POLITECNICO
MILANO 1863

Dipartimento di Elettronica, Informazione e Bioingegneria

Master Degree in Computer Science and Engineering

A model selection method for room shape classification based on mono speech signals

by:
Gabriele Antonacci

matr.:
915289

Supervisor:
Prof. Fabio Antonacci

Co-supervisors:
Ing. Clara Borrelli

Academic Year
2020-2021

Abstract

Each audio recording contains a huge amount of forensic traces. In principle, from the analysis of a speech recording, it is possible to extract details that range from the estimation of the source device used for its acquisition, to environmental characteristics. However, studies in the acoustic analysis and audio forensic fields throw light on the lack of tools for geometrical classification based on reverberant speech audio.
Despite acoustic analysis algorithms for estimating parameters like volume, reverberation time and enclosure type (e.g. small room, hall, stadium) were investigated in the literature, there is still room for improvement.

This study aims at widening the set of possible room parameters which can be considered for audio analysis. In detail, it introduces the topic of room shape classification based on reverberant speech signals.

The proposed methodology fixes some volume and reverberation time bands to enhance the performances of the shape classifiers and to establish a relationship among volume and geometry estimation performances. To do so, either a preliminary volume or a reverberation time estimation is performed to retrieve a band index estimate. Depending on the band index estimate, we retrieve the best shape classification model. Such estimator is used to prove whether a speech signal has been acquired in a room of a certain shape.

Our research shows promising results even in the reverberant speech study case. However, we observe some difficulties in dealing with rooms of small size.
Future work might consider introducing accurate free decay region extractors or using time-aware neural networks.

Sommario

Ogni acquisizione audio contiene grandi quantità di tracce forensi.
Dall'analisi di registrazioni vocali è possibile estrarre sia informazioni relative al dispositivo con qui l'acquisizione è effettuata che informazioni sulle caratteristiche dell'ambiente in cui la registrazione è effettuata. In ogni caso, gli studi nei campi di analisi acustica ed audio forense gettano luce sull'assenza di tools per la classificazione geometrica basata su segnali vocali riverberanti.
Nonostante nella letteratura siano stati indagati algoritmi di analisi acustica per stimare parametri come volume, tempo di riverberazione e tipo di ambiente (p.es. se una piccola stanza, una hall od uno stadio), c'è ancora vasto margine di miglioramento.

Questo studio mira ad amplicare l'insieme dei parametri di una stanza che possono essere presi in considerazione per effettuare un'analisi audio. In dettaglio, introduce l'argomento di classificazione della forma di una stanza basata su segnali vocali riverberanti.

Il metodo proposto fissa delle bande di volume e di tempo di riverberazione per migliorare le performance dei classificatori di forma e per stabilire una relazione tra le metriche nella stima del volume e nella stima della geometria.
Per poterlo fare, volume e tempo di riverberazione sono stimati in via preliminare per determinare la stima di un indice di banda. Dipendentemente da questo indice, siamo in grado di determinare il miglior modello per la classificazione della forma. Questo stimatore è usato per asserire se un segnale vocale sia o meno stato acquisito in una stanza di una determinata forma.

La nostra ricerca mostra risultati promettenti anche nel caso di segnali vocali riverberanti. In ogni caso, riscontriamo difficoltà nel gestire stanze di piccole dimensioni.
Futuri sviluppi potrebbero contemplare l'introduzione di estrattori di regioni di decadimento libero accurati o, ancor meglio, l'uso di time-aware neural networks.

Acknowledgements

Firstly, I would like to thank my whole family.
My parents, for their sacrifices and encouragements.
My sister, for her presence, and my nephew, popping up from nowhere with joy and tenderness.
My uncles for the demonstrated interest and sustain, my cousins and my grandma.

I am grateful to all the Professors and assistants of this masters degree, especially to the supervisors, who spent time in helping me through this experience, although with some disadvantages related to the pandemics.

What is more, my thanks go to my friends. The nearest ones: the Parampa'mpoli-maniacs and the Pirates, the farther, the ones belonging to my memories, and the colleagues, with whom I shared swaying moments of full excitement and deepest anxiety.

Last, but not least, my special thanks go to the students association Polifonia. I am glad I had the opportunity to meet new musicians and to share lively emotions with them while playing.
I am so thankful to the members of the board of direction, with whom I worked and grew during the past years.

I will always hold all of you in my heart.

"We must let go of the life we have planned,
so as to accept the one that is waiting for us."
Joseph Campbell

Contents

List of Figures

List of Tables

Acronyms

γ-FB Gammatone Filter-Bank. 12, 18, 32, 74, 81, 86

Adam Adaptive Moment Estimation. 25, 57

AEI Acoustic Environment Identification. 13

ANN Artificial Neural Network. 19

ASC Acoustic Scene Classification. 13

BP Back-Propagation. 25

Cepst Cepstum. 18, 32

CNN Convolutional Neural Network. 13, 14, 22–24, 26, 31, 88

COTA COmmon-TAngent. 16

DCT Discrete Cosine Transform. 18

DFT Discrete Fourier Transform. 17, 18, 32

DL Deep Learning. 19, 27

DNN Deep Neural Network. 22

DRR Direct-to-Reverberation Ratio. 12

EDC Energy Decay Curve. 11

EDM Euclidean-Distance Matrix. 17

ELR Early-to-Late Reverberation Ratio. 12

ENV Envelope. 19, 32

ESS Exponential Sine Sweep. 6

FB Filter-Bank. 18

FC Fully-Connected. 22, 23, 33

FDR Free-Decay Region. 11–13, 89

FIR Finite Impulse Response. 7

GA Genetic Algorithm. 14

Gabor-FB Gabor Filter-Bank. 12, 24, 35, 57

GAP Global Average Pooling. 26

GD Gradient Descent. 25, 41, 42

GMM Gaussian Mixture Model. 13

Grad-CAM Gradient-based Class Activation Mapping. 26, 27, 31, 58, 84–87

H Cross-entropy. 21, 58, 65, 75

IIR Infinite Impulse Response. 7

IS Image-Source. 7, 9, 14, 40

ISM Image-Source Model. 7–9, 17, 40, 42, 45

KLD Kullback-Leibler Divergence. 21

LEAF LEarnable Audio Front-end. 23, 36

LR Learning Rate. 25, 26, 58, 62, 64, 74

LSTM Long Short-Term Memory. 89

LTI Linear Time-Invariant. 4, 6

MAE Mean Absolute Error. 20, 57

MBConv Mobile inverted Bottleneck Convolutional. 36

MFC Mel-Frequency Cepstrum. 18

MFCC Mel-Frequency Cepstral Coefficient. 13

MLP Multi-Layer Perceptron. 12, 23

MLS Maximum Length Sequence. 6

MSE Mean Squared Error. 20, 57, 59

NN Neural Network. 23, 26, 85

PCEN Per-Channel Energy Normalization. 24, 35

PRA Pyroomacoustics. 38–40, 42

ReLU Rectified Linear Unit. 24, 35

RGIDA Room Geometry Inference - Deterministic front-end Architecture. 57–59, 62, 65, 67, 70, 73–75, 80, 83, 84, 86, 88, 89

RGILA Room Geometry Inference - Learnable front-end Architecture. 57, 58, 65, 67, 70, 73, 75, 80, 83, 84, 86, 88

RIR Room Impulse Response. 5–7, 10, 11, 13–17, 28, 31, 38, 40–43, 45–47, 59, 62, 84, 86, 89

RNN Recurrent Neural Network. 12, 89

RT Reverberation Time. 10–13, 62, 74

SGD Stochastic Gradient Descent. 25

STFT Short-Time Fourier Transform. 18, 89

TDoA Time-Difference of Arrival. 15, 16

ToA Time of Arrival. 16, 17

ToF Time of Flight. 6

UML Unified Modeling Language. 38

VLGA Variable Length Genetic Algorithm. 26

1
Introduction

The purpose of this master thesis is to provide a methodology for the classification of the shape of a room environment starting from a reverberant speech signal.
Our work might have a multiplicity of applications, however the most relevant ones can be found in the acoustic analysis and audio forensics fields.

Speaking about acoustic analysis, we might refer to room imaging and to the related room geometry inference techniques adopted for it. Such techniques exploit the reverberant properties of audio files to perform a room model reconstruction. Indeed, the geometry of an enclosure affects the way in which sound is perceived. The reflective behavior of soundwaves in air and their affection on audio signals have been extensively described in [5, 6, 7, 8].
The existing methods for room imaging [2, 3, 9, 10] are complex, therefore we might think of a simplified break-down strategy for the geometrical model retrieval. The floor plan shape and the shape variables could be estimated, for instance.
Other possible applications regard the currently existing field synthesis techniques. They could be extended to automatically adapt the speakers output depending on room parameters estimates to improve the surround experience of the listeners.

Changing perspective, we can also consider the audio forensics study field. Following this branch we take into account the integrity checking techniques. These algorithms are focused on proving whether an audio file has been compromised for malicious sake. For example it might be useful to certify whether an audio track has been truly acquired in a cer-

tain environment for legal purposes. Alternatively, if the environmental properties of a recording are detected as changing overtime, the considered audio document could have been forged.
Nowadays, the main environmental parameters which are considered are the reverberation time T_{60} [1, 11], the type of acoustic environment [12, 13, 14] (e.g. small room, hall, stadium) and the room spatial volume [15]. Our work goes in the direction of augmenting the set of environmental characteristics which might be considered. To do so, we introduce a shape estimator.

In our work we estimate the room volume and the T_{60} parameters using a couple of different architectures for the estimators. These estimators are based on recent works [15, 16]. However, many different approaches for estimating such properties have been proposed in the past.

In detail, our approach is structured according to the following outline. Firstly, we present a data generation framework which allows the generation of the datasets which are fundamental for the validation of the algorithmic pipeline. Such datasets are generated via simulation because we could not find datasets labeled with rooms shapes.
In a second step, we adopt a solution which works per bands. Either a preliminary volume or reverberation time estimation is performed to retrieve a band index estimate. Then, given the band index estimate, we consider a set of specific estimators trained on volume or reverberation time bands and we choose the best estimator for the shape classification depending on the band index estimate. We perform the classification task by considering two different architectures for the specific in-band estimators to compare their performances.
The results reported from the application of our methodology to reverberant speech signals show promising results. We observe a classification performance improvement getting higher as the volume of the considered rooms enlarges.
Our proposal accounts for a variety of advantages. Mainly, using a band approach we are able to gain estimation accuracy and to establish a relationship among volume and shape classification performance. Furthermore, we introduce a new room parameter among the ones considered by audio forensics. This implies the need of new efforts by malicious parties in manipulating audio data.

This thesis is organized as follows.

In the second chapter (Chapter 2), we are going through the literature related to room geometry inference and room parameters estimation techniques. We start from the concept of reverberation: we begin with the definition of reverberant signal and we analyze the relationship of reverberation with the environment. Afterwards, we describe related state of the art works which are linked to reverberation parameters and room parameters estimation in an audio forensics frame. Considering the geometrical purpose of our research, we also dig into documents related to room geometry inference, discovering the absence of a statistical tool capable of estimating the floor plan shape of an inner-space. Finally, we provide a brief walk-through in the theory of deep learning techniques to put the ground for the comprehension of the whole work of thesis.

In Chapter 3, we give a formal problem formulation and we describe our methodology developed to solve it. In detail, the methodology encompasses our band-based algorithmic pipeline for room shape estimation.

Chapter 4 contains details about the simulations for data generation, the experiments and the resulting reports of our method implementation for the problem resolution.

In the end, within the context of conclusions, Chapter 5 exposes the summary of the results of our research and puts forward possible developments for future improvements and extensions.

2

State of the Art and Theoretical Background

This chapter introduces the core topic of reverberation and the models which are useful for the comprehension of our work of thesis.
Furthermore, in this chapter we highlight the purpose of room geometry estimation, its relevance, its complexity and the potentialities of our approach. In detail, our work could be used both in the audio forensics field for integrity checking and in the acoustic analysis field for room reconstruction.
Finally, we glance in some theoretical aspects of artificial neural networks.

2.1 Reverberation and spatial cues

The reverberation is the physical phenomenon describing the propagation of a sound-wave (in space and time) in relation with the boundaries of the surrounding space.
Indeed, the wave might impinge against walls, being reflected or scattered around. As a simplification, in our work we are going to neglect scattering and to model reflection and absorption as independent on frequency.
Considering the environment among source and destination as a Linear Time-Invariant (LTI) system, its acoustic response can be described by the room impulse response, which affects the sound perceived at the listener location.

2.1.1 Reverberant signals

The signal received from a microphone in position $\mathbf{m}$ is the sound played by a source in position $\mathbf{s}$ on which is applied the effect of the room between source and microphone. Let $x_{\mathbf{s}}(t)$ be the signal emitted by the source and $h_{\mathbf{s}\to\mathbf{m}}(t)$ be the effect of the room, then the signal received by the microphone in ideal conditions is:

$$y_{\mathbf{m}}(t) = (x_{\mathbf{s}} \underline{*} h_{\mathbf{s}\to\mathbf{m}})(t) = \int_{-\infty}^{+\infty} x_{\mathbf{s}}(t') h_{\mathbf{s}\to\mathbf{m}}(-t'+t)dt' \qquad (2.1)$$

where $\underline{*}$ is the 1D convolution operator.
Such signals can either be directly acquired from a physical setup with loudspeakers and microphones, in which case the recorded signal would encode even the equipment response, or digitally generated given the knowledge of the Room Impulse Response (RIR).

2.1.2 Room impulse response

The RIR is an audio signal which encodes the room acoustical behavior in between a pair of points belonging to the environment.
In its simplest form, the RIR can be described as the superposition of delayed peaks, each of which with a different amplitude, acquired at the fixed receiver position with a fixed transmitter position. The phase-shift of each peak mainly depends on the space traveled by the wave or reflected wave to reach the microphone, while the amplitude depends primarily on the number of reflections on the path and on the absorption of each reflector.

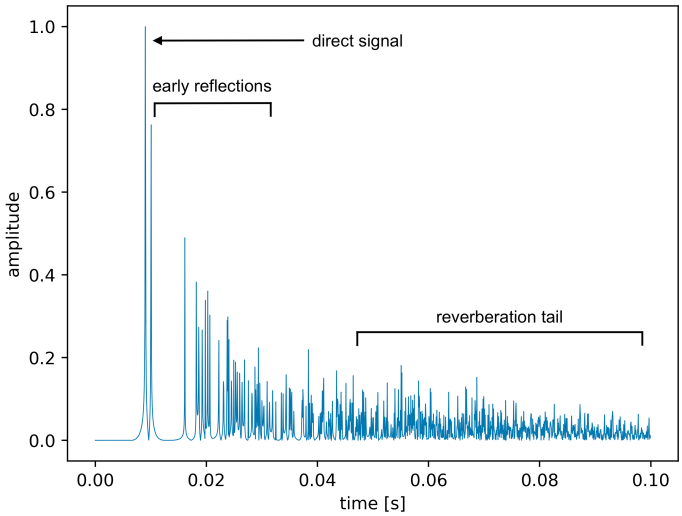


Figure 2.1: RIR temporal regions

As can be seen in Figure 2.1, the density of peaks in a RIR increases with time. Such an enhancement of density is related to the number of

replicas reaching the receiver. The longer the elapsed time, the more the reflections which are intercepted.
A RIR acquired in a polyhedral geometry (with no convexity in it) can be fairly divided in three regions:

1. Direct path echo: appears at the Time of Flight (ToF) instant, which in terms is proportional to the distance in between the source and the microphone (given stable environmental conditions);

2. Early reflection echoes: encode the time-shifts related to the first reflections of the wave hitting against the walls. This region encodes relevant spatial cues related to the perception of space around the listener, including the shape of the room;

3. Late reverberation tail: encoding the time-shifts related to the higher order reflections.

For what concerns convex geometries, the visibility of source and receiver cannot be ensured, therefore the direct path echo might be missing.
The impulse response measures the temporal behavior of a LTI system among an input and an output point. The RIR among spatial points can be expressed as:

$$h_{\mathbf{s}\to\mathbf{m}}(t) = h(t \,|\, \mathbf{s}, \mathbf{m}) = \sum_{i=0}^{i=+\infty} \alpha_i \delta(t - \tau_i) \tag{2.2}$$

where $\mathbf{s}$ and $\mathbf{m}$ represent respectively the known source and microphone positions, τ_i is the time lag at which the i^{th} echo is received and α_i is an amplitude attenuation factor. In the digital domain, fixed a sampling frequency, h is a finite tapped delay-line approximating the real infinite response.
The RIRs signals can be either physically acquired or obtained via algorithmic simulation.

2.1.2.1 Physical acquisition

In the past, the RIRs were acquired fixing the position of omni-directional microphones in a room and emulating a pulsive input signal for the spatial LTI system. This signal, coming from the explosion of balloons or from the firing of blank pistols, resembled a Dirac signal. Therefore, it allowed to measure the response from source to microphones. Other approaches exploited a loudspeaker (emitting either Maximum Length Sequence (MLS), Exponential Sine Sweep (ESS) or other signals) and a microphone, both independently moving in space, followed by appropriate deconvolutive techniques.
Nowadays, the applied techniques are in principle similar to the ones adopted in the past, but they exploit also new technological improvements. Indeed, dodecahedron loudspeakers and directional and soundfield microphones are used for multi-channel playbacks and acquisitions.

What is more, the acquisition procedures are steering towards the usage of loudspeakers and microphone arrays capable of synthesizing and analysing soundfields by encoding weights of spherical harmonics patterns (lower and higher order Ambisonics techniques).
We recall that our aim is to estimate unconventional geometries (considering other room shapes aside the rectangular one). To our knowledge the existing datasets containing RIRs have been acquired in rooms of different sizes generally with a rectangular floor plan, thereby it is worth digging into methods for simulating RIRs.

2.1.2.2 Virtual generation

The following techniques for RIRs generation belong to the geometrical acoustic modeling techniques category and can be exploited in a hybrid fashion to build the full response. Although in Equation 2.2 we gave an Infinite Impulse Response (IIR) definition of the RIR, with full response we mean a sufficiently long digital Finite Impulse Response (FIR) approximating the former one.

Early reverberation

Image-Source Model For Allen and Berkley in [5], the Image-Source Model (ISM) allows to discard the usage of counter-intuitive algorithms for the computation of a RIR, preferring instead an image method.

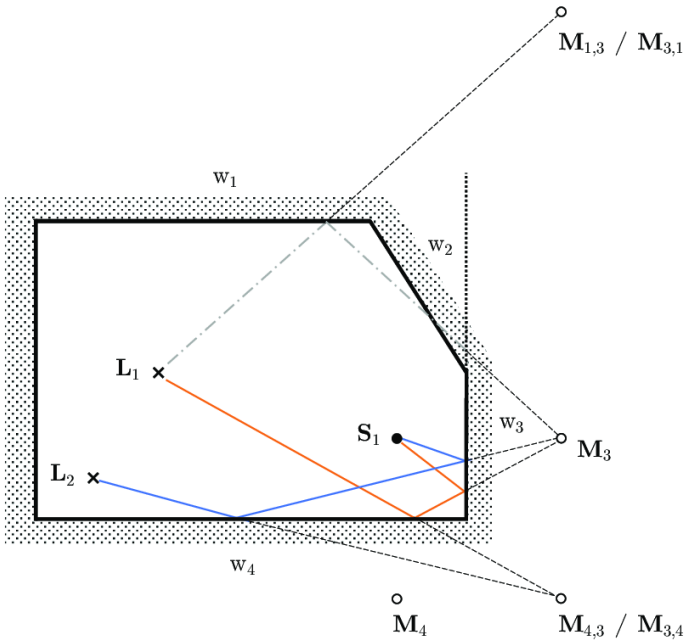


Figure 2.2: ISM application example

In Figure 2.2, we show some of the first ($\mathbf{M_i}$) and second ($\mathbf{M_{j,k}}$ for $\mathbf{j \neq k}$) order Image-Sources (ISs) generated from the primary source $\mathbf{S_1}$

and affecting the listeners $\mathbf{L_{1,2}}$ by impinging against an ordered set of walls $\mathbf{w}$.
The soundfield within the boundaries, at the listener's position, can be approximated by the superposition of the direct field and the field coming from the secondary sources which are out of the room emitting the same signal of the primary source. Explained away the delay factors in Equation 2.2 due to the distance of each primary or secondary source with respect to the receiver, we are left with the attenuation factors. The wave generated from the reflector at the bouncing point is going to have lower energy because of the wall energetic absorption. This loss is accumulated along the path at each reflection, resulting in a final peak amplitude perceived at the destination.
Interestingly enough, fixed a desired sampling rate F_s for the RIR and the number of reflectors of a given geometry W, the computational time follows $t_{comp} \in \mathcal{O}(W^o)$ for o a variable maximum order of reflection. As a consequence, the longer the desired response, the greater the maximum order and the computational cost. A similar issue can be found also in [17] for arbitrary polyhedra.
Apart from optimizations for the shoebox shaped room, which allow to save time, we forsee the need of another approach for the approximate generation of responses in rooms of different geometries.

Beam Tracing In the *beam tracing* algorithms category, we can find approaches close to the ISM approach (or to the *ray tracing* approach described in the following paragraph).

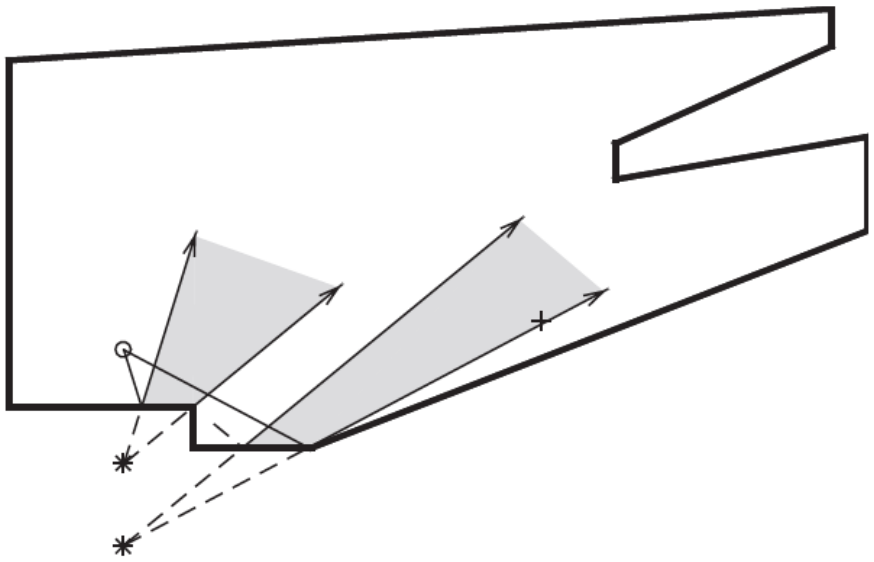


Figure 2.3: Beam tracing example

As we can see from Figure 2.3, these algorithms are based upon the firing of beams (volumetric objects, generally of pyramidal or conical cross-section) in different spatial directions from the source position, in

order to determine possible mirroring paths.
With respect to ISM, the goal is to reduce the computational cost for the calculation of the early reflections, by limiting the growth of ISs. Similarly to ISM, in these approaches the number of beams increases at each reflection, however, for efficiency, the number of beams is minimized by pruning the ISs tree as soon as possible.
The early implementations are discussed by Funkhouser *et al.* in [6, 18] to construct an auralization system considering both reflections and diffraction.
F. Antonacci *et al.* in [7, 19] present a tracing technique from the visibility standpoint and apply a parametrization, so that a beam in 2D is represented by a segment in a dual space (the *ray space*), thus reducing the dimensionality of the problem and gaining in performance.
Further solutions based on *beam tracing* can be adopted to emulate later reflections. However, they are generally poorer than *ray tracing* implementations in accuracy and resources consumption.

Late reverberation

Ray Tracing Whereas the ISM provides an exact geometrical solution that consists of all of the specular reflection paths, the premises of *ray tracing* methods are different. Instead of looking for all the paths deterministically, this class of methods is stochastic and performs a Monte Carlo sampling of possible reflection paths, leading to the introduction of relevant approximations (that we consider acceptable for the modeling of the reverberant tail).

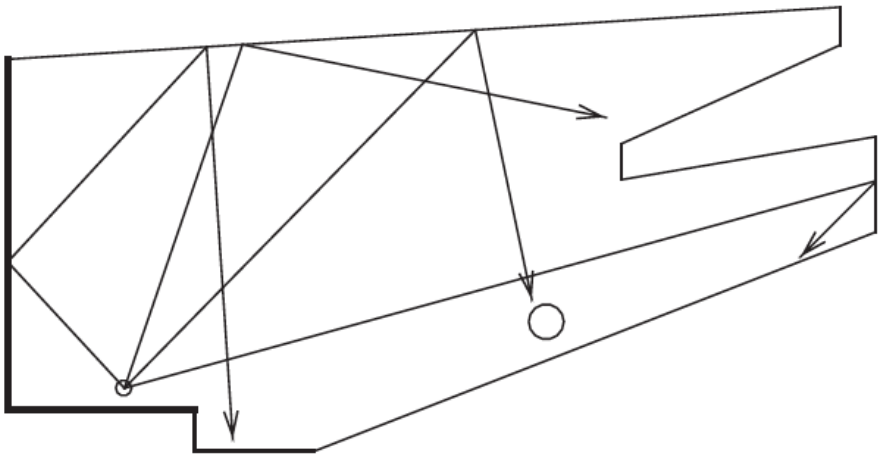


Figure 2.4: Ray tracing example

In Figure 2.4, a set of rays is cast from the source with a certain initial spatial distribution. The rays do bounce against the walls until they reach the listener.
As reported in [8] by Vorländer, each ray carries energetic information

encoded per-band and, depending on the materials of the walls, the energetic content is altered until the contribution of each band decreases underneath a certain threshold, therefore leading to the ray negligibility. What is more, we can imagine that the receiver should be a sphere-like spot and not a point as highlighted in [20, 21].

2.1.2.3 Reverberation time

The Reverberation Time (RT) is a time quantity characterizing the decay time of a sound within an environment. It is, in fact, the time interval required to a sound to evanish or, more specifically, the interval required to the sound pressure level to dacay underneath a predefined threshold. The RT can be estimated from theoretical formulations given the knowledge of room properties or inferred starting from an audio signal (a RIR or a reverberant signal).

Theoretical formulations In some cases, the RIR might be unknown, while environmental information (such as room shape, shape variables and volume) is known. In such cases, the RT can be estimated exploiting either Sabine's or Eyring's formulas.

Sabine's formula Under the assumption that the traveling wave does not hit the room reflectors at the same time, the Sabine's formula [22] can be exploited to estimate the RT. It corresponds to

$$T_{60} = \frac{24\ \ln(10)}{c} \frac{V}{\sum_{i=1}^{W} a_i S_i} \tag{2.3}$$

where a_i and S_i are respectively the energetic absorption and surface of the i^{th} reflector, c is the speed of sound in air and V is the volume of the room.

Eyring's formula Under the assumptions that the traveling wave does hit the room reflectors all at the same time and that all the surfaces share the same absorption coefficient, the following Eyring's formula [23] can be exploited to estimate the RT.

$$T_{60} = -\frac{24\ \ln(10)}{c} \frac{V}{S\ \ln(1-\overline{a})} \tag{2.4}$$

where $\overline{a} = \frac{1}{W} \sum_{i=1}^{W} a_i$ and $S = \sum_{i=1}^{W} S_i$.

Schroeder's integration The very first attempt of estimating the RT from a RIR was performed by Schroeder in 1965 [24].

Starting from the definition of Energy Decay Curve (EDC):

$$h^{(EDC)}(t) = \int_t^{+\infty} h^2(t')dt' \tag{2.5}$$

where h is an impulse response and

$$h_{dB}^{(EDC)}(t) = 10\log_{10}(h^{(EDC)}(t)) - 10\log_{10}(h^{(EDC)}(0)), \tag{2.6}$$

the T_{60} could be derived as:

$$T_{60} = \arg\min_{t\in\mathbb{R}}\{h_{dB}^{(EDC)}(t) \leq -60\} \tag{2.7}$$

or, to gain robustness, substituting $h_{dB}^{(EDC)}$ in Equation 2.7 with $\hat{h}_{dB}^{(EDC)}$, where the latter is a linear interpolation of the former.
The above expressions can be easily translated in the digital domain.

2.2 Audio forensics and integrity checking

The RT estimation techniques with many other audio parameter estimation strategies can be wrapped in a general framework for *integrity checking* within the context of audio forensics.
The *integrity checking* field focuses on the retrieval of information from a given document or file to check whether it has been affected by a content modification performed by unauthorized third parties. As a consequence, the intention is to check if an audio file has been modified introducing malicious or undesired content.
To give an intuition, in a simple scenario we might have at our disposal an audio file of which some acoustical properties are declared. We would like to be able to certify the reliability or unreliability of such declared parameters performing some tests.

2.2.1 Reverberation time inference

As a first audio parameter we consider the RT. As said, it can be estimated by performing measurements on the RIR, exploiting theoretical formulas given information of the acquisition environment or, for instance, applying algorithms on speech audio signals. The forensics field is focused on the latter option.

2.2.1.1 Reverberation time estimation from speech signals based on sub-band decomposition

In [1], the authors perform inference on the T_{60} by performing a sub-band decomposition of a speech signal and extrapolating a set of Free-Decay

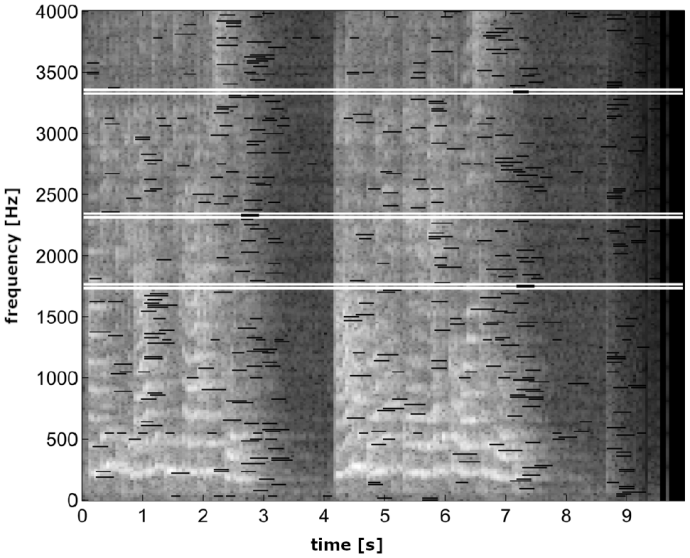


Figure 2.5: FDR samples from an energetic spectrum [1]

Regions (FDRs) from each sub-band. The sub-bands are determined extending what described by Vieira in [25].

As shown in Figure 2.5, an energetic *time-frequency* spectrum $E(t, \omega_k)$ (t on the horizontal axis is the time and conceals a frame index, ω_k on the vertical one represents frequency bins) of a reverberant speech signal is considered, and a set of FDRs is extracted from each band considering bundles of consecutive frames with decreasing energy. The regions which are too short under a temporal perspective or with a negligible energetic content are discarded.
A final T_{60} estimate is evaluated for each region, then the final $\hat{T}_{60}$ is evaluated as a linear combination of applications over $\hat{T}_{60}^{(k)}$ band estimates.

2.2.1.2 Learning estimation techniques

Statistical learning approaches aim at retrieving an I/O relation (model) for solving complex problems by looking for patterns in the data. They are based on convex optimization applied on *high-dimensional spaces*.
The main methods for RT estimation do differ mainly for exploited feature-sets, architecture and problem typology, either regressive or categorical. Here we briefly report some references (in a non exhaustive fashion) which are worth to be mentioned.
Xiong *et al.* in [11] exploit a two-dimensional Gabor Filter-Bank (Gabor-FB) over a Mel-scaled Spectrogram to extract a feature-set which is then fed to a Multi-Layer Perceptron (MLP) architecture for a 7-band-split T_{60} classification. A similar work [26] from an ACE challenge extends the architecture to perform a classification also on the Direct-to-Reverberation Ratio (DRR) and again a further work [27] considers both RT and Early-to-Late Reverberation Ratio (ELR) employing Gammatone Filter-Banks (γ-FBs) to build the feature-set. In [28], instead, a time-aware Recurrent

Neural Network (RNN) is used.

2.2.2 Acoustic environment identification

As mentioned, beyond RT, there is a great variety of room parameters which can be estimated to determine integrity. An example is to estimate over-time whether an audio signal has been acquired within a small room, a studio, a church, a hall, a concert room etc. with Acoustic Environment Identification (AEI) methods.
Results in this direction were initially drawn by Peters, Lei and Friedland in [12] where feature-maps built upon reverberant signals coming from certain room classes were used to train different Gaussian Mixture Models (GMMs) (one model per class). For them, the feature-map is constituted by a time dependent set of Mel-Frequency Cepstral Coefficients (MFCCs) and the predicted class is selected with a strategy depending on the GMMs results.
In [13], Malik proposes an alternative approach to FDRs extraction presenting an Automatic Decaying-Tail Selection algorithm to extract decay regions by retrieving peak and valley markers from the velocity of the energetic profile derived from the input signal. Depending on the selected regions, a couple of environmental parameters are estimated through a Maximum Likelihood statistical model and are then exploited to identify the ambience.
A paper by Moore, Brookes and Naylor [29] formulates a statistical room-print model based on an octave-band analysis of the reverberation time retrieved from real RIRs. Henceforth, in this case, the identification is performed with a T_{60}-estimate-based strategy. T_{60} estimates are retrieved from a third-octave-band representation and are used to fit a statistical model for the identification.
For M. Marković and J. Geiger in [14], a variety of baseline classification systems for Acoustic Scene Classification (ASC) (see AEI) are compared and a reverberation-based algorithmic pipeline is introduced to extend commonly adopted feature-sets.

2.2.3 Spatial volume inference

Further experiments related to room parameters inference can be found in the literature. In the past, the volume estimation issue was addressed as a classification problem by splitting a considered volume range in bands. Here we'd like to mention [15], in which Genovese *et al.* attempt in estimating the room volume with a regressive approach exploiting a Convolutional Neural Network (CNN) architecture.
In this work, a set of real RIRs acquired in shoebox-shaped rooms is augmented with simulated data trying to reach an almost uniform volume distribution. A reverberant speech set is retrieved from the set of responses and the one of anechoic voices. What is more, a realistic second

noise source within the room is modeled.
From the reverberant signals, a set of 25 feature vectors is extracted in order to build a feature-map to be fed to a CNN architecture. Such vectors provide a compressed representation of the initial signal, which is primarily relevant for the problem, secondarily useful to reduce memory consumption and, again, beneficial to enhance the training speed.

2.2.4 Shape variables inference

In similarity with the volume inference case, we might think of performing a direct estimation to retrieve the shape variables of a room. (i.e. for the shoebox room, width, height and depth).
In [30], D. Marković exploits a single RIR to assess a shape variables inference with a generative search. Contrary to previous works, his estimation is based upon the knowledge of a single response.
Here, the room shape (either with rectangular or L-shaped floor plan), the position of a loudspeaker and a microphone are fixed a-priori to acquire a RIR. Then, this response is compared with a template in order to minimize a loss function with a Genetic Algorithm (GA). The mentioned loss function is designed in order to minimize the distance among corresponding peaks within the real and template responses, by performing a search within the *shape variables space* of the given room.
An alternative approach going in the data-driven direction by Yu and Kleijn [31] exploits a corpus of RIRs and a 1D CNN to address the same problem with a three-output regressor.
Here, noticeably, the real strength of the network is the ability of determining useful features in an automated fashion, with a learning procedure which increases in abstraction.

2.3 Room reconstruction methods

After this brief walk-through into audio forensics, we highlight the lack of a tool capable of recognizing the shape of a room (e.g. with a rectangular, L-shaped or house-shaped floor plan). In the literature there are various papers and articles focusing on a geometry reconstruction. However we believe that new methods based on learning techniques may give new clues on the room acoustics field.

2.3.1 Spatial maps building

D. Aprea, F. Antonacci, A. Sarti and S. Tubaro in [2], perform experiments to estimate a polar map indicating the walls of the enclosure. To do so, they fix a single microphone in a room and they change repeatedly the position of a loudspeaker in space. At each repetition, they perform an acquisition. Changing the loudspeaker position, depending on the reflector, the position of the first order IS changes accordingly. We define

the point of reflection as the intersection of the wall with the segment linking microphone and image source. For the experiments, completely absorbing floor and ceiling are assumed.
Let $x_{\mathbf{s}}(t)$ be the signal of a speaker in a certain position acquired by a microphone and $s_{\mathbf{p}}(t)$ be a template signal. The template signal is defined as the convolution of the source signal with a simulated RIR for a single reflector orthogonal to the segment from the microphone to the point of reflection $\mathbf{p}$ on the wall.
Defining $m_{\mathbf{s}}(\mathbf{p}) = \langle x_{\mathbf{s}}, s_{\mathbf{p}} \rangle$ as the cross-correlation with all the template signals, the polar map is $m(\mathbf{p}) = \mathbb{E}_{\mathbf{s}}[m_{\mathbf{s}}(\mathbf{p})]$, where $\mathbb{E}_{\mathbf{s}}$ is the expectation operator over the speaker positions. The point of reflection is estimated as $\hat{\mathbf{p}} = \arg\max_{\mathbf{p}}\{m(\mathbf{p})\}$.
Introducing more reflectors, the superposition of the maps results in the following map.

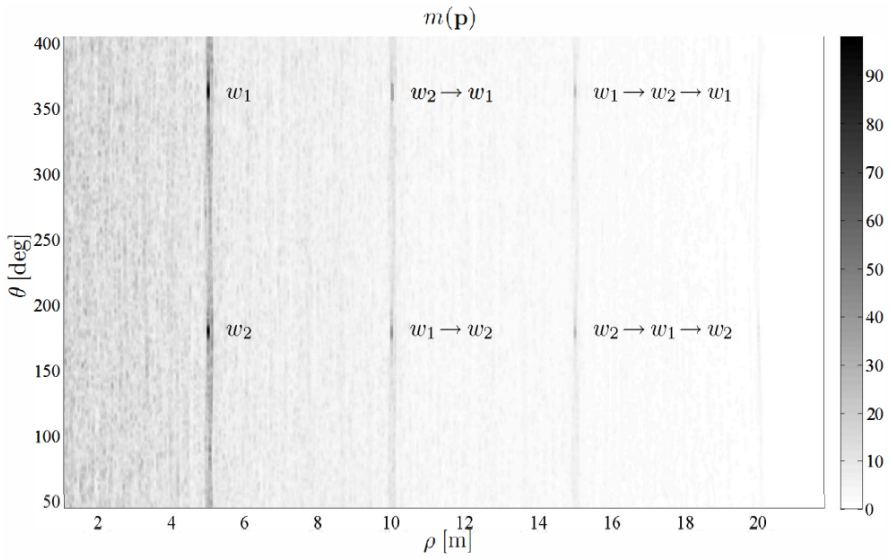


Figure 2.6: Room reflectors polar map [2]

Figure 2.6 shows the polar map in which peaks indicate the position of the first order points of reflection with respect to the microphone. Hence, the map allows to retrieve the floor plan geometry.
A further work [32] assumes no knowledge of the source signal and exploits more speakers and receivers. As a first step, the secondary sources locations are estimated. From the cross-correlation of pairs of signals incoming from the microphones, an experimental Time-Difference of Arrival (TDoA) matrix is computed obtaining $\tau(\mathbf{r_i}, \mathbf{r_j})$.
At the same time, a template TDoA matrix from a generic point in space to each couple can be determined as $\tau_{tpl}(\mathbf{r_i}, \mathbf{r_j}; \mathbf{x})$. Then, defining the cross-correlation between the TDoA matrices as $\mathbf{R}_{\mathbf{x_i},\mathbf{x_j}}(\tau_{tpl})$, the cartesian likelihood map is a combination of such estimation functions $\mathbf{R}_{\mathbf{x_i},\mathbf{x_j}}$ over all unique (i, j) microphone couples.

2.3.2 Common tangent algorithm

In the works which led to [3], F. Antonacci *et al.* fix a microphone and move a speaker at known positions. The playback signal is known, the devices are synchronized, and a single reflector is considered. Each source-microphone pair has an associated RIR which embeds the Time of Arrival (ToA) of the first-order reflection. Given the ToA, the source and the microphone positions, an ellipse can be described. Such ellipse is the space of possible reflective points.

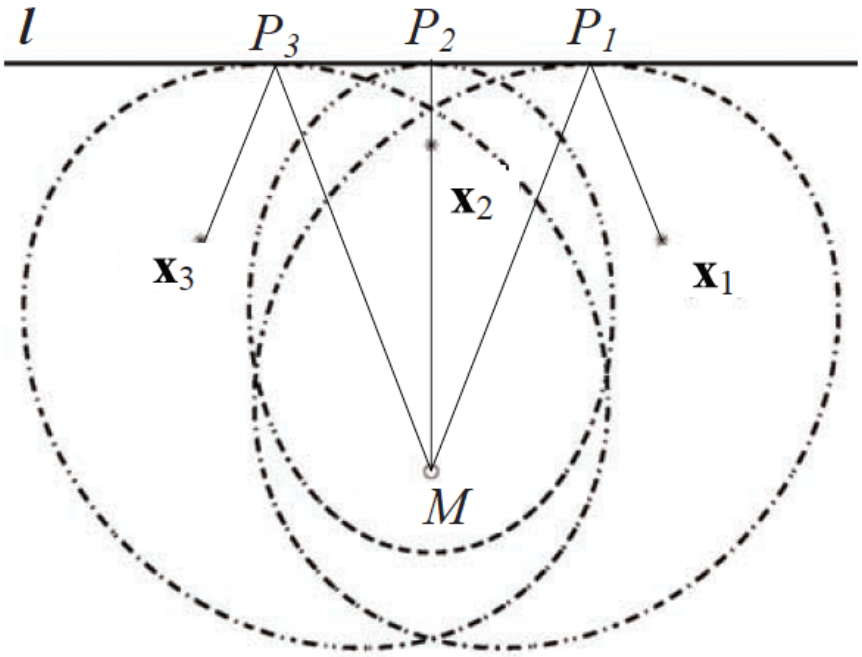


Figure 2.7: Ellipses from COTA algorithm [3]

By considering all the ellipses (Figure 2.7), we can uniquely identify two linear reflectors as their common tangent lines. To do so, the sets of bundle lines tangent to each ellipse are intersected to obtain two estimate lines resulting from an optimization (the COmmon-TAngent (COTA) algorithm considers a line parametrization to cope with errors). The basic method is revised to gain robustness using the Hough transform and further extended to use ellipsoids in order to deal with 3D spaces.

2.3.3 Euclidean distance matrix based algorithm

In [9], the authors first assume to be in a polyhedral room with a variety of walls. They use a minimum of 4 microphones to estimate the geometry. What is more, the source position is estimated exploiting TDoAs, while the receivers position is known.
It is the first relevant case in which all the walls are considered in one shot from the beginning, therefore the RIRs do encode the first-order reflection

incoming from all the reflectors. This implies the need of disambiguating the first-order reflection contributions coming from a wall or from another (echo-labeling problem). This is addressed using a Euclidean-Distance Matrix (EDM).
Basically, an EDM is squared, symmetric, contains positive values and zeros on the main diagonal. Such positive values are the squared distances of each microphonic couple. From the RIR of each microphone, the ToA of early reflections (possible image source distances) are considered, and the EDM is augmented with each combination of distances. Within the set of augmented EDMs, a test is accomplished to retain the sole real EDMs, i.e., the sole ToA grouping combinations related to a single image source. The authors are left with a set of image sources related to their ToAs in each RIR.
To retrieve the geometry, the non-first-order image sources are filtered out with a strategy and the first-order ones are exploited to apply an inverse-ISM algorithm.

2.3.4 Reflector search methods

As reported by Crocco, Trucco and Del Bue [10], their work goes in the direction of neglecting knowledge about the position of sources and receivers. A convex polyhedral room, with known number of walls and where sources and receivers are distributed, is considered. The playback of the speakers is a chirp signal. A matched filter is adopted to extract peaks from the recordings. From each source-receiver couple, a set of signal ToAs is extracted. Then, by exploiting a matricial approach, the positions of sources and receivers are inferred, together with times of emission and offset.
After the exclusion of approaches requiring excessive computational effort, a greedy iterative approach is proposed to retrieve the reflectors and to build the room model.

2.4 Audio descriptors

Moving towards a theoretical introduction of neural networks, we cannot omit a digression into data processing. The pre-processing is a stage in which data is transformed to obtain a feature representation which is useful for the network.
With feature we might either refer to a property of the environment where the input was acquired or a parameter of the input signal itself or a characteristic alternative representation of the signal.
Some standard audio features which can be extracted from a digital audio signal $s[n]$ with temporal index n are:

- Discrete Fourier Transform (DFT): describes the magnitude and phase in frequency of a signal excerpt obtained windowing the ini-

tial audio signal. Let $w[n]$ be a window of a certain type with length L_{win} and hop size L_{hop}, then $s_r[n] = s[n]w[n - rL_{hop}]$ is the excerpt for r the frame index.
Fixing the excerpt r_0, its DFT is:

$$S_{r_0}[k] = |S_{r_0}[k]|\, e^{j\angle S_{r_0}[k]} = \mathcal{F}\{s_{r_0}[n]\}[k] \tag{2.8}$$

where $\mathcal{F}$ is the DFT functional in N_{fft} frequency bins (indexed k) defined as:

$$\mathcal{F}\{s_{r_0}[n]\}[k] = \sum_{n=0}^{N_{fft}-1} s_{r_0}[n]e^{-j\omega_k n} \tag{2.9}$$

where s_{r_0} is assumed zero-padded until N_{fft} samples and $\omega_k = \frac{2\pi}{N_{fft}}k$ is the normalized frequency wrt the sampling frequency at which s was acquired.

- Short-Time Fourier Transform (STFT): describes magnitude and phase in frequency of an excerpts sequence.
  Then, for N_{fr} number of frames within s, the STFT is the concatenation of DFTs over consecutive frames:

$$S[r, k] = S_r[k] = |S_r[k]|\, e^{j\angle S_r[k]} = \mathcal{F}\{s_r[n]\}[k] \tag{2.10}$$

- Mel-scaled Spectrogram: represents in an alternative way the magnitude of the STFT by logarithmically scaling the frequency axis (following the Mel scale) and dB scaling the magnitude values.

- Filter-Bank: is a set of parallel filters each of which dividing the frequency range of the input signal in contiguous frequency bands. The STFT can be regarded as a special case of Filter-Bank (FB) in which the linearly distributed frequency bins represent the bands. Depending on the needs, we might opt for the STFT to give a representation of the signal or for a FB with fewer and wider bands but preserving the temporal resolution.
  A special case of FB is the γ-FB which tries to reproduce the human ear logarithmic functioning. The frequency band centers sometimes follow the ERB scale.

- Cepstum (Cepst): gives a further spectral visualization of the spectrum over l^{th} *quefrencies* $[s]$ and is defined as:

$$\zeta_{r_0}[l] = \mathcal{F}^{-1}\{\, \ln(\, |\mathcal{F}_{s_{r_0}}|\,) + j\angle\mathcal{F}_{s_{r_0}}\,\}[l] \tag{2.11}$$

  Even the Cepst vector can be generalized to obtain a time-varying cepstral map.
  The Mel-Frequency Cepstrum (MFC), scales the frequency axis of an inner power spectrum to be in Mel-scale and exploits a Discrete Cosine Transform (DCT) as outer transform.

- Envelope (ENV): describes the upper or lower amplitude profile of a signal as:
$$\epsilon[n] = \sqrt{s^2[n] + \mathcal{H}(s[n])} \tag{2.12}$$
where $\mathcal{H}$ is the Hilbert transform functional which can be seen as the application of an Hilbert filter over the signal via convolution.

2.5 Artificial neural networks and deep learning

To quickly summarize what we've written so far, dealing with a shape classification problem can valuably affect both the audio forensics field and the room imaging one. In fact, in the first case we underscore the absence of a room shape recognition tool and in the second case we shall deal with complex algorithms which we aim at automatize and fasten.
Nowadays, the research is increasingly moving towards the usage of statistical learning models to cope with supervised and unsupervised problems. With this work of thesis we would like to investigate the potential of supervised Deep Learning (DL) techniques applied to the task of room floor plan classification, choosing among three different shape classes.
The DL techniques are based upon Artificial Neural Networks (ANNs): they try to find the best solution in a search space exploiting an architecture in which nodes are interconnected. Such architectures try to emulate the brain behavior: the nodes represent the neurons, while the connections in the architectural graph represent synapses. The nodes are grouped into layers, the more the layers, the deeper the network and the greater is its ability to abstract concepts.
The network complexity must be attently evaluated. Models which result too simple tend to have great error because they have a bias which is misleading for the learning and visualization of patterns. *We shall notice that the term "bias" is adopted also in cognitive psychology to describe the propensity to believe as truthful a certain personal belief depending on a flawed logic generally based upon a small set of observations (prejudice, generalization);*
On the other hand, too complex models might require expensive resources, wide datasets, and fail anyways while considering the noise in the data as relevant for the problem.

2.5.1 Supervised problems and loss function

Essentially, the supervised problems for which we have data, are problems in which the input x (e.g. a 1D audio signal or 2D image signal) and the output y are known across the whole sampleset (of length $|\mathcal{D}|$), while the I/O relation is unknown.
The *loss function* describes the distance between two variables while time passes by (the discrete time instants at which the loss is updated

are ordinarily the u^{th} training epochs).
In general we can write the loss funcion as:

$$\mathcal{L}_{y,\hat{y}}[u] \tag{2.13}$$

where the two pedix variables are omogeneously either scalar or encoded label vectors: y is a set of groung-truth target values drawn from the real distribution p and $\hat{y}$ contains the samples drawn from the estimated distribution q (modeled by the estimator).
The goal of these learning techniques is to retrieve an I/O model which minimizes the loss.
For sake of notation compactness, let us fix an epoch $u = u_0$.

2.5.1.1 Regression

In the regression problem, the variables $(y, \hat{y})$ are a set of scalar values representing a feature of the input sample x, and are generally rounded to some decimal units.
Depending on the scale of the target values y, different loss functions might be chosen in order to spread and remark the variables differences. Nevertheless, we are going to consider the most commonly used loss functions for the regressive scenario: the Mean Absolute Error (MAE) and the Mean Squared Error (MSE).
We define the error as:

$$e = y - \hat{y} \tag{2.14}$$

where for the i^{th} sample the error is $e_i = y_i - \hat{y}_i$.

Loss functions The MAE averages the distances of ground and predicted samples. It is defined as:

$$\mathcal{L}_{y,\hat{y}} = MAE(y, \hat{y}) = \mathbb{E}[|e|] = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} |e_i| \tag{2.15}$$

and it represents the range around the prediction in which, in average, is possible to find the expected value.

The MSE averages the squared-distances of ground and predicted samples. It is defined as:

$$\mathcal{L}_{y,\hat{y}} = MSE(y, \hat{y}) = \mathbb{E}[e^2] = \mathbb{E}^2[e] + \sigma^2_{\hat{y}} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} e_i^2 \tag{2.16}$$

and it encompasses the contribution of the estimator bias and variance. To be thorough, Equation 2.16 should also include the acquisition irreducible error.

2.5.1.2 Classification

For what concerns the classification scenario, instead, the variables are a set of labels identifying the class of a sample. Such labels are encoded into integer class identifiers or categorical identifiers.
The most ordinary loss for classification problems is the Cross-entropy (H). Even the Kullback-Leibler Divergence (KLD) is sometimes used to measure the distance between the cross-entropy H and the discrete entropy of the true distribution p.

Loss functions Considering p (see ground-truth y) and q (see estimate $\hat{y}$) the true and estimated distributions over a support of classes Γ such that $y_i \sim p(\Gamma)$ and $\hat{y}_i \sim q(\Gamma)$, the cross-entropy is defined as:

$$\mathcal{L}_{y,\hat{y}} = H(p,q) = -\mathbb{E}_p[\log_2(q)] = -\sum_{G \in \Gamma} p(G) \log_2(q(G)) \tag{2.17}$$

and measures the average number of bits needed to encode a symbol $G \in \Gamma$ with an encoder build upon q. Therefore, H is a measure of the estimator uncertainty.
Depending on the number of classes $|\Gamma|$ we might opt for a binary cross-entropy function. What is more, the type of encoding adopted for the target value, whether with an integer class identifier or a categorical identifier, affects the usage of sparse categorical cross-entropy or categorical cross-entropy losses.

Other metrics The confusion matrix $\underline{\underline{\mathbf{C}}} = \{c_{i,j}\} \in \mathcal{M}_{|\Gamma|\times|\Gamma|}$ where i indexes ground values and j indexes predictions, organizes the counting of the predicted classes depending on the ground values: $\underline{c}_i = q(\Gamma \mid \mathcal{D}^{(i)})$. Such matrix highlights the number of samples which are correctly classified ($c_{i,i}$) and the distribution of the misclassified samples.

From the confusion matrix, the accuracy

$$ACC = \frac{\sum_{i \in \{1,\dots,|\Gamma|\}} c_{i,i}}{|\mathcal{D}|} \tag{2.18}$$

can be derived. A high accuracy measures and sums up the goodness of the estimator.

2.5.2 Network architecture

So far we discussed about the best-fitting loss for a problem typology. The *network parameters space* defined by the architecture is loosely mapped onto the loss function, hence the loss minimization procedure corresponds to building trajectories into the parameters space looking for the optimal parameters vector, i.e. for the optimal estimator.

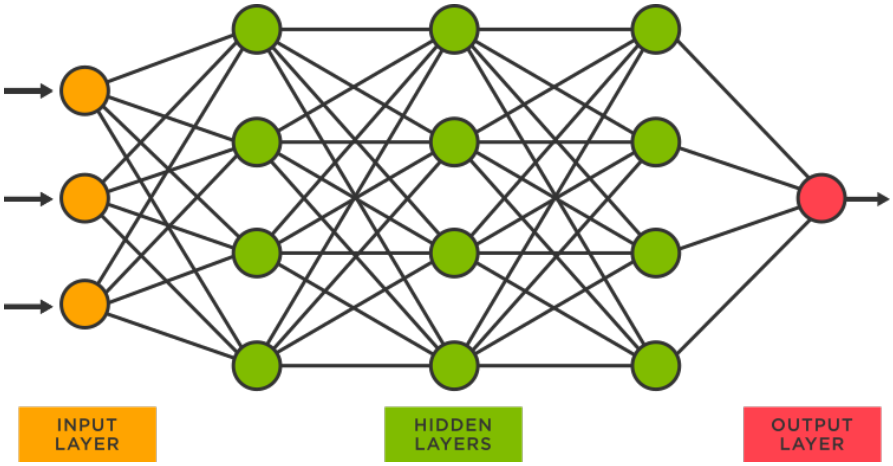


Figure 2.8: Example of NN architecture

Let us exploit Figure 2.8 to be clear: here we can see a Deep Neural Network (DNN) with a depth of five and with four Fully-Connected (FC) layers. Each FC layer contains a certain amount of neurons (so called FC because each of them receives and sends connections from each neuron of the previous layer to each neuron of the following one).
The input layer has a dimensionality which matches the one of the input tensor. It is followed by a set of hidden layers which abstract a latent representation of the input datum.
Finally, there is the output layer which may contain a single neuron or a multiplicity of them. The role of the output layer is to give a predicted value $\hat{y}_i$ in case of regression or a set of a-posteriori likelihood confidences $q(y_i = G \,|\, x)\ (\forall G \in \Gamma)$ in case of classification.

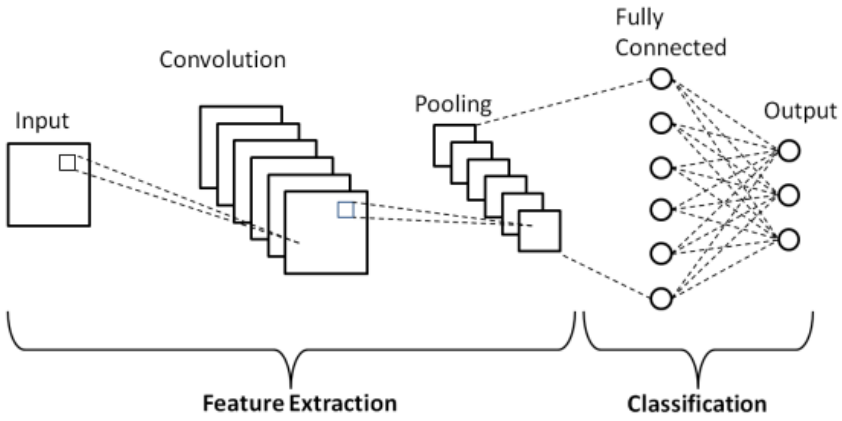


Figure 2.9: Example of CNN architecture

A slightly more complex example of architecture could be the CNN in Figure 2.9.

CNNs are habitually split in two main blocks:

1. A first block has the role of abstracting functional features from the input signal;
2. A second block is a FC sub-architecture (e.g. a MLP) which exploits the highlights of the preceeding block to perform the task;

2.5.2.1 Features abstraction layers

As brought up before, Neural Networks (NNs) can autonomously abstract features from an input signal. Bearing in mind that CNNs can be used also for 1D and 3D application, they are widely used to frame relevant portions of 2D images (with a certain representation convention, e.g. NHWC). That's why we think they are a good toy example to imagine the concepts of features extraction. Going back to our foregoing narration, here we are going to consider a convolutional abstraction block as an example.
The most common layer types which take part to this block are:

1. 2D-Convolution layer: this layer is the one that really evidences the portion of images that are relevant for the task. It is made-up by a certain number of cells which are bi-dimensional filters (aka kernels) of a certain size. The weights of each kernel are learned during the training phase. The output of a convolutional layer (indexed by l) is a set of images (one per kernel) where each image $\underline{\underline{\mathbf{O}}}_{l,r}$ (from filter r) is the result of the 2D convolution $\underline{\underline{*}}$ applied to input image $\underline{\underline{\mathbf{X}}}$ and filter $\underline{\underline{\mathbf{W}}}_{l,r}$ operands (with an optional stride: directional hop size);

$$\underline{\underline{\mathbf{O}}}_{l,r} = \underline{\underline{\mathbf{X}}} \underline{\underline{*}} \underline{\underline{\mathbf{W}}}_{l,r} \tag{2.19}$$

2. 2D-Pooling layer: this layer has a down-sampling role and does not introduce learnable parameters. Depending on the pool size and on the type of layer, whether max or average pool, a 2D down-sampling is performed retaining the max or average of each pooling block. This layer follows a convolution layer. The conv-pooling sub-structure is repeated as many times as needed;
3. Flatten layer: is the final layer preceding the FC block and has the role of flattening the incoming tensor into a 1D vector.

Learnable audio front-end Another specific example about features generalization is reported by Zeghidour *et. al* in [16] where LEarnable Audio Front-end (LEAF) is presented. LEAF is a deep neural front-end for a CNN audio classifier which dynamically adapts a parametric 2D representation of an input signal while the network undergoes the training.

Firstly, the audio signal x is filtered with a complex Gabor-FB with a fixed number of bands and retaining the squared modulus of the output signal. Here each filter has two parameters: band center and band-width;
Then, a low-pass pooling is performed introducing one low-pass filter (with gaussian kernel) per band. This pool layer is parametrized by the filter band;
Finally, a Per-Channel Energy Normalization (PCEN) compression is performed by normalizing the time-frequency representation with an exponential moving average filter similarly to what was done by the authors of [33].
The authors test their front-end network with an audio classifier based on a lite EfficientNet. Specifically, they opt for the least complex network, EfficientNet-B0. The EfficientNets have been introduced by Tan and Le in [4] as CNNs improving the performances of state of the art convolutional neural networks with the need of fewer parameters.

2.5.2.2 Fully-connected layers and neural activation

Considering the second macro-block, instead, the output of each neuron (of layer l, indexed by r) is constrained by an activation function a which receives a combination of applications over each input of the neuron itself. Each input is the output of a neuron of layer $l^- = l-1$ indexed by ρ. The concept of activation is transversal to layers of whichever kind, however we give an explanation here for sake of clarity.

$$o_{l,r} = a_l(b_{l,r}) \text{ where } b_{l,r} = \sum_{\rho} w_{l^-,\rho,r}\, o_{l^-,\rho,r} \tag{2.20}$$

Here the weight $w_{*,*,*}$ is scalar.
Relevant examples of activation functions can be:

- Rectified Linear Unit (ReLU): causes the neuron to fire $b_{l,r}$ iff $b_{l,r} > 0$ (practical for the output of regressive problems over a positive support);
- Sigmoid: is a soft unit-step function. If the unit-step fires 1 iff $b_{l,r} > 0$, the sigmoid considers input fluctuation and models a certainty (practical for the output of classification problems);

2.5.3 Training and parameters optimization via back-propagation

Given a supervised problem for which we have labeled data, the loss function and an architecture, we shall start training to determine the optimal weights for our estimator. This procedure is performed using the training data to learn patterns and the validation data to evaluate the goodness of the model on unseen data. Finally, the test data is used to have a final clue on the overall quality of the trained model.

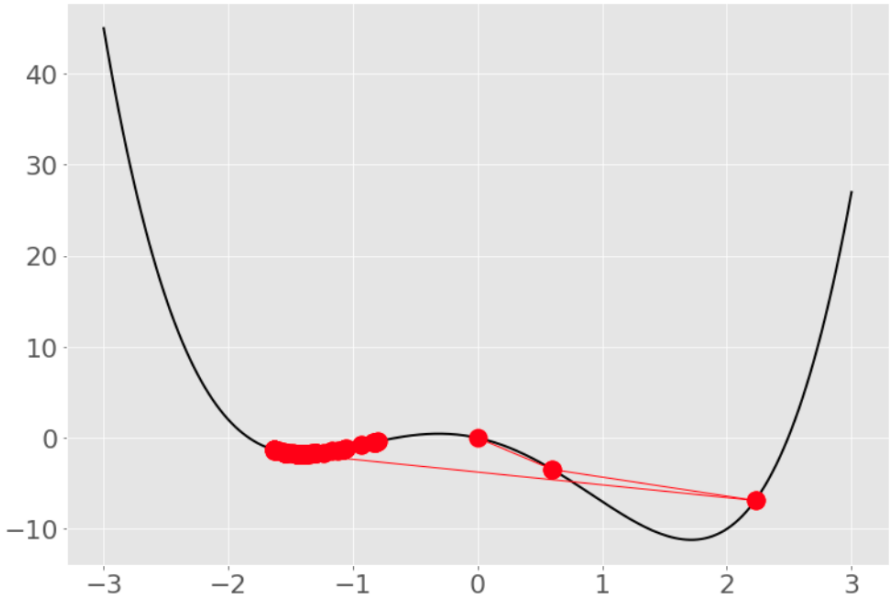


Figure 2.10: Example of GD solution - $(\mathbf{w}, \mathcal{L}(\mathbf{w}))$ in black

Let us assume to have the dataset $\mathcal{D} = \mathcal{D}_{train} \bigcup \mathcal{D}_{val} \bigcup \mathcal{D}_{test} = \{(x_i, y_i)\}$ where the samples are retrieved with a noise-prone measurement from the true distribution. Let us assume to have an architecture with initial parameters state $\mathbf{w} = \mathbf{w}_0$ and a Gradient Descent (GD) optimizer with an initial Learning Rate (LR) $\eta = \eta_0$. Figure 2.10 represents the loss among true data and predictions in the parameters space for a simple problem during a learning epoch.
Considering this first epoch, there are many steps in which splits (batches) of the training set are considered. For each sample, the Back-Propagation (BP) procedure computes the gradient of the loss with respect to each of the parameters (using the chain rule) going backwards from the output to the input layers and leading to the update of the parameters following the equation:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\partial \mathcal{L}(y(x), \hat{y}(\mathbf{w}|x))}{\partial \mathbf{w}} \tag{2.21}$$

When the gradient decreases, the parameters are left almost unchanged and an optimum has been found.
In case the loss function exhibited local minima, as in figure, beginning the learning with a small LR, given an unlucky initial state, might lead to sub-optimal solutions. Hence, initially the LR is set to a quite high value (wider descent step) to increase the chance of reaching the global optimum and is then decreased overtime following a certain logic. Therefore, there could be a dependence like $\eta(u)$ which reduces the training loss fluctuations overtime.
Aside GD for BP, other possible gradient optimizers are Stochastic Gradient Descent (SGD), RMSprop, Adaptive Moment Estimation (Adam) etc.

2.5.4 Validation and hyper-parameters optimization

The validation set retrieved from the whole dataset is often used to manually alter the architecture complexity to prevent over-fitting.
The architectural parameters, for example depth, number of neurons in each layer, activation of neurons in a layer together with LR and many other knobs are hyper-parameters. The automated assembly of a NN architecture for a problem (via hyper-parameters tweaking) is an active field of research which evidently involves dealing with the bias-variance trade-off. Even if we do not report it here, we tried to improve some of our results following the work of Xiao *et al.* [34] in this direction. In their work, they adopt a Variable Length Genetic Algorithm (VLGA) to build a CNN architecture optimized for a classification task.

2.5.5 Test and Grad-CAM feature spotlights

The relevance of the features derived from an input vector via processing or convolutional abstraction can be emphasized with a variety of instruments. Some of such instruments require to absorb parameters (e.g. Attention layers), other can be put to use during the test phase.
An example of these latter tools is Gradient-based Class Activation Mapping (Grad-CAM), presented by Selvaraju *et al.* in [35]. Essentially, for each class, the spotlight map called the heat map, is the weighted superposition of the activation maps coming from a number of consecutive convolutional layers, from a starting one going to the output score (before the softmax deformation) for the same class. The class weight for the activation map of each layer is computed as the Global Average Pooling (GAP) of the gradient of the output score with respect to the activation map.
Let us try to give an intuitive explanation by considering a network already trained for an animal recognition task. Given a new sample, perhaps we'd like to highlight a dog in such image.

Figure 2.11: Grad-CAM spotlighting example

Grad-CAM is capable of highlighting the dog (i.e. the portion of the input image relevant for the task) through an heat map. Such map can be resized and superposed to the input image to create a view like the one in Figure 2.11.

2.6 Conclusive remarks

In this chapter we gave an overview of all the concepts on which we are going to operate in the following chapters. Starting from the definition and retrieval of audio signals and room impulse responses, we went through the discussion of environment parameters estimation focusing on the lack of a shape inference technique based on learning techniques. Then, we gave a condensed introduction of data processing and DL theory to clarify its applicability in our case of study.
In the following chapter we are going to propose our method for the classification of an environment shape based on mono speech signals.

3
Proposed Method

In this chapter we formally present the problems of estimating three room parameters: the spatial volume, the reverberation time and the floor shape.
Then, we describe our solution pipeline in which we present two approaches for estimating the acoustic environment properties. The first one aims at retrieving a preliminary estimation of either the room volume or the reverberation time. These problems have been often addressed in the literature. The second approach tackles the problem of room shape classification, which is less investigated in the literature.
Finally, we are going to draw some remarks on the chapter also putting forward possible architectural improvements.

3.1 Problem formulation

Let $x^{(in)}[n] \in \mathbb{R}^N$ be a digital anechoic audio signal of length N sampled at F_s. Morover, let us consider a room space $\mathcal{R} = \langle \Gamma, \Lambda_\Gamma \rangle$ where Γ is the set of possible room shapes and Λ_Γ is the space in which exist the room class variables for all the classes.
To better explain the meaning of Λ_Γ, let $G \in \Gamma$ be a room shape, then Λ_G is the space of the room parameters fixed the shape G, and $\lambda_G \in \Lambda_G$ is an instance point of such space. Let $r = \langle G, \lambda_G \rangle \in \mathcal{R}$ be a room sample, for which vertices, shape $G(r)$ and volume $V(r)$ are defined.
Let $\mathbf{s} \in \mathbb{R}^3, \mathbf{m} \in \mathbb{R}^3$, $(\mathbf{s}, \mathbf{m})$ is a source-microphone pair fixed in r so that $|\mathbf{m} - \mathbf{s}|$ belongs to a constrained range.
Let $h^{(r)}_{\mathbf{s}\to\mathbf{m}}[n] \in \mathbb{R}^M$ be a RIR of length M acquired at F_s into room r with an associated T_{60} estimation.

Then, from the digital implementation of Equation 2.1, we obtain $x_{\mathbf{m}}^{(r)}[n] \in \mathbb{R}^{N+M-1}$.

$$x_{\mathbf{m}}^{(r)}[n] = (x_{\mathbf{s}}^{(in)} \underline{*} h_{\mathbf{s}\to\mathbf{m}}^{(r)})[n] = \sum_{m=-\infty}^{m=+\infty} x_{\mathbf{s}}^{(in)}[m]\, h_{\mathbf{s}\to\mathbf{m}}^{(r)}[-m+n] \tag{3.1}$$

where $\underline{*}$ is the discrete 1D convolution operator. From the considerations above (V, T_{60}, G) can be inferred.

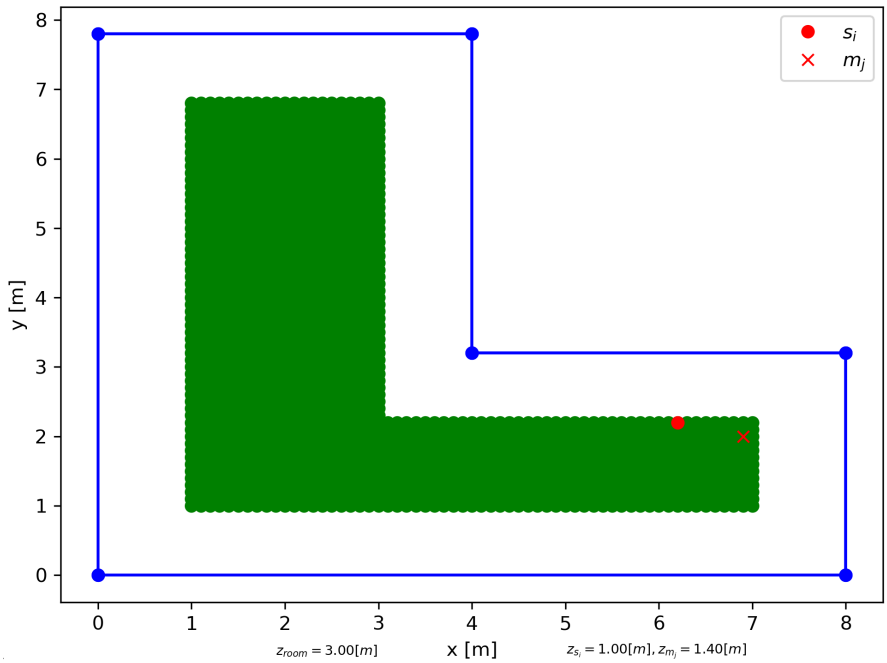


Figure 3.1: An acquisition setup scenario

In Figure 3.1, we represent a setup scenario in which we fix a source (red dot) and a microphone (red cross) for the signal acquisition.
With a notational abuse, let x be the reverberant signal or a compact representation of it.
Starting from the reverberant signal, we would like to retrieve the best estimate for the room shape class. To do so, we are going to use either a preliminary volume or a preliminary reverberation time estimator.

Volume regression In the volume regression scenario, we want to design an estimator for the volume V, using as input a reverberant audio signal. If we assume to know the relation $(x, V(x))$, where V is the volume of the room in which the x signal is acquired, we search for an estimator M_V which approximates $V(x)$ with $\hat{V} = M_V(x)$.

Reverberation time regression In the T_{60} regression scenario, we want to design an estimator for the reverberation time T_{60}, using as input

a reverberant audio signal. If we assume to know the relation $(x, T_{60}(x))$, where T_{60} is the reverberation time within a room in which the x signal is acquired, we search for an estimator $M_{T_{60}}$ which approximates $T_{60}(x)$ with the estimate $\hat{T}_{60} = M_{T_{60}}(x)$.

Shape classification In the shape classification scenario, we strive for evaluating the optimal estimator for the room shape class G. If we consider known $(x, G(x))$, where G is the shape of the room in which x has been acquired, we then look for the estimator M_g so that $\hat{G} = M_g(x)$ provides the estimate of $G(x)$.

3.2 Solution pipeline

Our problem is to estimate in the most accurate way the shape of a room starting from a reverberant signal.
To do so, we combine the usage of a group of estimators. Some estimators allow to retrieve a preliminary estimation of the volume and of the reverberation time depending on the input signal. The other ones are specific estimators which have been trained and optimized either on a volume or on a T_{60} band for the shape classification task. Each band is identified by an index b and with the preliminary estimation we are able to retrieve a band index estimate $\hat{b}$ for the specific estimators. We decide to work on a band-basis to provide the best room shape estimate. Indeed, the results of our experiments show that an estimator focused on a certain band has better performances.
Once we have chosen the specific estimator, we apply it to the input signal to retrieve an estimate.

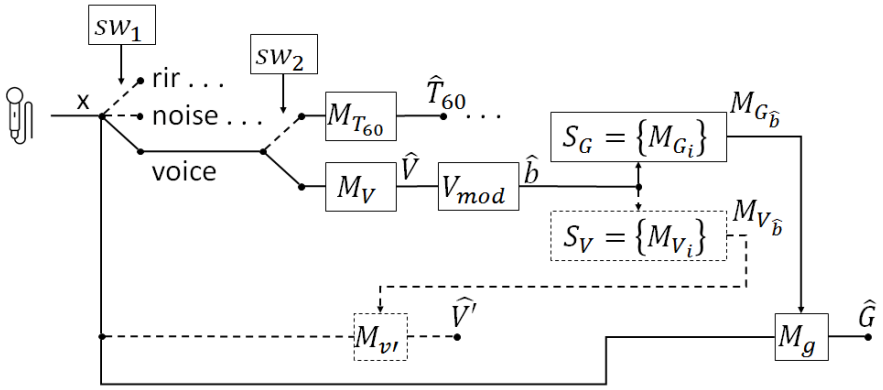


Figure 3.2: Solution pipeline

Figure 3.2 contains the core logic of our work.
$M_{G_{\hat{b}}}$, is the best model for the retrieval of the best shape estimate $\hat{G}$.
$M_{G_{\hat{b}}} \in S_G^{(voice,V)}$, where $S_G^{(voice,V)}$ is a set of estimators for shape classification, each of which focuses on a different volume band.
As mentioned, $\hat{b}$ is an index which represents the volume band on which

a model has been trained. Our models are parametrized on a specific band and with $\hat{b}$ we choose the best one. $\hat{b}$ is retrieved from a coarse estimation of the volume of the room.

Digging deeper, another band index estimate can be retrieved depending on the $\hat{T}_{60}$ estimate through $M_{T_{60}}$. That's why we parametrized $S_G^{(sw_1,sw_2)}$ to reference models. The ones trained on different input signals (among *rir*, *reverberant white noise* or *reverberant speech* signals) are chosen via sw_1 as in Figure 3.2. And a sub-set of these is obtained via sw_2 by considering the domain on which the band subdivision is performed (a *volume band subdivision* or a T_{60} *band subdivision*).
The reverberation time band subdivision is similar to the volume band subdivision and uses the coarse $M_{T_{60}}$ model in Figure 3.2 to retrieve the band index estimate which might be used as an alternative to the one obtained with M_V.
What is more, to have a clue on the usefulness of features, we experimented the usage of Grad-CAM.

In the following, we provide additional details in terms of analysis of our proposal.

Preliminary volume estimation Speaking about the M_V preliminary estimator, it has the role of retrieving a coarse volume estimation. The ground-truth volume property is retrieved from the room instance associated to the signal acquired in it.

Preliminary reverberation time estimation $M_{T_{60}}$ is treated similarly to M_V. However, for whichever kind of input signal (sw_1), the T_{60} considered as ground value is calculated on the RIR from which the signal derives.

Specific in-band estimators and their selection The overall volume range for which we have room examples is considered to be split in a number I of disjoint volume sub-ranges: the volume bands. Given the $\hat{V}$ estimate coming from the preliminary estimation, we can choose one of the specific models trained specifically for that volume band.
Parallelly, the same considerations hold for the branch which considers $\hat{T}_{60} = M_{T_{60}}(x)$.

3.2.1 Architectures for the estimators

For what concerns the estimators, we adopted a data driven approach because of the complexity of our study case. We used a couple of different CNN architectures. The first architecture puts at use a deterministic front-end for the computation of features, while the second one has a learnable front-end. For the preliminary volume estimation, we exploited

the deterministic front-end architecture, while for the room shape estimation we compared the two architectures.

3.2.1.1 Room Geometry Inference - Deterministic front-end Architecture

In this section we are going to fill the pending narrative gaps of Section 2.4 and Section 2.5.2 describing part of the estimation techniques used in the proposed solution.
First, we compute a compact 2D representation of audio signals through a pre-processing stage. Later, such 2D representation is fed to a neural architecture similarly to what done in [15] to retrieve volume and shape estimations.

Data pre-processing and input pipeline The feature map representation is evaluated on an adaptation of each reverberant signal. A segment of length N is obtained from the original signal using a rectangular window. The adapted signal x of length N is the superposition of the segment with a thermal noise (of -120dB deviation). The thermal noise is used to grant the log-energy computability (see below). The features are computed from the adapted signal.
A feature-map contains a subset of the listed features.

- A γ-FB sub-map parametrized by $(B_1, F_s, F_{lo}, F_{hi}, L_{win}, L_{hop})$:
  $\underline{\underline{\mathbf{\Gamma}}} : \mathbb{N}|_{B_1-1} \times \mathbb{N}|_{N_{fr}(L_{win},L_{hop})-1} \to \mathbb{R}$
  where B_1 is the number of desired ERB-scaled bands within $[F_{lo}, F_{hi}]$, L_{win} and L_{hop} are the window parameters for the log-energy computation and N_{fr} is the number of r^{th} frames of the input signal on which a log-energy coefficient $E_r^{(log)} = \ln(\sum_{j=0}^{L_{win}-1} x_r^2[j])$ is computed;

- A DFT vector parametrized by (F_s, F_{hi}, N_{fft}):
  $\underline{\phi} : \mathbb{N}|_{N_{fft}-1} \to \mathbb{R}$ calculating the DFT on $2N_{fft}$ points and returning half of the magnitude spectrum. The DFT is computed on a downsampled version of x determined to have F_{hi} as maximum frequency to fit the final image shape;

- A magnitude sorted DFT vector $\underline{\phi}_{ms}$ which retrieves an ascendent ordering of the previous vector;

- A Cepst vector with parameters (F_s, F_{hi}, N_{quef}):
  $\underline{\zeta} : \mathbb{N}|_{N_{quef}-1} \to \mathbb{R}$ on N_{quef} quefrencies, of which the magnitude is retained (see DFT feature vector downsampling);

- An ENV vector depending on (F_s, L_{win}, L_{hop}):
  $\underline{\epsilon} : \mathbb{N}|_{N_{fr}(L_{win},L_{hop})-1} \to \mathbb{R}$ (see γ-FB log-energy compression);

- The time domain signal with a dependence on (F_s, L_{win}, L_{hop}):
  $\underline{\chi} : \mathbb{N}|_{N_{fr}(L_{win},L_{hop})-1} \to \mathbb{R}$ (see γ-FB log-energy compression).

From the subset above, a bi-dimensional feature map is computed by stacking the mentioned features.
Once the maps are built on all the input signals, constituting an ideal tensor of shape $(N, H, W, C) = (|\mathcal{D}|, B_1 + 5, N_{fr}, 1)$, a min-max normalization is performed to coherently scale each feature sub-map/vector in the $[0, 1]$ range. The procedure leads to a representation reported in Figure 3.3.

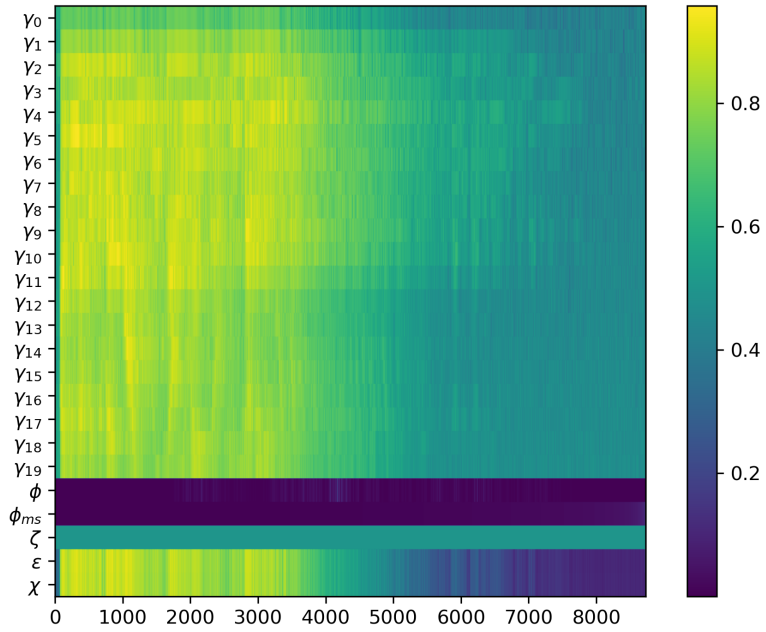


Figure 3.3: A normalized feature-map example

In particular, in Figure 3.3 we can observe a feature map built on a reverberant speech signal with quite small windows. On the horizontal axis we find the indices representing log-energy temporal frames or frequencies or quefrencies, while on the vertical one we represent the features.
Some feature vectors have very slight variation from the average, therefore their contribution is almost invisible in the figure.

Network architecture As explained in Section 2.5.2.1, the feature extraction block is common in both the regression and classification scenarios. Actually, what changes is the output FC layer.
The input batch traverses a batch normalization layer. There the normalization of the batch is performed to improve the network stability. Then, the tensors traverse a series of ConvPool blocks. Afterwards, a dropout regularization is performed and finally the FC block is reached. A description of the layers functioning is available in Section 2.5.2.1.
This architecture has been trained within the regression context to have a preliminary estimator of the volume and of the reverberation time. In addition, in the classification scenario, it has been trained for the in-band

room shape classification.
The feature extraction block is represented in Figure 3.4.

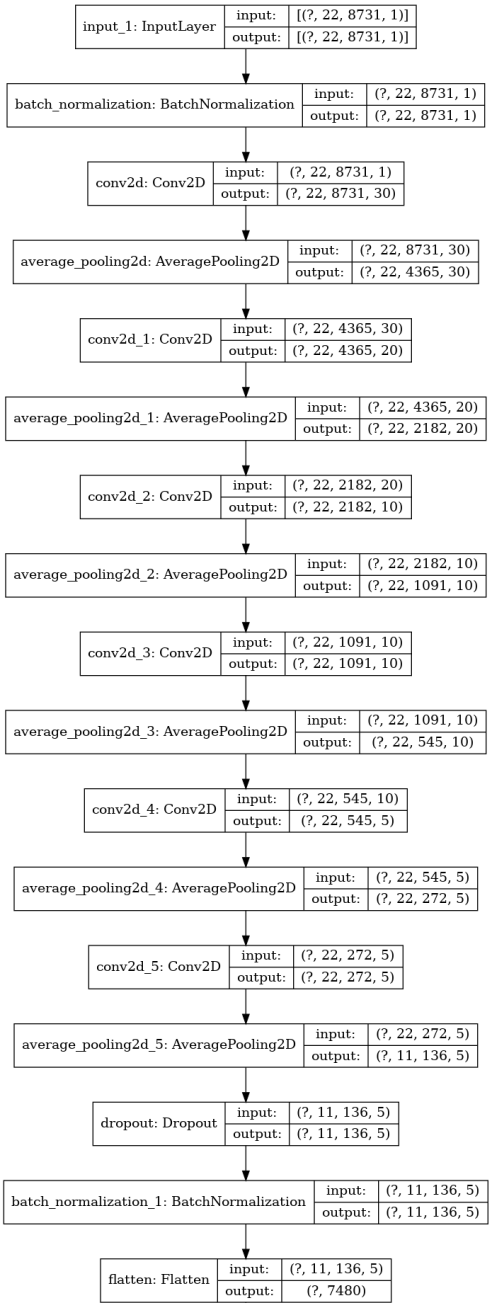


Figure 3.4: Network feature extraction block

The following Figure 3.5 represents the output block of the architecture with deterministic front-end.

dense: Dense	input:	(?, 7480)
	output:	(?, 1)

(a) Regressive block

dense: Dense	input:	(?, 7480)
	output:	(?, 3)

(b) Classification block

Figure 3.5: Network FC block

The 3.5a and 3.5b blocks respectively use ReLU and Softmax activations.

3.2.1.2 Room Geometry Inference - Learnable front-end Architecture

In our second proposal, the specific estimators belonging to the pipeline are built by filling the gaps of Section 2.5.2 and of the LEAF paragraph in Section 2.5.2.1.
In detail, we make use of a front-end network to compute an alternative representation of audio signals. The front-end network is connected to a standard back-end architecture, following the example in [16], to define the desired estimators.

Network architecture In this case, the neural architecture back-end is extended by introducing a front-end block.

Figure 3.6: LEAF front-end

Differently from the architecture with deterministic front-end, this architecture has been trained only within the classification context for the in-band room shape classification to improve some poor results of the aforementioned alternative.

Learnable audio front-end Here, fixed B_2 the number of Gabor-FB bands, the Gabor filtering stage introduces $2B_2$ parameters: center frequency and bandwidth of each of the B_2 filters.
Then, the gaussian low-pass stage introduces the cut-off frequency of each gaussian filter (one per band, B_2 parameters).
Finally, the sPCEN stage introduces $4B_2$ parameters for an overall amount of $7B_2$ trainable parameters.
The functioning of this front-end architecture has been more extensively explained in Section 2.5.2.1.

Follows an example of feature-map retrieved using LEAF.

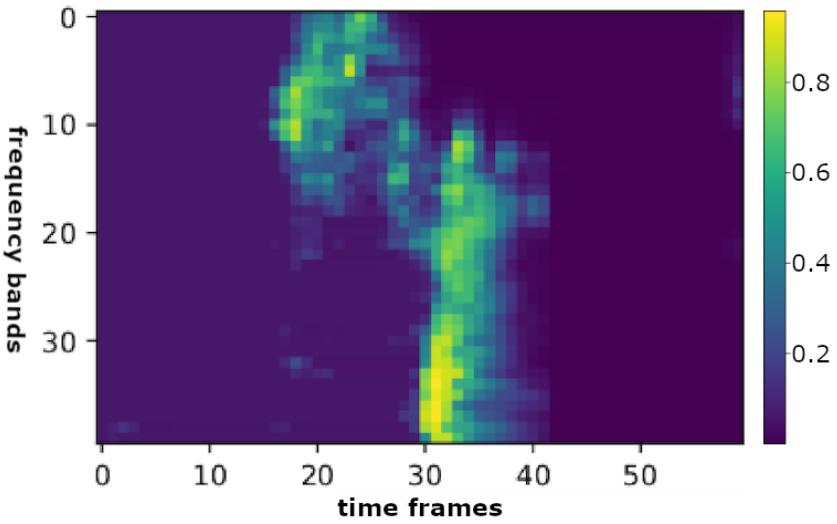


Figure 3.7: A LEAF feature-map example

Concerning Figure 3.7, on the horizontal axis we find temporal frame indices, while on the vertical one we find frequency band indices.

Back-end Similarly to what done by the authors of [16], we choose the lite EfficientNet-B0 as a standard back-end architecture for our work.

Stage	Operator	Resolution	#Channels	#Layers
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	28×28	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Figure 3.8: EfficientNet-B0 architecture [4]

Figure 3.8 has been extracted from the official reference [4] of Tan and Le and summarizes the network architecture.
Each *Stage* of the network wraps a certain number of layers *#Layers*. The Mobile inverted Bottleneck Convolutional (MBConv) stage has been defined in [36]. There k stands for the kernel size of the depthwise convolution layer belonging to the block. Finally, *Resolution* and *#Channels* represent the shape of the output tensor for each given *Stage*.

3.3 Conclusive remarks

In this chapter we described the problem formulation and the complete architecture of our proposed solution.
As far as the problem formulation is involved, we gave a description of the regression and classification problems applied to our scenario.
For what concerns the proposed solution, we analyzed our algorithmic pipeline to determine the best specific in-band shape estimator depending on a coarser volume or reverberation time estimation.

4
Simulations, Tests and Results

In this chapter we give details about the experimental setup and the results obtained using the proposed method.
First, we introduce our data generation framework designed to deal with big amount of data in a structured fashion. The framework is used for the generation of the datasets.
Then, we illustrate the details related to the experimental setup used for the validation stage.
Afterwards, we are both going to give details about our architectures and to analyze the results of our method implementation on the generated data.
Finally, we are drawing some observations and hypotheses based upon our experimental results.

4.1 Data generation framework

The data driven method described in the previous chapter requires the availability of a large dataset. To the best of our knowledge, a RIR dataset labeled with floor shapes does not exist. Therefore, we create an ad-hoc dataset to validate the proposed method. In the next section we describe the data model and creation pipeline.

4.1.1 Room shapes and data model

The data model is built to fit the Pyroomacoustics (PRA) python package [37] and extend it to automate the construction of some room classes. Concerning Figure 4.1 and Figure 4.2, they represent Unified Modeling Language (UML) class diagrams which summarize the structure of our

data model. There, the blocks represent classes, while the arrows represent inheritance.

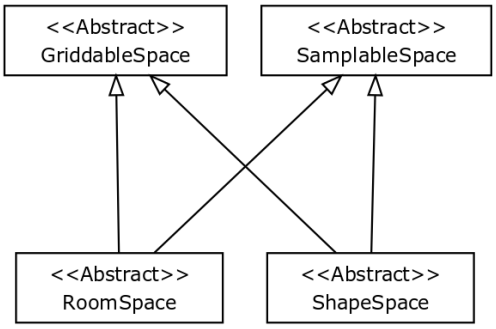


Figure 4.1: Sample spaces model

A *GriddableSpace* is a space on which a grid can be constructed. A *SamplableSpace*, instead, is a space from which samples can be extracted. The 2D *ShapeSpace* class is an abstract descriptor of the space containing floor plan maps built from vertices. With floor plan map we mean the 2D geometry from which a 3D room can be constructed via extrusion. In a similar way, *RoomSpace* describes the space containing all the possible rooms.
To gather the instances of a certain class, the room variables for each class can be limited to obtain realistic rooms. What is more, a step is fixed to have a discrete number of possible values for each class variable. For example, if we consider a room with a rectangular shape, each of its variables (width, height and depth) can be limited to a range. At the same time, a space can be sampled to extract a certain number of instances from the grid.

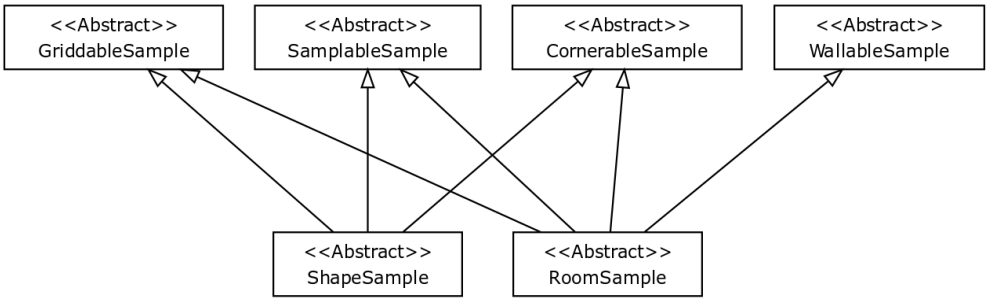


Figure 4.2: Sample model

The *GriddableSample* class exposes the method which allows to retrieve a grid of points within a room. Similarly, *SamplableSample* allows to retrieve a certain number of points coming from the grid. *CornerableSample* defines the abstract method which allows to retrieve the spatial vertices of a room. Finally, the methods encoded by *WallableSample* permit the retrieval of the set of walls of the room (as a set of PRA *pyroomacoustics.wall.Wall* instances). *RoomSample* schematizes the properties of an instance extracted from a *RoomSpace* instance

through sampling. The sampling procedure fixes the variables encoded by the classes which inherit from *RoomSample*. These classes do not appear in the diagram of Figure 4.2 for sake of compactness, however they are specific implementations of *RoomSample* for the geometries which we consider in our experiments: RectangleRoomSample with a rectangular floor plan, LRoomSample with an L-shaped floor plan and HouseRoomSample with a House-shaped floor plan. Finally, the variables belonging to the specific room class are exploited to retrieve the above mentioned properties, for instance the volume, the room vertices, the room walls and their surface.
A room sample can be newly meshed to retrieve all the geometrical points belonging to an admitted region. Otherwise it can be sampled to retrieve some examples from the grid.
The above model describes the data structure of each room sample. The samples can be generated with the volume factory.

4.1.2 Volume factory

Let a class, a volume range, a volume bandwidth and a number of desired room samples for each band in the range be given. In this context, with volume bandwidth we identify a narrow slice of the volume range in which we count a constant number of room samples (the desired ones). With the volume factory, we retrieve the number of desired samples for each bandwidth within the range by performing an exhaustive search in the room sub-space related to the geometry class. This allows to have an almost uniform volume distribution for the rooms sample-set in the overall range.

4.1.3 RIRs and reverberation time factory

RIRs generation Speaking about the generation of the RIRs in a room, we observe that PRA provides an implementation of the ISM algorithm. It is important to highlight that ISM algorithm reaches very high computational complexity for rooms with shape different from the common shoebox. In fact, to obtain an impulse response of, for instance, $T_{60} = 2s$ in a room of quite high volume (e.g. $1000m^3$), it is necessary to select a high maximum order (e.g. 20 or 30, modeling as relevant for the generation all the ISs within a maximum distance from the microphone). While the computation in shoebox rooms for quite high maximum order and volume can be optimized, the same does not hold for different shapes, where the computational time for one room might require even weeks.
For these reasons, we adopted an hybrid solution involving ISM to model early-reverberation and ray-tracing to model the reverberant tail, a trade-off between response quality and computational cost.
Given a room sample generated with the volume factory, its sampling

allows to select spatial points over which the response is computed, also parameterised by walls absorption coefficient and on desired T_{60}.

Reverberation time factory Let us consider a room with controllable absorption behavior, a target $T_{60}^{(tgt)}$ with a maximum acceptable temporal error T_ϵ. Assuming that $|\mathbf{m}-\mathbf{s}|$ is constrained, we consider the walls absorption coefficient as our parameter for controlling the measured $T_{60}^{(meas)} \in T_{60}^{(tgt)} \mp T_\epsilon$.
To do so, we perform a GD search in the absorption coefficient space, considering the distance among $T_{60}^{(meas)}$ and $T_{60}^{(tgt)}$ as our loss gradient. We start with an average absorption coefficient estimated from the inversion of Equation 2.3. Then $T_{60}^{(meas)}$ is measured from the response. The measurement at each iteration is performed using interpolation over Equation 2.7 on a third-octave-band basis and weighting the estimates with the sub-band energy normalized by the overall energy. If the temporal error is below the maximum acceptable error, the exit condition is matched. Otherwise, the absorption coefficient is updated depending on the loss gradient and the iterations continue.
This factory allows to generate RIRs with a T_{60} belonging to a certain range starting from the rooms set generated with the volume factory.

4.1.4 Generative framework

Considering the fact that we were going to deal with big amount of data, we have built a framework for data storage.
The advantages of such data storage framework can be summarized in the points below.

- prevention of data inconsistency;
- prevention of data mix-up;
- storage of an experimental history;
- lower RAM consumption during data generation.

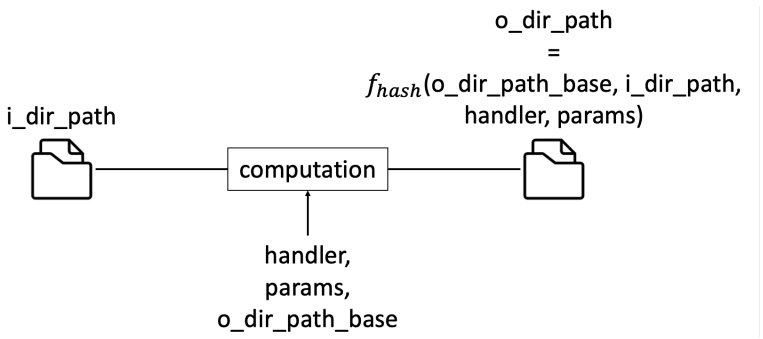


Figure 4.3: Generative framework wrap-up

To sum up, the *computation* block computes either serially or with multi-threading the *handler* function with the related *parameters* on each of the example files contained in the relative *input folder* (*i_dir_path*). Such input files are split among threads to grant load-balancing. The examples resulting from the computation are stored in the *output directory* (*o_dir_path*, relative to the output directory base path *o_dir_path_base*), which name depends on input and parameters of the *computation* block. Each output file stores relational references to input signals, some computational parameters, output signal and optionally further characteristic values.
The main drawback here is the overhead introduced by I/O operations, however we can consider it as negligible while generating big amount of data. Other approaches based on more structured data sources (e.g. NoSql DB [38]) could be considered.

4.2 Experimental setup and datasets

In this section we illustrate the implementation deatils relative to data generation framework. For what concerns the experimental setup, we start from the generation of rooms and to constrain the search, we fix the room space step and each class variable range. Moreover, we define a volume range which is split in volume bands in which a number of rooms must be found. We highlight that our almost exhaustive implementation can be easily extended to perform a search with a GD method.
Then, for each room, a certain number of RIRs is retrieved by randomly fixing a source $\mathbf{s} \in \mathbb{R}^3$ and a microphone $\mathbf{m} \in \mathbb{R}^3$ so that $|\mathbf{m} - \mathbf{s}| \in [0.8, 1.5]m$.
Furthermore, during the acquisition of the RIRs with ISM and ray-tracing, a V range is re-mapped in a T_{60} range to obtain the desired $T_{60}^{(tgt)}$s as references for the measured values (more details in Section 4.1.3). Here, fixed the source-microphone distance, the walls absorption is our main toy parameter to obtain rooms of a certain volume with a desired (and possibly realistic) reverberation time.
Finally, our study signals (reverberant white noises dataset and reverberant speeches dataset) are retrieved performing convolution between the RIRs in our responses set (generated with PRA) and randomic source input signals. If the desired source signal is a white noise, a random signal is generated for each response. Otherwise, if the desired source signal is an anechoic input speech signal, for each RIR a random sample is extracted from ∼7K audio signals belonging to the "CMU_ARCTIC" databases [39].

4.2.1 Rooms and setups

The generation of the rooms is constrained to be quite realistic. In table Table 4.1 we summarize the parameters for rooms construction.

Parameter	Parameter value
Floor map variables range	$[3, 20]m$
Floor map variables step	$0.2m$
Room height range	$[3, 6]m$
Volume range	$[50, 1050]m^3$
Generation volume bandwidth	$50m^3$
Samples per generation volume bandwidth	63

Table 4.1: Rooms generation parameters

In Figure 4.4 we observe the volume distribution considering all the samples resulting from the search.

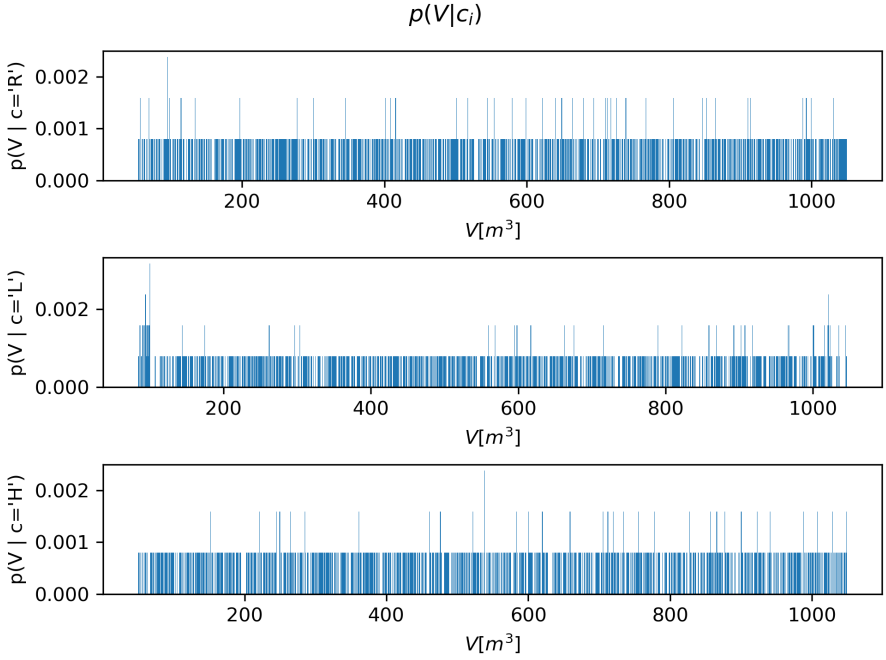
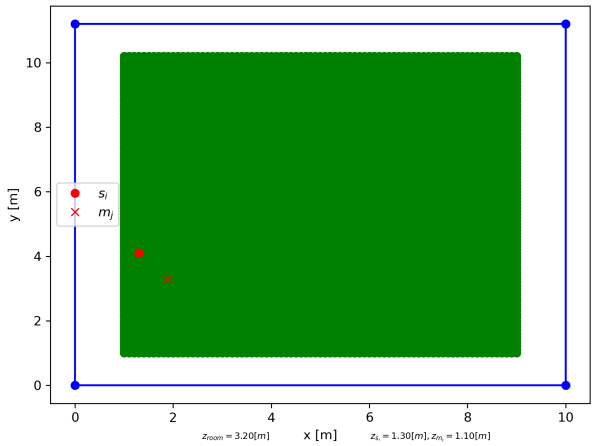


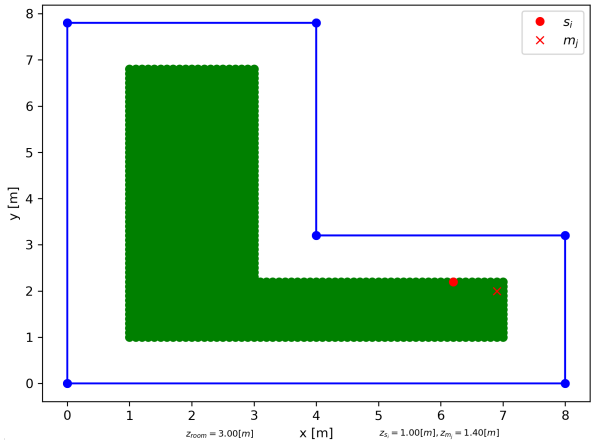
Figure 4.4: Volume distribution given the room class

An amount of 1260 unique rooms is extracted for each class, therefore we consider 3780 rooms. As visible, the volume distribution is almost uniform to avoid an unbalanced dataset.

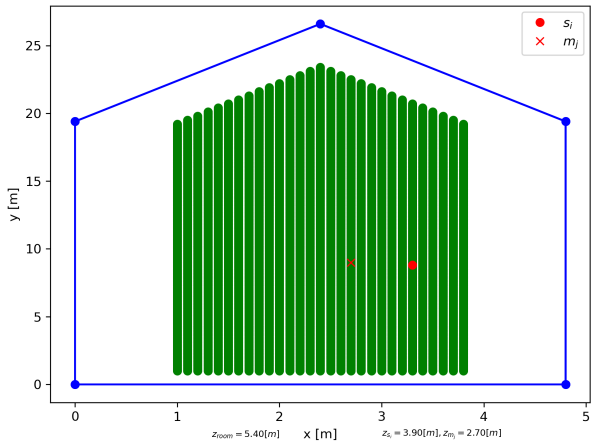
As a following step, we would like to retrieve approximately 10K RIRs from each room class, therefore for each class we perform 8 acquisitions per room from a source to a microphone. However, we would like to underline the fact that the number of extracted rooms, the number of acquisitions per room and the number of sources (with related input signals) and microphones per acquisition can be modified.
In Figure 4.5 we report some examples of setup in which we color in green the projection in the 2D plane of the admissible region in which spatial samples can be extracted.



(a) In a RectangleRoomSample



(b) In a LRoomSample



(c) In a HouseRoomSample

Figure 4.5: Setup examples

The admissible region is the hull containing green points which distance from the walls is of at least $1m$. This distance is selected to prevent the loss of first order peaks in the response, especially if we had considered co-located source and receiver. The sampling frequency is fixed to $F_s = 16KHz$. In the admissible regions examples in Figure 4.5, we drew

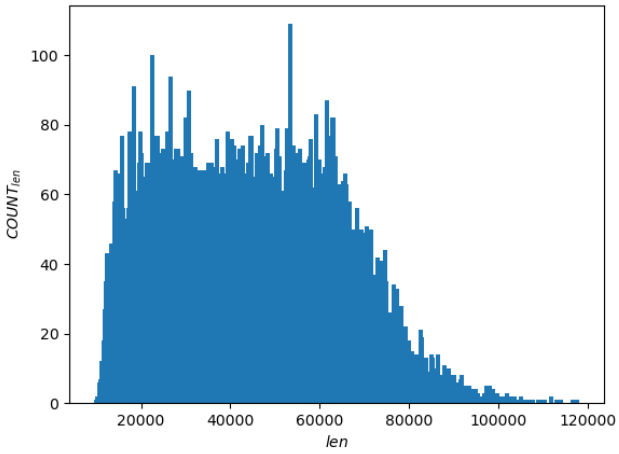
sources as red dots and microphones as red crosses.
Given the setup parameters, we obtain 30240 unique setup configurations (from which the RIRs are retrieved).

4.2.2 RIRs and reverberant signals

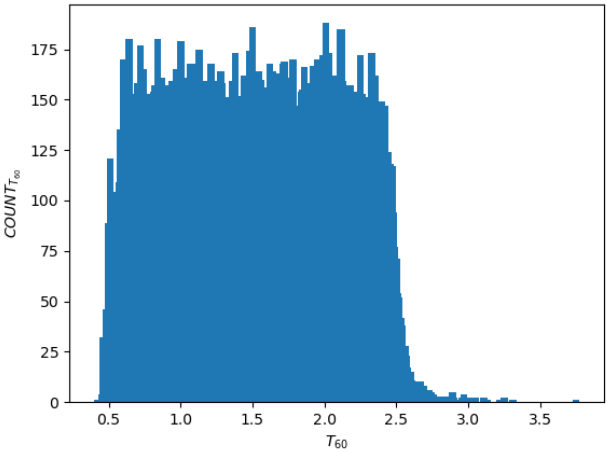
The above configurations allow to create an equal amount of responses and acquired signals ($|\mathcal{D}| = 30240$). If, for instance, we parametrized two microphones per acquisition instead of one, we would have ∼30K response tensors in the associated experimental folder (with source, microphone, time axes), but ∼60K mono microphone signals in the folder related to the acquisition handler for the current parameters.

For what concerns our description in Section 4.1.3 about the reverberation time manipulation, we decided to map our whole volume range in a $T_{60}^{(tgt)}$ range from $0.5s$ to $2.5s$ so that the absorption coefficients are not too high. In this way, the room samples corresponding to the volume distribution are not too far away from the room samples corresponding to the reverberation time distribution. We fix $T_\epsilon = 0.05s$ as maximum acceptable error for the measured $T_{60}^{(meas)}$ relative to the target $T_{60}^{(tgt)}$. What is more we adopted an hybrid ISM and ray-tracing approach to compute the responses. We fixed the maximum order of reflection for ISM to 10.

Figure 4.6 considers the whole RIRs set and shows the overall distribution of two properties: the length in samples and the T_{60}. Given the great number of samples per room class, we have proof that such distributions follow the same flat profile when considering each class separately. The advantage of digitally generating our data performing a search is that we have control on the desired distribution of the properties. The volume of examples obtained by randomly sampling a room space instance would be positively skewed, thus leading to the need of applying a transformation on the target (e.g. $\log(y)$ or $\sqrt[3]{y}$) to normalize its distribution.
However, we must take in account that it is not strictly necessary to have a normally distributed target variable for training. The prediction residuals must be normally distributed, instead. Indeed, we opt for approximately uniformly distributed targets and in the regressive scenario we are going to check for the error normality.



(a) Distribution of the lengths



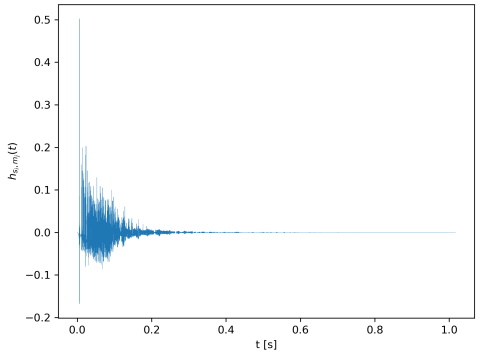
(b) Distribution of the $T_{60}^{(meas)}$s

Figure 4.6: Properties distribution of the whole RIRs set

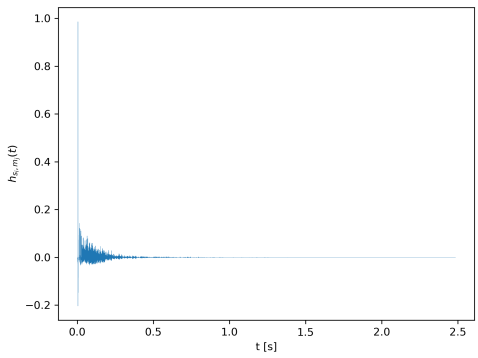
In Figure 4.6a the length variable is expressed in samples, remembering that for us $F_s = 16$KHz, while in Figure 4.6b the variable is expressed in seconds.
From the entire RIRs set, the reverberant signals are obtained convolving each response with a random source audio signal. As mentioned, each response belonging to the set is convolved with a random unique white noise or with a random anechoic speech from the CMU dataset to obtain reverberant white noises and reverberant speech signals respectively.
In the following, we present an analytical report of the reverberant signals.

4.2.2.1 Reverberant audio signals analysis

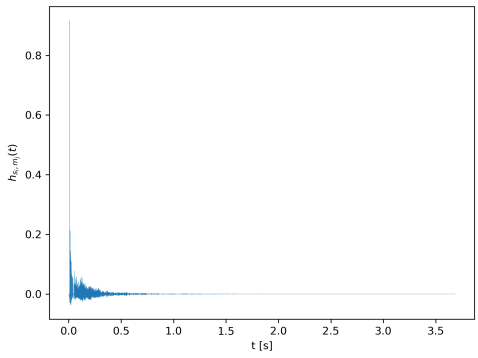
RIRs examples In Figure 4.7, Figure 4.8 and Figure 4.9 we show some examples of RIRs acquired in rooms of different shapes with increasing volume.



(a) $V \in [50, 100]m^3$

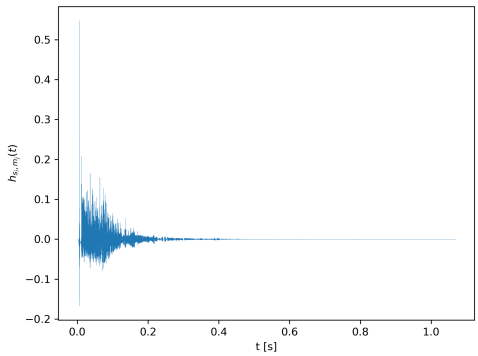


(b) $V \in [100, 700]m^3$

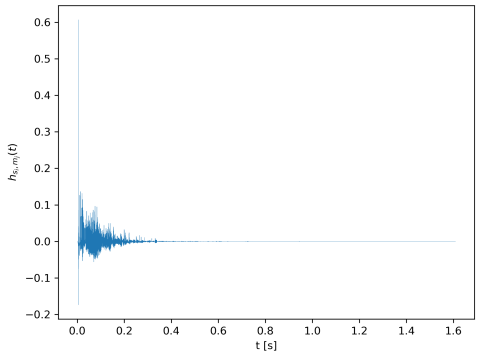


(c) $V \in [700, 1050]m^3$

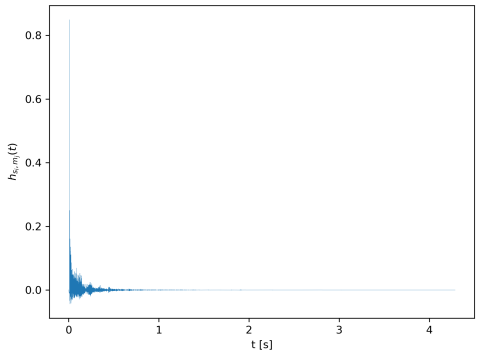
Figure 4.7: RIRs - RectangleRoomSample



(a) $V \in [50, 100]m^3$

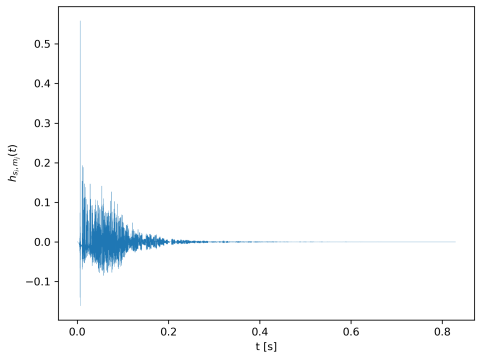


(b) $V \in [100, 700]m^3$

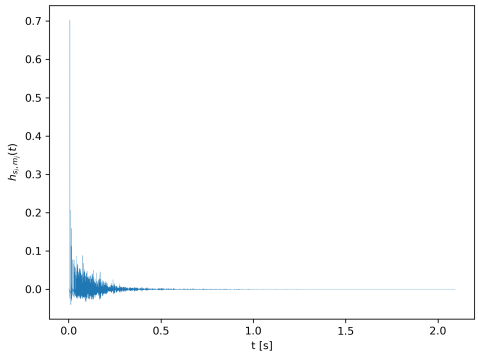


(c) $V \in [700, 1050]m^3$

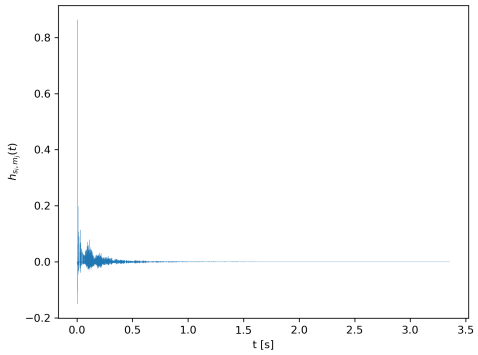
Figure 4.8: RIRs - LRoomSample



(a) $V \in [50, 100]m^3$



(b) $V \in [100, 700]m^3$

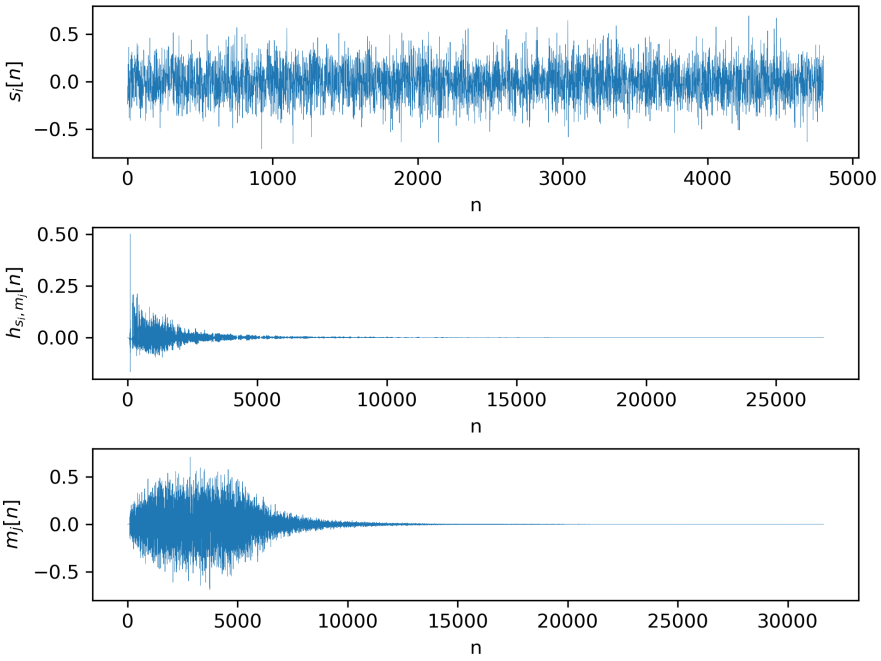


(c) $V \in [700, 1050]m^3$

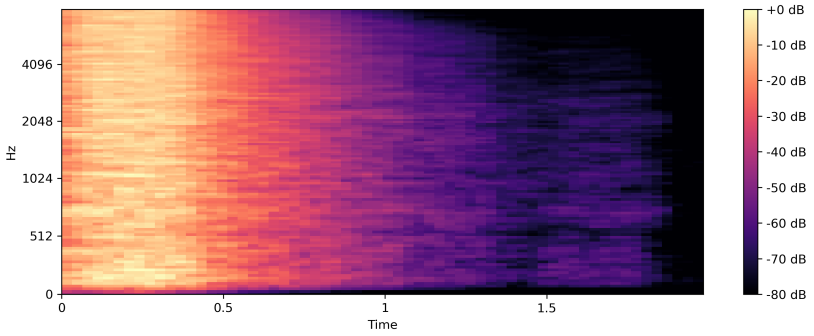
Figure 4.9: RIRs - HouseRoomSample

For the dataset construction, we observe that for these samples and in most cases the response length and the measured T_{60} become higher when the volume increases, but in general this is not always true. Our framework allows to map whichever volume range in whichever reverberation time range.

White noise examples In Figure 4.10, Figure 4.11 and Figure 4.12 we show a relevant extract of reverberant white noises together with their Mel-scaled spectrogram representation.

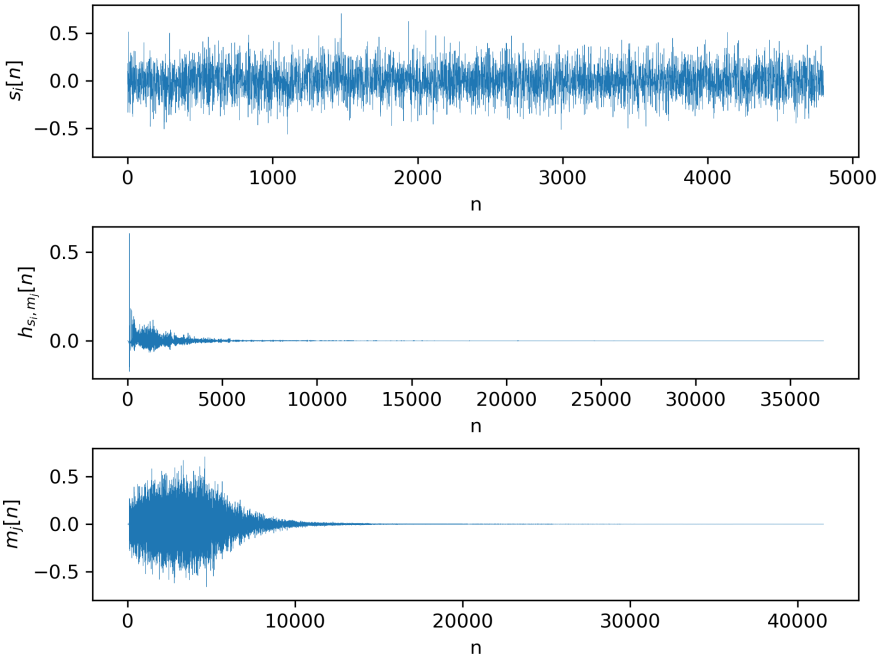


(a) Resulting signal

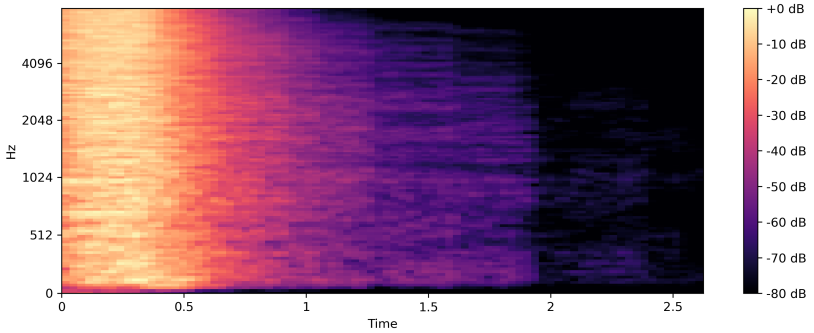


(b) Related Mel-spectrogram

Figure 4.10: White noises - RectangleRoomSample - $V \in [50, 100]m^3$

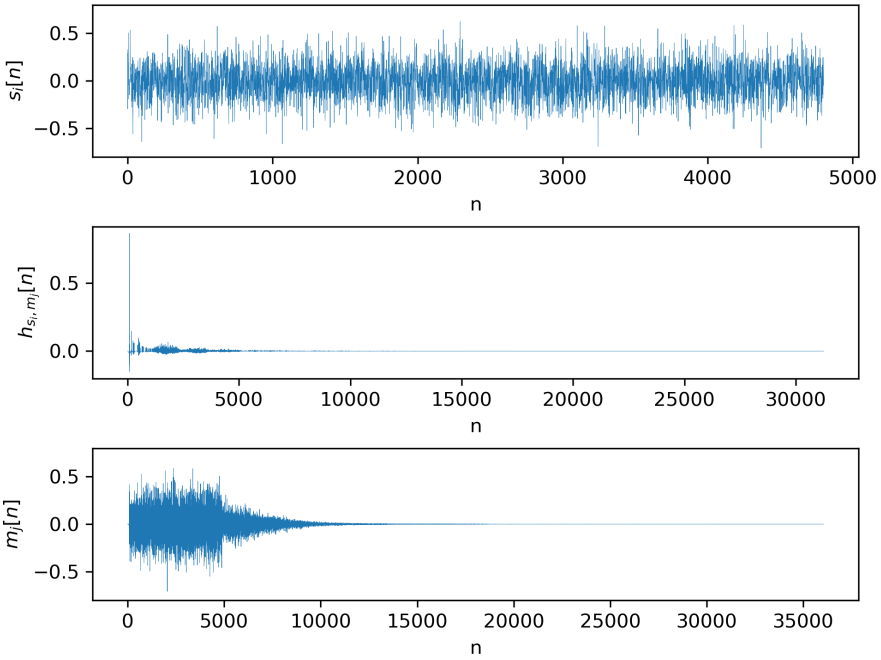


(a) Resulting signal

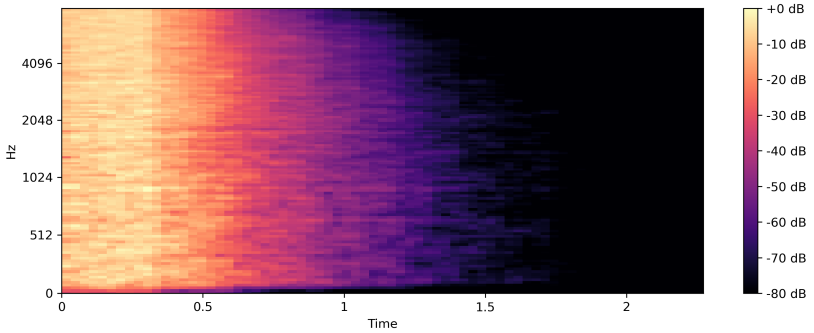


(b) Related Mel-spectrogram

Figure 4.11: White noises - LRoomSample - $V \in [100, 700]m^3$



(a) Resulting signal

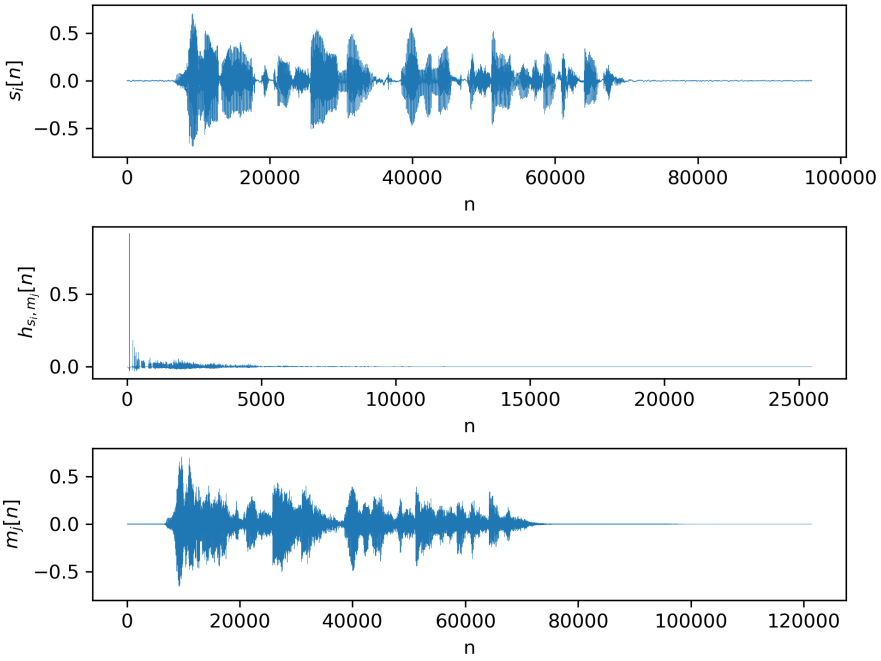


(b) Related Mel-spectrogram

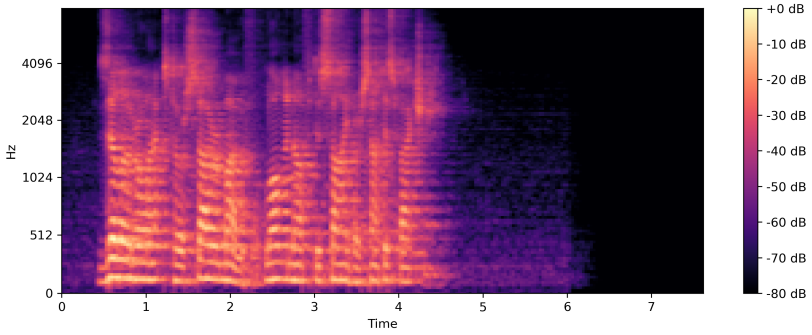
Figure 4.12: White noises - HouseRoomSample - $V \in [700, 1050]m^3$

Into Figure 4.10, Figure 4.11 and Figure 4.12 we observe the reverberant white noise signals m_j resulting from the convolution of an anechoic white noise s_i and a response h_{s_i,m_j}.
Such signals are acquired in rooms of different shape and different volume.
From the Mel-spectrograms we can glance at the sound decay behavior in the time-frequency domain.

Speech examples Following the outline of the previous paragraph, we report also samples of reverberant speech signals.

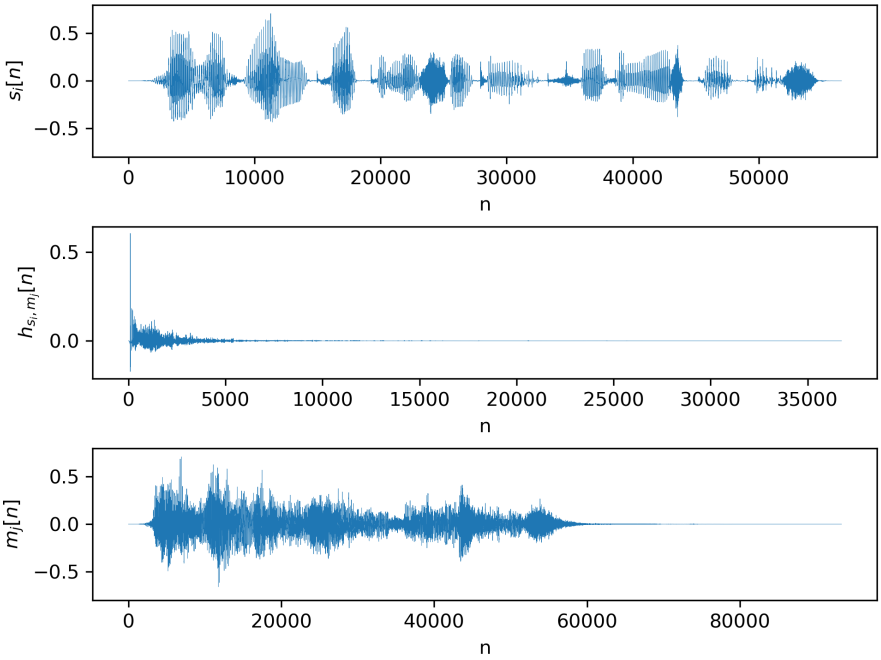


(a) Resulting signal

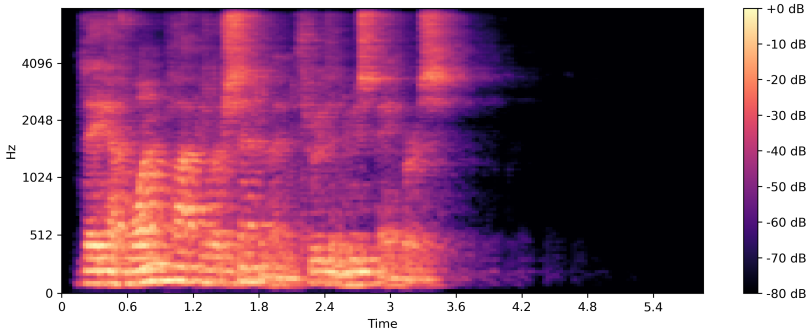


(b) Related Mel-spectrogram

Figure 4.13: Voices - RectangleRoomSample - $V \in [700, 1050]m^3$

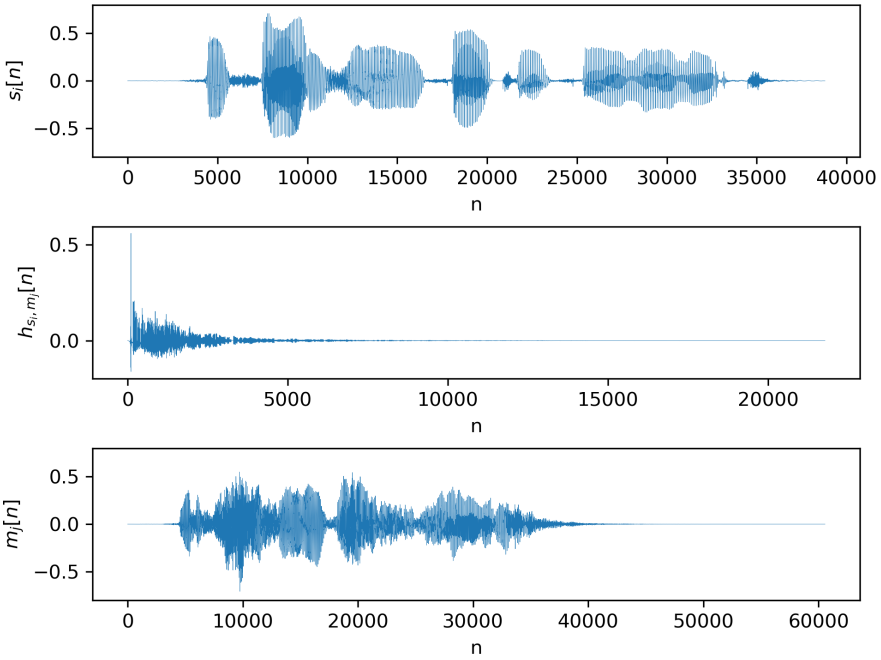


(a) Resulting signal

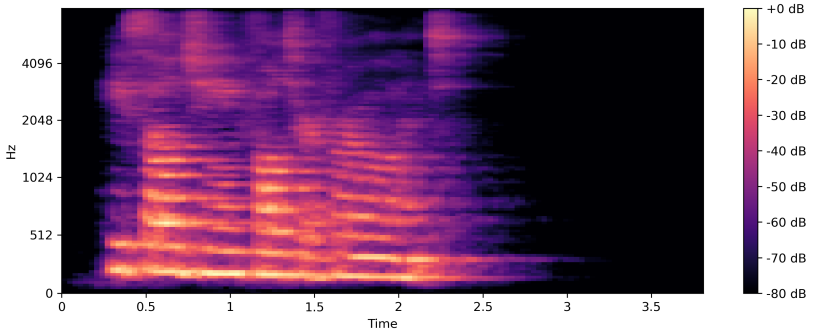


(b) Related Mel-spectrogram

Figure 4.14: Voices - LRoomSample - $V \in [100, 700] m^3$



(a) Resulting signal



(b) Related Mel-spectrogram

Figure 4.15: Voices - HouseRoomSample - $V \in [50, 100]m^3$

Similarly to the observations of the previous paragraph, into Figure 4.13, Figure 4.14 and Figure 4.15 we observe the reverberant speeech signals m_j resulting from the convolution of an anechoic speech s_i and a response h_{s_i,m_j}.
Such signals are acquired in rooms of different shape and different volume.
From the Mel-spectrograms we glance at the sound decay behavior in the time-frequency domain.

4.2.3 Datasets

The samples that we just analyzed come from one of the three datasets (one per signal type) that we built for our research. The main properties included in our final version of the datasets are the signal itself, the volume of the room in which the signal is acquired, the related T_{60} and the label of the room shape class.

Splits Each dataset of $|\mathcal{D}| = 30240$ samples is split in training, validation and test sets. The training set contains 70% of the samples, the validation set contains 20% of them and the test set contains the remaining samples. Depending on the band subdivisions, the main dataset is dynamically filtered while the split percentages are unvaried.

Band subdivisions As mentioned, the dataset for each source signal type contains ∼30K samples. To understand the relation among volume or T_{60} and shape we decide to analyze the behavior of a set of shape estimators ($S_G^{(sw1,sw2)}$ in Figure 3.2). Each of them has been trained on different input signals and either on a volume or reverberation time band.
The V_{mod} or T_{mod} module selects $M_{G_{\hat{b}}} \in S_G^{(sw1,sw2)}$ depending on the preliminary estimation coming from the previous computational block. At the moment, the training of the specific estimators is performed on the bands represented in Table 4.2.

V-bands $[m^3]$	**RT-bands** $[s]$
$[50, 250]$	$[0.5, 0.9]$
$[450, 650]$	$[1.3, 1.7]$
$[850, 1050]$	$[2.1, 2.5]$

Table 4.2: Band splits

Within each dataset, we decide to perform a filtering to retrieve ∼6K samples for a certain property in a certain band. As for the preliminary estimation, these samples are split in training, validation and test sets. Each split still follows an almost uniform distribution although we have less samples at our disposal.

Alternatively, we could have exploited our factories to extract more samples for a narrow volume band (instead of considering range and sub-bands), map them in a desired narrow reverberation time band, produce a final different dataset for the training of each specific specific model.
Following this alternative procedure, the pro would be the presence of many more samples in each band, while the primary con would be the time consumption.

4.2.4 Front-end parameters definition

We described the details which are foundation for the generation of our datasets. We are left with the description of the initialization parameters of our estimators architectures.

RGIDA For what concerns Room Geometry Inference - Deterministic front-end Architecture (RGIDA), used for the preliminary volume estimation and for shape classification, we used the parameters reported below in Table 4.3.

Feature	Parameters	Parameters value
$\underline{\underline{\mathbf{\Gamma}}}$	$(B_1, F_s, F_{lo}, F_{hi}, L_{win}, L_{hop})$	$(20, 16000, 50, 2000, 32, 16)$
$\underline{\phi}$	(F_s, F_{hi}, N_{fft})	$(16000, 500, N_{fr})$
$\underline{\phi}_{ms}$	()	()
$\underline{\zeta}$	(F_s, F_{hi}, N_{quef})	$(16000, 500, N_{fr})$
$\underline{\epsilon}$	(F_s, L_{win}, L_{hop})	$(16000, 32, 16)$
$\underline{\chi}$	(F_s, L_{win}, L_{hop})	$(16000, 32, 16)$

Table 4.3: RGIDA front-end parameters

Depending on the signal type and on the maximum length of the signals belonging to each dataset, the parameters in Table 4.3 might undergo slight variations.
We remind that N_{fr} has been defined in the pre-processing paragraph of Section 3.2.1.1 and depends on the maximum length of the signals belonging to a certain type and both on L_{win} and L_{hop}.

RGILA As far as Room Geometry Inference - Learnable front-end Architecture (RGILA) is involved, apart from other specific parameters mainly inherited from [16], we set $B_2 = 128$ filters for the Gabor-FB.

4.2.5 Back-end parameters and metrics definition

In this section we are giving some specifications about the losses, metrics and optimizers used for the training of the networks in the different scenarios.
We highlight that such parameters might undergo slight variations depending on the specific cases.
All the estimators do exploit Adam optimizer and train over 1000 epochs considering early stopping as soon as there are no performance improvements. The batch size is fixed to 64 or 32 depending on the number of training examples.

4.2.5.1 RGIDA

Preliminary estimators The preliminary regressors do minimize a MSE loss as described by Equation 2.16. Furthermore, a MAE metric is

considered.
In Table 4.4 we present the initial network parameters.

Parameter	Parameter value
Initial LR	1×10^{-3}
LR reduction factor on plateau	0.2
Plateau patience	10
Significative loss variation	$100m^6$ (V) / $0.01s^2$ (RT)
Early-stopping patience	33

Table 4.4: Initial RGIDA parameters for preliminary estimators

Specific estimators The specific classifiers do minimize a H cross-entropy loss as described by Equation 2.17. What is more, a confusion matrix metric is taken into account.
In Table 4.5 we summarize the initial network parameters.

Parameter	Parameter value
Initial LR	1×10^{-1}
LR reduction factor on plateau	0.1
Plateau patience	30
Significative loss variation	0.01
Early-stopping patience	93

Table 4.5: Initial RGIDA parameters for specific estimators

4.2.5.2 RGILA

Specific estimators The main parameter for the specific RGILA shape estimators is the initial LR fixed to 1×10^{-4}.

4.3 Experiments and results

Explained the different aspects of data generation, of the evaluation setup and of the architectures initialization, we are going to compare the results obtained from the experiments conducted with estimators based on RGIDA (Section 3.2.1.1) and RGILA (Section 3.2.1.2) for shape estimation.
Firstly, we are going to analyze the results of the RGIDA preliminary estimators for volume and reverberation time depending on the input signal type. Then, we are going to compare the results of the classification task for both the RGIDA and RGILA architectures depending on source signal type and volume or reverberation time band subdivision.
Finally, we are going to provide some insights into the features useful for our classification task with Grad-CAM.

4.3.1 Preliminary estimators

In this section we are going to report the performances of the estimators which allow to retrieve the in-band estimators. For the training, in this case, we use the whole dataset for each signal type and we split it into training, validation and test sets. The goal of preliminary estimators is to perform regression on either volume or reverberation time. To do it, the sole RGIDA architecture has been used and a MSE loss is minimized.

4.3.1.1 Volume prediction

In detail, we are going to cover the block M_V of Figure 3.2.
In Table 4.6 we report the performances of our preliminary volume estimators considering three types of signals, the RIRs, the reverberant white noises and the reverberant speeches. Such performances are retrieved on the validation set.

RGIDA regression metrics	**RIRs**	**White noises**	**Voices**
MAE $[m^3]$	27	75	143
MSE $[m^6]$	2422	12139	31959

Table 4.6: Preliminary volume estimation metrics per signal type

From the summary table we see that we obtain better results from the RIRs. The performances degrade when we take in account the reverberant speech signals. This is due to the hardness introduced by the convolution and to the characteristics of the input signals that we consider.

Reverberant speech signals details As expected, we observe that our estimator based on RGIDA encounters increasing hardness while facing the problem with the different signals.
For completeness, below we show some more details of the behavior of the architecture on speech signals.

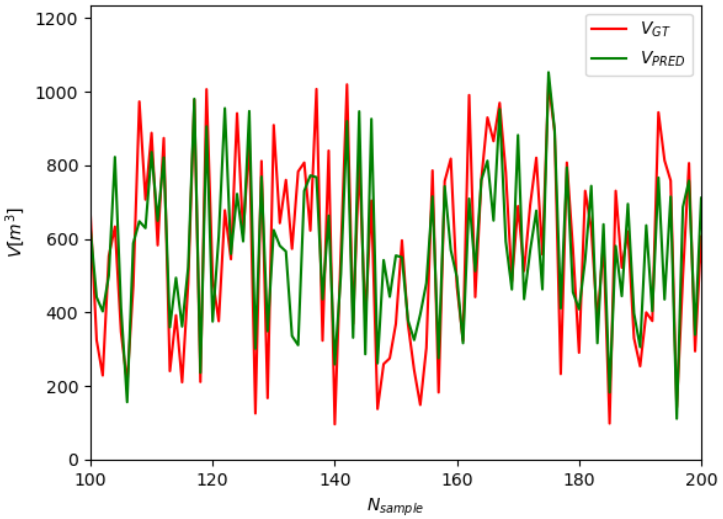


Figure 4.16: Preliminary volume estimation on vocal signals - Prediction samples

In Figure 4.16 we show some ground-truth volume samples coming from the test dataset (red) compared with their prediction (green).

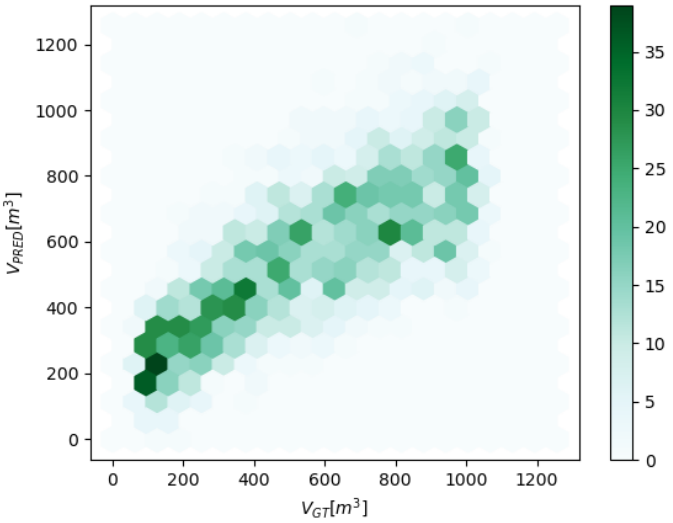
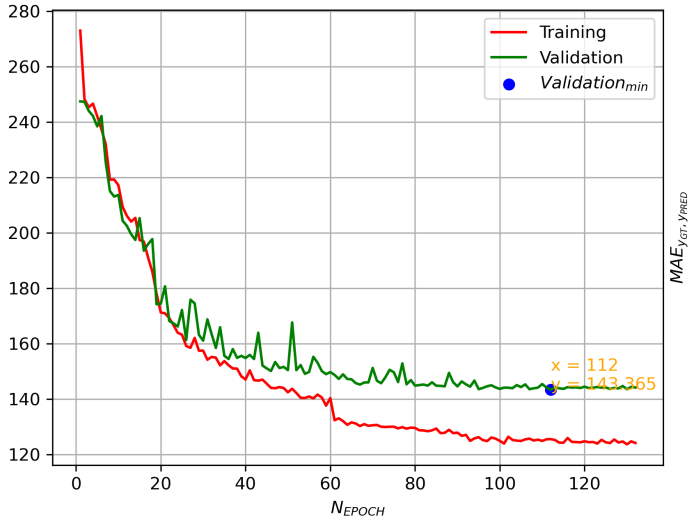


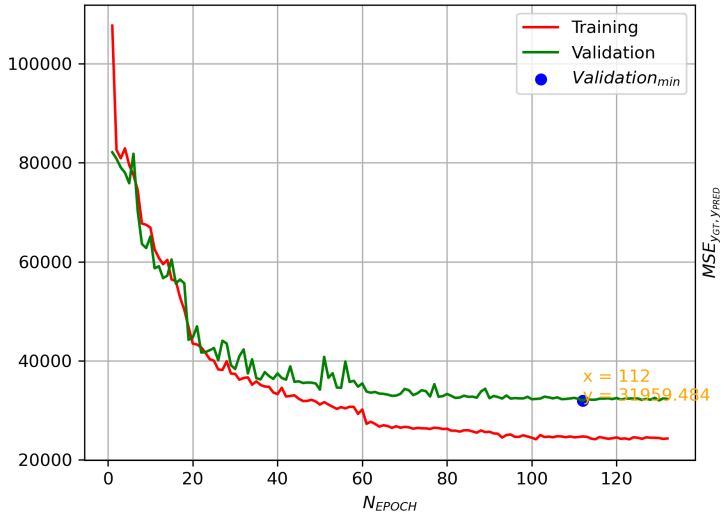
Figure 4.17: Preliminary volume estimation on vocal signals - Predictions spread

In Figure 4.17, instead, we observe that the spreading of the predicted values with respect to the ground-truth values does increase when the volume grows.

Although we are not going to include an extensive report here, from our experiments we see that the same observation seems to hold true also for the specific in-band volume estimators: the greater the volume, the harder the volume prediction and the easier the shape classification. This might depend on the problem, on the number of samples per band, on the quality of our simulated responses, or on many other factors. Furthermore, we observe that in the case of preliminary volume prediction on signals other than voice, the spreading is less pronounced and the dependence on the volume is slighter.



(a) MAE



(b) MSE

Figure 4.18: Preliminary volume estimation on vocal signals - Losses samples

Finally, for what concerns the training phase, in Figure 4.18 we would like to report the loss functions overtime over both the training (red) and validation (green) sets, while the LR decreases.

4.3.1.2 Reverberation time prediction

In detail, we are going to cover the block $M_{T_{60}}$ of Figure 3.2.
In Table 4.7 we report the performances of our preliminary RT estimators considering three types of signals, the RIRs, the reverberant white noises and the reverberant speeches. Such performances are retrieved on the validation set.

RGIDA regression metrics	RIRs	White noises	Voices
MAE $[s]$	0.06	0.04	0.09
MSE $[s^2]$	0.009	0.005	0.015

Table 4.7: Preliminary RT estimation metrics per signal type

Reverberant speech signals details We observe that, our estimator based on RGIDA has more difficulty in facing the problem for speech signals.
For completeness, below we show some more details of the behavior of the first architecture on speech signals.

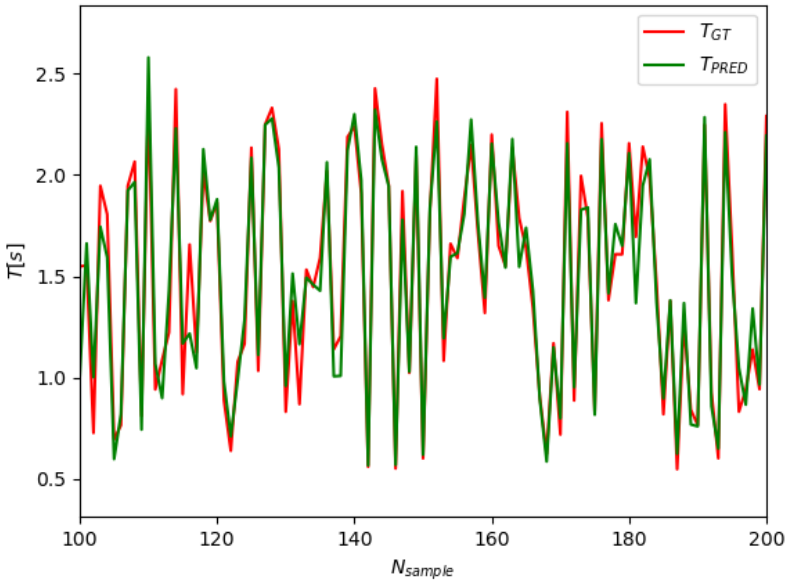


Figure 4.19: Preliminary RT estimation on vocal signals - Prediction samples

In Figure 4.19 we show some ground-truth T_{60} samples coming from the test dataset (red) compared with their prediction (green). In this case, our prediction error is pretty much reduced.

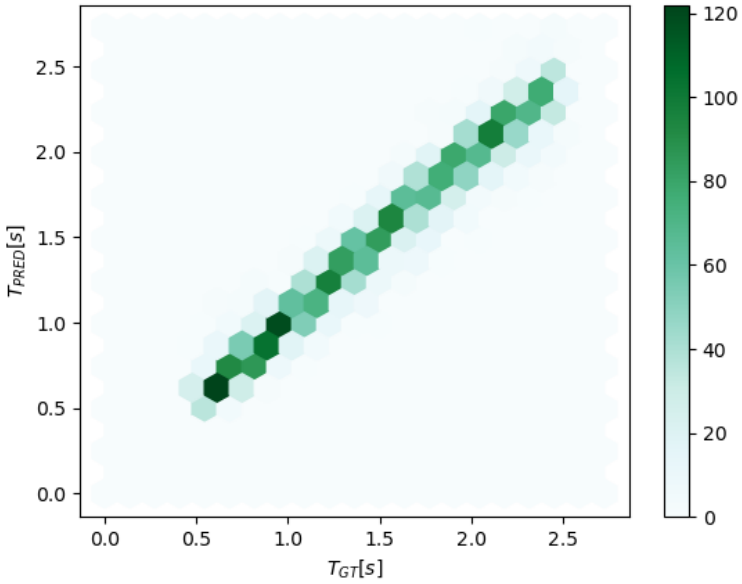
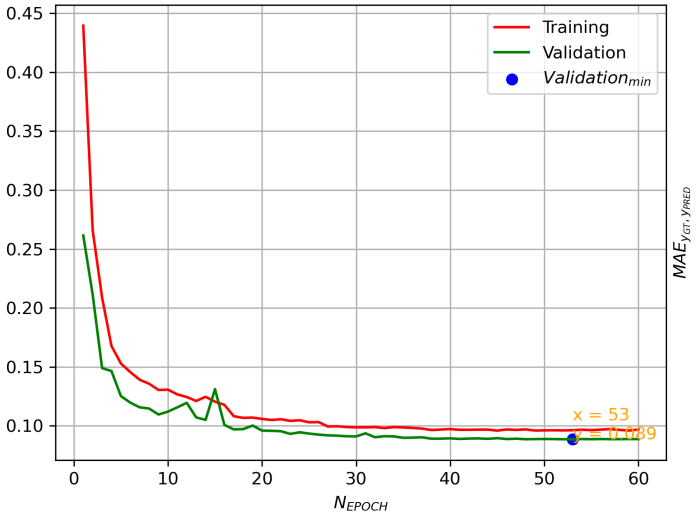
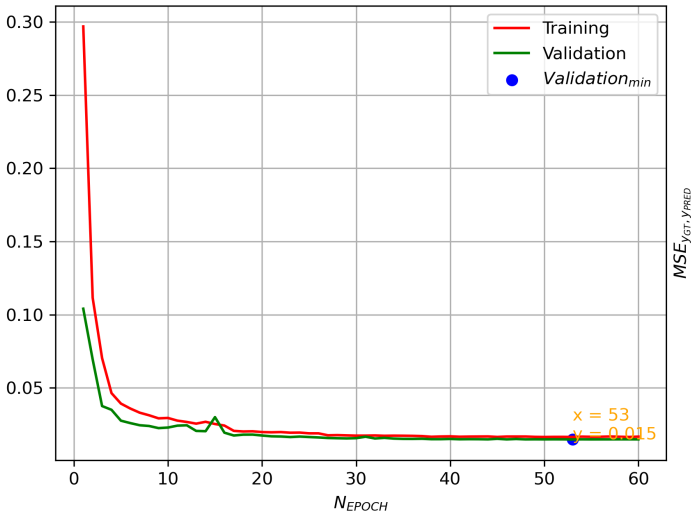


Figure 4.20: Preliminary RT estimation on vocal signals - Predictions spread

Contrary to what observed in the volume estimation scenario, it is interesting to notice from Figure 4.20 that the spreading of the predicted values is way more reduced and better follows the ground values.
This holds true even in the case of reverberant speech signals.



(a) MAE



(b) MSE

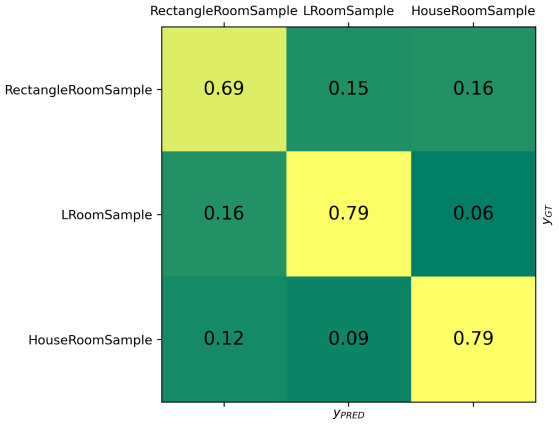
Figure 4.21: Preliminary RT estimation on vocal signals - Losses samples

In conclusion, in Figure 4.21 we report the loss functions overtime while the LR decreases.
Contrary to the volume case, the T_{60} regression seems a simpler task. This conclusion is quite intuitive and is partly proved by the fact that the predictions are better spread around the true values. We also observe that, in this case, the validation curve (green) lies underneath the training one. Considering the vast amount of samples used for the preliminary estimation and the balanced dataset (i.e. almost uniform distribution of the target T_{60} in the dataset splits), we hypotize that this trend might be due to a strong affection of the regularization (performed by the dropout layer during the training) on the current estimation task.

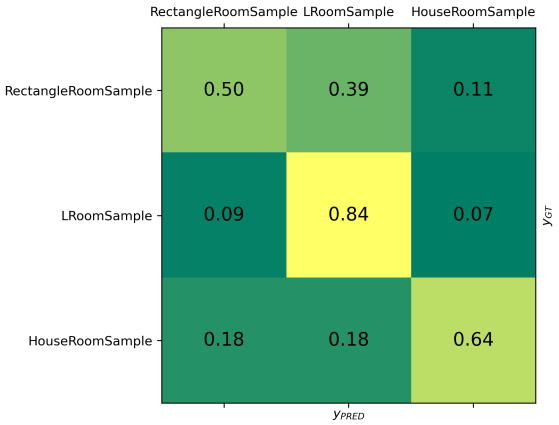
4.3.2 Shape Classification - In-volume-band specific estimators

In this section we are going to dig into the $S_G^{(sw1,V)}$ specific estimators. Speaking about the in-band estimators, we recall that the whole range of the volume variable is $[50, 1050]m^3$, while the bands that we are considering are $[50, 250]m^3$, $[450, 650]m^3$ and $[850, 1050]m^3$. The competence of these estimators is to perform shape classification on a volume-band basis. To do it, we compare the RGIDA and RGILA architectures and a H categorical cross-entropy loss is minimized.
This subsection compares the architectures at the variation of signal type and specific volume band.

Using RIR signals Here we compare the performances of both the architectures over the impulse responses with estimators trained on separate and increasing volume-bands.

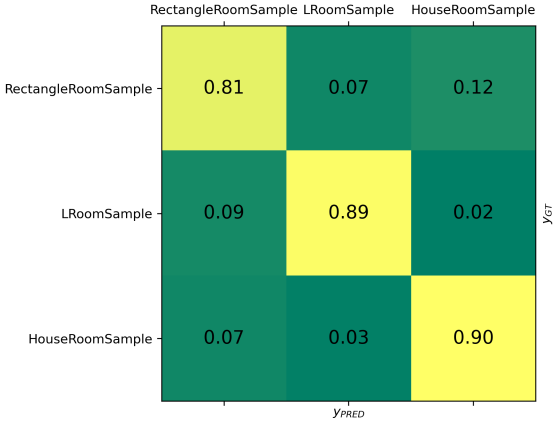


(a) RGIDA

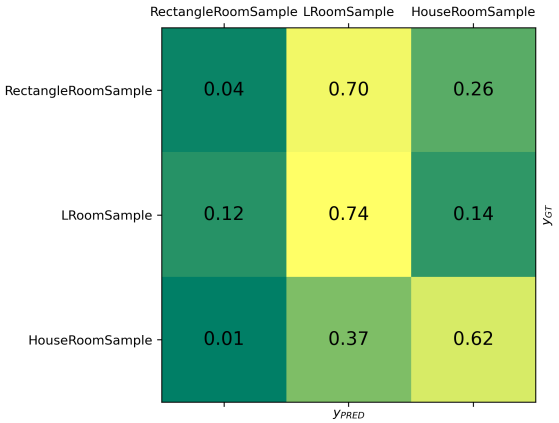


(b) RGILA

Figure 4.22: Specific in-volume-band shape classification on RIR signals - $V \in [50, 250]m^3$

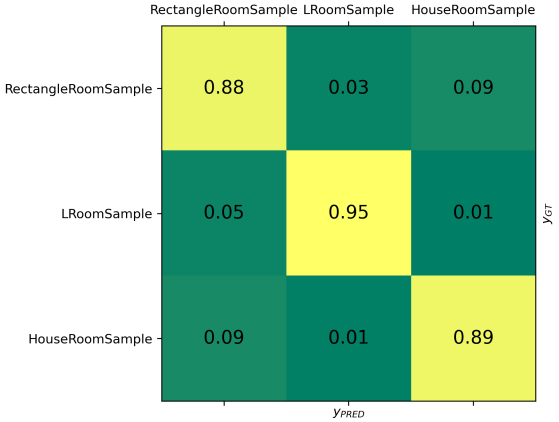


(a) RGIDA

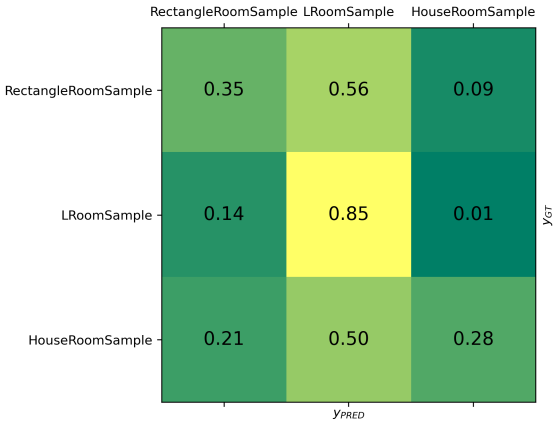


(b) RGILA

Figure 4.23: Specific in-volume-band shape classification on RIR signals - $V \in [450, 650]m^3$



(a) RGIDA



(b) RGILA

Figure 4.24: Specific in-volume-band shape classification on RIR signals - $V \in [850, 1050] m^3$

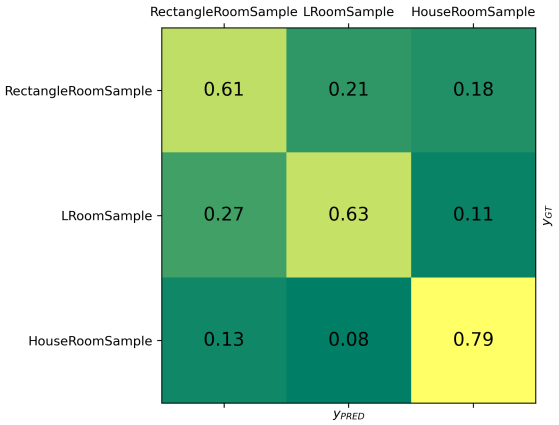
	ACC [%]	
V-band $[m^3]$	**RGIDA**	**RGILA**
$[50, 250]$	75	65
$[450, 650]$	88	50
$[850, 1050]$	89	52

Table 4.8: Specific in-volume-band shape classification on RIR signals - RGIDA vs RGILA comparison

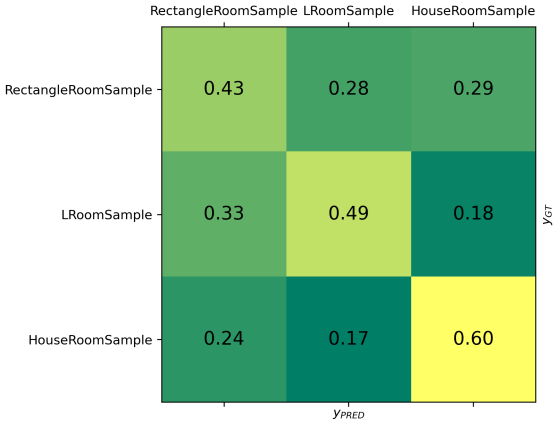
From the results of Table 4.8 we observe that the RGIDA architecture performs better than the RGILA one.
In particular, it seems harder for the architecture with LEAF front-end to dicriminate among rectangular and L-shaped classes for great volumes. This might be due to the nature of its pooling and compression strategies leading to a less informative representation.
Interestingly enough, we also observe that the shape estimation task for the RGILA performs better for wider rooms (i.e. increasing volume):

greater volumes lead to a wider early reverberation region in the response with peaks which are more separated and patterns which are more easily recognizable.
We detect that the rectangular room requires more effort for the recognition. What is more, to lower volume bands correspond poorer accuracy results.

Using reverberant white noise signals Here we compare the performances of both the architectures over the reverberant noises with estimators trained on separate and increasing volume-bands.

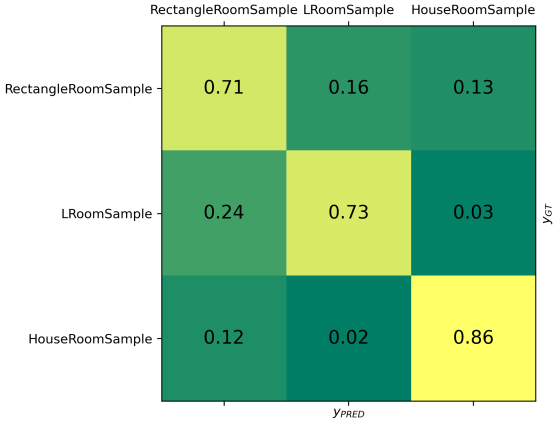


(a) RGIDA

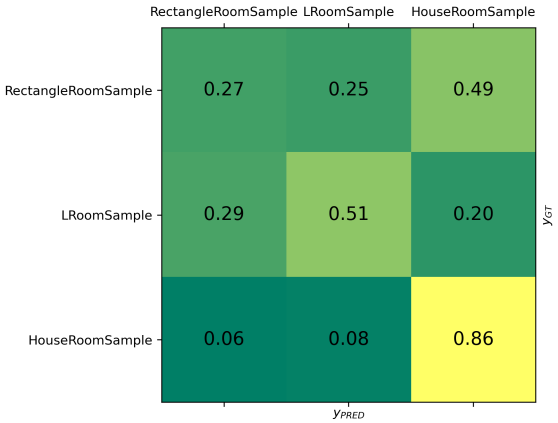


(b) RGILA

Figure 4.25: Specific in-volume-band shape classification on white noise signals - $V \in [50, 250]m^3$

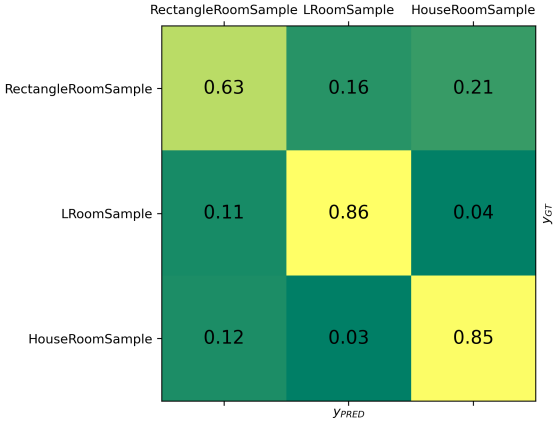


(a) RGILA

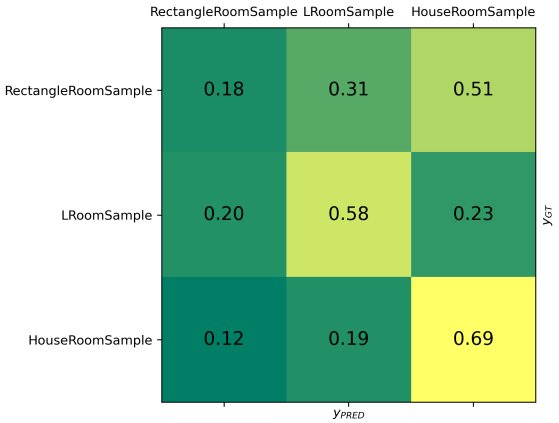


(b) RGILA

Figure 4.26: Specific in-volume-band shape classification on white noise signals - $V \in [450, 650]m^3$



(a) RGIDA



(b) RGILA

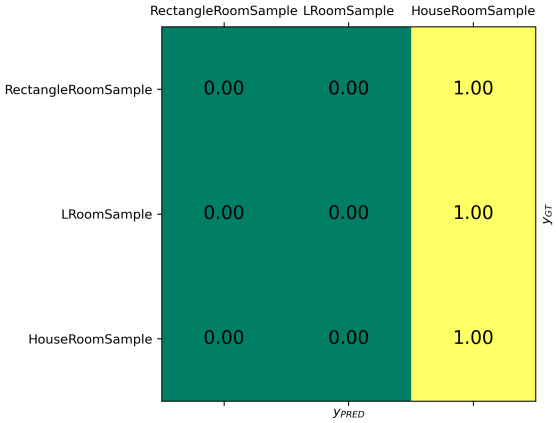
Figure 4.27: Specific in-volume-band shape classification on white noise signals - $V \in [850, 1050] m^3$

	ACC [%]	
V-band $[m^3]$	**RGIDA**	**RGILA**
$[50, 250]$	69	56
$[450, 650]$	75	63
$[850, 1050]$	80	52

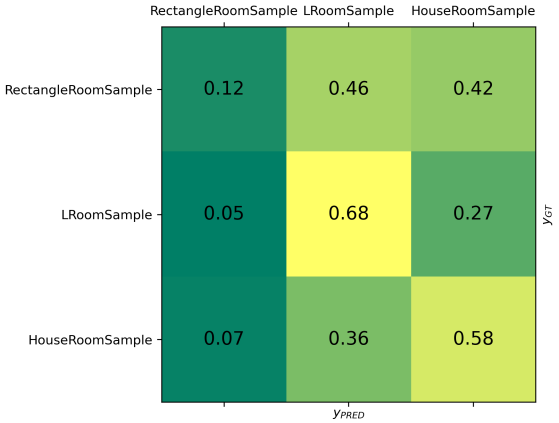
Table 4.9: Specific in-volume-band shape classification on white noise signals - RGIDA vs RGILA comparison

As for the previous input signal, we have evidence of a performance improvement as the volume expands for the first proposed RGIDA architecture. In this case, though, the second RGILA network seems more confused by the rectangular and House-shaped classes.

Using reverberant speech signals Here we compare the performances of both the architectures over the reverberant voices with estimators trained on separate and increasing volume-bands.

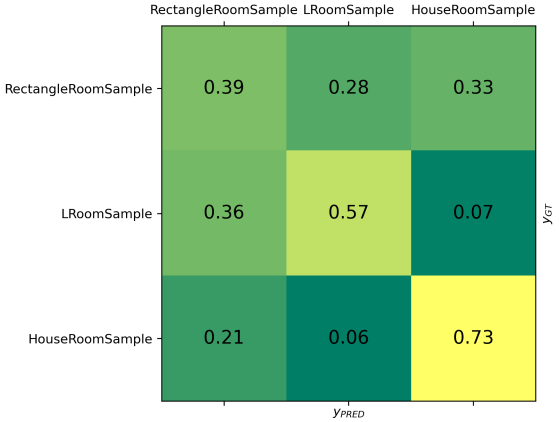


(a) RGIDA

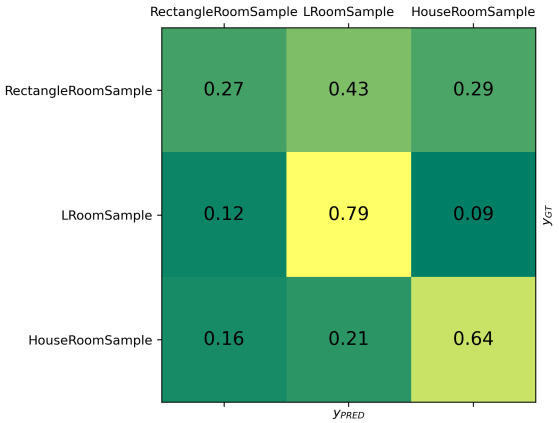


(b) RGILA

Figure 4.28: Specific in-volume-band shape classification on vocal signals - $V \in [50, 250] m^3$

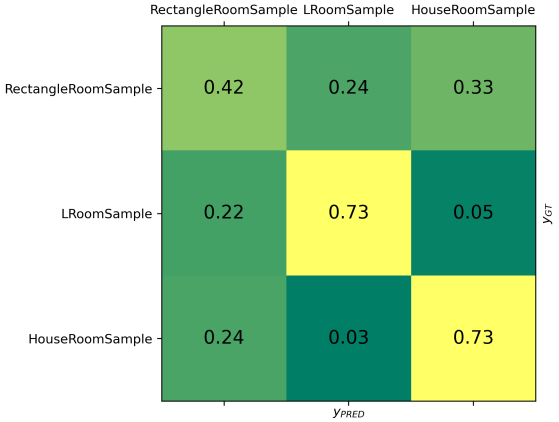


(a) RGIDA

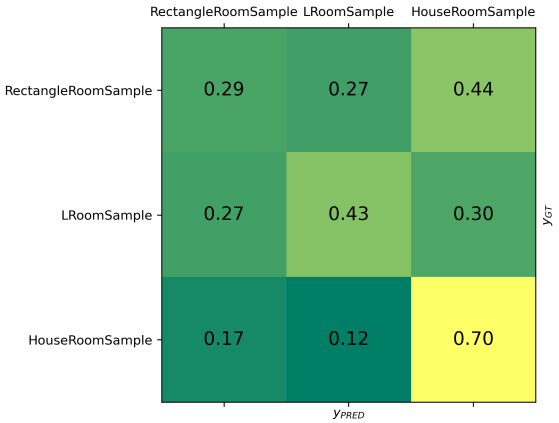


(b) RGILA

Figure 4.29: Specific in-volume-band shape classification on vocal signals - $V \in [450, 650]m^3$



(a) RGIDA



(b) RGILA

Figure 4.30: Specific in-volume-band shape classification on vocal signals - $V \in [850, 1050]m^3$

	ACC [%]	
V-band $[m^3]$	**RGIDA**	**RGILA**
$[50, 250]$	38	46
$[450, 650]$	58	59
$[850, 1050]$	67	53

Table 4.10: Specific in-volume-band shape classification on vocal signals - RGIDA vs RGILA comparison

In this case, there is a performance enhancement as the volume gets bigger, however, for what concerns the first volume band, the accuracy of RGIDA is worse than the one of RGILA.
Again, similarly to the noisy case, RGILA seems more confused among rectangular and House-shaped rooms.
A relevant observation which can be moved is that with reverberant voices, the L and House-shaped rooms can be easily recognized, while there seems to be no clue which allows to discriminate them from the

rectangular shape.

To summarize, what is common to all the source signals is a greater difficulty in the classification of the rectangular shape room and an accuracy improvement which follows the volume growth for all the classes. What differences them, instead, is a greater complexity of the speech signal type with respect to the other ones.
A motivation to this latter behavior could be found in the characteristics of the vocal signals. Indeed, a speech signal has a great variability in time and new spoken words might mask the reverberant content of previous words in certain frequency bands. In addition, the frequency content of the voice at the end of a word, generating a complete decay region, might not be sufficient for the stress of room modes which might be informative for the task.

Finally, we would like to mention our unfortunate attempt while trying to improve our results in the first volume or RT band with RGIDA. We tried to alter the feature-maps content by retaining the sole γ-FB since we assumed it to be most relevant feature for our task.

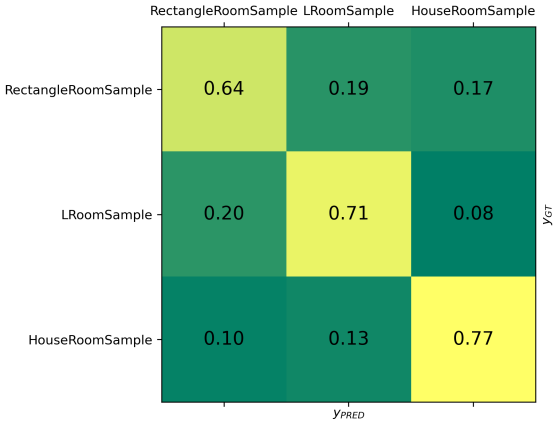
We tried to consider the final portion of 4s of each audio signal (this should not be a strict constraint for the first volume band because the considered reverberation time is much lower than 4s: we did so to limit memory consumption) while maintaining the bank range in $[50, 2000]Hz$ or changing it to $[50, 4000]Hz$ and enhancing in both cases the filters density (from 20 to 30 bands).
We also reduced the log-energy window size to $L_{win} = 8$ and the hop size to $L_{hop} = 4$ to soften the compression.
Furthermore, we augmented the initial LR for the network to prevent the possibility to get stuck in local minima.

And again, we tried considering binary classifiers on each possible couple of classes with the hope of improving the results on such critical band. We desired to search for a weighting strategy over binary distributions to retrieve an improved distribution for the three-classes case, but even the binary classifiers could not provide any advantage. Hence, we were not able to improve the results.

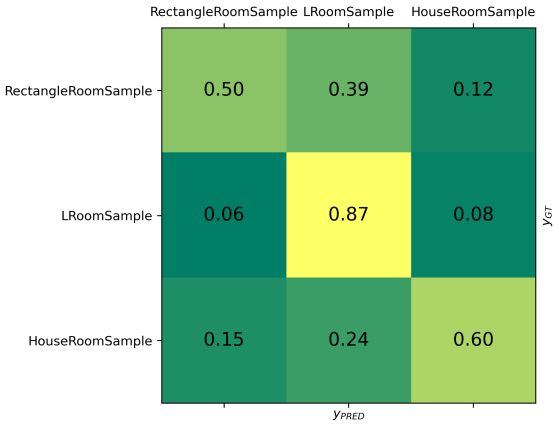
4.3.3 Shape Classification - In-RT-band specific estimators

In this section we are going to dig into the $S_G^{(sw1,T_{60})}$ specific estimators. Speaking about the in bands estimators, we recall that the whole range of the T_{60} variable is $[0.5, 2.5]s$, while the bands that we are considering are $[0.5, 0.9]s$, $[1.3, 1.7]s$ and $[2.1, 2.5]s$. The goal of these estimators is to perform shape classification on a RT-band basis. To do it, we compare the RGIDA and RGILA architectures and a H categorical cross-entropy loss is minimized.
This subsection compares the architectures at the variation of signal type and specific reverberation time band.

Using RIR signals Here we compare the performances of both the architectures over the impulse responses with estimators trained on separate and increasing RT-bands.

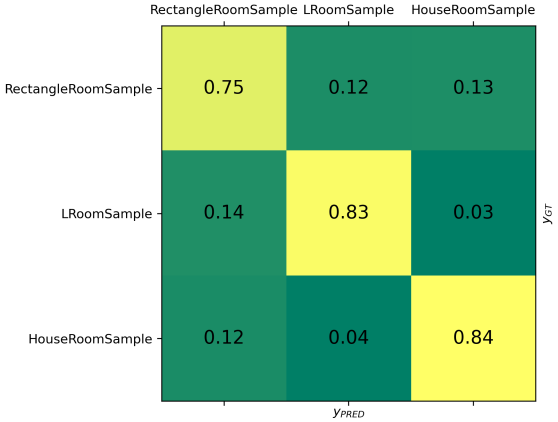


(a) RGIDA

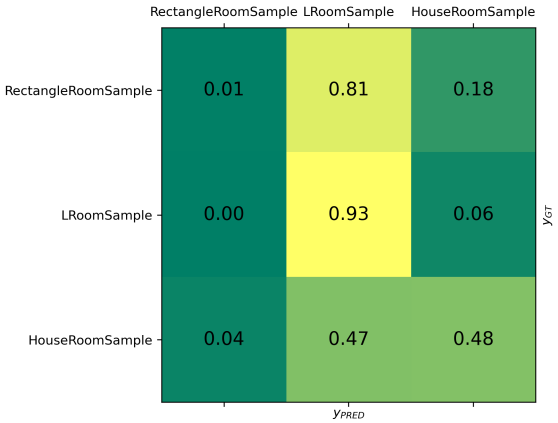


(b) RGILA

Figure 4.31: Specific in-RT-band shape classification on RIR signals - $T_{60} \in [0.5, 0.9]s$

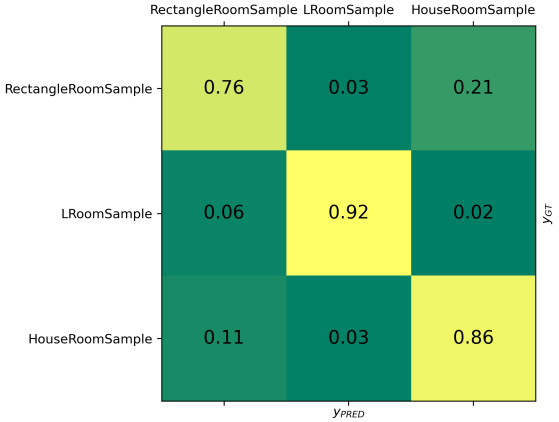


(a) RGIDA

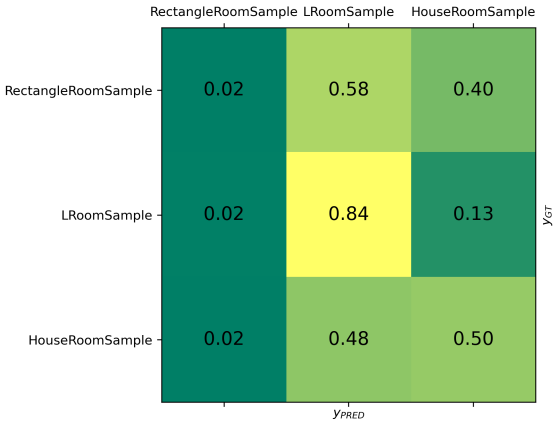


(b) RGILA

Figure 4.32: Specific in-RT-band shape classification on RIR signals - $T_{60} \in [1.3, 1.7]s$



(a) RGIDA



(b) RGILA

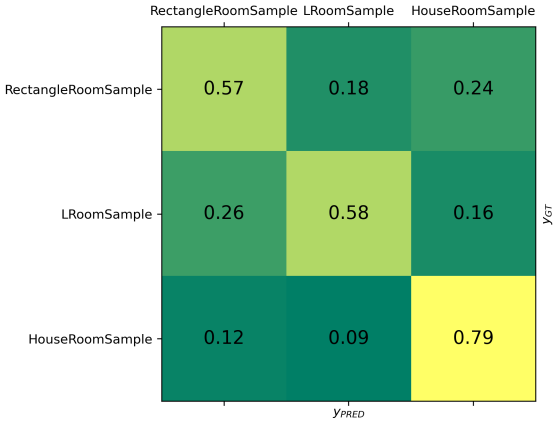
Figure 4.33: Specific in-RT-band shape classification on RIR signals - $T_{60} \in [2.1, 2.5]s$

	ACC [%]	
RT-band $[s]$	**RGIDA**	**RGILA**
$[0.5, 0.9]$	74	68
$[1.3, 1.7]$	83	47
$[2.1, 2.5]$	86	48

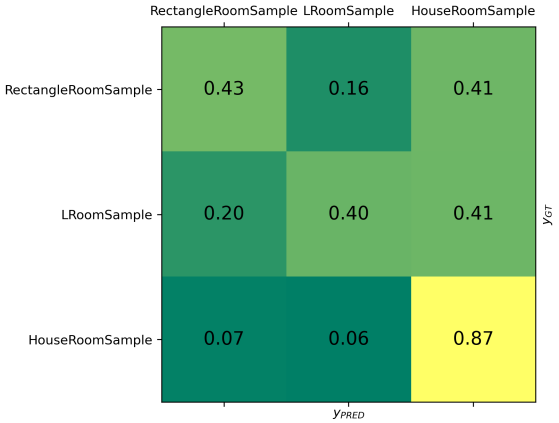
Table 4.11: Specific in-RT-band shape classification on RIR signals - RGIDA vs RGILA comparison

Our comments, in this case, are similar to the ones of the first paragraph of Section 4.3.2.

Using reverberant white noise signals Here we compare the performances of both the architectures over the reverberant noises with estimators trained on separate and increasing RT-bands.

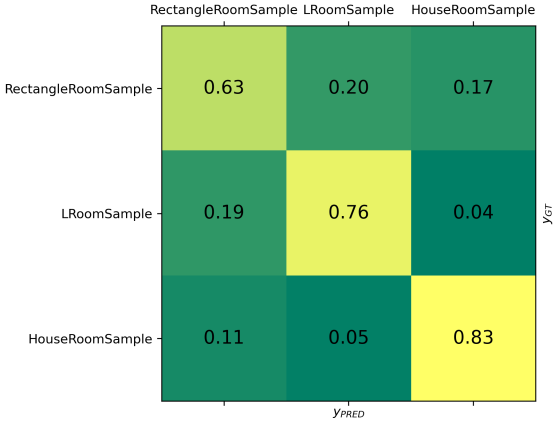


(a) RGIDA

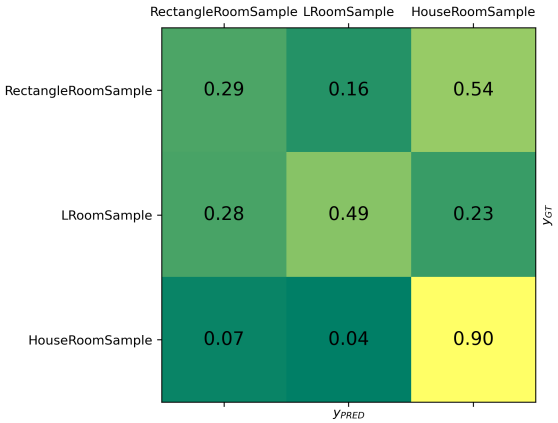


(b) RGILA

Figure 4.34: Specific in-RT-band shape classification on white noise signals - $T_{60} \in [0.5, 0.9]s$

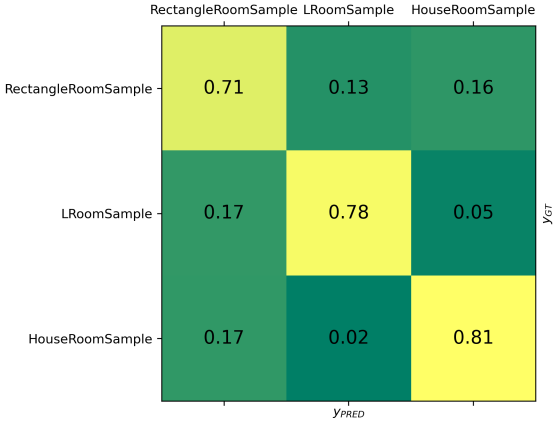


(a) RGIDA

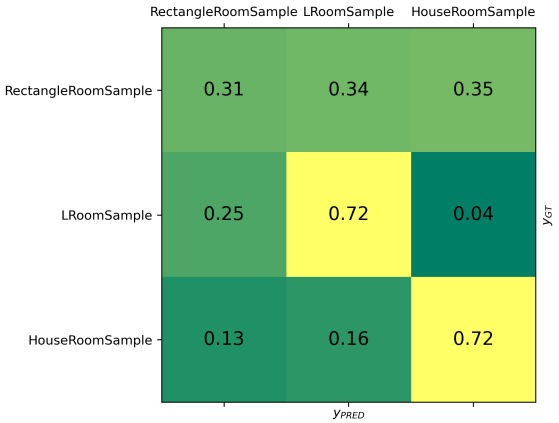


(b) RGILA

Figure 4.35: Specific in-RT-band shape classification on white noise signals - $T_{60} \in [1.3, 1.7]s$



(a) RGIDA



(b) RGILA

Figure 4.36: Specific in-RT-band shape classification on white noise signals - $T_{60} \in [2.1, 2.5]s$

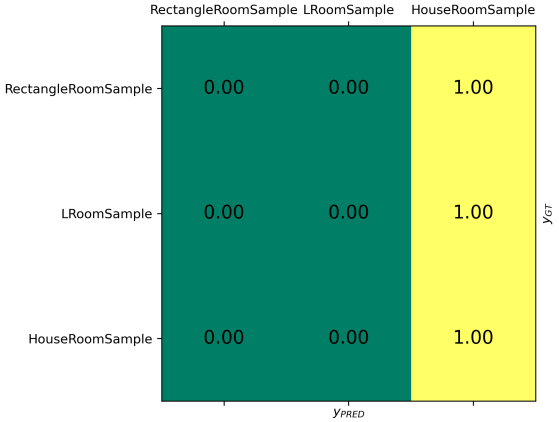
	ACC [%]	
RT-band $[s]$	**RGIDA**	**RGILA**
$[0.5, 0.9]$	65	57
$[1.3, 1.7]$	77	59
$[2.1, 2.5]$	79	56

Table 4.12: Specific in-RT-band shape classification on white noise signals - RGIDA vs RGILA comparison

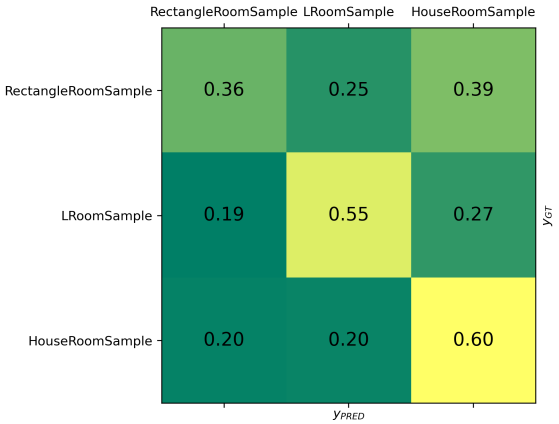
Our comments, in this case, are similar to the ones of the second paragraph of Section 4.3.2, apart from the fact that for T_{60} values belonging to the last band the rectangular room seems equally confused for the L-shaped and the House-shaped ones by the estimator built on RGILA.
Furthermore, we report that in the case of $T_{60} \in [0.5, 0.9]s$ for RGIDA, with the feature-map described in the methodology within the first paragraph of Section 3.2.1.1, we obtained a 35% accuracy versus the current

65% obtained by considering the sole γ-FB feature-map and a filter density of 30 bands.

Using reverberant speech signals Here we compare the performances of both the architectures over the reverberant voices with estimators trained on separate and increasing RT-bands.

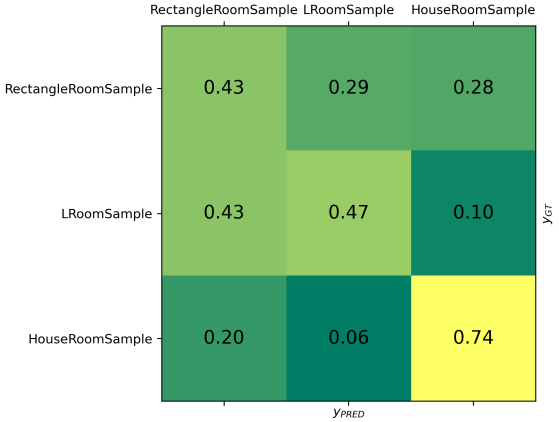


(a) RGIDA

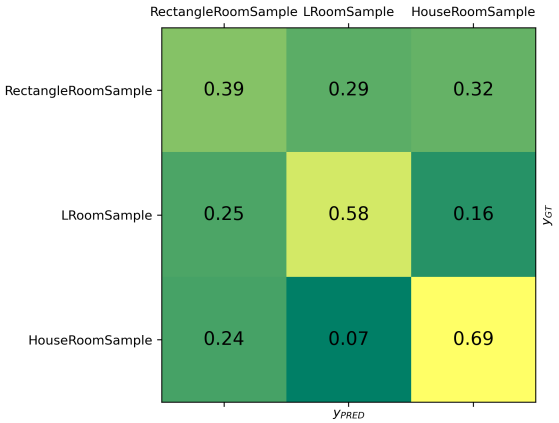


(b) RGILA

Figure 4.37: Specific in-RT-band shape classification on vocal signals - $T_{60} \in [0.5, 0.9]s$

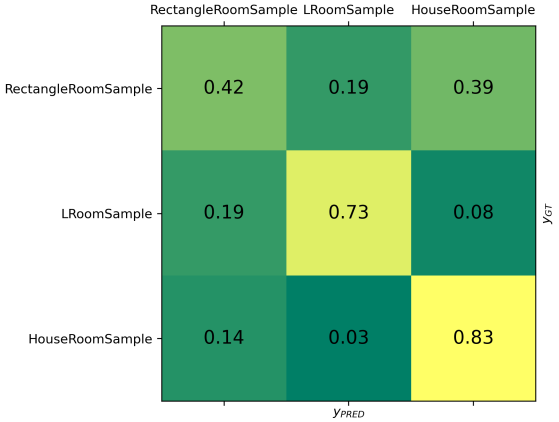


(a) RGIDA

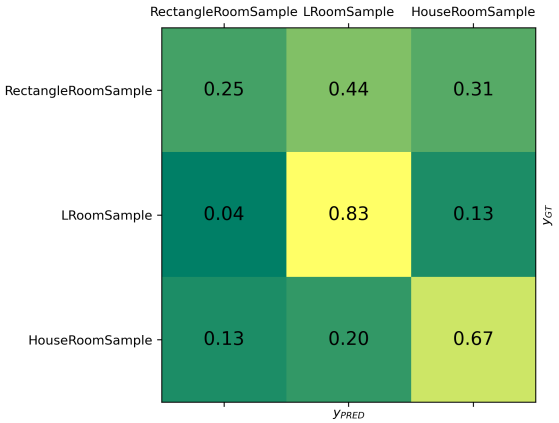


(b) RGILA

Figure 4.38: Specific in-RT-band shape classification on vocal signals - $T_{60} \in [1.3, 1.7]s$



(a) RGIDA



(b) RGILA

Figure 4.39: Specific in-RT-band shape classification on vocal signals - $T_{60} \in [2.1, 2.5]s$

	ACC [%]	
RT-band $[s]$	**RGIDA**	**RGILA**
$[0.5, 0.9]$	36	52
$[1.3, 1.7]$	57	60
$[2.1, 2.5]$	66	56

Table 4.13: Specific in-RT-band shape classification on vocal signals - RGIDA vs RGILA comparison

Our comments, in this case, are similar to the ones of the third paragraph of Section 4.3.2, aside the fact that RGILA tends to confuse a rectangular room for a House-shaped one for smaller T_{60}s and rectangular for L-shaped ones in the third considered reverberation time band. Even in this case we tried to better the feature-set for RGIDA, but we could not gain any boost.

A couple of words are worth to be spent on the comparison of specific shape classifiers parametrized either on volume or reverberation time. We remark that the observation for the volume and reverberation time bands are similar. However, the general results coming from the volume band analysis demonstrate higher accuracy and lower confusion.

4.3.4 Grad-CAM feature spotlights

In this section we put at work the Grad-CAM instrument explained in Section 2.5.5 to gain some insights into the usefulness of features for the specific shape classificators based on RGIDA.
Such architecture outperforms RGILA in almost all the cases.

4.3.4.1 Using RIR signals

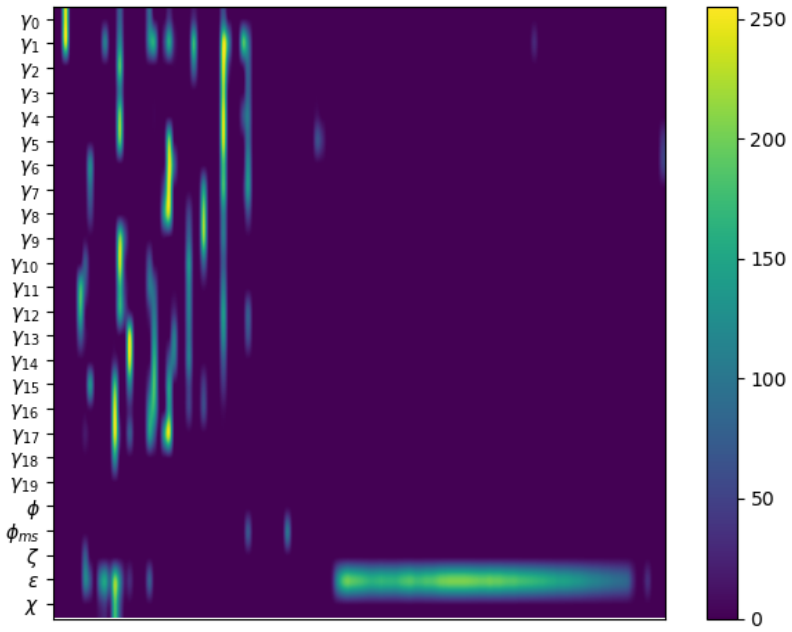


Figure 4.40: An heat-map built on a RIR feature-map

Figure 4.40 represents an heat-map retrieved with Grad-CAM during the analysis of some RIR test signals.
Such image can be considered as a combination of activation maps retrieved from the network for a specific sample. If we superposed this image to the related input RGIDA feature-map, we would highlight the portions of the original feature-map which are relevant for our classification task.

From Figure 4.40 we can observe that the most part the informative content is spread in the initial part of the response in the whole range of frequency bands. However, considering the response from which feature-map and heat-map are determined, the network considers as relevant even echoes which are not strictly belonging to the early region. In particular, a good portion of the tail seems relevant in the envelope feature vector. From other samples at our disposal, we can claim that even the features which appear irrelevant in Figure 4.40, are actually relevant.
Furthermore, extending our knowledge about the importance of lower-frequencies for the volume estimation task, for the shape classification task, even higher frequency modes of the rooms are involved.
This means that also higher frequency bands are useful for the task.
We might also allege that Grad-CAM is spotting some spikes in each band (echo patterns in the reverberant signals) which help the NN distinguish among the classes.

4.3.4.2 Using reverberant white noise signals

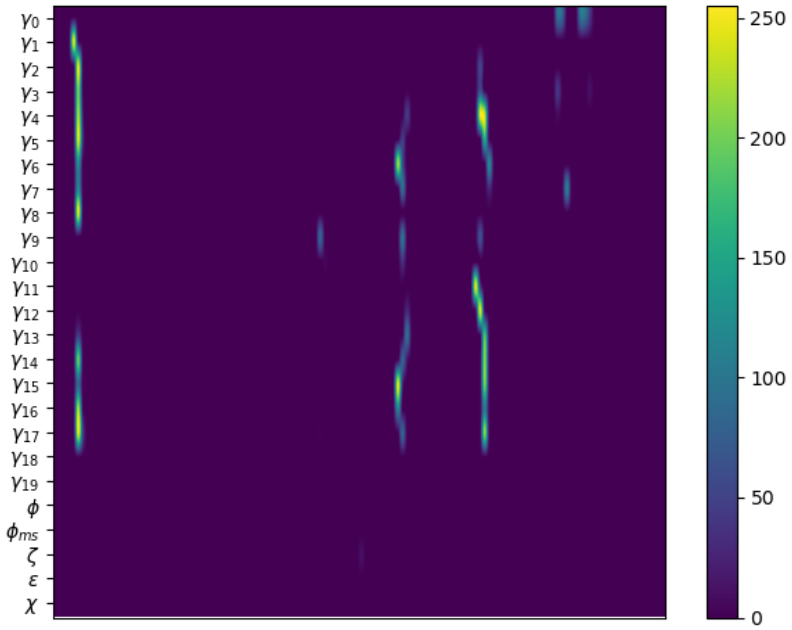


Figure 4.41: An heat-map built on a reverberant white noise feature-map

The analysis of the reverberant noise signal highlights a narrow component in the attack phase as important.
Once again the classifier agilely finds possible repetitive patterns in the signal reverberation.

4.3.4.3 Using reverberant speech signals

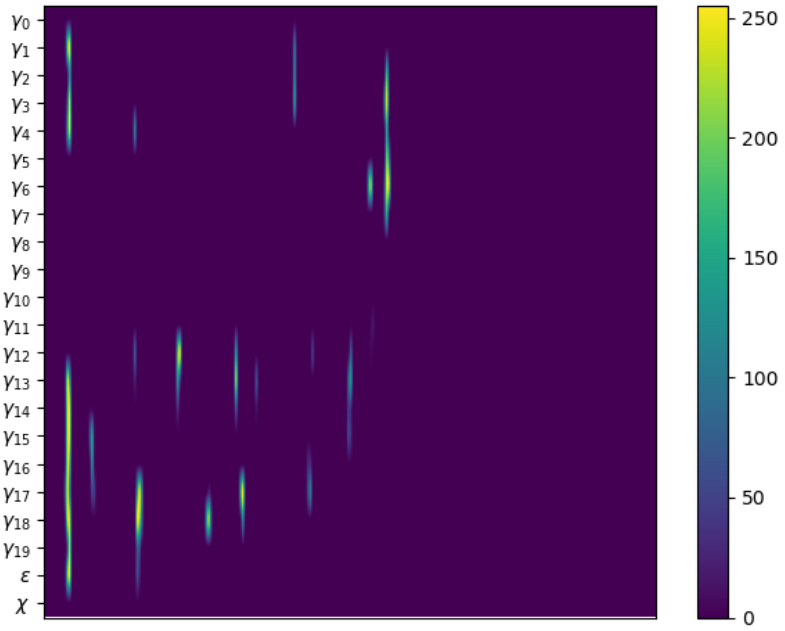


Figure 4.42: An heat-map built on a reverberant voice feature-map

For what concerns the voices, we find that both attack and release related to each spoken word might influence our estimate.
In this case, the most relevant feature seems to be the γ-FB. In it, echoing repetitions due to reverberation appear to be detected by Grad-CAM.

4.4 Conclusive remarks

In this chapter we have evaluated the proposed methodology through simulations and experiments.
We've first built our datasets, then we provided the estimators for preliminary volume and T_{60} estimation and specific shape prediction.
There, we discovered that the strain for the volume estimation increases as the volume raises.
On the other hand, the shape classification retrieval seems simpler with greater volumes or reverberation times.
Depending on the input signal type, we obtain better performances with RIRs and white noises, while we evidence greater efforts with voices for both the preliminary and specific estimations.
Considering the specific shape classifiers, the RGIDA architecture proposal overcomes the results of RGILA in all the cases, aside for the first volume and reverberation time with speech signals.

For what concerns the case of reverberant voices, neglecting the preliminary estimator, the first volume/RT band cannot be correctly classified.

This might be due to a way too brief reverberation. It might be interesting to investigate the mapping of the volume range of certain volume bands in longer reverberation times.

With Grad-CAM we observed that, as expected, the early reflections contain a good part of the informative content which results useful for the classification. However, it seems that even part of the diffuse tail carries information for the neural network.

5 Conclusions and Future Works

5.1 Conclusions

This work of thesis proposes a methodology for inferring the floor plan shape of a room starting from a reverberant speech signal. To our knowledge this work shows great novelty, indeed we could not find other works with a similar aim starting from the mono audio hypothesis. The purpose of this research is mainly focused on audio forensics and integrity checking, but interesting implications might affect also the room reconstruction field and the soundfield rendering techniques adopted by loudspeaker systems.
The devised methodology is based on a learning strategy which adopts and compares two CNN architectures to build the estimators performing the classification task. Such architectures are inspired to the literature. Here we proposed the estimation of an unstudied room parameter. The main advantages are related both to the room acoustics field and to the audio forensics one.
Indeed, the introduction of a new integrity technique requires greater efforts in camouflaging recorded audio signals, therefore requiring new skills to be acquired by malicious attackers. The forgery of a room shape requires much more analysis and endeavour than the falsification of either the reverberation time or the volume. However, it is evident that sofisticated dereverberation techniques followed by synchronous superpositions of anechoic signals and re-spatialization could put in serious trouble the most advanced forensics instruments.
In addition, as far as the experiments are concerned, we observe our architectural solution based on RGIDA has better performances than the RGILA in almost all the cases. Furthermore, the coarse volume esti-

mation performances do decrease for greater volumes, while the coarse reverberation time estimation performance does not seem to have a dependence on the RT itself. Speaking about the shape classification, we observe that to bigger volumes or reverberation times do correspond better performances. However, especially in the reverberant voice scenario, we observe that the rectangular room classification undergoes difficulties which might be due to the rectangular floor room peculiar modal superpositions.
Having that said, we claim that the proposed approach has shown promising results both in simulations and in experiments, apart from the case of small rooms.

5.2 Future Works

It is easy to see that our solution pipeline can be used on fragments of audio files of some seconds (the assumption of the availability of a speech signal of some minutes is not so strict) as demonstrated during the experiments. The segment samples from an original track can be exploited to obtain a time-dependent shape confidence profile. Then, neglecting the probability distributions with an entropy above a certain threshold, we retain significant estimates. From their analysis, we can check for incoherent segments.
As mentioned, replacing a fixed segmentation with functioning FDRs would be beneficial, both in terms of resources consumption and in completeness of the input pipeline for both architectures.
As far as data generation is concerned, an optimized finite RIR generation algorithm would be worthwhile. At this regard, a hybrid beam tracing - ray tracing technique could be investigated to improve the quality of the responses and to reduce the generation computational costs. A physically acquired dataset of responses could be used to test our solver.
Furthermore, the results could be improved expanding the feature-set of the primary RGIDA architecture. Indeed, considering other reverberation-related or phase-related features (e.g. STFT phase) or cepstral maps could help greatly. The drawback here is that the feature-maps sizes would considerably increase, thus implying greater memory consumption.
A trade-off should consider once again the usage of decay regions, especially in the case of small rooms, in which the temporal resolution seems crucial. For what concerns RGIDA, a good idea would be to consider such regions and their representations without log-energy compression.
Going to the estimators, hyper-parameters optimization is worth to be more deeply investigated. What is more, it would be of great interest to generate a binary dataset of real and spliced tracks in which environments of different shapes and volumes do contribute to each audio signal. Then, the research could move towards the usage of time-aware neural networks (for example Long Short-Term Memory (LSTM) RNNs).
For what concerns the room imaging field, we might consider speech sig-

nals, then think of retrieving a shape class and of estimating such class variables to build an approximate 3D model of the enclosure.

Bibliography

[1] T. M. Prego, A. A. Lima, S. L. Netto, B. Lee, A. Said, R. W. Schafer, and T. Kalker, "A blind algorithm for reverberation-time estimation using subband decomposition of speech signals," *The Acoustical Society of America (ASA)*, vol. 131, no. 4, pp. 2811–2816, 2012.

[2] D. Aprea, F. Antonacci, A. Sarti, and S. Tubaro, "Acoustic reconstruction of the geometry of an environment through acquisition of a controlled emission," in *European Signal Processing Conference (EUSIPCO)*, 2009.

[3] F. Antonacci, J. Filos, M. Thomas, E. Habets, A. Sarti, P. Naylor, and S. Tubaro, "Inference of room geometry from acoustic impulse responses," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 20, no. 10, pp. 2683–2695, 2012.

[4] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, PMLR, 09–15 Jun 2019.

[5] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Acoustical Society of America (ASA)*, vol. 65, no. 4, pp. 943–950, 1979.

[6] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West, "A beam tracing approach to acoustic modeling for interactive virtual environments," in *Proceedings of the 25th Conference of Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 21–32, 1998.

[7] F. Antonacci, M. Foco, A. Sarti, and S. Tubaro, "Fast tracing of acoustic beams and paths through visibility lookup," *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 16, no. 4, pp. 812–824, 2008.

[8] M. Vorländer, "The accuracy of calculations using the room acoustical ray-tracing model and its dependence on the calculation time," *Acustica*, vol. 66, no. 2, pp. 90–96, 1988.

[9] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 110, no. 30, pp. 12187–12191, 2013.

[10] M. Crocco, A. Trucco, and A. Del Bue, "Uncalibrated 3d room geometry estimation from sound impulse responses," *Journal of the Franklin Institute*, vol. 354, pp. 8678–8709, 2017.

[11] F. Xiong, S. Goetze, and B. T. Meyer, "Blind estimation of reverberation time based on spectro-temporal modulation filtering," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 443–447, 2013.

[12] N. Peters, H. Lei, and G. Friedland, "Name that room: room identification using acoustic features in a recording," in *Proceedings of the 20th ACM international conference on Multimedia*, 2012.

[13] H. Malik, "Acoustic environment identification and its applications to audio forensics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1827–1837, 2013.

[14] M. Marković and J. Geiger, "Reverberation-based feature extraction for acoustic scene classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[15] A. F. Genovese, H. Gamper, V. Pulkki, N. Raghuvanshi, and I. J. Tashev, "Blind room volume estimation from single-channel noisy speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[16] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi, "Leaf: a learnable frontend for audio classification," in *arXiv:2101.08596 [cs.SD]*, 2021.

[17] J. Borish, "Image method for efficiently simulating small-room acoustics," *The Acoustical Society of America (ASA)*, vol. 75, no. 6, pp. 1827–1836, 1984.

[18] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, M. Sondhi, J. E. West, G. Pingali, P. Min, and N. A., "A beam tracing method for interactive architectural acoustics," *The Acoustical Society of America (ASA)*, vol. 115, no. 2, pp. 739–756, 2004.

[19] F. Antonacci, A. Sarti, and S. Tubaro, "Two-dimensional beam tracing from visibility diagrams for real-time acoustic rendering," *EURASIP Journal on Advances in Signal Processing*, 2010.

[20] M. Vorländer, "Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm," *The Acoustical Society of America (ASA)*, vol. 86, no. 1, pp. 172–178, 1989.

[21] H. Lehnert, "Systematic errors of the ray-tracing algorithm," *Applied Acoustics*, vol. 38, no. 2-4, pp. 207–221, 1993.

[22] W. C. Sabine *Collected Papers on Acoustics*, 1923.

[23] H. Eyring, "Reverberation time in dead rooms," *The Acoustical Society of America (ASA)*, vol. 1, p. 168, 1930.

[24] M. R. Schroeder, "New method of measuring reverberation time," *The Acoustical Society of America (ASA)*, vol. 37, pp. 409–412, 1965.

[25] J. Vieira, "Automatic estimation of reverberation time," *Audio Engineering Society (AES)*, pp. 1–7, 2004.

[26] F. Xiong, S. Goetze, and B. T. Meyer, "Joint estimation of reverberation time and direct-to-reverberation ratio from speech using auditory inspired features," in *Proceedings of the ACE Challenge Workshop, a satellite event of IEEE-WASPAA*, 2015.

[27] F. Xiong, S. Goetze, B. Kollmeier, and B. T. Meyer, "Joint estimation of reverberation time and early-to-late reverberation ratio from single-channel speech signals," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 27, no. 2, pp. 255–267, 2019.

[28] P. P. Parada, D. Sharma, T. Waterschoot, and P. A. Naylor, "Evaluating the non-intrusive room acoustics algorithm with the ace challenge," in *Proceedings of the ACE Challenge Workshop, a satellite event of IEEE-WASPAA*, 2015.

[29] A. H. Moore, M. Brookes, and P. A. Naylor, "Roomprints for forensic audio applications," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2015.

[30] D. Marković, F. Antonacci, A. Sarti, and S. Tubaro, "Estimation of room dimensions from a single impulse response," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013.

[31] W. Yu and W. B. Kleijn, "Room geometry estimation from room impulse responses using convolutional neural networks," in *arXiv:1904.00869 [eess.AS]*, 2019.

[32] S. Tervo and T. Korhonen, "Estimation of reflective surfaces from continuous signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[33] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Sauroud, "Trainable frontend for robust and far-field keyword spotting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[34] X. Xiao, M. Yan, S. Basodi, C. Ji, and Y. Pan, "Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm," in *arXiv:2006.12703 [cs.NE]*, 2020.

[35] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and B. D., "Grad-cam: visual explanations from deep networks via gradient-based localization," in *arXiv:1610.02391 [cs.CV]*, 2016.

[36] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[37] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: a python package for audio room simulation and array processing algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[38] Y. Li and S. Manoharan, "A performance comparison of sql and nosql databases," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 15–19, 2013.

[39] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *5th ISCA Speech Synthesis Workshop*, 2004.