121] is shown a further method always based on the parallel connection of all-pass filters to obtain stable digital IIRs according to the Tsypkin stability criterion and therefore not affected by *finite wordlength effects*. The last two mentioned papers focus respectively on obtaining an equiripple magnitude response and

on the quadratic phase error minimization. However, no attention is paid to the pass band amplitude behaviour, leading to a not negligible ripple that makes these filters unsuitable for realizing crossovers. Concerning fractional calculus, and in particular the Fractional Derivatives (FD), it was recently applied in different scientific fields, with a special effort also for designing digital FIR filters. The method illustrated in [123] employs fractional calculus applied at the prescribed frequency point for improving the filter response. In addition, in the last decade different approaches have been proposed to design digital IIR filters based on evolutionary techniques [124, 125, 126], however no constraints were applied for the phase response. Further advancements in the design of FIR and IIR filters were achieved by employing neural networks [127, 128, 129, 130].

Contribution

The main objective of this research is the study and realization of an audio crossover relying on digital IIR filters with quasi-linear phase response. In details, crossovers are particular filter banks dividing the audio spectrum so that each speaker reproduces the range of frequencies for which it is designed. Crossovers are indispensable in multi-way speaker systems. The development of digital audio crossover systems commonly relies on IIR filters, since FIR filters require a higher computational cost, plus they introduce a remarkable delay in the filtered signal. A novel design method has been recently introduced [131] for quasi-linear phase IIR filters, which exploits fractional derivative theory and counts on Swarm Intelligence (SI) algorithms to explore the solution space. However, this approach is not suitable for a flat response crossover design, since the single filter transition band behaviour is not predictable neither controllable. In fact, the primary objective for the audio crossovers design is that the intersections between the magnitude responses of the filters take place in correspondence with their cut-off frequencies, so as to give a flat sum [132].

This research aims to crossover characterized to quasi-linear phase response, for this reason an advanced version of the Particle Swarm Optimization (PSO) algorithm is proposed. In particular, the new proposed method introduces additional degrees of freedom to the PSO, plus relies on a diverse fitness function piloting the exploration of the solution space (e.g., the desired cut-off frequency of the filter amplitude response). Computer simulations show that one of the proposed methods is able to achieve a passband error being 60 dB lower than the reference solution. In addition, low order filters can be achieved compared to the reference solution, passing from a 27-th order filter to a 9-th. Further experiments are conducted for the design of a 2-way and a 4-way crossover, with a cut-off frequency of 50 Hz for the 2-way, and 500, 1500, 6000 Hz for the 4-way, at sampling frequency of 96 kHz. As result, the achieved crossovers have flat band magnitude response and quasi-linear phase.

8.1.2 Theoretical Background

The purpose of this section is to provide a general description of the mathematical tools employed for the quasi-linear phase IIR design. In details, FD introduced in [123] are discussed in the first paragraph, while evolutionary techniques are shown in the second one.

Fractional Derivatives

Considering a generic function $f(x)$, its n order derivative is defined as $D^n f(x)$, being n an integer value. In the field of fractional calculus, the definition of derivative is extended through FD, firstly introduced in [123]. In details, considering the real value u, the generic FD is defined as $D^u f(x)$. In [133] conventional integer derivative order were employed for designing digital filters, with the purpose of improving the filter magnitude response. The introduction of fractional derivatives in the design procedure represents a further step forward.

The mathematicians Grünwald-Letnikov define the generic order derivative as:

$$D^u f(x) = \frac{d^u f(x)}{dx^u} = \lim_{\Delta \to 0} \sum_{k=0}^{\infty} \frac{(-1)^k C_k^u}{\Delta^u} f(x - k\Delta) \tag{8.1}$$

where coefficient C_k^u is given by:

$$\begin{aligned} C_k^u &= \frac{\Gamma(u+1)}{\Gamma(k+1)\Gamma(u-k+1)} = \\ &= \begin{cases} 1 & k = 0 \\ \dfrac{u(u-1)(u-2)\cdots(u-k+1)}{1,2,3\cdots k} & k \geq 1 \end{cases} \end{aligned} \tag{8.2}$$

In the proposed work, FD are applied to two trigonometric functions (i.e., sine and cosine), leading to:

$$D^u A\sin(\omega x + \phi) = A\omega^u \sin(\omega x + \phi + \frac{\pi}{2}u) \tag{8.3}$$

$$D^u A\cos(\omega x + \phi) = A\omega^u \cos(\omega x + \phi + \frac{\pi}{2}u) \tag{8.4}$$

Evolutionary Techniques

Evolutionary techniques generally rely on the formulation of a search space, being subjected to an update rule depending on a fitness function. In details,

the search space is defined as:

$$[\boldsymbol{U}]^k = \begin{bmatrix} v_0^1, & v_1^1, & \cdots, & v_g^1 \\ v_0^2, & v_1^2, & \cdots, & v_g^2 \\ \vdots & \vdots & \ddots & \vdots \\ v_0^a, & v_1^a, & \cdots, & v_g^a \end{bmatrix}^k, \; 1 \leq x, y \leq a, g \tag{8.5}$$

where each one of the a rows in Equation 8.5 represents a generic individual of the evolutionary technique, composed by the g parameters (i.e., the number of columns), defined as v_y^x, while k indicates the iteration number. Furthermore, the search space goes through an update at each iteration. Several techniques have been proposed for the update strategy. In this case study PSO is employed, whose update relies on the following equations:

$$V_{x,y}^{k+1} = \chi \left\{ \begin{matrix} w \cdot V_{x,y}^k + c_1^k \cdot rand_1 \cdot \left(pbest_{x,y}^k - U_{x,y}^k \right) + \\ + c_2^k \cdot rand_2 \cdot \left(gbest_{1,y}^k - U_{x,y}^k \right) \end{matrix} \right\} \tag{8.6}$$

$$U_{x,y}^{k+1} = V_{x,y}^{k+1} + U_{x,y}^k \tag{8.7}$$

$$w = w_{max} - k \cdot (w_{max} - w_{min})/maxite \tag{8.8}$$

in which V_x is the speed related to a single particle in $\boldsymbol{U}$; x and y indicate the row and column indexes within the various matrices of particle elements; w is the inertia weight evaluated as Equation 8.8 and varying between $[w_{min}, w_{max}]$ and depending on the maximum number of epochs (*maxite*); χ is a constrained factor; c_1 and c_2 are constants; $rand_i$ is a uniformly distributed random number [0,1]; *pbest* and *gbest* are respectively the local best solution (for present iteration) and global best solution.

8.1.3 Quasi-Linear Phase IIR Design

In this section the reference solution for quasi-linear phase IIR filter design [131] is discussed. An IIR filter design is possible with a parallel configuration of two All-Pass Filters (APFs). Once the IIR constraints are defined, the problem is solved by employing FD, of which a general description is given in the previous section. Finally, the application of PSO for filter design is discussed, which allows to optimize the filter parameters through the minimization of the errors defined in the next paragraphs.

IIR filter by means of APFs parallel connection

Given two APFs in the time domain defined as $t_1[n]$, $t_2[n]$ and their response in the frequency domain $T_1(e^{j\omega})$, $T_2(e^{j\omega})$, their parallel connection leads to the

IIR filter defined as:

$$H_r\left(e^{j\omega}\right) = \frac{\left[T_2\left(e^{j\omega}\right) \pm T_1\left(e^{j\omega}\right)\right]}{2} \tag{8.9}$$

where the + sign is intended for a Low-Pass Filter (LPF) and − for a High-Pass Filter (HPF). In details, $t_1[n]$ is pure delay function of order M_1, while $t_2[n]$ is the M_2 order APF to be determined, defined as:

$$T_2(e^{j\omega}) = z^{-M_2}\left(\frac{\sum_{m=0}^{M_2} t_2[m]z^m}{\sum_{m=0}^{M_2} t_2[m]z^{-m}}\right) = z^{-M_2}\frac{K_2(z^{-1})}{K_2(z)} \tag{8.10}$$

The phase response of the delay function is $\varphi_1(\omega) = -M_1\omega$, whereas that of the second APF is given by:

$$\varphi_2(\omega) = -M_2\omega + \theta_2 \qquad \theta_2 = 2 \cdot arg\left\{\frac{1}{K_2(e^{j\omega})}\right\} \tag{8.11}$$

After that, the frequency response for the LPF and HPF is computed as:

$$H(e^{j\omega}) = \frac{\left[e^{j\varphi_1(\omega)} \pm e^{j\varphi_2(\omega)}\right]}{2} \tag{8.12}$$

and rearranging the above equations, the LPF response is:

$$H\left(e^{j\omega}\right) = e^{j\left(\theta_2(\omega) - \frac{M_1+M_2}{2}\omega\right)} \cdot \cos\left(\begin{array}{c}\frac{M_2-M_1}{2}\omega \\ -\theta_2\left(\omega\right)\end{array}\right) \tag{8.13}$$

and for the HPF:

$$H\left(e^{j\omega}\right) = j \cdot e^{j\left(\theta_2(\omega) - \frac{M_1+M_2}{2}\omega\right)} \cdot \sin\left(\begin{array}{c}\theta_2\left(\omega\right) - \\ \frac{M_2-M_1}{2}\omega\end{array}\right) \tag{8.14}$$

From Equation 8.13 and Equation 8.14 the phase of APF $T_2(e^{j\omega})$ is then adjusted to achieve the desired magnitude response of a LPF or HPF, respectively. In particular the desired phase response θ_D is defined as:

$$\theta_D\left(\omega\right) = \begin{cases} \dfrac{(M_2 - M_1)\cdot\omega}{2} & \omega \in [0, \omega_p] \\ \dfrac{(M_2 - M_1)\cdot\omega \pm \pi}{2} & \omega \in [\omega_s, \pi] \end{cases} \text{ for LPF} \tag{8.15}$$

$$\theta_D(\omega) = \begin{cases} \dfrac{(M_2 - M_1)\cdot\omega}{2} & \omega \in [0, \omega_s] \\ \dfrac{(M_2 - M_1)\cdot\omega \pm \pi}{2} & \omega \in [\omega_p, \pi] \end{cases} \quad \text{for HPF} \tag{8.16}$$

Where ω_s, ω_p are the pulsation of stop band and pass band, respectively. In addition, it is required that the difference among the APFs orders is equal to one. Then according to $M_2 \lessgtr M_1$, the sign $\pm$ is chosen in Equation 8.15 and in Equation 8.16.
Furthermore, in order to evaluate the IIR phase response, it is sufficient to calculate the phase response of the denominator polynomial $T_2(e^{j\omega})$ because the APF transfer function has symmetry property between numerator and denominator. For the sake of conciseness the mathematical details are not reported. The following equation is finally obtained:

$$\sum_{m=0}^{M_2} t_2[m] \cdot \sin\left(\theta_2(\omega) - m\omega\right) = 0 \;\Rightarrow\; (\boldsymbol{T}') \cdot \boldsymbol{S}(\omega) = 0 \tag{8.17}$$

where $\boldsymbol{T} = [t[0], t[1], t[2], \ldots, t[M_2]]$, and
$\boldsymbol{S}(\omega) = [\boldsymbol{s}(0), \boldsymbol{s}(\omega), \boldsymbol{s}(2\omega), \ldots, \boldsymbol{s}(M_2\,\omega)]$, whose each element is computed as:

$$\boldsymbol{s}(m\omega) = \sin\left[\theta_2(\omega) - m\omega\right] \tag{8.18}$$

Hence, the condition $\theta_2(\omega) = \theta_D(\omega)$ is imposed. Therefore the design problem of a flat response IIR filter becomes a minimization problem of the following error function [121]:

$$E_0 = \int_{\omega=0}^{\omega=\pi} W(\omega) \cdot \left[\boldsymbol{T}' \cdot \widehat{\boldsymbol{S}}(\omega)\right]^2 d\omega = \boldsymbol{T}'\boldsymbol{Q}\boldsymbol{T} \tag{8.19}$$

where:

$$\widehat{\boldsymbol{S}}(\omega) = \sin\left[\theta_D(\omega) - m\omega\right] \tag{8.20}$$

and $W(\omega)$ is a weighting function defined as follows:

$$W(\omega) = \begin{cases} \dfrac{1}{\cos^{-1}(1-\delta_p)} & \text{for } \omega \in pb \\ \dfrac{1}{\sin^{-1}(\delta_s)} & \text{for } \omega \in sb \end{cases} \tag{8.21}$$

The parameters δ_p and δ_s represent the maximum passband and stopband ripple magnitudes. With pb and sb pass band and stop band are indicated. The IIR filter design problem, defined in Equation 8.19, is solved by extracting

the eigenvectors of $\boldsymbol{Q}$ which is computed as:

$$\boldsymbol{Q} = \int W^2(\omega) \cdot \sin\left[\theta_D(\omega) - k\omega\right] \times \sin\left[\theta_D(\omega) - l\omega\right] d\omega \qquad 0 \leq k, l \leq M_2 \tag{8.22}$$

Here the passband accuracy is improved through the application of fractional derivative as discussed in Section 8.1.2.

The error function Equation 8.19 for digital filter design problem, can be solved by Lagrange multipliers [123]:

$$E_0 = \boldsymbol{T}'\boldsymbol{Q}\boldsymbol{T} - 2\boldsymbol{p}'\boldsymbol{T} + \alpha \tag{8.23}$$

where:

$$\boldsymbol{p} = \int_{\omega_l}^{\omega_u} \left[\boldsymbol{T}'\widehat{\boldsymbol{S}}(\omega)\right] \cdot \widehat{\boldsymbol{S}}(\omega)\, d\omega \tag{8.24}$$

and

$$\alpha = \int_{\omega_l}^{\omega_u} \left[\boldsymbol{T}'\widehat{\boldsymbol{S}}(\omega)\right]^2 = 0 \tag{8.25}$$

in which ω_l, ω_u are the lower and upper bound in the passband. After that, zero phase error is imposed between $T_1(e^{j\omega})$ and $T_2(e^{j\omega})$ at the prescribed frequency point (ω_0), leading to:

$$\boldsymbol{T}'\widehat{\boldsymbol{S}}(\omega_0) = 0 \tag{8.26}$$

and

$$D^u\left(\boldsymbol{T}'\widehat{\boldsymbol{S}}(\omega_0)\right) = 0 \tag{8.27}$$

Application of Fractional Derivative Constraints

Fractional Derivative Constraints (FDCs) are applied on $\widehat{\boldsymbol{S}}(\omega)$ using Equation 8.3 plus defining $M = M_2 - M_1$. Thus, from Equation 8.26 Equation 8.27 is obtained the system of constraints, writeable in matrix form as:

$$\boldsymbol{C}\boldsymbol{T} = \boldsymbol{f} \tag{8.28}$$

where $\boldsymbol{f}$ is a null vector of length equal to number of fractional constraints. Furthermore, the minimization of error function Equation 8.23, subjected to condition defined in Equation 8.28, is obtained by Lagrange multiplier method, and the optimized solution is:

$$\boldsymbol{T}_{opt} = \boldsymbol{Q}^{-1}\boldsymbol{p} - \boldsymbol{Q}^{-1}\boldsymbol{C}'\left(\boldsymbol{C}\boldsymbol{Q}^{-1}\boldsymbol{C}'\right)^{-1}\left[\boldsymbol{C}\boldsymbol{Q}^{-1}\boldsymbol{p} - \boldsymbol{f}\right] \tag{8.29}$$

The vector $\boldsymbol{T}_{opt}$ contains the denominator coefficients for the APF $T_2(e^{j\omega})$ For the sake of conciseness, further details are provided in [131].

IIR Filter Error Measure

In the previous paragraph the procedure to achieve a quasi-linear phase IIR is described, nonetheless a set of metrics is necessary to evaluate the accuracy of the obtained filter. On purpose the following errors are defined:

$$E_p = \int_{\omega \in pb} \left(1 - \left|H_r\left(e^{j\omega}\right)\right|\right)^2 d\omega \tag{8.30}$$

$$E_s = \int_{\omega \in sb} \left(\left|H_r\left(e^{j\omega}\right)\right|\right)^2 d\omega \tag{8.31}$$

$$Er_0 = E_p + E_s \tag{8.32}$$

$$A_s|_{dB} = 20\log_{10}\left(\left|H_r\left(e^{j\omega}\right)\right|\right)_{\omega=\omega_s} \tag{8.33}$$

being respectively the passband error, stopband error, total error, minimum sb attenuation. Er_0 will be employed as fitness function for the PSO.

On the application of PSO to IIR Filter Design

In the addressed case study, the PSO generic particle is defined as follows:

$$\boldsymbol{U}^x = \left[\omega_0^{\mathrm{x}}, u_0^{\mathrm{x}}, u_1^{\mathrm{x}}, ..., u_g^{\mathrm{x}}\right] \tag{8.34}$$

where ω_0 is the frequency where FDCs are applied, and $u_{0,\cdots,g}^{\mathrm{x}}$ are the FDCs values. The search space is limited by upper and lower bounds, defined as:

$$\boldsymbol{UB} = \left[\omega_0^{\max}, u_0^{\max}, u_1^{\max}, ..., u_g^{\max}\right] \tag{8.35}$$

$$\boldsymbol{LB} = \left[\omega_0^{\min}, u_0^{\min}, u_1^{\min}, ..., u_g^{\min}\right] \tag{8.36}$$

The search space initialization relies on uniform distribution within the defined limits. The most important features of PSO algorithm are summarized below: the first variable in each particle represents the pulsation ω_0 and it is chosen in the filter passband $[\omega_p^{max}; \omega_p^{min}]$, the limits 14.99 and -14.99 are set up for each FDC that is decided to use. The fitness function of the PSO consists in the error Er_0 defined in Equation 8.32.

8.1.4 Proposed Method for Audio Crossover Design

The purpose of [131] is the design of quasi-linear phase IIR. Differently, here the objective is to develop a flat band crossover relying on quasi-linear phase IIR. In details, considering a generic two way crossover, a flat band crossover is achieved if and only if the two filters of the crossover have a symmetric response. This statement is valid for linear or quasi-linear phase filters. Hence, when the

reference theory was initially employed to design a flat band crossover, the main issue raised. In particular, while in [131] the pass band and stop band errors are minimized in the filter design procedure, no attention is paid to the transition band. This result will be accurately shown in Section 8.1.5. The proposed method tackles the transition band issue, relying on the introduction of several advancements in the PSO algorithm. In details, two variants are proposed, where the first one allows an accurate control of the passband behaviour, and the second a fine tuning in terms of cut-off frequency. They are simply referred as *Strategy 1* and *Strategy 2*.

Strategy 1

With this approach, ω_p and ω_s become two more grades of freedom for the PSO algorithm, with the purpose of a finer tuning. Hence, the PSO needs to be piloted differently. Two new frequencies are thus defined, which are $\omega_{p,des}$, $\omega_{s,des}$, substituting ω_p and ω_s in Equation 8.30, Equation 8.31, Equation 8.32. As a consequence Equation 8.34 becomes:

$$\boldsymbol{U}^x = \left[\omega_p^{\mathrm{x}}, \omega_s^{\mathrm{x}}, \omega_0^{\mathrm{x}}, u_0^{\mathrm{x}}, u_1^{\mathrm{x}}, \ldots, u_g^{\mathrm{x}}\right] \tag{8.37}$$

The new frequencies are selected in the range $[0, \pi]$, however the following conditions are imposed: $\omega_p < \omega_s$ must be valid for LPF design, and $\omega_p > \omega_s$ for HPF. In addition, ω_0 must lie in the passband region. The drawback of this approach consists in particles leading to non-stable filters. In this context it is introduced the concept of *dead* or *alive* as a condition of the generic particle of the PSO. In details, for each PSO particle, the $\boldsymbol{T}_{opt}$ vector is calculated, from which the poles of the filter are evaluated, being the roots of the polynomial. If just one root lies outside the unitary circle, the particle is declared as *dead*. A *dead* particle goes through regeneration, i.e., it is randomly re-initialized, and declared as *alive*.

Strategy 2

Besides of $\omega_{p,des}$, $\omega_{s,des}$ used in Equation 8.30, Equation 8.31, Equation 8.32, a new error is here introduced in order to be the fitness function of the PSO. It is defined as the difference in omega between the passage to -3 dB of the drawn filter and the desired cut-off frequency ω_{3dB}. Imposing this constraint, the sum of two adjacent filters (HPF and LPF) guarantees a flat band response for the transition band. Conversely, a wider space of solutions means a more difficult task in terms of convergence. Thus, the solution is to initialize the PSO with a diverse strategy. In details, for the initialization stage of the PSO, the piloting error is evaluated as in Equation 8.32, imposing $\omega_p = \omega_s = \omega_{3dB}$.

8.1.5 Comparison with the Reference Solution

Quasi-linear phase IIR filter: the transition band issue

Here experiments with the method proposed in [131] are conducted, focusing on the behaviour of the transition band, with the purpose of showing the unsuitability of the reference solution for designing a flat band crossover. The test reported concerns the development of two LPFs, the first with a passband from 0 to 50 Hz, the second from 0 to 250 Hz. While a set of parameter is fixed, the remaining ones are varied in order to proof the independence of the obtained results from any of the above.

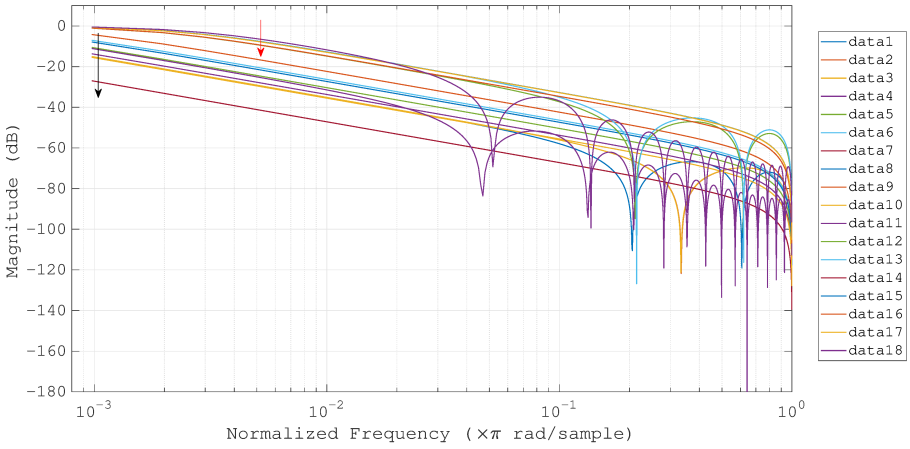


Figure 8.1: Plot of the experiments conducted for testing the achieved transition band by the reference solution for quasi-linear phase IIR filter design. The black arrow represents the normalized theoretical cut-off frequency of 50 Hz for that filter family; the red arrow indicates frequency of 250 Hz.

In details, for the LPF specifications M_2 ranges between 3 and 14, while $M_1 = M_2 - 1$; $f_{sampling}$ is 96 kHz, f_p and f_s vary from 50 Hz to 500 Hz; δ_p and δ_s are equal to 0.02 dB and -60 dB, respectively. For the PSO, lower and upper bounds of ω_0 are 0 and ω_p, the number of FDC goes from 1 to 10, FDC values range in [-14.99, 14.99], the population is 100 and iterations 50. In addition, update parameters of Equation 8.6 and Equation 8.8, i.e., c_1, c_2, χ, w_{max} and w_{min}, are set to 1, 1, 0.7, 0.9 and 0.4, respectively. Finally, the integrals of Equation 8.22 and Equation 8.24 are computed on a grid size of 1000 equally spaced points for each design normalized frequency band. The STFT of the frequency response $H_r(e^{j\omega})$ of Equation 8.30 and Equation 8.31 is evaluated over 4096 points. The result of the experiments in terms of the errors defined in Equation 8.32 and achieved cut-off frequencies are reported in Table 8.1, while

their plots are shown in Fig. 8.1. Results are obvious: the obtained cut-off frequencies shows a high variance. Therefore the approach proposed by article [131] does not guarantee any constraint for the transition band.

n° Exp	$f_p = f_s$ (Hz)	M2, M1	n° FDC	ω_0 (PSO)	E_p (dB)	E_s (dB)	Er_0 (dB)	$\lvert H(w_p)\rvert$	A_s (dB)	f_{3dB} (Hz)
1	50	3 , 2	1	0.0032	-34.183	-37.696	-32.583	0.3026	-10.383	7.91
2	50	3 , 2	2	0.0012	-34.178	-37.695	-32.579	0.3022	-10.394	7.9
3	50	3 , 2	3	0.0021	-34.185	-37.679	-32.579	0.3027	-10.379	7.914
4	50	3 , 2	5	0.0015	-35	-36.023	-32.471	0.3619	-8.828	9.383
5	50	3 , 2	10	0.0018	-37.477	-32.547	-31.336	0.5177	-5.718	14.253
6	fp=50; fs=80	3 , 2	10	0.001	-41.701	-31.442	-31.051	0.7032	-7.056	23.183
7	fp=50; fs=100	3 , 2	10	0.0007	-30.189	-60.691	-30.185	0.0386	-37.772	starts from -9 dB
8	50	9 , 8	10	0.0011	-40.504	-29.86	-29.501	0.6594	-3.617	20.438
9	50	14 , 13	10	0	-47.212	-26.097	-26.063	0.8427	-1.487	36.516
10	fp=50; fs=100	14 , 13	10	0.0027	-33.968	-43.125	-33.47	0.2866	-19.969	7.577
11	fp=50; fs=100	14 , 13	1	0.0014	-36.987	-38.2	-34.541	0.4899	-14.641	13.369
12	250	3 , 2	1	0.0109	-25.779	-27.045	-23.356	0.3537	-9.028	88.416
13	fp=250; fs=500	3 , 2	1	0.0056	-27.506	-28.024	-24.747	0.4374	-12.547	113.741
14	250	3 , 2	10	0.0096	-27.34	-24.94	-22.967	0.4297	-7.338	111.254
15	fp=250; fs=500	3 , 2	10	0.0107	-27.337	-27.79	-24.547	0.4295	-12.715	111.202
16	250	14 , 13	10	0.0053	-25.851	-26.734	-23.259	0.3573	-8.939	89.414
17	fp=250; fs=500	14 , 13	10	0.0083	-27.271	-27.867	-24.548	0.4264	-12.789	110.189
18	fp=250; fs=500	14 , 13	1	0	-28.822	-28.756	-25.779	0.4952	-11.545	134.386

Table 8.1: Experiments concerning the transition band of the reference solution for quasi-linear phase IIR filter design. Varying parameters are reported in the second, third and fourth columns.

Comparison with the reference solution

The so-called method *Strategy 1* proposed in Section 8.1.4 is here tested against the reference solution [131]. The experiments have been conducted as follows. The employed $\omega_{p,des}$, $\omega_{s,des}$ of the proposed method have been set equal to ω_p, ω_s used by the reference solution for quasi-linear phase IIR filter design, in details 0.3π and 0.4π. The errors E_p and E_s have been evaluated on the ω_p, ω_s found by the PSO. Results are reported in Table 8.2. The LPF specifications are: $M_2 = 14$ and $M_1 = 13$; δ_p and δ_s are equal to 0.02 dB and -60 dB. For the PSO, lower and upper bounds of ω_p, ω_s and ω_0 are $[0, \pi]$, $[0, \pi]$ and $[0, \omega_{p,des}]$ respectively; the number of FDC goes from 1 to 10, FDC values range in [-14.99, 14.99], the population is 200 and iterations 20. In addition, update parameters, integral points and STFT points are set as described in the previous paragraph.

Parameters	1-FDC	2-FDC	3-FDC	4-FDC	5-FDC	6-FDC	7-FDC	8-FDC	9-FDC	10-FDC
CFI-PSO (reference paper best results)										
E_p (dB)	-65.838	-69.963	-69.516	-72.324	-72.595	-72.749	-72.402	-72.171	-72.618	-72.581
E_s (dB)	-67.471	-67.988	-67.269	-68.476	-68.938	-68.992	-69.25	-69.464	-69.123	-69.218
$\|H(\omega_p)\|$	0.994	0.995	0.996	0.998	0.996	0.996	0.996	0.997	0.996	0.997
A_s (dB)	-44.891	-49.281	-49.61	-47.246	-47.831	-47.719	-48.102	-49.134	-47.935	-48.335
ω_p (PSO)	1.0187	1.0316	1.0566	0.3698	0.6261	0.2491	0.5108	0.248	0.8218	0.3968
ω_s (PSO)	1.2461	1.2463	1.2394	1.2687	1.7162	1.9855	1.859	2.0033	1.4275	2.1416
E_p (dB)	-34.392	-33.068	-22.552	-82.703	-76.598	-103.02	-94.927	-98.034	-96.809	-118.779
E_s (dB)	-61.067	-61.393	-55.479	-68.44	-79.482	-44.986	-48.367	-48.864	-56.238	-55.946
$\|H(\omega_p)\|$	0.8358	0.8079	0.5281	0.9998	0.99937	0.99997	0.99994	0.99994	0.99984	1
A_s (dB)	-43.243	-43.669	-38.519	-51.416	-67.563	-53.157	-42.135	-40.076	-51.047	-49.874

Table 8.2: Comparison between the reference solution for quasi-linear phase IIR filter design and the proposed approach.

In Table 8.2 a general improvement is observed in terms of E_p employing four or more FDCs, which is a considerable results, since a flat band crossover must rely on filters having maximally flat pass band behaviour.

Another set of simulations is carried out with the purpose of reducing the filter order (i.e., M_2, M_1), by always employing the method *Strategy 1.* M_2 is varied from 2 to 14, with $M_1 = M_2 - 1$. The number of FDCs is 10. The population and iterations of the PSO is randomly chosen, and reported in Table 8.3. The remaining parameters are equal to the parameters employed for the set of experiments previously described in this section.

Filter specifications				Results: errors computed with respect to ω_p and ω_s chosen by PSO						
M_2	M_1	Population	Iteration	ω_p by PSO	ω_s by PSO	E_p (dB)	E_s (dB)	Er_0 (dB)	$\|H(\omega_p)\|$	A_s (dB)
14	13	200	50	0.3658	2.2819	-132.386	-52.466	-52.466	1	-46.742
13	12	200	50	0.3367	2.4257	-76.575	-47.124	-47.119	0.99955	-40.08
12	11	30	500	0.6532	1.6488	-72.744	-48.087	-48.072	0.99956	-45.011
11	10	30	500	0.7557	1.5166	-63.975	-50.293	-50.111	0.99904	-40.712
10	9	30	500	0.1517	2.2732	-90.23	-58.756	-58.753	0.99983	-52.654
9	8	30	500	0.1471	2.6561	-86.179	-45.087	-45.087	0.99971	-39.236
8	7	200	20	0.9152	1.6559	-53.702	-52.346	-49.961	0.98505	-47.928
7	6	200	20	0.9874	1.6435	-29.263	-36.206	-28.463	0.82574	-32.475
6	5	30	500	0.6662	1.9066	-49.519	-31.597	-31.527	0.98744	-33.692
5	4	30	500	0.1338	2.7998	-102.777	-38.68	-38.68	0.99996	-30.289
4	3	200	20	0.8516	2.2256	-45.715	-37.429	-36.828	0.97022	-41.05
3	2	30	500	1.7228	2.8991	-3.323	-48.213	-3.323	0.00787	-37.699
2	1	200	20	1.6689	3.003	-4.076	-45.026	-4.076	0.03684	-31.723

Table 8.3: Results obtained with *Strategy 1.* Test are conducted varying the filter order M_2, M_1. Low order filters still achieve lower passband error compared to the reference solution.

In Table 8.3 the achieved results are reported. In the best case study of a 27th order filter ($M_2 = 14$; $M_1 = 13$), the proposed method outperforms the reference solution for quasi-linear phase IIR filter design in term of E_p, achieving an error value of -132.386 dB, which is 60 dB lower compared to [131] (i.e., -72.749 dB). Moreover, a considerable reduction of the filter order allows

to achieve better results in terms of passband error: specifically, with a 9th order filter ($M_2 = 5$; $M_1 = 4$) the error is -102.777 dB, that is 30 dB lower than the reference one. The order reduction is an important result aiming to real world applications.

8.1.6 Results for Crossover Design

Both the proposed approaches described in Section 8.1.4 are tested for the purpose of crossover design. The first method employs the amplitude error (Er_0) to pilot the PSO, while the second makes use of the ω_{3dB} error. The LPF-HPF pairs composing a crossover are designed by employing the same value for cut-off frequency. Furthermore, as shown in Section 8.1.5, the objective here to develop filters with a lower order compared to the ones proposed in [131]. In particular, when employing the technique named *Strategy 2*, encouraging results are achieved also with $M_2 = 3$ and $M_1 = 2$ for a 4-way crossover design.

2-way Crossover Design

As first test, a 2-way crossover with an expected cut-off frequency equal to 50 Hz is designed. Adopting *Strategy 1* exposed in the previous section, where the -3 db error is not yet used by the PSO, two filters are obtained: the real cut-off frequencies are 17.8 Hz and 61.5 Hz for LPF and HPF respectively. The two filters are summed to realize the 2-way crossover: as shown in Fig. 8.2 the overall amplitude response is not flat and has a ripple of -6.88 dB near the crossing region.

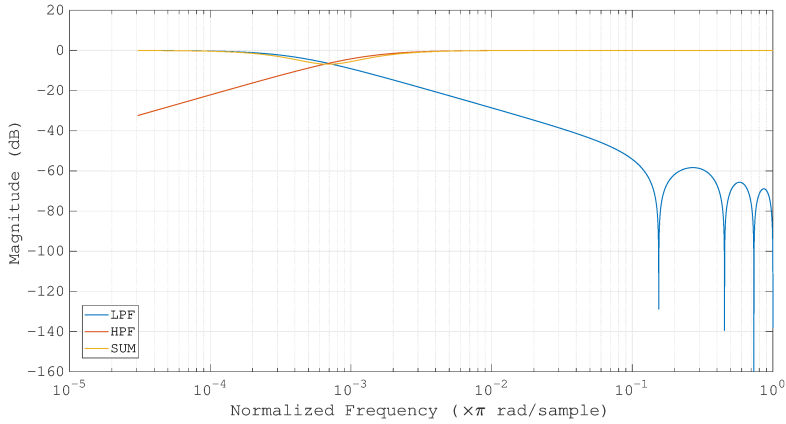
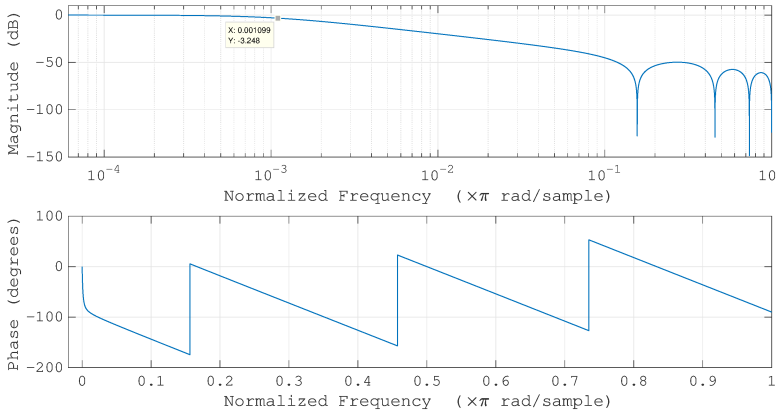
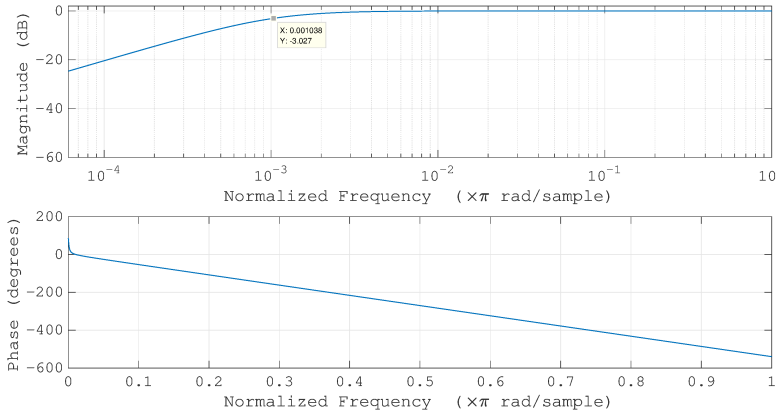


Figure 8.2: 2-way crossover with 50 Hz expected cut-off frequency: only the amplitude error (Er_0) is computed and minimized for each filter (*Strategy 1*). Their sum, i.e. the crossover response, is not flat.

Then the same crossover is designed by means of *Strategy 2*, presented in Section 8.1.4, using the -3 db error that allows to obtain exactly the set cut-off frequency. The low-pass and high-pass filters are separately plotted in Fig. 8.3, where both shows a quasi-linear phase behaviour in their pass and transition band.



(a) Quasi-linear phase IIR LPF



(b) Quasi-linear phase IIR HPF

Figure 8.3: Filters designed with the proposed method (*Strategy 2*): magnitude response in logarithmic scale (8.3a) and phase response in linear scale (8.3b).

The crossover composed by these two filters is depicted in Fig. 8.4, where a maximally flat band behaviour is achieved. Although *Strategy 1* minimizes the error in passband region, this constraint is not sufficient for a flat band crossover design, thus *Strategy 2* demonstrates to be a winning solution. Specifications

for the employed filters are: $M_2 = 4$, $M_1 = 3$; $f_{sampling} = 96\,\text{kHz}$; $f_p = f_s = f_{3dB} = 50\,\text{Hz}$; $\delta_p = 0.02\,\text{dB}$; $\delta_s = -60\,\text{dB}$. For the PSO, ω_p, ω_0 lies in the range $[0, \omega_{3dB}]$ and ω_s is set equal to ω_{3dB} for the LPF, while for the HPF ω_s, ω_0 lies in the range $[0, \omega_{3dB}]$ and ω_p is set equal to ω_{3dB}; the number of FDC is 1 that ranges in [0, 14.99]; the population is 100 and iterations 20. In addition, update parameters of Equation 8.6 and Equation 8.8, which are c_1, c_2, χ, w_{max} and w_{min}, are set to 1, 1, 1, 0.9 and 0.4, respectively. Finally, the integrals are computed on a grid size of 10000 equally spaced points. The STFT is evaluated over 65536 points. As a drawback for the quasi-linear phase response, the filters show a low slope in transition band, being $5 \div 6\,\text{dB/oct}$.

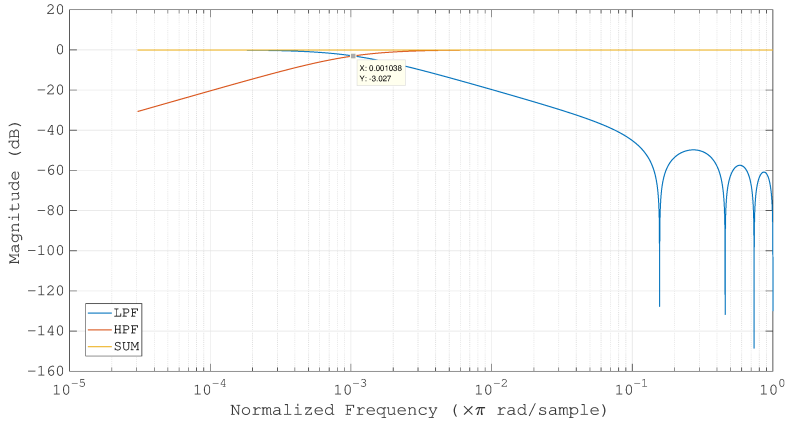
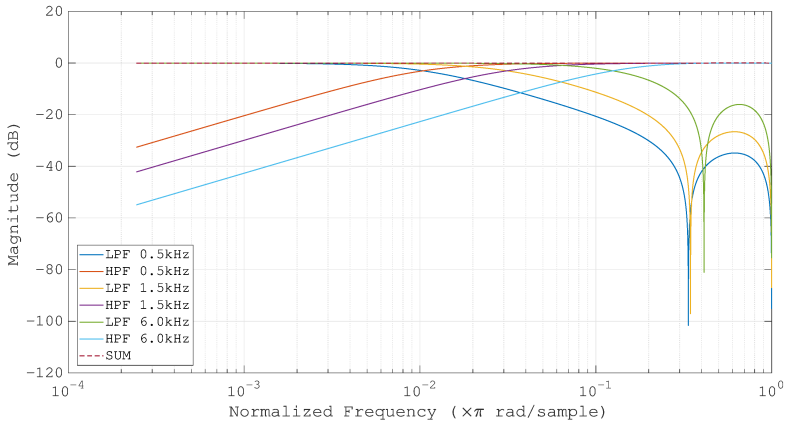


Figure 8.4: 2-way crossover with 50 Hz cut-off frequency designed with *Strategy 2*.

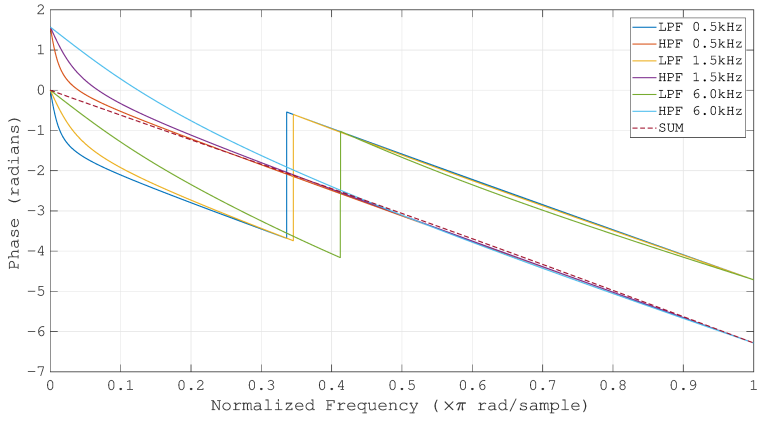
4-way Crossover Design

The design of a 4-way crossover is proposed. Three LPF and three HPF filters are designed by means of the proposed approach named *Strategy 2*, with the following cut-off frequencies: 500 Hz; 1500 Hz; 6000 Hz. The sampling frequency is 96 kHz. Low order filters are employed ($M_2 = 3$, $M_1 = 2$) in the perspective of real-world applications. Computer simulations show that a number of FDCs equal to 1 do not guarantee the algorithm convergence, hence 2 FDCs have been employed. In details, the others specifications for the LPFs are: $M_2 = 3$ and $M_1 = 2$, $f_p = f_s = f_{3dB}$; δ_p and δ_s are equal to 0.02 dB and -60 dB, respectively. ω_p, ω_s and ω_0 ranges in $[0, \omega_{3dB}]$; the number of FDCs is 2 and their value lies in the range [-14.99, 14.99], the population is 100 and iterations 50. In addition, update parameters of the PSO are the same of the previous experiment of 2-way crossover, whereas the integrals are computed on a grid size of 1000 equally spaced points and the STFT is evaluated over 4096

points. The HPFs specifications are the same of the LPFs. For the high-pass filters, the PSO lower and upper bounds of ω_s and ω_0 are $[0, \omega_{3dB}]$ while ω_p is set equal to ω_{3dB}; the number of FDCs is 2 and they range in [-14.99, 14.99], the population is 50 and iterations 150. In addition, update parameters of the PSO, along with the integrals and STFT points, are the same of the previous LPFs. The obtained crossover is depicted in Fig. 8.5.



(a) Filters and crossover magnitude responses

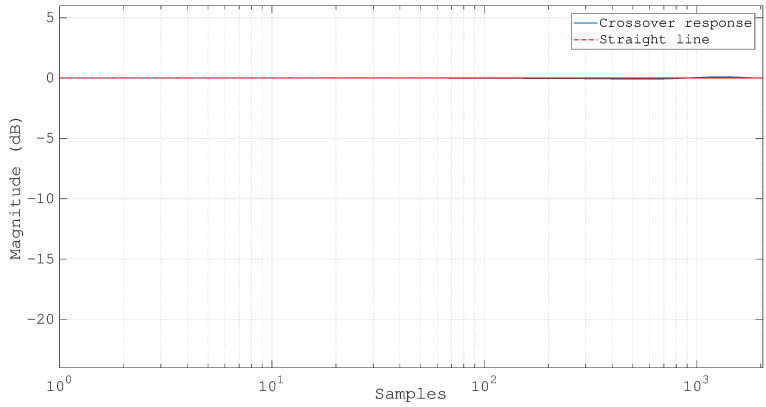


(b) Filters and crossover phase responses

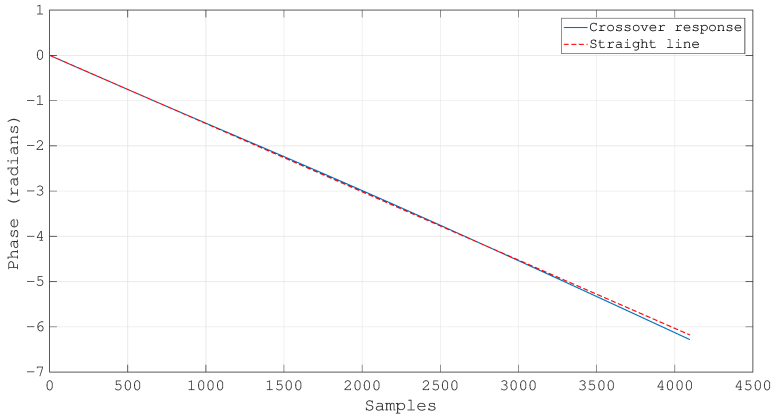
Figure 8.5: Quasi-linear phase IIR 4-way crossover designed with *Strategy 2*: magnitude responses in logarithmic scale (8.5a); phase responses in linear scale (8.5b).

It relies on 5-th order filters with quasi-linear phase. The slope of each filter is about 5÷6 dB/oct and the crossover amplitude response is flat. Phase response of the so designed crossover has a quasi-linear behaviour. To highlight

the behaviour of the crossover, its amplitude response and phase response are depicted together with straight lines, as shown in Fig. 8.6.



(a) Crossover magnitude response



(b) Crossover unwrapped phase response

Figure 8.6: Quasi-linear phase IIR 4-way crossover designed with *Strategy 2*: FFT of the 4096 points impulse response. Magnitude in logarithmic scale (8.6a); unwrapped phase in linear scale (8.6b).

Moreover, in order to evaluate the goodness of the crossover response, two further errors definitions are here introduced. The first is the integral of the squared difference between the crossover amplitude response and a straight line (Fig. 8.6a): the integral is evaluated over the same number of points as the STFT. The second is the integral of the squared difference between the crossover phase response and a straight line (Fig. 8.6b) evaluated over the same number of STFT points. For the proposed 4-way crossover, the above

errors are respectively: amplitude error = 0.2078; phase error = 4.9912.

Conclusions

In this contribution, the development of an audio crossover relying on quasi-linear phase IIR filters is addressed. Recently, a new technique for quasi-linear phase IIR filter design has been proposed, where FDCs are employed to optimize the filter response, and evolutionary techniques explore the solution space. Specific tests demonstrate the unsuitability of this reference solution for an efficient crossover design. Hence a new design methodology is proposed, relying on an advanced version of the PSO algorithm, where new degrees of freedom are introduced. In details, two PSO optimization strategies are here discussed, depending on two diverse errors employed as fitness function, where the first one aims to guarantee an accurate pass band behaviour, while the second consists in the mismatch between the desired and the obtained cut-off frequency. The first strategy is tested against the reference solution, achieving a remarkable reduction (60 dB) in the passband error. After that, both the proposed techniques have been tested for the purpose of a 2-way crossover design, where the second strategy allows to achieve a flat band crossover with quasi-linear phase. Furthermore, encouraging results are achieved with low order filters, which was not addressed by the reference solution. This undoubtedly represents a remarkable results for real-world applications where the computational complexity is always an issue. Finally, the case study of a more complex 4-way crossover is addressed, where a flat band crossover is achieved by employing the cut-off frequency error as PSO fitness function.

Chapter 9

Conclusions and Future Works

9.1 Conclusions

This thesis addresses the development of data-driven models for VAD and SLOC in reverberant environments. Thus, DNNs have been largely investigated and exploited in this work, driven by the promising results achieved with DNNs for audio processing in the recent years. Indeed, the choice of these models is motivated by their generalization capability, which is commonly weak in classical algorithms proposed for VAD and SLOC.

The first three chapters of this work aim to give an overview of the main motivations behind this work. Indeed, recent results in DNN-based approaches for VAD and SLOC are vastly discussed. After that, main aspects of DNNs are explored, followed by detailed descriptions of employed audio features and datasets.

Chapter 5 discusses a DNN-based approach for VAD in a multi-room environment. A preliminary study compares several neural architectures, where multiple audio features feed the neural models. As result, the CNN-based approach shows the most reliable results in terms of statistical behaviour, due to the possibility of exploiting the temporal evolution of the signal. For this reason, the CNN-based VAD goes through further investigations, by considering multiple audio channels as input data, and relying on audio captured from multiple rooms.

SLOC systems are investigated in Chapter 6. The multi-room environment is initially addressed, where a first model based on MLP outperforms a classical SLOC algorithm highly suffering from reverberation and noise present in the environment. A further study makes use of CNNs as well, and proposes extensive experiments to assess the effectiveness of the employment of a temporal context along with the processed data. Binaural sound localization is then addressed by means of two CNN-based frameworks. An end-to-end approach estimates the speaker azimuth, whose design is inspired by the human auditory system. Experiments show that the proposed model robustly localize the speaker, behaving as our hearing system does in reverberant conditions.

The elevation of the speaker is then estimated by means of a novel approach relying on the frequency domain magnitude and phase of the audio signals. This system outperforms the state-of-the-art DNN-based algorithm for sound elevation estimation present in literature.

Chapter 7 proposes a novel framework for joint detection and localization of a speaker. The idea driving this study aims to the possibility of virtuously exploit localization and detection features in order to increase the overall performance of the system. Hence a CNN model with multiple input and multiple output is initially investigated, when the issue of localizing an absent speaker is dealt with. After that, a further work assesses the model performance with respect to the only framework present in literature capable of joint VAD and SLOC in a multi-room environment. Within this study, an ad hoc data augmentation technique is proposed to adequately train the neural model.

Another contribution is presented in Chapter 8, where the design of quasi-linear phase IIR filters for audio crossover is addressed. This work makes use of FDCs to impose the constraint of a quasi-linear phase, after that the definition of a new error allows to control the ω_{3dB} of the filter, to guarantee a proper flat band crossover. Finally, two case studies show that the proposed method achieves promising results when applied for the design a 2-way and a 4-way crossover.

9.2 Future Works

Future works will concern the testing of the proposed algorithms for VAD and SLOC against new environments. Along with this new scenarios, ad hoc solutions for data augmentation must be considered as well, since as shown for the development of a joint VAD and SLOC framework, the discussed models are sensible to data employed for their training.

Furthermore, it is interesting to adapt the proposed SLOC models inspired by the human hearing system to a multi-room context, where microphone arrays are available instead of binaural ones. Again, important results can be demonstrated by a observing the typology of features extracted by end-to-end systems in reverberant environments. Last but not least, models similar to the ones discussed for binaural SLOC can be easily extended for VAD, deserving further investigations. Even the development of a novel framework for joint VAD and SLOC can be addressed, where input features are the frequency domain magnitude and phase of the signals.

On the other hand, the investigation of a unique system acting in a binaural scenario and capable of localizing a speaker in terms of azimuth and elevation deserves particular attention, driven by the promising results achieved in this work.

Differently, the end-to-end approach proposed here could be adapted for other tasks different from VAD and SLOC, where the same feature extraction procedure performed by the network could be suitable for other audio-related case studies.

Complete Publications List

- Fabio Vesperini, Paolo Vecchiotti, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "Deep neural networks for multi-room voice activity detection: Advancements and comparative evaluation," in *Neural Networks (IJCNN), 2016 International Joint Conference on.* IEEE, 2016, pp. 3391–3398.

- 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*, pp. 161–170. Springer, 2018.

- Fabio Vesperini, Paolo Vecchiotti, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "A neural network based algorithm for speaker localization in a multi-room environment," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on.* IEEE, 2016, pp. 1–6.

- Fabio Vesperini, Paolo Vecchiotti, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "Localizing speakers in multiple rooms by using deep neural networks," *Computer Speech & Language*, vol. 49, pp. 83–106, 2018.

- Ferdinando Foresi, Paolo Vecchiotti, Diego Zallocco, and Stefano Squartini, "Designing quasi-linear phase iir filters for audio crossover systems by using swarm intelligence," in *Audio Engineering Society Convention 144.* Audio Engineering Society, 2018.

- Paolo Vecchiotti, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "Deep neural networks for joint voice activity detection and speaker localization," 2018.

- Paolo Vecchiotti, Giovanni Pepe, Emanuele Principi, and Stefano Squartini, "A deep learning based method exploiting data augmentation for joint voice activity detection and speaker localization in residential environments," *Submitted to Speech communication*, 2018.

- Paolo Vecchiotti, Ning Ma, Stefano Squartini, and Guy J. Brown, "End-to-end sound localisation from the raw waveform," in *Submitted to International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2019.
- Ning Ma, Paolo Vecchiotti, and Guy J. Brown, "A convolutional neural network for estimating sound source elevation in reverberation using phase and magnitude spectra," in *Submitted to International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2019.

Bibliography

[1] Thad Hughes and Keir Mierle, "Recurrent neural networks for voice activity detection," in *Proc. of ICASSP*, Vancouver, BC, Canada, Mar. 26-31 2013, pp. 7378–7382.

[2] Diego Augusto Silva, José Augusto Stuchi, Ricardo P Velloso Violato, and Luís Gustavo D Cuozzo, "Exploring convolutional neural networks for voice activity detection," in *Cognitive Technologies*, pp. 37–47. Springer, 2017.

[3] Yuuki Tachioka, Tomohiro Narita, Shinji Watanabe, and Jonathan Le Roux, "Ensemble integration of calibrated speaker localization and statistical speech detection in domestic environments," in *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2014 4th Joint Workshop on*. IEEE, 2014, pp. 162–166.

[4] Mohammad J Taghizadeh, Philip N Garner, Hervé Bourlard, Hamid R Abutalebi, and Afsaneh Asaei, "An integrated framework for multi-channel multi-source localization and voice activity detection," in *Proc. of HSCMA*, 2011, pp. 92–97.

[5] Adil Benyassine, Eyal Shlomot, H-Y Su, Dominique Massaloux, Claude Lamblin, and J-P Petit, "Itu-t recommendation g. 729 annex b: a silence compression scheme for use with g. 729 optimized for v. 70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64–73, 1997.

[6] Robert E Yantorno, Kasturi R Krishnamachari, Jereme M Lovekin, Daniel S Benincasa, and Stanley J Wenndt, "The spectral autocorrelation peak valley ratio (sapvr)-a usable speech measure employed as a co-channel detection system," in *Proceedings of IEEE International Workshop on Intelligent Signal Processing (WISP)*. Citeseer, 2001, vol. 21.

[7] 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.

[8] Akinobu Lee, Keisuke Nakamura, Ryuichi Nisimura, Hiroshi Saruwatari, and Kiyohiro Shikano, "Noice robust real world spoken dialogue system using gmm based rejection of unintended inputs," 2004.

[9] Ji Wu and Xiao-Lei Zhang, "An efficient voice activity detection algorithm by combining statistical model and energy detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, pp. 1–10, 2011.

[10] Jong Won Shin, Joon-Hyuk Chang, and Nam Soo Kim, "Voice activity detection based on statistical models and machine learning approaches," *Computer Speech & Language*, vol. 24, no. 3, pp. 515–530, 2010.

[11] Xiao-Lei Zhang and Ji Wu, "Deep belief networks based voice activity detection," *IEEE Audio, Speech, Language Process.*, vol. 21, no. 4, pp. 697–710, 2013.

[12] Xiao-Lei Zhang and DeLiang Wang, "Boosted deep neural networks and multi-resolution cochleagram features for voice activity detection," in *Proc. of Interspeech*, Singapore, Singapore, Sep. 14-18 2014, pp. 1534–1538.

[13] Neville Ryant, Mark Liberman, and Jiahong Yuan, "Speech activity detection on youtube using deep neural networks.," in *Proc. of Interspeech 2013*, Lyon, France, 2013, pp. 728–731.

[14] Giacomo Ferroni, Roberto Bonfigli, Emanuele Principi, Stefano Squartini, and Francesco Piazza, "A deep neural network approach for voice activity detection in multi-room domestic scenarios," in *Proc. of IJCNN*, 2015, pp. 1–8.

[15] Alberto Abad, Miguel Matos, Hugo Meinedo, Ramon F Astudillo, and Isabel Trancoso, "The L2F system for the EVALITA-2014 speech activity detection challenge in domestic environments," in *Proc. of EVALITA*, 2014, pp. 147–152.

[16] Samuel Thomas, Sriram Ganapathy, George Saon, and Hagen Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. Citeseer, 2014, pp. 2519–2523.

[17] Tribikram Kundu, "Acoustic source localization," *Ultrasonics*, vol. 54, no. 1, pp. 25 – 38, 2014.

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

[19] B. Champagne, S. Bedard, and A. Stephenne, "Performance of time-delay estimation in the presence of room reverberation," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 2, pp. 148–152, 1996.

[20] A. Tsiami, A. Katsamanis, P. Maragos, and G. Potamianos, "Experiments in acoustic source localization using sparse arrays in adverse indoors environments," in *Proc of EUSIPCO*, Lisbona, Portugal, Sep 1-5 2014, IEEE, pp. 2390–2394.

[21] M. S. Brandstein and H. F. Silverman, "A robust method for speech signal time-delay estimation in reverberant rooms," in *Proc. of ICASSP*, Munich, Germany, Apr 1997, vol. 1, pp. 375–378 vol.1.

[22] Alessio Brutti, Maurizio Omologo, Piergiorgio Svaizer, and Christian Zieger, "Classification of acoustic maps to determine speaker position and orientation from a distributed microphone network," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2007, vol. 4, pp. IV–493.

[23] Dongsuk Yook, Taewoo Lee, and Youngkyu Cho, "Fast sound source localization using two-level search space clustering," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 20–26, 2016.

[24] Hoang Do, Harvey F Silverman, and Ying Yu, "A real-time SRP-PHAT source location implementation using stochastic region contraction (SRC) on a large-aperture microphone array," in *Proc. of ICASSP*, 2007, vol. 1, pp. I–121.

[25] P. Transfeld, U. Martens, H. Binder, T. Schypior, and T. Fingscheidt, "Acoustic event source localization for surveillance in reverberant environments supported by an event onset detection," in *Proc. of ICASSP*, Brisbane, Australia, 19-24 Apr. 2015, pp. 2629–2633.

[26] Pierre Zakarauskas, John M Ozard, and Peter Brouwer, "Artificial neural networks for simultaneous and independent range and depth discrimination in passive acoustic localization.," *The Journal of the Acoustical Society of America*, vol. 90, no. 4, pp. 2366–2366, 1991.

[27] Mehdi Banitalebi Dehkordi, Hamid Reza Abutalebi, and Hossein Ghanei, "A compressive sensing based compressed neural network for sound source localization," in *Proc. the Int Symp. on Artificial Intelligence and Signal Processing*, 2011, pp. 6–10.

[28] Soumitro Chakrabarty and Emanuël AP Habets, "Multi-speaker localization using convolutional neural network trained with noise," *arXiv preprint arXiv:1712.04276*, 2017.

[29] Eric L Ferguson, Stefan B Williams, and Craig T Jin, "Sound source localization in a multipath environment using convolutional neural networks," *arXiv preprint arXiv:1710.10948*, 2017.

[30] Michael S Datum, Francesco Palmieri, and Andrew Moiseff, "An artificial neural network for sound localization using binaural cues," *The Journal of the Acoustical Society of America*, vol. 100, no. 1, pp. 372–383, 1996.

[31] E. Mumolo, M. Nolich, and G. Vercelli, "Algorithms for acoustic localization based on microphone array in service robotics," *Robotics and Autonomous Systems*, vol. 42, no. 2, pp. 69 – 88, 2003.

[32] John C Murray and Harry R Erwin, "A neural network classifier for notch filter classification of sound-source elevation in a mobile robot," in *Proc. of IJCNN*. IEEE, 2011, pp. 763–769.

[33] Ning Ma, Tobias May, and Guy J Brown, "Exploiting deep neural networks and head movements for robust binaural localization of multiple sources in reverberant environments," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 12, pp. 2444–2453, 2017.

[34] N. Ma, J. A. Gonzalez, and G. J. Brown, "Robust binaural localization of a target sound source by combining spectral source models and deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 11, pp. 2122–2131, 2018.

[35] Weipeng He, Petr Motlicek, and Jean-Marc Odobez, "Deep neural networks for multiple speaker detection and localization," *arXiv preprint arXiv:1711.11565*, 2017.

[36] K. D. Martin, "Estimating azimuth and elevation from interaural difference," in *Proc. 1995 Workshop on Applications of Signal Processing to Audio and Accoustics*, 1995, pp. 96–99.

[37] C. Lim and R. Duda, "Estimating the azimuth and elevation of a sound source from the output of a cochlear model," in *Proc. 28th Asilomar Conference on Signals, Systems and Computers*, 1994.

[38] H. O'Dwyer, E. Bates, and F. M. Boland, "A machine learning approach to detecting sound-source elevation in adverse environments," in *Audio Engineering Society Convention 144*, May 2018.

[39] H. O'Dwyer, E. Bates, and F. M. Boland, "Machine learning for sound elevation detection," in *Proc. 4th Workshop on Intelligent Music Production*, Sep 2018.

[40] Tobias May, Steven van de Par, and Armin Kohlrausch, "A binaural scene analyzer for joint localization and recognition of speakers in the presence of interfering noise sources and reverberation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 20, no. 7, pp. 2016–2030, 2012.

[41] Rupayan Chakraborty and Climent Nadeu, "Joint model-based recognition and localization of overlapped acoustic events using a set of distributed small microphone arrays," *arXiv preprint arXiv:1712.07065*, 2017.

[42] Giuseppe Valenzise, Luigi Gerosa, Marco Tagliasacchi, Fabio Antonacci, and Augusto Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *Proc. of AVSS*, 2007, pp. 21–26.

[43] S. Adavanne, S. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *arXiv preprint arXiv:1807.00129*, 2018.

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

[45] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

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

[47] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[48] Ian McLoughlin and Yan Song, "Low frequency ultrasonic voice activity detection using convolutional neural networks," in *Proc. of Interspeech*, Dresden, Germany, Sep. 6-10 2015.

[49] François Chollet et al., "Keras," `https://keras.io`, 2015.

[50] Jun Han and Claudio Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *International Workshop on Artificial Neural Networks*. Springer, 1995, pp. 195–201.

[51] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947, 2000.

[52] Nasser M Nasrabadi, "Pattern recognition and machine learning," *Journal of electronic imaging*, vol. 16, no. 4, pp. 049901, 2007.

[53] John Duchi, Elad Hazan, and Yoram Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[54] Matthew D Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[55] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[56] 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.

[57] A. r. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc of ICASSP*, Kyoto, Japan, Mar 25-30 2012, pp. 4273–4276.

[58] Karen Ullrich, Jan Schlüter, and Thomas Grill, "Boundary detection in music structure analysis using convolutional neural networks," in *Proc of ISMIR*, Taipei, Taiwan, Oct 27-31 2014, pp. 417–422.

[59] D. J. Hermes, "Measurement of pitch by subharmonic summation," *The journal of the acoustical society of America*, vol. 83, no. 1, pp. 257–264, 1988.

[60] Erik Marchi, Giacomo Ferroni, Florian Eyben, Leonardo Gabrielli, Stefano Squartini, and Bjorn Schuller, "Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2164–2168.

[61] Hynek Hermansky and Nelson Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, 1994.

[62] Niko Moritz, Jörn Anemüller, and Birger Kollmeier, "Amplitude modulation spectrogram based features for robust speech recognition in noisy

and reverberant environments," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5492–5495.

[63] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 4, pp. 320–327, 1976.

[64] C. Zhang, D. Florencio, and Z. Zhang, "Why does PHAT work well in lownoise, reverberative environments?," in *Proc. of ICASSP*, Las Vegas, USA, 2008, pp. 2565–2568.

[65] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," in *Proc. of ICASSP*, Brisbane, Australia, Apr 19-24 2015, IEEE, pp. 2814–2818.

[66] Luca Cristoforetti, Mirco Ravanelli, Maurizio Omologo, Alessandro Sosi, Alberto Abad, Martin Hagmüller, and Petros Maragos, "The dirha simulated corpus.," in *LREC*, 2014, pp. 2629–2634.

[67] C. Hummersone, R. Mason, and T. Brookes, "Dynamic precedence effect modeling for source separation in reverberant environments," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 18, no. 7, pp. 1867–1871, 2010.

[68] G. Kearney and T. Doyle, "An HRTF database for virtual loudspeaker rendering," in *Audio Engineering Society Convention 139*, Oct 2015.

[69] D. T. Murphy and S. Shelley, "OpenAIR: An interactive auralization web resource and database," in *Audio Engineering Society Convention 129*, Nov 2010.

[70] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT Acoustic-phonetic continuous speech corpus CD-ROM," *National Inst. Standards and Technol. (NIST)*, 1993.

[71] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[72] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely,

"The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[73] Florian Eyben, Felix Weninger, Stefano Squartini, and Björn Schuller, "Real-life Voice Activity Detection with LSTM Recurrent Neural Networks and an Application to Hollywood Movies," in *Proc. of ICASSP*, Vancouver, BC, Canada, May 26-31 2013, pp. 483–487.

[74] Florian Eyben, Felix Weninger, Florian Groß, and Björn Schuller, "Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor," in *Proc. of ACM Int. Conf. on Multimedia*, Barcelona, Spain, Oct. 21-25 2013, pp. 835–838.

[75] Felix Weninger, Johannes Bergmann, and Björn Schuller, "Introducing CURRENNT, the Munich open-source CUDA RecurREnt Neural Network Toolkit," *Journal of Machine Learning Research*, pp. 547–551, 2014.

[76] Noel Lopes and Bernardete Ribeiro, "Towards adaptive learning with improved convergence of deep belief networks on graphics processing units," *Pattern Recogn.*, vol. 47, no. 1, pp. 114–127, Jan. 2014.

[77] Joseph H DiBiase, Harvey F Silverman, and Michael S Brandstein, "Robust localization in reverberant rooms," in *Microphone Arrays*, pp. 157–180. Springer, 2001.

[78] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[79] A. Stéphenne and B. Champagne, "A new cepstral prefiltering technique for estimating time delay under reverberant conditions," *Signal Processing*, vol. 59, no. 3, pp. 253–266, 1997.

[80] Y. Sun, J. Chen, C. Yuen, and S. Rahardja, "Indoor sound source localization with probabilistic neural network," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6403–6413, 2018.

[81] N. Ma, T. May, and G. J. Brown, "Exploiting deep neural networks and head movements for robust binaural localisation of multiple sources in reverberant environments," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 12, pp. 2444–2453, 2017.

[82] J. Blauert, *Spatial hearing - The psychophysics of human sound localization*, The MIT Press, Cambride, MA, USA, 1997.

[83] B. Grothe, M. Pecka, and D. McAlpine, "Mechanisms of sound localization in mammals," *Physiol. Rev.*, vol. 90, pp. 983–1012, 2010.

[84] Tara N Sainath, Ron J Weiss, Kevin W Wilson, Bo Li, Arun Narayanan, Ehsan Variani, Michiel Bacchiani, Izhak Shafran, Andrew Senior, Kean Chin, et al., "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, 2017.

[85] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio.," in *SSW*, 2016, p. 125.

[86] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," *arXiv preprint arXiv:1612.07837*, 2016.

[87] R. D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C.Zhang, and M. H. Allerhand, "Complex sounds and auditory images," *Auditory Physiology and Perception, (Eds.) Y. Cazals, L. Demany, K.Horner, Pergamo*, pp. 429–446, 1992.

[88] D. L. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*, Wiley/IEEE Press, 2006.

[89] G. Ehret, "Stiffness gradient along the basilar membrane as a basis for spatial frequency analysis within the cochlea," *The Journal of the Acoustical Society of America*, vol. 64, no. 6, pp. 1723–1726, 1978.

[90] W. Yost, "Structure of the inner ear and its mechanical response," in *Fundamentals of Hearing: An Introduction (5th edition)*, chapter 7, pp. 83–101. Brill, 2013.

[91] T. May, S. van de Par, and A. Kohlrausch, "A probabilistic model for robust localization based on a binaural auditory front-end," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 19, no. 1, pp. 1–13, 2011.

[92] T. M. Nguyen, *The effects of target spectrum, noise, and reverberation on auditory cue weighting in sound localization*, Ph.D. thesis, University of Western Ontario, 2014.

[93] V. R. Algazi, C. Avendano, and R. O. Duda, "Elevation localization and head-related transfer function analysis at low frequencies," *J. Acoust. Soc. Amer.*, vol. 109, no. 3, pp. 1110–1122, 2001.

[94] H. A. Schnyder, D. Vanderelst, S. Bartenstein, U. Firzlaff, and H. Luksch, "The avian head induces cues for sound localization in elevation," *PLOS ONE*, vol. 9, no. 11, pp. 1–8, 11 2014.

[95] CL Searle, LD Braida, DR Cuddy, and MF Davis, "Binaural pinna disparity: another auditory localization cue," *The Journal of the Acoustical Society of America*, vol. 57, no. 2, pp. 448–455, 1975.

[96] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *Proc. of ICASSP*, 2016, pp. 405–409.

[97] S. Chakrabarty and E. A. P. Habets, "Multi-speaker localization using convolutional neural network trained with noise," in *NIPS 2017 Workshop on Machine Learning for Audio Processing*, 2017.

[98] T. Rodemann, G. Ince, F. Joublin, and C. Goerick, "Using binaural and spectral cues for azimuth and elevation localization," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2008.

[99] Roland Maas, Sree Hari Krishnan Parthasarathi, Brian King, Ruitong Huang, and Björn Hoffmeister, "Anchored speech detection.," in *INTERSPEECH*, 2016, pp. 2963–2967.

[100] Panagiotis Giannoulis, Alessio Brutti, Marco Matassoni, Alberto Abad, Athanasios Katsamanis, Miguel Matos, Gerasimos Potamianos, and Petros Maragos, "Multi-room speech activity detection using a distributed microphone network in domestic environments," in *Proc. of EUSIPCO*, 2015, pp. 1271–1275.

[101] Dan Goodman and Romain Brette, "Learning to localise sounds with spiking neural networks," in *Adv. Neural Inf. Process. Syst.*, 2010, pp. 784–792.

[102] Kuba Lopatka, Jozef Kotus, and Andrzej Czyzewski, "Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations," *Multimedia Tools and Applications*, vol. 75, no. 17, pp. 10407–10439, 2016.

[103] 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.

[104] 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.

[105] Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney, "Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[106] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 9, pp. 1469–1477, 2015.

[107] Anton Ragni, Kate M Knill, Shakti P Rath, and Mark JF Gales, "Data augmentation for low resource languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[108] Huy Dat Tran, Wen Zheng Terence Ng, and Yi Ren Leng, "Data augmentation, missing feature mask and kernel classification for through-the-wall acoustic surveillance," *Proc. Interspeech 2017*, pp. 3807–3811, 2017.

[109] Matthias Zöhrer and Franz Pernkopf, "Virtual adversarial training and data augmentation for acoustic event detection with gated recurrent neural networks," *Proc. Interspeech 2017*, pp. 493–497, 2017.

[110] Justin Salamon and Juan Pablo Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[111] Masakiyo Fujimoto and Kentaro Ishizuka, "Noise robust voice activity detection based on switching kalman filter," *IEICE transactions on information and systems*, vol. 91, no. 3, pp. 467–477, 2008.

[112] Kohei Hayashida, Masanori Morise, and Takanobu Nishiura, "Near field sound source localization based on cross-power spectrum phase analysis with multiple microphones," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[113] Hoang Do, Harvey F Silverman, and Ying Yu, "A real-time srp-phat source location implementation using stochastic region contraction (src) on a large-aperture microphone array," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on.* IEEE, 2007, vol. 1, pp. I–121.

[114] Sunit Sivasankaran, Emmanuel Vincent, and Douglas R Campbell, *Room Impulse Response Generator*, https://github.com/sunits/rir_simulator_python.

[115] Jont B Allen and David A Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[116] Wu-Sheng Lu, Soo-Chang Pei, and Chien-Cheng Tseng, "A weighted least-squares method for the design of stable 1-D and 2-D IIR digital filters," *IEEE Transactions on Signal Processing*, vol. 46, no. 1, pp. 1–10, Jan 1998.

[117] T. I. Laakso, M. Lang, and T. Saramaki, "Design of limit-cycle-free recursive transfer functions for fixed-point direct form implementation," in *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, May 1994, vol. 2, pp. 477–480 vol.2.

[118] P. Vaidyanathan, S. Mitra, and Y. Neuvo, "A new approach to the realization of low-sensitivity IIR digital filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 2, pp. 350–361, Apr 1986.

[119] Markku Renfors, "A class of approximately linear phase digital filters composed of allpass subfilters," in *Proc. IEEE Int. Symp. Circuits & Syst.*, 1986, pp. 678–681.

[120] Xi Zhang and Hiroshi Iwakura, "Design of iir digital allpass filters based on eigenvalue problem," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 554–559, 1999.

[121] A Djebbari, Jean Michel Rouvaen, AL Djebbari, M Faouzi Belbachir, and Sid Ahmed Elahmar, "A new approach to the design of limit cycle-free iir digital filters using eigenfilter method," *Signal Processing*, vol. 72, no. 3, pp. 193–198, 1999.

[122] B. Jaworski and T. Saramaki, "Linear phase IIR filters composed of two parallel allpass sections," in *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, May 1994, vol. 2, pp. 537–540 vol.2.

[123] Chien-Cheng Tseng and Su-Ling Lee, "Design of linear phase fir filters using fractional derivative constraints," *Signal Processing*, vol. 92, no. 5, pp. 1317–1327, 2012.

[124] Sheng Chen and Bing L. Luk, "Digital IIR filter design using particle swarm optimisation," vol. 9, 05 2010.

[125] Nurhan Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328 – 348, 2009.

[126] N. Agrawal, A. Kumar, and Varun Bajaj, "Design of digital IIR filter with low quantization error using hybrid optimization technique," *Soft Computing*, Mar 2017.

[127] Alok Pandey and Santosh Sharma, "FIR Filter Design and Analysis Using Neural Network," *International Journal of Engineering Research and General Science*, vol. 3, no. 1, pp. 297–301, Jan-Feb 2015.

[128] Lo-Chyuan Su, Yue-Dar Jou, Fu-Kun Chen, and Chao-Ming Sun, "Neural Network-Based IIR All-Pass Filter Design," *Circuits, Systems, and Signal Processing*, vol. 33, no. 2, pp. 437–457, Feb 2014.

[129] Amir A. Bature and Sunusi S. Adamu, "Design of Digital Recursive Filter Using Artificial Neural Network ," *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, March 2012.

[130] A. G. Parlos, S. K. Menon, and A. Atiya, "An algorithmic approach to adaptive state filtering using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1411–1432, Nov 2001.

[131] Nikhil Agrawal, Anil Kumar, and Varun Bajaj, "A new design method for stable iir filters with nearly linear-phase response based on fractional derivative and swarm intelligence," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 6, pp. 464–477, 2017.

[132] Henri Korhola and Matti Karjalainen, "Perceptual study and auditory analysis on digital crossover filters," *Journal of the Audio Engineering Society*, vol. 57, no. 6, pp. 413–429, 2009.

[133] Soo-Chang Pei, Chien-Cheng Tseng, and Wen-Sing Yang, "Fir filter designs with linear constraints using the eigenfilter approach," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 2, pp. 232–237, 1998.